

Privacy Protection for Authentication Protocols

Vom Fachbereich Informatik der
Technischen Universität Darmstadt genehmigte

Dissertation

zur Erlangung des Grades
Doktor rerum naturalium (Dr. rer. nat.)

von

Dipl.-Inf. Dipl.-Math. Bertram Poettering

geboren in Guatemala Stadt



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Referenten: Prof. Dr. Mark Manulis
Technische Universität Darmstadt
Prof. Kenneth G. Paterson, Ph.D.
Royal Holloway, University of London

Tag der Einreichung: 18. November 2011
Tag der mdl. Prüfung: 17. Januar 2012
Hochschulkennziffer: D 17

Darmstadt 2012

Dieses Dokument wird bereitgestellt von tuprints, E-Publishing-Service der TU Darmstadt.

<http://tuprints.ulb.tu-darmstadt.de>

tuprints@ulb.tu-darmstadt.de

Bitte zitieren Sie dieses Dokument als:

URN: [urn:nbn:de:tuda-tuprints-28676](https://nbn-resolving.org/urn:nbn:de:tuda-tuprints-28676)

URL: <http://tuprints.ulb.tu-darmstadt.de/2867>

Die Veröffentlichung steht unter folgender Creative Commons Lizenz:

Namensnennung-Keine kommerzielle Nutzung-Keine Bearbeitung 2.0 Deutschland

<http://creativecommons.org/licenses/by-nc-nd/2.0/de/>



Abstract

In our highly computerized and networked society, privacy of individuals is precious and becomes increasingly important. Problems particularly arise in the context of authentication protocols where, as a general rule, entities actively reveal their respective identities to each other. To encounter this issue, different privacy-preserving authentication methods have been developed in the last decades. The list of these techniques comprises, apart from identity escrow, ring authentication, hidden and anonymous credentials, and several others, the concept of *affiliation-hiding authentication* (AHA). Such protocols offer the appealing and seemingly contradictory service to enable users to authenticate each other as members of a certain group without revealing their affiliation to group outsiders.

In AHA protocols (also known as *Secret Handshakes*), users become group members by registering with group authorities (GAs) and obtaining individual membership credentials. Group members then use their credentials to privately authenticate each other, optionally also establishing a secure session key. The pivotal privacy property that contrasts AHA with classical authentication or authenticated key establishment is that parties learn each other's affiliations to groups and compute common session keys if and only if their groups match.

Prior work has succeeded in constructing AHA protocols that offer different degrees of security, privacy, and efficiency. However, a set of essential problems have been left open. These include a close study of the level of trust that intrinsically has to be placed into participants of such systems (including into GAs), the extension of the single-group setting with only one GA to a setting where users are affiliated to multiple groups and, through AHA, want to discover matching ones, and certainly the question of efficient implementability. We argue that all these topics are highly relevant for practical deployment of privacy-preserving authentication in general, and AHA in particular. In this thesis, the author concretizes and cryptographically models these challenges, and offers provably secure solutions.

Furthermore, this thesis treats privacy-related challenges that are posed in the context of network-based social interactions. Without doubt, online social networks, that help participants to build and reflect their social relations to other participants, have taken an essential role in people's daily life. A key step in the constitution of new links between participants consists of the reconciliation of shared contacts or friends. The author develops techniques to discover common contacts in social networks in a privacy-aware manner, i.e., without disclosing non-matching contacts. Besides formalizing this task and offering appropriate solutions, the thesis analyzes an interesting connection between AHA protocols and the challenge of private discovery of common contacts.

By identifying and solving a variety of relevant open problems in the context of privacy-aware authentication, this thesis contributes to wide-scale deployment of methods that respect and regain user privacy in p2p systems, mobile ad hoc networks, and social networking applications.

Zusammenfassung

In unserer zunehmend computerisierten und vernetzten Gesellschaft gewinnt der Schutz der Privatsphäre der Menschen zunehmend an Bedeutung. In dieser Hinsicht können insbesondere Authentikationsprotokolle problematisch sein, da es ihre Aufgabe ist, Nutzer und ihre jeweiligen Identitäten eindeutig zu bestimmen. Um dieser Problematik zu begegnen, sind in den vergangenen Jahren verschiedene privatsphären-schützende Authentikationsmethoden entwickelt worden, etwa *identity escrow*, *ring authentication*, *hidden* und *anonymous credentials*. Ein weiteres interessantes Konzept ist die *zugehörigkeit-versteckende Authentikation* (*affiliation-hiding authentication*, AHA). Solche Protokolle bieten Nutzern die scheinbar widersprüchliche Möglichkeit, sich gegenseitig als Mitglieder bestimmter Gruppen zu authentisieren, ohne jedoch zugleich Außenseitern diese Zugehörigkeit preiszugeben.

In AHA-Protokollen (auch bekannt als *Secret Handshakes*) werden Nutzer zu Mitgliedern von Gruppen durch Registrierung bei sog. Gruppenautoritäten (*group authorities*, GAs) und erhalten dabei entsprechende Zugehörigkeitsnachweise. Gruppenmitglieder verwenden diese Nachweise, um sich gegenseitig zu authentisieren, und um evtl. sichere Sitzungsschlüssel auszutauschen. Die entscheidende Eigenschaft, die AHA bzgl. Privatsphärenschutz von klassischer Authentikation unterscheidet, liegt jedoch in der Tatsache, dass Teilnehmer nur bei Übereinstimmung ihrer Gruppen die jeweiligen Gruppenzugehörigkeiten erfahren (bzw. gleiche Sitzungsschlüssel berechnen).

Die bisherige Forschung in diesem Bereich hat verschiedene AHA-Protokolle hervorgebracht. Diese genügen unterschiedlichen Sicherheits- und Effizienzanforderungen. Einige wichtige Probleme sind jedoch weiterhin ungelöst. Dazu gehört u.a. eine genaue Vermessung des Vertrauens, das immanent den Mitgliedern von Gruppen und ihren GAs entgegengebracht werden muss, sowie die Erweiterung des Ein-Gruppen-Settings mit nur einer GA zu einem Setting, in dem es Nutzern möglich ist, mehreren

Gruppen zugehörig zu sein und vermöge AHA übereinstimmende Gruppen zu erkennen. Zudem stellt sich natürlich die Frage nach effizienter Implementierbarkeit von AHA-Verfahren. Die Bearbeitung dieser Themen ist Voraussetzung für den praktischen Einsatz von privatsphären-schützender Authentikation im Allgemeinen, und von AHA insbesondere. In der vorliegenden Dissertation werden die genannten Fragestellungen diskutiert und kryptographisch bearbeitet. In diesem Rahmen werden zudem entsprechende AHA-Protokolle entworfen und als sicher bewiesen.

Diese Dissertation befasst sich des weiteren mit Problemen des Schutzes der Privatsphäre, die im Bereich Netzwerk-gestützter sozialer Interaktion entstehen. Online-geführte soziale Netzwerke unterstützen ihre Nutzer beim Aufbau und Verwalten sozialer Verbindungen und haben sicherlich eine wesentliche Rolle im täglichen Leben der Menschen eingenommen. Eine Schlüsselfunktion kommt beim Aufbau neuer Verbindungen zwischen Teilnehmern dem Abgleich *gemeinsamer Kontakte* oder *Freundschaften* mit anderen Teilnehmern zu. Die in dieser Arbeit entwickelten Methoden erlauben es, in sozialen Netzwerken solche gemeinsamen Kontakte unter Einhaltung der Privatsphäre der Teilnehmer zu identifizieren, d.h., ohne dass nicht-gemeinsame Kontakte preisgegeben werden. Neben der Formalisierung dieser Aufgabe und dem Entwickeln geeigneter Lösungen, analysiert die vorliegende Arbeit einen interessanten Zusammenhang zwischen AHA-Protokollen und Protokollen zur Entdeckung gemeinsamer Kontakte.

Durch das Identifizieren und Lösen einer Reihe von relevanten offenen Problemen im Bereich der kryptographischen Authentikation trägt diese Arbeit zum breiten praktischen Einsatz von Methoden bei, die die Privatsphäre von Nutzern schützen bzw. bei ihrer Wiedererlangung helfen.

Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

I am especially indebted to my supervisor Mark Manulis who gave me the opportunity to study and write this thesis under his encouraging guidance. I am very grateful to him for the support and assistance he offered from the initial to the concluding level. He proposed ideas for research projects and established contacts with numerous excellent researchers in cryptography. His time-intensive supervision has helped me during all the time of research and of writing this thesis.

I also want to thank Marc Fischlin and Stefan Katzenbeisser and the students of their research groups. I particularly remember innumerable fruitful and enlightening discussions with Paul Baecher, Christina Brzuska, Özgür Dagdelen, Pooya Farshim, Cristina Onete, Andreas Peter, and Heike and Dominique Schröder. Over the past years of common studying you have become close friends! Also attending the reading group organized by Marc and his students contributed substantially to my formation as a cryptographer. I thank Andrea Püchner and Heike Meissner for their organizational support.

During my studies at TU Darmstadt I had the opportunity to meet and work with many brilliant international researchers. I treasure the discussions and joint research with my co-authors and co-researchers Colin Boyd, Emiliano De Cristofaro, Pooya Farshim, Benoît Libert, Juan Manuel González Nieto, Kenny Paterson, Benny Pinkas, Douglas Stebila, and Gene Tsudik as highly inspiring and illuminating.

I would like to make a special reference to the excellent working conditions I found at the Department of Computer Science at TU Darmstadt and the support I received from the Center for Advanced Security Research Darmstadt (CASED).

Last, not least, I want to express my gratitude to my family for their constant encouragement.

Contents

1. Introduction	1
1.1. Summary of results	2
1.2. Related work	5
1.3. Classification of AHA	7
1.4. Related concepts	9
1.4.1. Ring signatures	10
1.4.2. Ring authentication, private authentication	10
1.4.3. Group signatures	10
1.4.4. Traceable signatures	11
1.4.5. Identity escrow	11
1.4.6. Anonymous signatures	11
1.4.7. Anonymous credentials	11
1.4.8. Predicate-based signatures and key agreement	12
2. Prerequisites and building blocks	13
2.1. Mathematical background	13
2.1.1. Number theory	13
2.1.2. Cryptography in cyclic groups	14
2.1.3. Cryptography in composite order groups	16
2.1.4. Cryptography in the pairing-based setting	16
2.2. Digital signature schemes	17
2.2.1. Schnorr's signature scheme	18
2.2.2. Full-domain hash RSA signature scheme	18
2.3. Non-interactive key distribution	19
2.3.1. Construction of NIKDS based on bilinear maps	20
2.4. Syntax and Correctness of AHA	21

2.5.	Basic AHA constructions	23
2.5.1.	AHA based on RSA	23
2.5.2.	AHA based on DLP	26
2.5.3.	AHA based on bilinear maps	28
3.	AHA with untrusted group authorities	29
3.1.	A security model for AHA that considers dishonest GAs	32
3.1.1.	Adapted syntax for AHA	32
3.1.2.	Adversarial queries	33
3.1.3.	Affiliation-hiding security in the presence of corrupt GAs	35
3.1.4.	Key security in the presence of corrupt GAs	37
3.1.5.	Untraceability	39
3.2.	A construction based on RSA	40
3.2.1.	Protocol specification	41
3.2.2.	Efficiency and optimizations	43
3.2.3.	Security analysis	44
3.2.4.	Proving well-formedness of RSA parameters	49
3.3.	Comparison of protocols	52
4.	Strategies towards multigroup AHA	53
4.1.	The naïve approach	54
4.2.	(Authorized) Private Set Intersection	54
4.3.	Reducing the overhead by using hashing	55
4.4.	An attempt to further improve the overhead	56
4.5.	Index-hiding message encoding	57
4.5.1.	A construction of IHME	59
4.5.2.	Multigroup AHA from IHME	60
4.6.	Comparison of strategies	60
5.	Multigroup AHA	63
5.1.	The mAHA scheme by Jarecki and Liu	63
5.2.	Syntax of mAHA	65
5.3.	A security model for mAHA	66
5.3.1.	Adversarial queries	66
5.3.2.	Affiliation-hiding security	67
5.3.3.	Key security	68
5.4.	A mAHA construction based on RSA	70
5.4.1.	Protocol specification	71
5.4.2.	Efficiency and optimizations	73
5.4.3.	Security analysis	73

5.5.	A mAHA construction based on NIKDS	79
5.5.1.	Protocol specification	80
5.5.2.	Efficiency analysis	81
5.5.3.	Security analysis	82
5.6.	Comparison of our mAHA solutions	87
5.7.	On the feasibility of DLP-based mAHA	88
6.	Multigroup AHA in practice	91
6.1.	Optimizing IHME	91
6.1.1.	Polynomial interpolation	92
6.1.2.	Lagrange interpolation	93
6.1.3.	Newton interpolation	93
6.1.4.	Interpolation without precomputation	93
6.1.5.	Interpolation with precomputation	95
6.1.6.	Performance comparison of interpolation algorithms	96
6.1.7.	Polynomial evaluation	96
6.2.	Interleaved IHME	97
6.2.1.	Efficiency analysis	98
6.2.2.	Interleaved IHME over the integers	99
6.3.	An optimized RSA-based mAHA protocol	99
6.3.1.	Optimized CreateGroup algorithm	100
6.3.2.	Optimized AddUser algorithm	101
6.3.3.	Optimized Handshake protocol	101
6.3.4.	Performance analysis and discussion	104
6.4.	Efficiency of NIKDS-based mAHA protocol	105
6.5.	Practical comparison of our mAHA solutions	106
7.	Applications of AHA: private contact discovery	107
7.1.	Approaches towards private contact discovery	108
7.2.	Contact discovery from mAHA?	109
7.3.	Syntax of CDS	110
7.4.	Security model for CDS	111
7.4.1.	Adversarial queries	112
7.4.2.	Contact-hiding security	112
7.5.	A CDS construction based on RSA	113
7.5.1.	Protocol specification	115
7.5.2.	Efficiency analysis	115
7.5.3.	Security analysis	117
7.6.	A CDS construction based on NIKDS	120
7.6.1.	Protocol specification	120

7.6.2. Efficiency analysis	121
7.6.3. Security analysis	121
8. Conclusion	125
8.1. Directions for future research	127
A. On the security of RSA-based AHA	149
B. Publication record	151

Traditional protocols for authentication [73,85,143] and authenticated key exchange (AKE) [28,70,111,112,114] reveal to observers the identities and certificates of users. In particular, in the PKI setting, the main goal is the establishment of unforgeable and publicly verifiable links between users' public keys and identities. Although this paradigm has proved to work reliably from a security point of view, it certainly raises questions about privacy of participants; for instance, if users want to communicate securely *without* leaving evidence that a conversation has ever taken place.

To cope with this issue, specialized privacy-aware authentication methods have been developed. What at first seems counter-intuitive (since authentication traditionally goes hand-in-hand with identification) turns out to be a very reasonable concept: Instead of authenticating users by identities, they are authenticated by properties or attributes. For example, in order to access a company's parking lot, it would be sufficient to authenticate as an (anonymous) employee of the company, instead of providing full name or worker's id. The variety of such property-based authentication methods that appeared in the literature is extensive (see Section 1.4 for a list). One interesting technique, *affiliation-hiding authentication* (AHA), where users authenticate as members of groups, emerged in 2003 (Balfanz *et al.* [9]) and received much attention [6, 47, 47, 95–97, 99, 150, 155, 161].

The idea behind AHA is simple: Prior to participation in the actual authentication session users become members of groups. These are administrated by specific group authorities (GAs) who issue individual membership credentials to registering users, but can also revoke users. The goal of AHA is to let participants mutually authenticate each other by *privately* comparing their respective affiliations. Authentication is deemed successful if the affiliations of participants are equal. Moreover, one of the posed privacy requirements is that affiliations of participants remain hidden to group outsiders, i.e., to members of non-matching groups, independently of whether authentication succeeds or not. Observe that this setting considerably deviates from classical authentication where roles are clearly separated in prover and verifier: in

AHA, equally equipped participants *mutually* authenticate each other.

Since the introduction of this concept by Balfanz *et al.*, several AHA constructions have been proposed, many of them bringing their individual tailor-made security models with them (compare, for instance, the four models from [6, 9, 95, 104]). However, in all cases, the essential security notions of AHA (that optionally comprises secure key establishment) are captured by the following intuitive description:

Affiliation-hiding security

It should be infeasible for an active adversary to learn the affiliation, i.e., the group membership, of any user from his authentication sessions or from knowledge of keys computed in these sessions.

Available publications model this requirement in various ways: For instance, while in [9] the (game-based) goals of *impersonator* and *detector resistance* have to be simultaneously fulfilled, Jarecki *et al.* [95] propose the more sophisticated notion of *affiliation-hiding* that encompasses both named sub-goals and additionally allows the adversary to schedule an unbounded number of concurrent AHA sessions (in contrast to [9]).

Authenticated key exchange security

It should be infeasible for an active adversary to distinguish the session key computed in a test session from a random value in the same range, with a probability non-negligibly exceeding $1/2$.

Key security in AHA is generally modeled following the general approach for key exchange protocols (e.g., [14, 46]), including the (sub)requirement of forward secrecy. The first AHA construction that formally considers key security is [95].

We anticipate that, in Chapter 3, we will introduce a third requirement: the *untraceability* of users. However, this goal plays a role only in the special context of untrusted group authorities.

1.1. Summary of results

The contributions of this thesis (apart from the introduction to existing concepts and constructions of AHA in Chapter 2) can be summarized as follows:

AHA with untrusted group authorities (Chapter 3)

After observing that, in current security models for AHA, group authorities are considered fully trustworthy, we investigate the effects that malicious/corrupt GAs can have on security of protocols. Our investigations reveal that, in most cases both, privacy of affiliations and security of established session keys,

are immediately lost after GA corruptions. To counter this problem, we propose an extended security model that factors in untrusted GAs, i.e., considers GAs as active adversaries. Amongst others, we introduce the new security notion of *untraceability* that, roughly speaking, requires that group members are not identifiable even by their own GA. We present an RSA-based AHA construction that is secure in this stronger model. Dealing with corrupt GAs is especially challenging in the RSA setting, as our model principally allows GAs to propagate malformed RSA parameters to group members. After giving a treatment on how well-formedness of such parameters can be enforced, the chapter concludes with a comparison of security and efficiency of our solution and other protocols.

This chapter bases on joint research with Mark Manulis and Gene Tsudik. Corresponding results were published in the proceedings of ACNS 2010 [124] and PETS 2010 [125].

Strategies towards multigroup AHA (Chapter 4)

This chapter focuses on AHA in a setting with multiple GAs who manage their groups independently of each other. In particular, users are allowed to be member of several groups, and corresponding authentication protocols privately discover the set of all common groups. Note that this setting is a very attractive and natural one; nevertheless, existing AHA constructions do not consider this problem and, as we discuss, (acceptably efficient) adaptations to this setting are not obvious. We conceptually separate the underlying problem of *group discovery* from more established cryptographic tools, like private set intersection (PSI, APSI). We describe several attempts to achieve constructions of multi-affiliation AHA protocols (mAHA), and propose a new cryptographic primitive — IHME — that proves itself path-breaking in the design of a first mAHA protocol. Concerning IHME, we specify corresponding security properties and give an information-theoretically secure construction. We highlight that this primitive may also be of independent interest and may find applications that are not necessarily related to preservation of users' privacy.

This chapter bases on a joint publication with Mark Manulis and Benny Pinkas that appeared on ACNS 2010 [121].

Multigroup AHA (Chapter 5)

Here we formally adapt the general syntax and security models of AHA schemes to the setting with multiple group authorities (mAHA). In addition, basing on observations and design strategies discovered in Chapter 4, we present two independent and efficient solutions to the mAHA challenge, generically building on the new IHME primitive and on non-interactive key distribution

(NIKDS), respectively. We highlight that the challenge of constructing such protocols was posed as an open problem by Jarecki, Kim, and Tsudik on CT-RSA 2008 [95, p. 356], and has not yet been solved in a satisfactory way. Efficiency of our protocols is impressive: it can be estimated with $O(n)$ public-key operations (i.e., exponentiations and pairing evaluations), where n is the number of affiliations per user. We provide detailed security reductions for our schemes and conclude the chapter with a comparison of all known approaches to mAHA in terms of security and efficiency.

Our mAHA constructions were also presented on ACNS 2010 [121] and WISTP 2011 [122], and result from joint work with Benny Pinkas (in the former case) and Mark Manulis.

Multigroup AHA in practice (Chapter 6)

This chapter evaluates whether efficiency of the mAHA constructions from Chapter 5 is in practice sufficient for deployment of affiliation-hiding authentication. In order to obtain a meaningful analysis, instead of comparing naïve implementations of the protocols, we combine a large set of known and novel techniques and tricks to further optimize the schemes and the underlying IHME primitive in several aspects. This leads to remarkable performance gains (in respect to runtime and bandwidth complexity, and key sizes). By presenting performance measurements obtained from concrete implementations of the protocols on different computing environments, we make evident that mAHA is ready for practical deployment, even on devices with constraint resources.

Our contributions on practical deployability of mAHA were also presented on ASIACCS 2011 [123], and were jointly achieved with Mark Manulis.

Applications of AHA: private contact discovery (Chapter 7)

We consider the concept of *private discovery of shared social contacts* in which two participants of an online social network interact and assess their social proximity by learning the set of contacts (friends) they have in common, without disclosing non-matching ones. We discuss the approaches to this challenge so far proposed in the literature — coming to the conclusion that none of them provides satisfying privacy guarantees. In particular, we formalize adequate security properties for private contact discovery and note that none of the analyzed schemes is secure in respect to this model. In contrast, we propose two solutions and prove their security in our model. Although our constructions share design ideas with our mAHA schemes from Chapter 5, we indicate why a generic conversion of mAHA protocols to contact-discovering schemes (CDS) is not feasible. We conclude the chapter evaluating and comparing the efficiency of our constructions. Note that, with private contact discovery, we identified

a further application of our IHME primitive from Chapter 4.

Our RSA-based construction for discovery of common contacts was published at ACNS 2011 [62] as joint work with Emiliano de Cristofaro and Mark Manulis. The NIKDS-based scheme was developed by the author especially for this thesis.

Chapter 8 concludes this thesis and, within others, enumerates some open problems and ideas for future research in the context of privacy-aware authentication.

1.2. Related work

We briefly describe the publications that appeared in the area of affiliation-hiding authentication. A more compact overview is also provided in Table 1.1. We do not consider schemes [156], [163], and [90], as they were broken (in [95], [103], and [152], respectively).

Early (linkable) AHA schemes [9, 47] provide group members with credentials composed of a pseudonym and additional secrets. Although authentication is in the main focus of these schemes ('secret handshakes'), both of them offer session key establishment as an additional service, but no formal security treatment of the latter is provided. All known linkable AHA schemes provide efficient revocation using certificate/pseudonym revocation lists. An extension to AHA that comprises (forward) secure session key establishment has been formally modeled and analyzed in [95]. The schemes by Balfanz *et al.* [9], Castelluccia, Jarecki, and Tsudik [47], and Jarecki, Kim, and Tsudik [95] are defined in the pairing-based, DLP, and RSA setting, respectively.

In unlinkable schemes, participants reuse their credentials across different AHA invocations, while the possibility of (adversarial) correlation of multiple sessions involving the same participant is precluded. In this setting, the challenging part is revocation of protocol participants, which is completely disregarded in the schemes by Ateniese, Kirsch, Blanton [6], Hoepman [89], Nasserian and Tsudik [131], and its refinement by Zhou, Susilo, and Mu in [168]. The unlinkable schemes [155] by Tsudik and Xu and [96] by Jarecki and Liu (partially) handle revocation by deployment of group key management (GKM) techniques based on Wallner's logical key hierarchy [158, 160]. More specifically, while [155] utilizes group signatures to achieve traceability of users, [96] adapts GKM to the key-private [10] public-key setting in which it is able to handle desynchronized revocation epochs. Jarecki and Liu [99] also construct a scheme that supports more efficient revocation, based on the verifier-local mechanism introduced in [26] (observe that each revocation check requires two pairing evaluations per revoked user). In particular, by publishing a

user-specific revocation token, GA turns unlinkable session transcripts into *traceable* ones, i.e., into transcripts that directly reveal the respective user. We remark that [99] builds on a ‘conditional oblivious transfer’ primitive (originally introduced in [66,67]), and that privacy of the scheme is not regarded in a model with concurrent sessions. The scheme proposed by Sorniotti and Molva in [150] adds revocation to the unlinkable AHA scheme [149] by the same authors. In particular, they claim that the ‘RevocationMatching’ (sub-)protocol they construct adds revocation to most linkable AHA schemes. As in [99], each revocation check takes two pairing evaluations per revoked user, assuming hardness of a tailor-made pairing-based ‘SM’ assumption. [150] also discusses why dynamic accumulators [40] are not suited for revocation in AHA schemes: the accumulator itself would reveal the affiliation of corresponding user.

A weaker flavor of unlinkability, *k-anonymity*, is explored by Xu and Yung in [161]. Intuitively, in *k-anonymous* schemes, participants hide behind sets of *k* users chosen ad-hoc, where *k* is an adjustable parameter, in contrast to classical unlinkability, where users hide behind the whole population. [161] also proposes a motivation for why *k-anonymity* might suffice to protect users’ privacy in practice: in modern western jurisdiction, 2-anonymity would impose ‘reasonable doubt’ at court, i.e., 2-anonymous AHA transcripts do not serve as evidence for committed online crimes¹. We point out that some technical problems might arise when deploying the protocol from [161]: not only that [161] assumes that all users are aware of all available groups, but in order to compute the ‘optimal’ value of *k*, the probability *p* that any given user is corrupted during its interactions, has to be known. Another critique on *k-anonymity* was raised by Jarecki and Liu [96], who notice that the affiliation of participants belongs to the intersection of the *k*-element sets released at each protocol invocation, such that sessions become linkable to users with high probability after a small number of recorded sessions. The notion of *k-anonymity* was originally coined in the context of database security [153].

The notions of *dynamic* and *fuzzy matching* were introduced to the AHA context by Ateniese, Kirsch, and Blanton in [6]. In the dynamic matching scenario, participants specify attributes that the protocol peer is expected to occupy (corresponding certificates are issued by the GA). In fuzzy matching, authentication is deemed successful if more than *d* of participants’ attributes match, where *d* is a session-dependent threshold. However, the matching algorithm proposed in [6] deploys standard techniques for private-set intersection [78,107], whose security is only proven in the semi-honest adversary model². As all participants possess the same

¹We argue, however, that the primary aim of deployment of privacy-preserving authentication should not lie in frustrating legal prosecution, but in hiding communication patterns of participants in general.

²Explicitly, the authors “assume that it is in the best interest of the players to authenticate

(possibly rerandomized) credential, [6] cannot offer revocation of users. Note that Ateniese *et al.* claim that their schemes supports multiple groups per user. However, close inspection of the specification reveals that the corresponding GAs need to share a single secret key, i.e., the different groups are not independent of each other. The notion of dynamic matching is strengthened to *dynamic controlled matching* by Sorniotti and Molva [149–151], where participants need specific certificates not only for the attributes they have, but also for the attributes they expect the peer to have.

Some schemes [93, 94, 155, 162] extend AHA’s execution model from two-party to multi-party authentication and key establishment. An approach by Jarecki, Kim, and Tsudik [93] uses credentials similar to those in (DLP-based) [47]. The same authors also lift the RSA-based scheme from [95] to the multi-party setting [94]. Both approaches achieve session group key establishment according to accepted security models [30] through a variant of the well-known Burmester-Desmedt technique [33]. Further on, Xu and Yung convert their k -anonymous AHA protocol from [161] to the multi-party setting in [162]. The only known unlinkable multi-party AHA scheme was given by Tsudik and Xu in [155], and is discussed above.

First results on privacy protection against misbehaving GAs are due to Kawai, Yoneyama, and Ohta [104] and Sorniotti and Molva [151]. They deviate from the traditional setting by splitting GA into a set of mutually distrusting authorities. However, as we will study in detail in Chapter 3, the setting with a single authority suffices to protect users’ privacy from corrupt GAs. Our line of work is hence more consistent with earlier AHA-related results.

Exclusively the protocol by Jarecki and Liu [97] offers support for multiple credentials per user, i.e., solves the problem of efficient group discovery. Yet, their scheme has the questionable property that, during the registration process, users have to surrender their secret keys to the GAs. We will identify concrete weaknesses of this approach in Sections 5.1 and 7.2.

1.3. Classification of AHA

We complete the presentation of related work from Section 1.2 and Table 1.1 by giving a classification of the notions of AHA that appeared in the literature so far.

Plain authentication vs. authenticated key agreement

While initially proposed AHA schemes [6, 9, 47, 155, 161] focused exclusively on privacy-aware authentication of participants (and were hence termed *secret handshakes* [9]), more recent constructions [94–97] also encompass secure

(and deviating from the prescribed behavior might prevent this), and the players follow the set intersection protocol correctly”. This argumentation, however, precludes adversaries from actively taking part in AHA sessions. This is certainly an unacceptable assumption in practice.

Publication	Conference	Linkability ¹	Revocation ²	AKE ³	Setting ⁴	Remark
Balfanz <i>et al.</i> [9]	S&P 03	L	✓	✗	BM	
Xu, Yung [161]	CCS 04	k	✓	✗	generic	
Castelluccia <i>et al.</i> [47]	Asiacrypt 04	L	✓	✗	DL	
Nasserian, Tsudik [131]	FC 06	L	✓	✗	DL	
Zhou <i>et al.</i> [168]	ISPEC 06	L	✓	✗	DL	
Tsudik, Xu [155]	PET 06	U	✓	✗	generic	multi-party AHA
Jarecki <i>et al.</i> [93]	ACNS 06	L	✓	✗	DL	multi-party AHA
Xu, Yung [162]	FC 07	k	✓	✗	generic	multi-party AHA
Hoepman [89]	ESAS 07	U	✗	✗	DL	
Ateniese <i>et al.</i> [6]	NDSS 07	U	✗	✗	BM	fuzzy matching, semi-honest
Jarecki, Liu [96]	ACNS 07	U	✓	✓	generic	delayed revocation ⁵
Jarecki <i>et al.</i> [94]	CT-RSA 07	L	✓	✓	RSA	multi-party AHA
Jarecki <i>et al.</i> [95]	CT-RSA 08	L	✓	✓	RSA	
Jarecki, Liu [97]	ICALP 08	L	✓	✓	DL	multiple groups/credentials
Jarecki, Liu [99]	CRYPTO 09	U	✓	✗	DL	verifier-local revocation
Sorniotti, Molva [149]	SEC 09	U	✗	✗	BM	controlled matching
Sorniotti, Molva [150]	ICISC 09	U	✓	✗	BM	controlled matching
Kawai <i>et al.</i> [104]	ISPEC 09	U	✗	✗	BM	untrusted GAs
Sorniotti, Molva [151]	ICICS 10	U	✓	✗	BM	untr. GAs, cntr. matching
Section 3.2	ACNS 10	L	✓	✓	RSA	single untrusted GA
Section 5.4	ACNS 10	L	✓	✓	RSA	multiple groups/credentials
Section 5.5	WISTP 11	L	✓	✓	BM	multiple groups/credentials
Section 6.3	ASIACCS 11	L	✓	✓	RSA	multiple groups/credentials

¹{L, U, k } = {linkable, unlinkable, k -anonymous}; ²support for revocation: all linkable schemes are marked as revocable, although this property is not always explicitly considered in the corresponding publication; ³protocol offers secure key establishment (with forward secrecy); ⁴{BM, RSA, DL} = {bilinear map, RSA, discrete logarithm} based setting; ⁵scheme tolerates delayed update of revocation lists by the users

Table 1.1.: Comparison of published AHA schemes and those from Chapters 3–6

establishment of session keys [14, 46]. As a design principle of such *affiliation-hiding key agreement* [95] protocols, we note that affiliations are guaranteed to remain hidden even if established session keys are leaked to adversaries.

Linkable vs. unlinkable schemes

In *linkable* AHA protocols [9, 47, 94, 95, 97], sessions of the same participant can easily be linked together. Such protocols are useful if participants wish to be recognized across different sessions, and usually employ re-usable pseudonyms that members obtain during the registration process to the group. Hiding of affiliations is considered valuable nonetheless, and remains an explicit security goal of those protocols. Linkable schemes are typically deployed in settings where users are identified by pseudonyms anyway (e.g., in instant messaging or in online social networks).

In contrast, *unlinkable* AHA protocols [6, 96, 99, 150] prevent any correlation among sessions of the same participant. This property is explicitly covered by the corresponding security models. Unlinkable schemes are typically deployed when linkability is considered a privacy threat (see discussions in the context of identity escrow [106], electronic cash [52], and anonymous credentials [38]). Clearly, the property of unlinkability contradicts revocation handling via re-

vocation lists. This makes unlinkable schemes often more difficult to design and implement, in comparison to linkable schemes.

An intermediate form of linkability is given by the notion of k -anonymity [161, 162], for an adjustable parameter $k \in \mathbb{N}$. Intuitively, a scheme is k -anonymous if an adversary can infer from a session transcript only that the monitored protocol participant is one out of k certain users.

Linkable schemes can trivially be turned into unlinkable ones (as proposed by [9, 93]), by letting users obtain a set of different credentials upon registration to the group, and letting them use each credential for exactly one session. However, such one-time credentials seem to be unsuitable in practice, as honest users' credentials can easily be easily depleted by adversaries.

Two-party vs. multi-party authentication

The concept of protocols involving two users that mutually authenticate each other finds a natural extension in the multi-party setting, i.e., in *group secret handshakes* [93, 94, 155, 162]. Here, a set of multiple users engages in a joint protocol session, and the authentication is deemed successful if all participants provide valid credentials for the same group. While this goal is achieved in the linkable setting [93, 94] by combining 2-party AHA protocols with standard multi-party key agreement techniques [33], the challenging task in the unlinkable case is that participants have to ensure that all their peers are distinct. However, [155, 162] succeed in finding solutions to this problem.

Role-based AHA

In most linkable AHA schemes, sessions can be linked through participants' pseudonyms that are exchanged in clear at every protocol invocation. In this setting, *roles* are supplements to participants' pseudonyms and express the social function that participants are expected to play (e.g., *teacher*, *doctor*, *policeman*). Early linkable and unlinkable AHA schemes support roles (cf. [6, 9] and [47, Appendix A.1]). However, this concept turned out to be rather trivial, as there is no logical distinction between pseudonyms and roles (neither in the protocol specifications, nor in the security models). More recent schemes [94, 95, 97, 99, 150] do not consider roles any more.

1.4. Related concepts

We give an overview about different established cryptographic schemes and constructions that are related to privacy-aware authentication. Observe that none of them comes close to the service that affiliation-hiding authentication offers: Even in those schemes (e.g., anonymous credentials, identity escrow) where users hide behind

groups that are managed by designated authorities, the presented schemes' focus lies on hiding the identities of authenticating users, while the identities of the managing authorities do not remain hidden. In contrast, in AHA schemes, keeping GA's (but not necessarily user's) identity hidden is the main goal. Moreover, AHA offers joint and mutual authentication: users learn about the validity of peer's credentials only if they provide valid credentials themselves.

1.4.1. Ring signatures

Ring signatures, introduced by Rivest, Shamir, and Tauman [140], allow a member of an ad-hoc group of users to sign a given message on behalf of that group, without revealing its own identity (i.e., public key). The only information that verifiers learn about the signer is that it is part of the group. The signatures are *unforgeable*, meaning that generation of signatures without knowledge of at least one secret key corresponding to a public key in the group must be infeasible. Observe that ring signatures offer *privacy* to the signer since the latter remains *anonymous* (in the group). Moreover, most schemes are also *unlinkable*, i.e., verifiers are not able to determine whether two signatures were produced by the same signer. Ring signatures can be used, for instance, to leak insider information to the press in a manner that authenticates the source as a knowledgeable insider, yet protecting its identity.

1.4.2. Ring authentication, private authentication

Naor's *deniable ring authentication* [130] bases on the concept of ring signatures and focuses on interactive authentication of messages by a member of an ad-hoc group, such that corresponding transcripts are additionally deniable (i.e., computable by an appropriate simulator without knowledge of secret keys).

Similarly, Abadi [1] explores *private authentication*: here, apart from hiding themselves behind ad-hoc groups of users, provers can also limit the set of verifiers that are able to check the validity of created signatures.

1.4.3. Group signatures

In *group signatures* [12, 53], introduced by Chaum and van Heyst, users register with group authorities (GAs) to obtain individual signing keys. Using these keys, they can sign arbitrary messages on behalf of the group. Resulting signatures can be verified in respect to group's (constant-size) public key, independently of the specific signer. Two security properties besides *unforgeability* are important in the context of group signatures: the property of *unlinkability* demands that adversaries, given a signature on a message, cannot identify the specific signer that produced the signature. In contrast, *traceability* formalizes the requirement that GAs have the ability to *open* signatures to recover signer's identity.

1.4.4. Traceable signatures

Traceable signatures, introduced by Kiayias, Tsiounis, and Yung [105] (see also [16, 55, 80]) are a special type of group signatures with a slightly modified traceability concept: for each member U_i of the group, GA is in possession of a special *tracing trapdoor*. By revealing this trapdoor, all signatures issued by U_i can be identified/linked independently by other parties, called *tracing agents*. However, tracing agents do not learn the actual identity of U_i . Traceable signatures hence offer anonymity (through the group-based authentication approach) and unlinkability of signatures, while the latter can be revoked through publication of the appropriate trapdoor. The group manager is the only authority that can identify the actual signer U_i through the corresponding opening procedure, as in group signatures.

1.4.5. Identity escrow

Also closely related to group signatures is *identity escrow* [39, 106]. In this setting, users (interactively) authenticate to verifiers as members of groups administered by specific authorities. These authorities are able to admit and revoke memberships, and to reveal identities of users from recorded session transcripts. Identity escrow schemes are mostly build from group signature schemes [39, 106], and the main security properties, *unlinkability* and *traceability*, are similarly defined.

1.4.6. Anonymous signatures

Yang *et al.* [164] introduced *anonymous signatures* that aim at achieving anonymity in the traditional setting of digital signatures, where signer's private key is used to generate signatures that are verifiable using the corresponding public key. At first sight, achieving anonymity in this setting seems contradictory, as signatures can readily be linked to signers using the verification routine. The crucial observation in [164] is that a signature scheme's verification procedure requires three inputs: verification key, candidate signature, and the message. The hope is to keep the signer anonymous as long as the message is not disclosed to the adversary. This intuition was formalized in [75, 164], under the assumption that signed messages have sufficiently high entropy.

Interestingly, techniques underlying anonymous signature schemes for high-entropy messages have recently [17] been used in the construction of group signatures, to achieve an efficiency improvement over many earlier schemes.

1.4.7. Anonymous credentials

Anonymous credential systems (ACS, proposed by Chaum [50]) provide strong authentication of users to verifiers, while protecting privacy of the former. ACS users

receive, as their specific secrets, credentials from organizations. These credentials are typically provided through certificates that are issued on unique identifiers of the users, e.g., on the identities under which the users are known to the particular organizations.

The core functionality of ACS is to allow users to prove possession of valid credentials without revealing their (certified) identities. Early ACS [50, 60, 119] were not efficient or required additional (trusted) parties to assist the execution of ACS sessions [51]. Modern ACS, designed with both security and efficiency in mind, were proposed only in the last decade [38, 40], and improved and extended in [35–37, 41]. Some ACS [36, 37] offer efficient support for revocation of issued credentials. The challenge in this case is to allow revocation without compromising revoked users' privacy.

ACS have several important security properties: within others, credentials should be *unforgeable* to prevent (colluding) users from claiming possession of credentials that were never issued; ACS should be *unlinkable* to guarantee user's privacy against third-party verifiers and colluding organizations. These requirements are satisfied by all modern ACS. Collusion-resistance is especially important for ACS that additionally support certification of *attributes* [35]. Attributes, such as age or address information, may allow for a finer form of access control. In contrast to group signatures, ACS lack the ability of organizations to identify/trace users based on recorded ACS executions.

Key agreement where authentication is based on anonymous credentials is explored in *credential-authenticated key exchange* [34].

1.4.8. Predicate-based signatures and key agreement

In *attribute-based signatures* (Li *et al.* [116]), users obtain certificates for (sets of) attributes from a designated attribute-certification authority. Generated signatures are verifiable in respect to policies consisting of expected attribute sets, independently of signer's identity. *Predicate-based signatures* (Maji *et al.* [120]) generalize this concept towards complex predicates on attributes, and allow for modeling more fine-grained access control. As an application of both attribute- and predicate-based signatures, *predicate-based key agreement* (Birkett and Stebila [19]) uses these methods to authenticate users for key agreement. Besides the classical goal of *session key security*, predicate-based key agreement offers *credential privacy*. This security property demands that nobody is able to distinguish between two users whose credentials satisfy the same predicate, even if they have different credentials. Observe that this notion covers both *anonymity* and *unlinkability* of users.

Prerequisites and building blocks

We briefly recall some well-known facts about standard cryptographic primitives and useful structures in number theory and algebra. In addition, we formalize hardness assumptions and instantiate building blocks on which we base our protocol designs in later chapters. We also give an exposition of important design ideas that drive some affiliation-hiding authentication (AHA) schemes that were reported in the literature so far. These will also serve as a foundation of our constructions in Chapters 3–7.

2.1. Mathematical background

2.1.1. Number theory

Although we assume that the reader is generally familiar with the concepts of basic number theory, we recall some important results needed in this thesis. We refer to [127] for a more detailed exposition.

A *safe prime* p is a prime number such that $p = 2p' + 1$, where p' is prime as well. For a safe prime p , the multiplicative group \mathbb{Z}_p^\times of the finite field $\mathbb{Z}_p \cong GF(p)$ has order $p - 1 = 2p'$, and each of its subgroups has order 1, 2, p' or $2p'$ (by Lagrange's theorem). The subgroup of order p' consists exactly of all squares in \mathbb{Z}_p^\times , it is hence called the subgroup of *quadratic residues mod p* , $QR(p)$ for short. Note that $QR(p)$ is generated by each square in \mathbb{Z}_p^\times , except by 1, and that $2|QR(p)| = |\mathbb{Z}_p^\times|$. Note also that about every second element g in \mathbb{Z}_p^\times is primitive, i.e., $\langle g \rangle_p = \mathbb{Z}_p^\times$.

For any prime p , the *Legendre symbol* $\left(\frac{\cdot}{p}\right) : \mathbb{Z}_p^\times \rightarrow \{-1, 1\}$ is defined by $\left(\frac{a}{p}\right) = 1 \Leftrightarrow a \in QR(p)$. By considering $\{-1, 1\} = \mathbb{Z}_3^\times$, this mapping becomes a group homomorphism. Observe $\ker\left(\frac{\cdot}{p}\right) = QR(p)$. It is known that $\left(\frac{-1}{p}\right) = -1$ for all $p = 3 \pmod{4}$, that $\left(\frac{2}{p}\right) = -1$ for all $p = 3 \pmod{8}$, and that $\left(\frac{2}{p}\right) = 1$ for all $p = 7 \pmod{8}$.

Euler's totient function $\varphi : \mathbb{N} \rightarrow \mathbb{N}$; $m \mapsto \varphi(m)$ indicates the number of invertible elements in \mathbb{Z}_m , i.e., $\varphi(m) = |\mathbb{Z}_m^\times|$. The related *Carmichael function* $\lambda : \mathbb{N} \rightarrow$

\mathbb{N} ; $m \mapsto \lambda(m)$ indicates the order of the largest cyclic subgroup in \mathbb{Z}_m^\times . Both functions can easily be computed if the factorization of its argument is known. In particular, if $n = pq$ is an RSA modulus, i.e., p, q are prime numbers, then $\varphi(n) = (p-1)(q-1)$ and $\lambda(n) = \text{lcm}(p-1, q-1)$. If n is moreover a *safe RSA modulus*, i.e., $p = 2p'+1$ and $q = 2q'+1$ are safe primes, then we have $\lambda(n) = 2p'q' = \varphi(n)/2$.

The *Chinese Remainder Theorem* (CRT) states that rings \mathbb{Z}_{pq} and $\mathbb{Z}_p \times \mathbb{Z}_q$ are isomorphic, for all primes p, q . We denote this by $\mathbb{Z}_{pq} \cong \mathbb{Z}_p \times \mathbb{Z}_q$. The corresponding (ring) isomorphism is given by $a_n \mapsto (a_n \bmod p, a_n \bmod q)$, and its inverse by $(a_p, a_q) \mapsto a_p + ph$ for $h = (a_q - a_p)/p \pmod{q}$. Note that it follows that groups \mathbb{Z}_{pq}^\times and $\mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$ are isomorphic as well, i.e., $\mathbb{Z}_{pq}^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$.

Let $n = pq$ be a safe RSA modulus. In particular, n is a Blum integer, i.e., $p = q = 3 \pmod{4}$, and it follows that $-1 \notin QR(p)$ and $-1 \notin QR(q)$. Consider an element $g \in \mathbb{Z}_n^\times$ that is primitive in $QR(p)$ but is not primitive in $QR(q)$ (or vice versa). Then $\text{ord}_n(g) = 2p'q' = \lambda(n)$ and $-1 \notin \langle g \rangle_n$ hold. In this case, we have $\mathbb{Z}_n^\times \cong \langle -1 \rangle_n \times \langle g \rangle_n$. A simple combinatorial argument shows that this property holds for about a half of the elements in \mathbb{Z}_n^\times .

2.1.2. Cryptography in cyclic groups

By (\mathcal{G}, g, q) we denote the setting where $\mathcal{G} = \langle g \rangle$ is a cyclic group of prime order q . We further denote algorithms that, on input security parameter 1^κ , generate (specifications of) such groups by $\text{GGen}(1^\kappa)$. It is easy to see that hardness of any of the problems given below implies that $q = q(\kappa)$ grows super-polynomially in κ .

Some well-known hardness assumptions defined in the cyclic group setting base on the Discrete Logarithm Problem (DLP), and the Computational and Decisional Diffie-Hellman problems (CDH and DDH, respectively). We briefly describe the adversary's task:

DLP: given g, g^x for $x \in \mathbb{Z}_q$, compute x

CDH: given g, g^x, g^y for $x, y \in \mathbb{Z}_q$, compute g^{xy}

DDH: given g, g^x, g^y, g^z for $x, y, z \in \mathbb{Z}_q$, decide whether $z = xy \pmod{q}$

The less known GapDH assumption states that the CDH problem stays hard even when the adversary is equipped with a DDH oracle. As the DLP and DDH problems are only marginally referred to in this thesis, we restrict ourselves to define just the CDH problem in full detail:

Definition 1 (CDH assumption) *Let GGen denote a (possibly deterministic) algorithm that outputs descriptions (\mathcal{G}, g, q) of cyclic groups. For an adversary \mathcal{A} , consider the experiment from Figure 2.1. Define \mathcal{A} 's success probability as*

$$\text{Succ}_{\text{GGen}, \mathcal{A}}^{\text{cdh}}(\kappa) = \Pr \left[\text{Expt}_{\text{GGen}, \mathcal{A}}^{\text{cdh}}(\kappa) = 1 \right] .$$

The CDH assumption states that there exists an algorithm GGen such that $\text{Succ}_{\text{GGen}, \mathcal{A}}^{\text{cdh}}$ is negligible for all efficient adversaries \mathcal{A} .

- $\text{Expt}_{\text{GGen}, \mathcal{A}}^{\text{cdh}}(\kappa)$:
- (a) $(\mathcal{G}, g, q) \leftarrow \text{GGen}(1^\kappa)$
 - (b) $x, y \leftarrow_R \mathbb{Z}_q$
 - (c) $h \leftarrow \mathcal{A}(\mathcal{G}, g, q, g^x, g^y)$
 - (d) Return 1 iff $h = g^{xy}$.

Figure 2.1.: cdh experiment

For each of the named problems (DLP, CDH, DDH), there exist candidate groups (\mathcal{G}, g, q) in which the corresponding assumptions are conjectured to hold. Classical examples [21, 86, 127] are given by prime order subgroups of \mathbb{Z}_p^\times , for primes p , and by subgroups of the set of rational points on elliptic curves, defined over finite fields. In particular, the group $QR(p)$ of quadratic residues modulo a safe prime p is a group in which all three problems are assumed to be hard to solve [24]. Indeed, in Definition 2, we capture the assumption that CDH is difficult in *all* such groups, provided that safe prime p has a certain minimal length.

Definition 2 (SCDH assumption) For an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, consider the experiment from Figure 2.2. Define \mathcal{A} 's success probability as

$$\text{Succ}_{\mathcal{A}}^{\text{scdh}}(\kappa) = \Pr \left[\text{Expt}_{\mathcal{A}}^{\text{scdh}}(\kappa) = 1 \right] .$$

The SCDH assumption states that $\text{Succ}_{\mathcal{A}}^{\text{scdh}}$ is negligible for all efficient adversaries \mathcal{A} .

- $\text{Expt}_{\mathcal{A}}^{\text{scdh}}(\kappa)$:
- (a) $(p, g, \text{state}) \leftarrow \mathcal{A}_1(1^\kappa)$
 - (b) If p is not a safe prime of length κ : return 0.
 - (c) If $\langle g \rangle_p \neq QR(p)$: return 0.
 - (d) $x, y \leftarrow_R \mathbb{Z}_{(p-1)/2}$
 - (e) $h \leftarrow \mathcal{A}_2(\text{state}, g^x \bmod p, g^y \bmod p)$
 - (f) Return 1 iff $h = g^{xy} \bmod p$.

Figure 2.2.: scdh experiment

2.1.3. Cryptography in composite order groups

Composite order groups are of the form \mathbb{Z}_n^\times , where modulus n is not prime. The intrinsic property of these groups is that their order is generally difficult to determine, unless some trapdoor information is known (namely, the factorization of n). The most prominent example is given by \mathbb{Z}_n^\times , for RSA moduli $n = pq$.

In this thesis, however, we will exclusively deploy *safe RSA moduli*, i.e., we assume that both factors of n are safe primes. As noted already in Section 2.1.1, the order of \mathbb{Z}_n^\times is then given by $\varphi(n) = 2\lambda(n) = 4p'q'$. Observe that, for any given element $m \in \mathbb{Z}_n \setminus \mathbb{Z}_n^\times$, a non-trivial factor of n is given by $\gcd(m, n)$. Hence, picking elements at random from \mathbb{Z}_n will yield non-invertible elements only with negligible probability, assuming hardness of the factorization problem.

We formalize the RSA assumption in the setting of safe RSA moduli. Note that condition (d) on SRSA-GEN algorithm is generally rarely needed, but required when dealing with RSA moduli that are provided by malicious users (cf. Chapter 3).

Definition 3 (RSA assumption on safe moduli) *Let SRSA-GEN be an efficient algorithm that, on input security parameter 1^κ , outputs tuples (n, e, d) such that (a) $n = pq$ for primes p and q , (b) $p = 2p' + 1$ and $q = 2q' + 1$ for primes p' and q' , (c) $e, d \in \mathbb{Z}_{\varphi(n)}^\times$ with $ed = 1 \pmod{\varphi(n)}$, and (d) $[n]_2 = \kappa$ and $[p]_2 \approx \kappa/2 \approx [q]_2$. The success probability of an adversary \mathcal{A} with respect to SRSA-GEN is defined as*

$$\text{Succ}_{\text{SRSA-GEN}, \mathcal{A}}^{\text{srsa}}(\kappa) = \Pr [\text{Expt}_{\text{SRSA-GEN}, \mathcal{A}}^{\text{srsa}}(\kappa) = 1] \quad ,$$

where $\text{Expt}_{\text{SRSA-GEN}, \mathcal{A}}^{\text{srsa}}$ is defined in Figure 2.3. The RSA assumption on safe moduli states that there exists an algorithm SRSA-GEN such that $\text{Succ}_{\text{SRSA-GEN}, \mathcal{A}}^{\text{srsa}}$ is negligible for all efficient adversaries \mathcal{A} .

$\text{Expt}_{\text{SRSA-GEN}, \mathcal{A}}^{\text{srsa}}(\kappa)$:

- (a) $(n, e, d) \leftarrow_R \text{SRSA-GEN}(1^\kappa)$
- (b) $z \leftarrow_R \mathbb{Z}_n^\times$
- (c) $m \leftarrow \mathcal{A}(n, e, z)$
- (d) Return 1 iff $m^e = z$.

Figure 2.3.: srsa experiment

2.1.4. Cryptography in the pairing-based setting

Let $\mathcal{G} = \langle g \rangle$ and $\mathcal{G}_T = \langle g_T \rangle$ be cyclic groups of prime order q . A *pairing* is an efficient bilinear map $\hat{e} : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}_T$ such that $\hat{e}(g^a, g^b) = g_T^{ab}$ for all $a, b \in \mathbb{Z}_q$ (see also [22, Chapter X]). This setting is usually called a TYPE I setting [79], for

which efficient constructions are known [145, 146]. The (computational) Bilinear Diffie-Hellman assumption [25] states that there exist bilinear groups such that the BDH problem is hard:

BDH: given g, g^x, g^y, g^z for $x, y, z \in \mathbb{Z}_q$, compute g_T^{xyz} .

Note that pairings $\hat{e} : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}_T$ are always *symmetric*, i.e., $\hat{e}(h_1, h_2) = \hat{e}(h_2, h_1)$ for all $h_1, h_2 \in \mathcal{G}$, as the simple observation $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab} = \hat{e}(g^b, g^a)$ shows.

2.2. Digital signature schemes

Digital signatures belong to the most fundamental primitives in public key cryptography. They will play explicit and implicit roles in the protocols we design in later sections. Here, we define syntax and security properties, and recall two important instantiations.

Definition 4 (Digital signature scheme) A digital signature scheme Σ is a set $\Sigma = \{\text{KGen}, \text{Sign}, \text{Verify}\}$ of three efficient algorithms:

$\text{KGen}(1^\kappa)$

On input of security parameter 1^κ , this algorithm outputs a secret signing key sk and a verification key pk .

$\text{Sign}(\text{sk}, m)$

On input signer's secret key sk and message $m \in \{0, 1\}^*$, this algorithm outputs a signature σ .

$\text{Verify}(\text{pk}, m, \sigma)$

On input verification key pk , message $m \in \{0, 1\}^*$, and candidate signature σ , this algorithm either accepts or rejects, i.e., outputs either **true** or **false**.

Definition 5 (Correctness of signature schemes) A digital signature scheme Σ is correct if for all $\kappa \in \mathbb{N}$, all $(\text{sk}, \text{pk}) \leftarrow \text{KGen}(1^\kappa)$, all messages $m \in \{0, 1\}^*$, and all signatures $\sigma \leftarrow \text{Sign}(\text{sk}, m)$, we have $\text{Verify}(\text{pk}, m, \sigma) = \text{true}$.

The principal security property of signature schemes is unforgeability. This notion is formalized as follows:

Definition 6 (Existential unforgeability, EUF-CMA) A digital signature scheme Σ is existentially unforgeable under adaptive chosen-message attacks (EUF-CMA) if for all efficient adversaries \mathcal{A} the success probability

$$\text{Succ}_{\Sigma, \mathcal{A}}^{\text{euf-cma}}(\kappa) = \Pr \left[\text{Expt}_{\Sigma, \mathcal{A}}^{\text{euf-cma}}(\kappa) = 1 \right]$$

is negligible, where $\text{Expt}_{\Sigma, \mathcal{A}}^{\text{euf-cma}}$ is the experiment specified in Figure 2.4.

- $\text{Expt}_{\Sigma, \mathcal{A}}^{\text{euf-cma}}(\kappa)$:
- (a) $(\text{sk}, \text{pk}) \leftarrow_R \text{KGen}(1^\kappa)$
 - (b) $(m, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}_S}(\text{pk})$
 - If \mathcal{A} queries $\mathcal{O}_S(m)$:
 - (a) $\sigma \leftarrow \text{Sign}(\text{sk}, m)$
 - (b) Append m to SList.
 - (c) Answer \mathcal{A} with σ .
 - (c) Return 0 if $m \in \text{SList}$.
 - (d) Return 1 iff $\text{Verify}(\text{pk}, m, \sigma) = \text{true}$.

Figure 2.4.: euf-cma experiment

2.2.1. Schnorr's signature scheme

We sketch the signature scheme by Schnorr [143, 144]. Its EUF-CMA security can be proven in the random oracle model, if the scheme is instantiated over a group where DLP is hard [137].

$\text{KGen}(1^\kappa)$

Run $(\mathcal{G}, g, q) \leftarrow \text{GGen}(1^\kappa)$ (cf. Section 2.1.2) and specify a hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_q$. We assume that these parameters are implicitly known to the signer and all verifiers. Pick a random element $x \leftarrow_R \mathbb{Z}_q \setminus \{0\}$, and output signing key $\text{sk} = x$ and verification key $\text{pk} = g^x$.

$\text{Sign}(\text{sk}, m)$

To sign message $m \in \{0, 1\}^*$, pick $r \leftarrow_R \mathbb{Z}_q$ and compute $\omega \leftarrow g^r$ and $t \leftarrow r + xH(\omega \| m)$. Output signature $\sigma = (\omega, t)$.

$\text{Verify}(\text{pk}, m, \sigma)$

Candidate signature $\sigma = (\omega, t)$ on message $m \in \{0, 1\}^*$ is accepted under verification key $\text{pk} = g^x$ iff $g^t = \omega \text{pk}^{H(\omega \| m)}$.

2.2.2. Full-domain hash RSA signature scheme

The FDH-RSA signature scheme is provably EUF-CMA-secure, in the random oracle model, provided that the RSA assumption holds [15].

$\text{KGen}(1^\kappa)$

Run $\text{SRSA-GEN}(1^\kappa)$ to obtain a tuple (n, e, d) of RSA parameters (cf. Definition 3). Specify a hash function $H_n : \{0, 1\}^* \rightarrow \mathbb{Z}_n$. Output signing key $\text{sk} = (n, d, H_n)$ and verification key $\text{pk} = (n, e, H_n)$.

$\text{Sign}(\text{sk}, m)$

To sign message $m \in \{0, 1\}^*$, compute and output signature $\sigma \leftarrow H_n(m)^d \bmod n$.

$\text{Verify}(\text{pk}, m, \sigma)$

Candidate signature σ on message $m \in \{0, 1\}^*$ is accepted under verification key $\text{pk} = (n, e, H_n)$ iff $\sigma^e \bmod n = H_n(m)$.

A blind version of this signature scheme was introduced by Chaum [49] (see also the specification of `AddUser` protocol in Section 3.2.1), but its security reduction requires a non-standard interactive ‘one-more’ assumption [13].

2.3. Non-interactive key distribution

In a multi-user setting, the purpose of a *non-interactive key distribution scheme* (NIKDS) [22, 71, 135, 141] is the assignment of a (fixed) symmetric key to each pair of users. The intrinsic property and advantage of NIKDS over (authenticated) key establishment protocols is that NIKDS are non-interactive, i.e., users can compute the particular keys shared with other users without any (prior) communication with them.

Typically, NIKDS are identity-based schemes where the identities may be arbitrary strings. In NIKDS, users first register their particular identity $\text{id} \in \{0, 1\}^*$ with an authority called *key generation center* (KGC) to obtain their specific credential $\text{sk}[\text{id}]$. Using this credential, they can compute a secure key shared between id and id' , for any other identity $\text{id}' \in \{0, 1\}^*$ and without any further communication. We formalize this in Definition 7.

Definition 7 (NIKDS) *A non-interactive key distribution scheme is a set $\text{NIKDS} = \{\text{NSetup}, \text{NRegister}, \text{NGetKey}\}$ of three efficient algorithms:*

$\text{NSetup}(1^\kappa)$

This algorithm is used to initialize a KGC. On input of security parameter 1^κ , it outputs a master secret key msk .

$\text{NRegister}(\text{msk}, \text{id})$

On input KGC’s master secret key msk and identity $\text{id} \in \{0, 1\}^$, this algorithm outputs credential $\text{sk}[\text{id}]$.*

$\text{NGetKey}(\text{sk}[\text{id}], \text{id}')$

On input credential $\text{sk}[\text{id}]$ and identity $\text{id}' \in \{0, 1\}^$, this algorithm outputs a key $K \in \{0, 1\}^\kappa$.*

We expect from a NIKDS that, if two users with identities id and id' compute $\text{NGetKey}(\text{sk}[\text{id}], \text{id}')$ and $\text{NGetKey}(\text{sk}[\text{id}'], \text{id})$, respectively, then the resulting keys always match. We consider this key as a *shared key* between parties id and id' , and, for convenience, denote it by $\text{NSharedKey}(\text{msk}; \text{id}, \text{id}')$. Observe that (registered) users with identities id, id' do not need KGC's key msk to compute $\text{NSharedKey}(\text{msk}; \text{id}, \text{id}')$.

Definition 8 (Correctness of NIKDS) *A NIKDS is correct if for all $\kappa \in \mathbb{N}$, all $\text{msk} \leftarrow \text{NSetup}(1^\kappa)$, all identities $\text{id}_1, \text{id}_2 \in \{0, 1\}^*$, all $\text{sk}[\text{id}_1] \leftarrow \text{NRegister}(\text{msk}, \text{id}_1)$ and $\text{sk}[\text{id}_2] \leftarrow \text{NRegister}(\text{msk}, \text{id}_2)$, and all $K_1 \leftarrow \text{NGetKey}(\text{sk}[\text{id}_1], \text{id}_2)$ and $K_2 \leftarrow \text{NGetKey}(\text{sk}[\text{id}_2], \text{id}_1)$, we have $K_1 = K_2$.*

The security notions we present next adopt the classical one-wayness and key indistinguishability requirements of interactive key agreement [14, 23, 28] to the (deterministic) non-interactive setting.

Definition 9 (OW-CIA security of NIKDS) *A NIKDS is one-way secure under adaptive chosen-identity attacks (OW-CIA) if for all efficient adversaries \mathcal{A} the success probability*

$$\text{Succ}_{\text{NIKDS}, \mathcal{A}}^{\text{ow-cia}}(\kappa) = \Pr [\text{Expt}_{\text{NIKDS}, \mathcal{A}}^{\text{ow-cia}}(\kappa) = 1]$$

is negligible, where $\text{Expt}_{\text{NIKDS}, \mathcal{A}}^{\text{ow-cia}}$ is the experiment specified in Figure 2.5.

Definition 10 (IND-CIA security of NIKDS) *A NIKDS is indistinguishable under adaptive chosen-identity attacks (IND-CIA) if for all efficient adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ the advantage*

$$\text{Adv}_{\text{NIKDS}, \mathcal{A}}^{\text{ind-cia}}(\kappa) = \left| \Pr [\text{Expt}_{\text{NIKDS}, \mathcal{A}}^{\text{ind-cia}, 0}(\kappa) = 1] - \Pr [\text{Expt}_{\text{NIKDS}, \mathcal{A}}^{\text{ind-cia}, 1}(\kappa) = 1] \right|$$

is negligible, where $\text{Expt}_{\text{NIKDS}, \mathcal{A}}^{\text{ind-cia}, b}$ is the experiment specified in Figure 2.5.

2.3.1. Construction of NIKDS based on bilinear maps

The first efficient NIKDS was constructed by Sakai, Ohgishi, and Kasahara in [141] (although the notion of NIKDS was introduced to cryptography about 20 years earlier, in [147]). We sketch their scheme:

NSetup(1^κ)

Specify cyclic groups $\mathcal{G} = \langle g \rangle$ and \mathcal{G}_T of prime order q , for which an efficient bilinear pairing $\hat{e} : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}_T$ is known (see Section 2.1.4). In addition, specify hash functions $H : \{0, 1\}^* \rightarrow \mathcal{G}$ and $H_1 : \mathcal{G}_T \rightarrow \{0, 1\}^\ell$, for fixed $\ell \in \mathbb{N}$. Randomly pick $s \leftarrow_R \mathbb{Z}_q \setminus \{0\}$ and return master secret key $\text{msk} = s$.

$\text{Exp}_{\text{NIKDS}, \mathcal{A}}^{\text{ow-cia}}(\kappa):$ <ul style="list-style-type: none"> (a) $\text{msk} \leftarrow \text{NSetup}(1^\kappa)$ (b) $(K, \text{id}_1, \text{id}_2) \leftarrow \mathcal{A}^{\mathcal{O}_R}(1^\kappa)$ <ul style="list-style-type: none"> – If \mathcal{A} queries $\mathcal{O}_R(\text{id})$: <ul style="list-style-type: none"> (a) $\text{sk}[\text{id}] \leftarrow \text{NRegister}(\text{msk}, \text{id})$ (b) Append id to RList. (c) Answer \mathcal{A} with $\text{sk}[\text{id}]$. (c) Return 0 if $\text{id}_1 \in \text{RList}$ or $\text{id}_2 \in \text{RList}$. (d) Return 1 iff $K = \text{NSharedKey}(\text{msk}; \text{id}_1, \text{id}_2)$. 	$\text{Exp}_{\text{NIKDS}, \mathcal{A}}^{\text{ind-cia}, b}(\kappa):$ <ul style="list-style-type: none"> (a) $\text{msk} \leftarrow \text{NSetup}(1^\kappa)$ (b) $(\text{id}_1, \text{id}_2, \text{state}) \leftarrow \mathcal{A}_1^{\mathcal{O}_R}(1^\kappa)$ <ul style="list-style-type: none"> – If \mathcal{A} queries $\mathcal{O}_R(\text{id})$: <ul style="list-style-type: none"> (a) $\text{sk}[\text{id}] \leftarrow \text{NRegister}(\text{msk}, \text{id})$ (b) Append id to RList. (c) Answer \mathcal{A} with $\text{sk}[\text{id}]$. (c) $K_0 \leftarrow_R \{0, 1\}^\kappa$ (d) $K_1 \leftarrow \text{NSharedKey}(\text{msk}; \text{id}_1, \text{id}_2)$ (e) $b' \leftarrow \mathcal{A}_2^{\mathcal{O}_R}(\text{state}, K_b)$ <ul style="list-style-type: none"> – Answer \mathcal{O}_R queries as above. (f) Return 0 if $\text{id}_1 \in \text{RList}$ or $\text{id}_2 \in \text{RList}$. (g) Return b'.
--	--

Figure 2.5.: OW-CIA and IND-CIA experiments for NIKDS

NRegister(msk, id)

On input master secret key $\text{msk} = s$ and identity $\text{id} \in \{0, 1\}^*$, credential $\text{sk}[\text{id}]$ is computed as $\text{sk}[\text{id}] \leftarrow H(\text{id})^s$.

NGetKey(sk[id], id')

This algorithm outputs key $K = H_1(\hat{e}(\text{sk}[\text{id}], H(\text{id}')))$.

Proof of correctness. For arbitrary $\text{id}_1, \text{id}_2 \in \{0, 1\}^*$, let $h_1 \leftarrow H(\text{id}_1)$ and $h_2 \leftarrow H(\text{id}_2)$. We then have $\text{sk}[\text{id}_1] = h_1^s$ and $\text{sk}[\text{id}_2] = h_2^s$, and correctness is implied by $\hat{e}(h_1^s, h_2) = \hat{e}(h_1, h_2)^s = \hat{e}(h_2, h_1)^s = \hat{e}(h_2^s, h_1)$. \square

Security of this NIKDS construction was determined in [71, 135] as follows:

Theorem 1 (Security of pairing-based NIKDS) *The NIKDS proposed above is OW-CIA-secure and IND-CIA-secure under the Bilinear Diffie-Hellman assumption (cf. Section 2.1.4), in the random oracle model.*

Observe that, in the NGetKey algorithm, user's first input element to the pairing operation is a fixed long-term parameter, namely credential $\text{sk}[\text{id}]$. This can be exploited to obtain efficient implementations: see [57, 145] for considerable optimizations on fixed-argument pairing evaluations.

2.4. Syntax and Correctness of AHA

Many different definitions of AHA (also known under the names of *secret handshakes* or *affiliation-hiding key exchange*) have been proposed in the literature [6, 9, 47, 95–97, 99, 150, 161]. This variety is a natural consequence of the fact that particular

schemes have specific properties, e.g., some are pure authentication protocols, while others also establish and output shared keys.

The following syntactical definition reflects the basic principles of (linkable) AHA and is general enough to cover most settings found in the literature. However, as the scenarios considered in subsequent Chapters 3 and 5 deviate considerably from this classical setting, we will have to slightly adapt the syntax to fit the new requirements. We defer this step to the particular sections.

Definition 11 (Affiliation-hiding authentication) *An affiliation-hiding authentication scheme is a set $\text{AHA} = \{\text{CreateGroup}, \text{AddUser}, \text{Handshake}, \text{Revoke}\}$ of four efficient algorithms and protocols:*

CreateGroup(1^κ)

This algorithm is executed by a GA to set up a new group G . On input of security parameter 1^κ , it generates a private key $G.\text{sk}$ and initializes the group's pseudonym revocation list $G.\text{prl}$ to \emptyset . The algorithm outputs revocation list $G.\text{prl}$ along with private key $G.\text{sk}$.

AddUser(G, id)

This algorithm is executed by GA of group G to admit a new user U to the group. On input secret key $G.\text{sk}$ and user's pseudonym id , a membership credential $\text{sk}_G[\text{id}]$ is computed and given to the user, enabling the latter to authenticate with pseudonym id in group G in future Handshake sessions. The communication channel between U and GA is assumed to be authentic.

Observe that our definition allows users to have several pseudonyms registered in the same group, and to have the same pseudonym registered in different groups. In case this is considered undesirable, GAs should be equipped with suitable admission policies.

Handshake($U_1 \leftrightarrow U_2$)

This protocol is executed between two users, U_1 and U_2 . User U_i , $i \in \{1, 2\}$, provides as input parameters $\text{params}_i = (\text{id}_i, \text{sk}_{G_i}[\text{id}_i], G_i.\text{prl}, r_i)$ and executes its individual part $\text{Handshake}'(\text{params}_i)$. It is expected that $(\text{id}_i, \text{sk}_{G_i}[\text{id}_i])$ is a valid pseudonym/credential pair for group G_i , obtained via AddUser algorithm. By $G_i.\text{prl}$ we denote the pseudonym revocation list of respective group G_i . Finally, we require $r_i \in \{\text{init}, \text{resp}\}$.

The protocol shall detect whether both users provide credentials for the same group (i.e., whether $G_1 = G_2$). If this is the case, the protocol shall accept with an established shared session key. Otherwise, it shall reject.

Users keep track of the state of created Handshake protocol sessions π through session variables that are initialized as follows: $\pi.\text{state} \leftarrow \text{running}$, $\pi.\text{id} \leftarrow$

id (where id is taken from params_i), $\pi.\text{key} \leftarrow \perp$, and $\pi.\text{partner} \leftarrow \perp$. At some point, the protocol completes and $\pi.\text{state}$ is updated to either *rejected* or *accepted*. In the latter case, $\pi.\text{key}$ is set to the established session key (of length κ) and the pseudonym of the *Handshake* partner is assigned to $\pi.\text{partner}$. State *accepted* cannot be reached if the protocol partner is revoked (i.e., $\pi.\text{partner} \in G.\text{prl}$).

$\text{Revoke}(G, \text{id})$

This is the revocation algorithm executed by GA of group G . It outputs the updated pseudonym revocation list $G.\text{prl} \leftarrow G.\text{prl} \cup \{\text{id}\}$.

Note that CreateGroup algorithm does not output a public key for created group G . The intuition behind this is that all parameters otherwise comprehended as ‘public keys’ play a role only for group insiders, i.e., for admitted users that have received their individual credential material. In the above definition, we may hence assume that all these ‘public’ parameters are embedded in users’ credentials.

We require from a useful AHA scheme that if two registered users run a *Handshake* honestly and without any interference by others, then the corresponding sessions accept and output the same session key. This idea is captured more precisely in Definition 12.

Definition 12 (Correctness of AHA) *Suppose that two users, U_1 and U_2 , register as members of groups G_1 and G_2 , respectively, and obtain their credentials $(\text{id}_1, \text{sk}_{G_1}[\text{id}_1])$ and $(\text{id}_2, \text{sk}_{G_2}[\text{id}_2])$ via corresponding AddUser executions. Further suppose that U_1 and U_2 use these credentials to engage in a *Handshake* protocol. Let π_1 and π_2 denote the corresponding sessions. The AHA scheme is correct if (a) π_1 and π_2 complete in the same state which is *accepted* iff $G_1 = G_2$ and $\text{id}_1 \notin G_2.\text{prl}$ and $\text{id}_2 \notin G_1.\text{prl}$ and $r_1 \neq r_2$, and (b) if both sessions accept, then $(\pi_1.\text{key}, \pi_1.\text{partner}, \pi_1.\text{id}) = (\pi_2.\text{key}, \pi_2.\text{id}, \pi_2.\text{partner})$.*

2.5. Basic AHA constructions

In the following sections, we expose in greater detail some pioneering AHA constructions found in the literature [9, 47, 95]. The presented schemes cover the most common cryptographic settings (cf. Sections 2.1.2–2.1.4): [95] is based on the RSA cryptosystem, [47] is defined in a cyclic group of prime order, while [9] requires availability of an efficient bilinear map.

2.5.1. AHA based on RSA

The scheme proposed in [95] by Jarecki, Kim, and Tsudik is RSA-based and builds on Okamoto’s identity-based key establishment protocol [81, 132, 133]. The idea to

turn Okamoto's protocol into an AHA scheme came initially from Vergnaud [156]. However, his scheme turned out to be flawed [95]. We describe a simplified version of [95]:

CreateGroup

To set up a group G , corresponding GA generates fresh parameters (n, g, e) , where n is a safe RSA modulus of length κ (where κ is the security parameter), $e \in \mathbb{Z}_{\varphi(n)}^\times$ is an RSA exponent, and $g \in \mathbb{Z}_n^\times$ is an element satisfying $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$ (i.e., $\text{ord}_n(g) \approx n/2$). See Sections 2.1.1 and 2.1.3 for more details about this setting. GA also specifies hash functions $H_n : \{0, 1\}^* \rightarrow \mathbb{Z}_n^\times$ and $H_1 : \{0, 1\}^* \rightarrow \{0, 1\}^\ell$, for some fixed $\ell = \ell(\kappa)$. Moreover, let $T = T(\kappa) \gg n$ be independent of n , e.g., $T = 2^{\kappa+\ell}$.

AddUser

The credential $\text{sk}_G[\text{id}] = (n, g, e, \sigma_{\text{id}})$ corresponding to pseudonym $\text{id} \in \{0, 1\}^*$ in group G consists of parameters (n, g, e) and the RSA signature $\sigma_{\text{id}} = H_n(\text{id})^d \bmod n$ on the full-domain hash of id (where $d = e^{-1} \bmod \varphi(n)$, cf. Section 2.2.2).

Handshake

The protocol is sketched in Figure 2.6. By `pad` we denote a probabilistic algorithm that maps its first argument θ' to an element θ within interval $[0, T - 1]$ such that $\theta = \theta' \pmod{n}$. For concreteness, let `pad` map θ' to $\theta = \theta' + kn$, where $k \leftarrow_R [0, \lfloor T/n \rfloor - 1]$. Protocol's correctness follows from $r_A = g^{2e_A x_A x_B} = r_B$, which holds if both participants deploy valid credentials and consistent group parameters (n, g, e) :

$$\begin{aligned} r_A &= ((\theta'_B)^{2e_A} H_{n_A}(\text{id}_B)^{-2})^{x_A} = ((g_B)^{2e_A x_B} (\sigma_{\text{id}_B})^{2e_A} H_{n_A}(\text{id}_B)^{-2})^{x_A} \quad (2.1) \\ &= \left((g_B)^{2e_A x_B} H_{n_B}(\text{id}_B)^{2e_A d_A} H_{n_A}(\text{id}_B)^{-2} \right)^{x_A} = (g_B)^{2e_A x_A x_B} \pmod{n_A} \end{aligned}$$

Note that, in this computation, we assume that $H_n(\text{id})$ is invertible \pmod{n} for all used pseudonyms id . However, the case that $H_n(\text{id})$ hits an element from $\mathbb{Z}_n \setminus \mathbb{Z}_n^\times$, in which the protocol would fail, occurs only with negligible probability, as we argue in Section 2.1.3.

Observe that hash function H_1 is used for both key derivation and key confirmation.

A comparison of this protocol with the original scheme by Okamoto [132, 133] reveals that Jarecki *et al.* consider trusted authorities from [132] as GAs in [95]. However, in order to achieve the property of affiliation-hiding, the authors introduce two essential modifications: First, RSA modulus n and generator g are now chosen such that $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$. Under this premise, we observe that the multiplicative

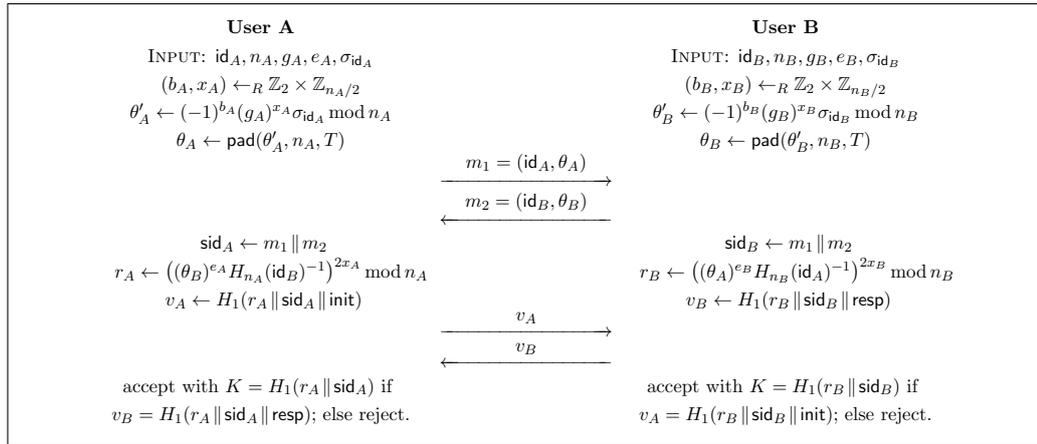


Figure 2.6.: RSA-based Handshake protocol from [95] (simplified)

blinding of intermediate element $g^x \in \langle g \rangle_n$ with a factor in $\{-1, 1\} = \langle -1 \rangle_n$ makes resulting value θ' (almost) uniformly distributed in \mathbb{Z}_n . Second, padding function pad is designed such that it sends a uniformly distributed element in $\mathbb{Z}_n = [0, n-1]$ to an element (almost) uniformly distributed in $[0, T-1]$, where $T \gg n$ is independent of n and hence independent of specific group G . This technique, that dates back to Desmedt [65], makes the distribution of transmitted messages θ (almost) independent of deployed group G . Jarecki *et al.* prove [95] that this blinding suffices to achieve an affiliation-hiding protocol.

In respect to key establishment, Jarecki *et al.* claim that their protocol offers key indistinguishability with forward secrecy. However, the corresponding proof is flawed¹. Independently of Jarecki *et al.*, Gennaro, Krawczyk, and Rabin [81, 82] analyze Okamoto's protocol in a slightly different setting: in their variant, group parameter g is chosen to be a generator of $QR(n)$. Under this condition, however, $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$ does not hold (recall that this property was essential to achieve affiliation-hiding security). In this thesis, we abstain from giving an adaption of the proof found in [81, 82] to the setting where g is chosen according to our requirements. Nevertheless, we have verified that the proof is convertible to the AHA setting in a sound way and enumerate all necessary modifications in Appendix A.

To prove forward secrecy of their *two-pass* scheme (that does not provide explicit key confirmation), Gennaro *et al.* require a strong non-standard interactive extractability assumption [82]. A weaker notion of key security, where the peer of the Test session under some circumstances may not be corrupted, is proven under SRSA assumption (cf. Definition 3), in the random oracle model.

¹Precisely, in [95], the proposed reduction fails to provide means to simulate the actions of user U_j in the analysis of 'TYPE III adversaries'. In other words: the proof assumes (in contrast to the model) that impersonated users stay completely offline.

However, the key confirmation step in Figure 2.6 has a healing consequence on key secrecy: Recall that, in strong models for key security that also regard forward secrecy [45], **Corrupt** queries on **Test** session’s peer may only be asked *after* session termination. In the AHA protocol, the latter requires reception of correct confirmation tag v , i.e., adversary has to pose a $H_1(r \parallel \dots)$ query *before* the session actually terminates. This implies that the solution to embedded SRSA challenge can be extracted before adversary gets permission to corrupt the peer. In other words: the case in which [82] cannot answer **Corrupt** queries is irrelevant in the three-pass protocol from Figure 2.6. In particular, the protocol is forward secure. We summarize this discussion as follows:

Theorem 2 *The protocol specified in Figure 2.6 is a secure authenticated key establishment protocol with (strong) forward secrecy in the sense of [45], under the RSA assumption on safe moduli, in the random oracle model.*

2.5.2. AHA based on DLP

Another approach towards AHA was pioneered by Castelluccia, Jarecki, and Tsudik in [47], basing on a new cryptographic building block: In a PKI setting with multiple certification authorities (CA), a *PKI-enabled encryption* scheme allows users to encrypt messages to other users with respect to a specific CA. Recipients, before being able to decrypt any ciphertext, have to obtain from the particular CA an individual (but ciphertext-independent) certificate that is bound to their identity. The scheme is *CA-oblivious* if ciphertexts do not leak information about the ‘addressed’ CA. Given such an encryption scheme, an AHA protocol can be constructed by identifying CAs with GAs, and by admitting users to groups by issuing corresponding certificates. Castelluccia *et al.* [47] give a construction of CA-oblivious encryption, based on Schnorr signatures (cf. Section 2.2.1) and ElGamal encryption [72]. We briefly describe their AHA scheme:

Let (\mathcal{G}, g, q) denote a cyclic group of prime order q (cf. Section 2.1.2). Let $H : \{0, 1\}^* \rightarrow \mathbb{Z}_q$ and $H_1 : \{0, 1\}^* \rightarrow \{0, 1\}^\ell$ denote hash functions, for some fixed $\ell = \ell(\kappa)$.

CreateGroup

To initialize a group G , GA picks as private key a random element $x \leftarrow_R \mathbb{Z}_q \setminus \{0\}$.

AddUser

A credential for pseudonym $\text{id} \in \{0, 1\}^*$ is issued by GA by computing a Schnorr signature (ω, t) on id , i.e., $(\omega, t) \leftarrow (g^r, r + xH(\omega \parallel \text{id}))$ for random $r \leftarrow_R \mathbb{Z}_q$, and handing out $\text{sk}_G[\text{id}] = (\omega, t, y)$ to the user, where $y = g^x$.

Observe that, in the context of Schnorr's signature scheme, signed message is id , while y serves as verification key.

Element $\omega \in \mathcal{G}$ is considered a (public) value associated with pseudonym id from which g^t can be computed via $g^t = \omega y^{H(\omega \parallel \text{id})}$, while t acts as a trapdoor for this value and is known only to the 'owner' of $\text{sk}_G[\text{id}]$. We remark that the CA-oblivious encryption scheme proposed in [47] is standard ElGamal encryption to public key g^t , where t is the decryption key.

Handshake

We reproduce a simplified version of the protocol from [47] in Figure 2.7, expanding it from a four-move to a six-move protocol for the sake of better readability. Observe that key K and confirmation messages v_A, v_B are derived from random nonces $r_A, r_B \in \mathcal{G}$. These nonces are transmitted from one party to the other via CA-oblivious encryption, in (ElGamal) ciphertexts $(C_{A,1}, C_{A,2})$ and $(C_{B,1}, C_{B,2})$, respectively.

We remark that Castelluccia *et al.* do not analyze key security of their AHA scheme. Observe that key establishment protocols, like the one from Figure 2.7, that authenticate participants by challenging their decryption capabilities, generally cannot be proven secure in a model that considers active adversaries, without explicitly assuming CCA security of the underlying encryption scheme (as decryption is needed in the simulation). As ElGamal encryption is not CCA-secure, it is unclear how Castelluccia's scheme could be proven secure in a strong model. However, the approach to construct AHA schemes from Schnorr signatures remains interesting and will serve as a basis for our constructions in later sections.

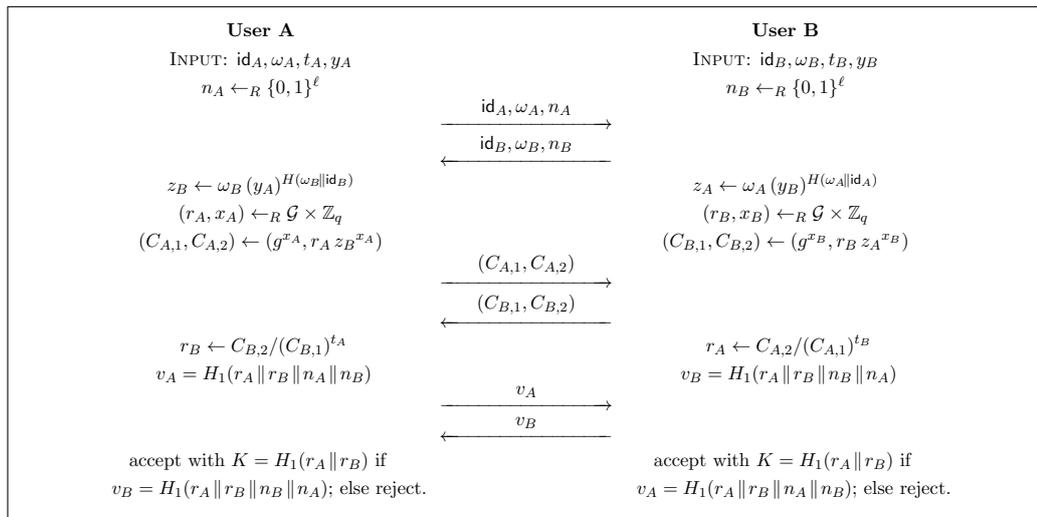


Figure 2.7.: DLP-based Handshake protocol from [47] (simplified)

2.5.3. AHA based on bilinear maps

The AHA scheme proposed by Balfanz *et al.* in [9] is defined in the pairing-based setting. More precisely, its central building block is the non-interactive key distribution scheme (NIKDS) by Sakai *et al.* [141] (cf. Section 2.3.1). The idea is to identify KGCs with GAs, i.e., the user with pseudonym id receives NIKDS's $\text{sk}[\text{id}]$ as credential. Implicitly, this establishes a fixed symmetric (per-group) key between each two users. The actual Handshake is a simple protocol that verifies whether both parties have knowledge of the same key. Note that it is the absence of network traffic in NIKDS that yields the affiliation-hiding privacy. The scheme from [9] is specified as follows:

CreateGroup

To initialize a group G , GA runs $\text{msk} \leftarrow \text{NSetup}(1^\kappa)$ and outputs secret key $G.\text{sk} = \text{msk}$.

AddUser

Admission of a user with pseudonym $\text{id} \in \{0, 1\}^*$ is done by computing id 's credential as $\text{sk}_G[\text{id}] \leftarrow \text{NRegister}(\text{msk}, \text{id})$.

Handshake

A simplified version of the protocol is given in Figure 2.8. Basically, it consists of the computation of the shared NIKDS key and a standard nonce-based equality check. The latter makes use of hash function $H_1 : \{0, 1\}^* \rightarrow \{0, 1\}^\ell$, for some fixed $\ell = \ell(\kappa)$.

Observe that the protocol outputs $K = \text{NSharedKey}(G.\text{sk}; \text{id}_A, \text{id}_B)$ as session key, which is independent of the actual Handshake session. Clearly, protocols that output fixed keys cannot offer key security in respect to a strong model [14, 45].

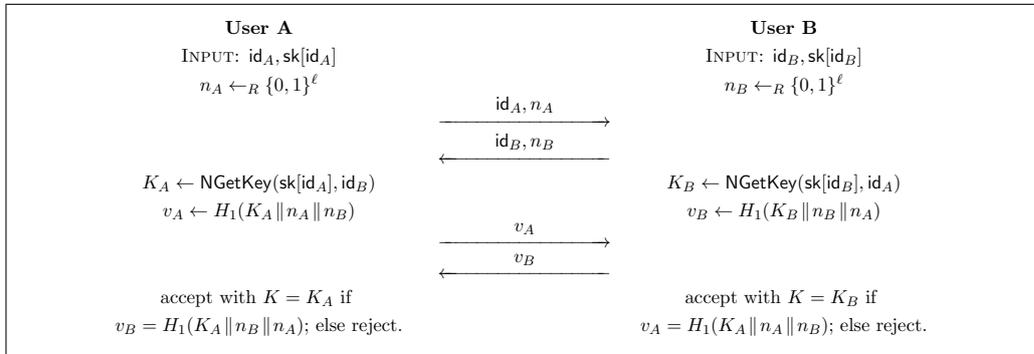


Figure 2.8.: NIKDS-based Handshake protocol from [9] (simplified)

AHA with untrusted group authorities

Authentication protocols that follow the classical AHA concept defend group members' privacy against group outsiders (i.e., non-members). This is clearly reflected in the accepted security models [9, 47, 95, 121]. Yet, the possibility of *insider attacks* has not been considered in the literature so far. This is mainly due to the fact that early constructions [9, 47] were pure authentication protocols, i.e., there was no point in defending from insiders as it was intended that these learned all relevant information anyway. However, observing the rise of AHA protocols that also encompass key establishment [94, 95, 121], we argue that there is the strong necessity to also formally treat security regarding attacks from the inside, most importantly to preclude attacks against key secrecy.

We even go a step further and ask for reasonable security notions that consider not only other members as potentially malicious, but also group authorities. For instance, if two honest members execute a **Handshake** in the presence of a malicious GA that mounts, say, a man-in-the-middle attack, then one should still require that both, the affiliations of participants and the computed session key, remain hidden from GA. In this section, we study and mitigate the consequences of potential GA misbehavior in AHA protocols. Within others, we show that unconditional trust of group members in their particular GAs is problematic in most schemes proposed in the literature so far. We also design a new scheme that tolerates malicious GAs, raising users' privacy to a new level.

Consider a social network that is operated in an oppressive regime as an example where trustworthiness of GAs cannot be guaranteed and, moreover, maliciousness even has to be assumed. Although AHA could be of great importance for members of an anti-government group to recognize each other and to hold discrete meetings and conversations, an AHA scheme with the classical security properties would not be safe to use: If the government raids the social network and confiscates the GA, then all parameters and records become exposed and the government can thereafter identify all members and unveil all (past) communication.

Considering possible insider attacks, we anticipate a malicious GA to attempt any of the following: generate group parameters in a rogue way, create phantom group members, misbehave during the registration process of honest members, and mount active attacks on sessions involving honest members. In the following paragraphs, we discuss these attacks in more detail. We also discuss the resulting challenge: Which security requirements for AHA protocols can be preserved in a meaningful way, even when dealing with corrupt GAs?

Security of registration/revocation

Among GA's duties is the registration and revocation of group members. Clearly, if GA misbehaves and introduces phantom members, then security of session keys computed by honest participants in sessions with phantom members can no longer represent a meaningful requirement. However, we argue that 'good' security models for AHA should at least leave GAs incapable of generating credentials for freely chosen pseudonyms, as this would keep GAs from impersonating admitted users by recomputing their credentials.

We believe that possible GA misbehavior during the registration procedure of new (honest) members must also be taken into account. Note that the registration process is the only step where GA interacts directly with users and that information obtained or issued by GA during registration may be later misused to the detriment of members' privacy.

Security of session keys

An AHA instance between two honest members should result in a secret session key. Often it is desirable that this key is secure even against a curious GA. More precisely, we might require that established keys offer forward secrecy with respect to any future corruption of GAs and regular users. Although this issue has not been formally addressed so far, it seems that some recent results [94, 95, 121] satisfy this extended form of forward secrecy. However, in many other constructions [9, 47], GA's private keys can be used to immediately recover session keys.

Privacy of group members

The central privacy property offered by AHA protocols is that group memberships of participants remain hidden from outsiders. However, it is also meaningful to extend this requirement towards the GA: As long as GA is trusted with the security of the registration, it makes sense to demand that transcripts of sessions among honest members do not reveal affiliations to GA. This requirement is of particular importance for linkable schemes where participants communicate via pseudonyms

and exchange these in the clear when executing `Handshake` sessions. Current linkable AHA schemes [9,47,94,95,121] do not provide this stronger privacy notion, since GA learns (or even picks) the pseudonyms of its members during the registration process and will recognize these when monitoring network traffic.

We argue, however, that there is an even more significant threat to privacy of group members. It stems from the fact that, during the registration process of users, GA learns users' real identities (in addition to users' pseudonyms). In particular, we believe that `Handshake` sessions involving honest members should not reveal to GA any information about participants' real identities (even though GA knows the real identities of all group members). In other words, users should remain *untraceable* throughout their communication sessions.

Untraceability is a new privacy requirement that does not appear in current AHA security models; in fact, none of the current linkable protocols we are aware of provides it. In linkable AHA protocols, hiding real identities of participants from GA appears to be especially challenging due to the use of pseudonyms created during the registration phase. Informally, we define the security goal of untraceability as follows (cf. Chapter 1). Note that untraceability is a property individual to a member, while affiliation-hiding is shared by all members of the same group.

Untraceability of members

It is infeasible for an (adversarial) GA to learn the real identity U of an honest group member from `AddUser` and `Handshake` sessions involving that member.

Intuitively, untraceability is related to GA's ability to obtain information during the registration phase of an honest member that allows it to later link back AHA sessions of that member to the registration process and, thus, to the real identity. A possible way to achieve untraceability is to prevent GA from learning pseudonyms of group members upon their registration. This can be achieved by *blinding* the registration process. For instance, in the case of AHA protocols sketched in Sections 2.5.1 and 2.5.2, where users' credentials are signatures on their respective pseudonyms, the `AddUser` algorithm could be replaced by a (multipass) protocol for blinded signature generation.

However, care has to be taken to prevent the adversary from registering pseudonyms of its choice with any given group. In particular, assuming a fully blinded registration process, the adversary would readily be able to obtain membership credentials for pseudonyms id that are already in use by honest group members, without explicitly corrupting any of them. This, in turn, would allow the adversary to mount impersonation attacks against key secrecy. We stress that this problem does not arise solely because of blind registration, but also due to specific constructions of pseudonyms. In fact, in Section 3.2, we will identify requirements on pseudonym registration that are sufficient to protect against this type of attack.

The concepts of untraceability and blind registration raise some concerns about membership revocation. Note that, in linkable AHA protocols, revocation can be understood in two ways: The first approach is *revocation of users*, where GA may want to revoke some particular user U . The second approach is *revocation of pseudonyms*, where GA may want to revoke some pseudonym id . In traceable protocols, i.e., in protocols that do not offer untraceability, there is usually no difference between these two approaches, since GA knows the mapping between U and id , and adds id in both cases to revocation list $G.prl$. However, in untraceable AHA, it is ensured that GA does not learn any pseudonym id , i.e., revocation of users is no longer possible. This is the price we have to pay for untraceability. However, users participate in group applications via pseudonyms. Therefore, if some misbehavior is noticed, the responsible pseudonym can be identified and revoked. This type of revocation is still meaningful, since, if GA revokes some pseudonym id that is owned by some user U , then U cannot communicate in that group anymore. Hence, revocation of pseudonyms can still be sufficient to prevent misbehaving users from further participation in groups.

3.1. A security model for AHA that considers dishonest GAs

3.1.1. Adapted syntax for AHA

In the general definition of AHA that we gave in Section 11, the `AddUser` procedure was considered to be an algorithm executed by GA. However, as discussed in the preceding paragraphs, the new security property of ‘untraceability’ can only be achieved by permitting a blinded registration, i.e., a registration via an interactive protocol. Hence, before proposing our security model for AHA schemes in the presence of corrupt authorities, we have to slightly adjust the syntax of AHA, by replacing the `AddUser` algorithm from Definition 11 by the following protocol:

`AddUser`($U \leftrightarrow G$)

This protocol is executed between the prospective group member U and the GA of group G . The algorithm on U 's side is denoted `AddUserU`(U, G), the algorithm on GA's side by `AddUserG`(G, U). Let π be a session of either `AddUserU` or `AddUserG`. The *state* of π is defined through the session variable $\pi.state$ and can take `running`, `accepted`, or `rejected` values. For both algorithms we initially assume $\pi.state = \text{running}$. Once `AddUserU` session π reaches $\pi.state = \text{accepted}$, its variable $\pi.result$ holds a pair $(id, sk_G[id])$, where id is a *pseudonym* and $sk_G[id]$ is a *membership credential* that enables user U to authenticate with pseudonym id in group G in future `Handshake` sessions.

3.1.2. Adversarial queries

In the security experiments defined below, adversary \mathcal{A} is modeled as a probabilistic algorithm that runs in polynomial time and interacts with the experiment via the following set of queries. Unless explicitly noted, we assume that, at any time, \mathcal{A} has access to exhaustive (system-wide) lists GLi and IDLi of available groups and registered pseudonyms, respectively. Note that these lists do not disclose the mapping between pseudonyms and groups.

CreateGroup

This query sets up a new group G and publishes its (empty) revocation list $G.\text{prl}$. The group is added to GLi .

AddUserU(U, G)

This query models the actions of user U initiating the `AddUser` protocol with target group G . A new protocol session π is started. Optionally, a first message M is output. Group G is added to GLi if it is a new group.

Note that we do not require $G \in \text{GLi}$ to hold before this query is asked. Basically, this allows the adversary to introduce its own groups with arbitrary (potentially maliciously chosen) parameters, and to populate these groups with honest members.

AddUserG(G, U)

This query differs from `AddUserU` in that it models GA's actions on the `AddUser` protocol. We require that G has been established through `CreateGroup` before this query is posed.

Handshake(id, G, r)

This query lets pseudonym $\text{id} \in \text{IDLi}$ start a new session π of the `Handshake` protocol. It receives as input the group G wherein the `Handshake` shall take place (given that id has credentials for that group) and a role identifier $r \in \{\text{init}, \text{resp}\}$ that determines whether the session shall act as protocol initiator or responder. Session variable $\pi.\text{revealed}$ is initialized to `false`. Optionally, this query returns a first protocol message M .

Send(π, M)

Message M is delivered to session π . After processing M , the eventual output is given to \mathcal{A} . This query is ignored if π is not waiting for input. Note that π is either an `AddUserU`, an `AddUserG`, or a `Handshake` protocol session. If π is an `AddUserU` session and is in state `accepted` after processing M , then id from $\pi.\text{result}$ is added to IDLi .

Reveal(π)

This query is defined only for Handshake sessions. If $\pi.\text{state} \in \{\text{accepted}, \text{rejected}\}$ it returns $(\pi.\text{state}, \pi.\text{key})$ and sets $\pi.\text{revealed} \leftarrow \text{true}$; otherwise, if $\pi.\text{state} = \text{running}$, the query is ignored.

Corrupt($*$)

The input is either a pseudonym id or a group G :

Corrupt(id)

If $\text{id} \in \text{IDLi}$, credential $\text{sk}_G[\text{id}]$ is given to \mathcal{A} , for any group G in which pseudonym id is registered.

Corrupt(G)

If $G \in \text{GLi}$ is a group created by a CreateGroup query, then G 's long term secret $G.\text{sk}$ and control over G 's revocation list $G.\text{prl}$ is handed over to \mathcal{A} .

Revoke(G, id)

This query lets GA of G include pseudonym id in its revocation list $G.\text{prl}$.

Remark 1 *In addition to factoring in potentially malicious GAs, our security model differs from that of, e.g., [95] in the way that user corruptions are handled. Indeed, unlike in [95], we do not require that Corrupt(id) queries automatically trigger the inclusion of pseudonym id in corresponding G.prl, i.e., we allow the adversary to actually use credentials obtained through corruptions. We argue that our model more comprehensively reflects some realistic attack scenarios, when compared to [95].*

Within others, our security model regards groups and pseudonyms that are introduced by the adversary. It is hence meaningful to distinguish between honest and potentially dishonest users and GAs:

Definition 13 (Honest generation of pseudonyms and groups) *A pseudonym id is honestly generated if it has been established through an AddUserU query, i.e., if $\text{id} \in \text{IDLi}$. It is honest if thereafter no Corrupt(id) query has been asked. Similarly, a group G is honestly generated if it has been established through a CreateGroup query. It is honest if thereafter no Corrupt(G) query has been asked.*

It is also convenient to introduce a notion for groups in which the adversary can obtain a valid membership credential in a trivial way:

Definition 14 (Intruded groups, intact groups) *A group G is intruded if at least one of the following holds:*

- G was not created via a CreateGroup query;
- G was created via a CreateGroup query, but later Corrupt(G) was asked;

- G was infiltrated via an $\text{AddUserG}(G, \cdot)$ query;
- G was infiltrated via a $\text{Corrupt}(\text{id})$ query, for some pseudonym $\text{id} \in \text{IDLi}$.

If a group G is not intruded, then it is intact.

As they prove useful also in the context of AHA protocols, we borrow some concepts from security models for key establishment [14, 45, 113]:

Definition 15 (Session id, partnered session, matching session) *The session id $\pi.\text{sid}$ of a Handshake session π with $\pi.\text{state} = \text{accepted}$ is a value that uniquely identifies π in the set of all protocol instances run by $\pi.\text{id}$. Two Handshake sessions π, π' are partnered if $\pi.\text{state} = \pi'.\text{state} = \text{accepted}$ and $(\pi.\text{sid}, \pi.\text{id}, \pi.\text{partner}) = (\pi'.\text{sid}, \pi'.\text{partner}, \pi'.\text{id})$. Sessions π, π' are matching if the groups G and G' associated with π and π' , respectively, are identical, and if the concatenation of the messages received by π is a prefix of the concatenation of messages sent by π' , and vice versa (i.e., π and π' have consistent transcripts).*

3.1.3. Affiliation-hiding security in the presence of corrupt GAs

In this section, we define the notion of (linkable) affiliation-hiding security (AH). Our model adapts the simulation-based approach from [95] to a setting that regards potentially malicious GAs. The idea is to require that the real protocol execution remains indistinguishable from an idealized one performed by a simulator \mathcal{SIM} that simulates Handshake executions without knowing participants' affiliations. This intuition is formalized through two experiments, $\text{Expt}^{\text{ah},0}$ and $\text{Expt}^{\text{ah},1}$, which adversary \mathcal{A} has to distinguish. More precisely, in security experiment $\text{Expt}_{\text{AHA}, \mathcal{A}, \mathcal{SIM}}^{\text{ah}, b}$, $b \in \{0, 1\}$, the adversary is run on input security parameter 1^κ and is provided access to a set of queries similar to those from Section 3.1.2. Eventually, the adversary stops and outputs a bit b' , which is the output of the experiment. While in $\text{Expt}^{\text{ah},1}$ all queries posed by \mathcal{A} are answered (almost) as described in Section 3.1.2, in $\text{Expt}^{\text{ah},0}$ some queries are answered by help of \mathcal{SIM} , as shown below. Clearly, deployment of \mathcal{SIM} can readily be detected by \mathcal{A} for intruded groups, simply by executing the Handshake protocol on behalf of an (impersonated) group member. Hence, the model lets \mathcal{SIM} simulate only those sessions that are run on behalf of honest pseudonyms in groups that are intact.

We specify in detail how adversary's queries are processed in experiment $\text{Expt}_{\text{AHA}, \mathcal{A}, \mathcal{SIM}}^{\text{ah}, b}$, $b \in \{0, 1\}$:

CreateGroup , $\text{AddUserU}(U, G)$, $\text{Revoke}(G, \text{id})$

These queries are answered as described in Section 3.1.2.

AddUserG(G, U)

This query is answered as described in Section 3.1.2, unless there exists a running **Handshake** session π invoked for a group G which is intact. In this case, **AddUserG** queries are ignored if their input is such that, after processing these queries, group G would become intruded.

Note that the named restriction preserves integrity of groups: It ensures that **Handshake** sessions that are started in intact groups continue running in intact groups until their termination.

Handshake(id, G, r)

We distinguish between two (complemental) cases:

- If G is intruded or $b = 1$, then the query is answered as described in Section 3.1.2 (i.e., without involving \mathcal{SIM}). We call the invoked session a *honest session*.
- If G is intact and $b = 0$, then $(st, M) \leftarrow \mathcal{SIM}.\text{Handshake}(1^\kappa, \text{id}, r)$ is invoked. While answer M is given to the adversary, simulator's state is stored in a session variable: $\pi.\text{sim} \leftarrow st$. Note that \mathcal{SIM} does not learn group G that was provided to the **Handshake** query¹. We call the invoked session a *simulated session*.

As mentioned above, in intruded groups, adversary \mathcal{A} could detect deployment of \mathcal{SIM} by trivial means. Hence, we let \mathcal{SIM} engage only in intact groups, and only in $\text{Expt}^{\text{ah},0}$.

Send(π, M) for AddUserU and AddUserG sessions

These queries are answered as described in Section 3.1.2.

Send(π, M) for Handshake sessions

Again, we distinguish between two cases:

- If π is an honest session, then the query is answered as described in Section 3.1.2 (i.e., without involving \mathcal{SIM}).
- If π is a simulated session, then algorithm $\mathcal{SIM}.\text{Send}(\pi.\text{sim}, M)$ is invoked and its answer is replied.

¹Here, for simplicity, we make the assumption that pseudonyms of users do not leak information about their groups, meaning that the statistical difference between the distributions $D_i = \{\text{id} \mid \text{id} \text{ was established by } \text{AddUser}(G_i, U)\}$ and $D_j = \{\text{id} \mid \text{id} \text{ was established by } \text{AddUser}(G_j, U)\}$ is bounded by a negligible function, for any two groups G_i, G_j . This is the case for most practical AHA protocols, including [9, 47, 94, 95], and also for our protocols presented in later sections. The assumption can be omitted by further refining the model such that $\mathcal{SIM}.\text{Handshake}()$ does not receive id , but instead some identifier $F(\text{id}, G)$ for a suitable function F that hides both the pseudonym and its affiliation from \mathcal{SIM} . This approach is discussed in detail in [95].

Observe that, in our definition, simulator STM is stateful. In particular, different invocations of $STM.Handshake$ and $STM.Send$ can be processed by the simulator in dependence of each other. Although it deviates from the requirements proposed in [95], we stress that this restriction still defines a meaningful setting.

Reveal(π)

Again, we distinguish between two cases:

- If π is an honest session, then the query is processed as described in Section 3.1.2.
- If π is a simulated session: The query is ignored if π did not receive enough messages to complete the protocol. Otherwise, if no session π' exists which is *matching* π , then (rejected, \perp) is returned. In all other cases the experiment returns (accepted, $\pi.key$), where $\pi.key$ is assigned according to the following rules:
 - If $\pi.key$ is not set but $\pi'.key$ is, then $\pi.key \leftarrow \pi'.key$;
 - If both $\pi.key$ and $\pi'.key$ are not set, then $\pi.key \leftarrow_R \{0, 1\}^\kappa$.

Observe that if sessions π and π' are matching then all requirements are satisfied for acceptance in the Handshake protocol. In this case, clearly, by revealing state and keys established in π and π' , \mathcal{A} will learn that id and id' belong to the same group. As noticed in [95], this is unavoidable and, even in this case, \mathcal{A} is not supposed to learn the affiliation of these members.

Observe that we do not allow the adversary to corrupt users or groups, i.e., to pose $Corrupt(*)$ queries. We are now ready to define affiliation-hiding security:

Definition 16 (Affiliation-hiding security) *Let $AHA = \{\text{CreateGroup}, \text{AddUser}, \text{Handshake}, \text{Revoke}\}$ and let $\text{Expt}^{\text{ah},0}$ and $\text{Expt}^{\text{ah},1}$ be the experiments described above. The advantage of adversary \mathcal{A} in respect to simulator STM is defined as*

$$\text{Adv}_{AHA, \mathcal{A}, STM}^{\text{ah}}(\kappa) = \left| \Pr \left[\text{Expt}_{AHA, \mathcal{A}, STM}^{\text{ah},0}(\kappa) = 1 \right] - \Pr \left[\text{Expt}_{AHA, \mathcal{A}, STM}^{\text{ah},1}(\kappa) = 1 \right] \right| .$$

We say that AHA is affiliation-hiding if there exists an efficient simulator STM such that $\text{Adv}_{AHA, \mathcal{A}}^{\text{ah}}$ is negligible for all efficient adversaries \mathcal{A} .

3.1.4. Key security in the presence of corrupt GAs

Security of session keys established in AHA protocols is defined analogously to that in classical key agreement schemes [14, 45]. The adversary's task is to distinguish a key established in a protocol execution from a randomly generated value of the same

length. To formalize this intuition, we introduce a new session flag, $\pi.\text{tested}$, that is set to `false` upon session initialization, and slightly adapt the `Reveal` query. We then define security experiments $\text{Expt}^{\text{ake},b}$, $b \in \{0, 1\}$, that make use of an auxiliary `Test` query (that is dependent on bit b).

We start by describing the new queries:

Reveal(π)

This query is answered as specified in Section 3.1.2 (in particular, it sets $\pi.\text{revealed} \leftarrow \text{true}$), unless $\pi.\text{tested} = \text{true}$ or $\pi'.\text{tested} = \text{true}$, for any session π' that is partnered with π . In the latter case, the query is ignored.

Test(π)

This query is ignored if π is not *fresh* (cf. Definition 17). Otherwise, $\pi.\text{tested}$ is set to `true` and a key is returned, according to the following rule: If $b = 1$, $\pi.\text{key}$ is returned. If $b = 0$, a random element drawn uniformly from $\{0, 1\}^\kappa$ is returned. The `Test` query may be invoked at most once.

The notion of session freshness is useful to exclude trivial attacks and simplifies the definition of key security:

Definition 17 (Session freshness) *A session π that is invoked in response to a `Handshake(id, G, r)` query is fresh if the following conditions are satisfied:*

- (a) $\pi.\text{state} = \text{accepted}$ and $\pi.\text{revealed} = \text{false}$ and $\pi'.\text{revealed} = \text{false}$ for all sessions π' that are partnered with π ;
- (b) in the moment that $\pi.\text{state} \leftarrow \text{accepted}$ was assigned, all of the following did hold:
 - (1) if $\pi.\text{partner}$ is honestly generated: no `Corrupt($\pi.\text{partner}$)` has been asked;
 - (2) if $\pi.\text{partner}$ is not honestly generated: G is honest and no `AddUserG(G, \cdot)` query has been asked;
 - (3) no `Corrupt($\pi.\text{id}$)` query has been asked.

We provide rationale for these constraints: Condition (a) prevents the trivial attack where adversary \mathcal{A} ‘computes’ the session key established by π or any partnered session π' by simply `Revealing` it. Conditions (1)–(3) model forward secrecy by allowing \mathcal{A} to corrupt participants *after* the computation of π ’s session key took place. Note that the specific conditions prevent impersonation attacks (of honest users, and of members of ‘honest groups’, respectively), and are identical to those in Definition 14. Observe that condition (3) permits the consideration of protocols that are not resilient to key compromise impersonation (KCI) attacks [23, 113]. Also note that the freshness conditions consider valid the attack where a malicious GA

sets up ‘odd’ group parameters and tries to break security of session keys established between honest members of that group.

Given the set of queries specified above and in Section 3.1.2, security experiments $\text{Expt}^{\text{ake},b}(\kappa)$, $b \in \{0, 1\}$, are defined as follows: adversary \mathcal{A} is run on security parameter 1^κ , and access to all queries is provided (where only **Test** query is dependent on bit b). Eventually, the adversary stops and outputs a bit b' , which is taken as output of the experiment. The definition of key security for AHA schemes is now straight forward:

Definition 18 (Key security with forward secrecy) *Let $\text{AHA} = \{\text{CreateGroup}, \text{AddUser}, \text{Handshake}, \text{Revoke}\}$ and let $\text{Expt}^{\text{ake},0}$ and $\text{Expt}^{\text{ake},1}$ denote the experiments described above. The advantage of adversary \mathcal{A} is defined as*

$$\text{Adv}_{\text{AHA},\mathcal{A}}^{\text{ake}}(\kappa) = \left| \Pr \left[\text{Expt}_{\text{AHA},\mathcal{A}}^{\text{ake},0}(\kappa) = 1 \right] - \Pr \left[\text{Expt}_{\text{AHA},\mathcal{A}}^{\text{ake},1}(\kappa) = 1 \right] \right| .$$

We say that AHA offers key security with forward secrecy if $\text{Adv}_{\text{AHA},\mathcal{A}}^{\text{ake}}$ is negligible for all efficient adversaries \mathcal{A} .

3.1.5. Untraceability

The idea behind untraceability is that (even malicious) GAs should not be able to trace back users’ pseudonyms to the identities of their owners. As discussed in the introduction to this section, untraceability is a new (individual) privacy requirement, orthogonal to key security and affiliation-hiding. We formalize it using the indistinguishability approach: In the corresponding security experiment, we let adversary \mathcal{A} first specify parameters for a group G . Two users, U_0 and U_1 , are enrolled into G , whereby their respective pseudonyms id_0 and id_1 (and corresponding credentials) are obtained. Untraceability reflects the inability of \mathcal{A} to trace these pseudonyms back to the two users.

Definition 19 (Untraceability) *Let $\text{AHA} = \{\text{CreateGroup}, \text{AddUser}, \text{Handshake}, \text{Revoke}\}$ and let $\text{Expt}^{\text{trace},0}$ and $\text{Expt}^{\text{trace},1}$ be the experiments specified in Figure 3.1, where, for obvious reasons, we do not allow adversary \mathcal{A} to access the list of pseudonyms IDLi . The advantage of \mathcal{A} is defined as*

$$\text{Adv}_{\text{AHA},\mathcal{A}}^{\text{trace}}(\kappa) = \left| \Pr \left[\text{Expt}_{\text{AHA},\mathcal{A}}^{\text{trace},0}(\kappa) = 1 \right] - \Pr \left[\text{Expt}_{\text{AHA},\mathcal{A}}^{\text{trace},1}(\kappa) = 1 \right] \right| .$$

We say that AHA is untraceable if $\text{Adv}_{\text{AHA},\mathcal{A}}^{\text{trace}}$ is negligible for all efficient adversaries \mathcal{A} .

Note that, in experiments $\text{Expt}^{\text{trace},b}$, $b \in \{0, 1\}$, corruption of id_0 and id_1 is not forbidden. Therefore, untraceable AHA schemes hide the real identity of group members even if their membership credentials are leaked.

$\text{Expt}_{\text{AHA}, \mathcal{A}}^{\text{trace}, b}(\kappa)$:

- (a) adversary $\mathcal{A}(1^\kappa)$, having access to the queries from Section 3.1.2, specifies parameters for a group G
- (b) let U_0 and U_1 be two users. The experiment executes the admission protocol $\text{AddUserU}(U_0, G)$, followed by $\text{AddUserU}(U_1, G)$, where all protocol steps on behalf of G are executed by \mathcal{A} . The experiment does not proceed until the corresponding protocol sessions executed on behalf of U_0 and U_1 accept and output pseudonyms id_0 and id_1 , respectively.
- (c) adversary \mathcal{A} continues its execution on input id_b (still having access to the queries from Section 3.1.2) and outputs bit b'
- (d) the output of the experiment is b'

Figure 3.1.: trace experiment

3.2. A construction based on RSA

The design of the protocol presented in this section is based on schemes by Vergnaud [156] and Jarecki, Kim, and Tsudik [95], whose ideas were sketched in Section 2.5.1. It turns out that an adaption of [95, 156] towards a scheme that is secure in the presence of corrupted GAs is rather challenging. This is due to the fact that the scheme is RSA-based, i.e., security heavily depends on the well-formedness of the RSA parameters (n, e) . For instance, the security argument for the classical protocol for blind generation of RSA signatures (cf. Section 2.2.2) through blinding to-be-signed message m with a blinding factor r^e (for random $r \in \mathbb{Z}_n^\times$) assumes that operation $r \mapsto r^e \bmod n$ is bijective (which is the case iff $\gcd(e, \varphi(n)) = 1$). Moreover, even if RSA parameters have been properly generated, GA's knowledge of the modulus' factorization can burden users' security. For example, solving the CDH problem in group \mathbb{Z}_n^\times appears to be much easier if the factorization of n is known, as the Chinese Remainder Theorem readily decomposes \mathbb{Z}_n^\times into $\mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$, i.e., CDH mod n can be computed by solving CDH mod p and mod q separately, and then combining the results. All in all, we see that special care has to be taken with RSA-related operations if the generation of parameters is not fully trusted, and that public verifiability of correct generation of group parameters is a necessary requirement of secure AHA protocols. In the RSA-based setting, the latter is generally achieved using (slight modifications of) the zero-knowledge proof framework from [42], although such proofs are rather expensive to generate and to check.

A different challenge, related to the goal of untraceability, is the following: If users' pseudonyms are arbitrary strings (as in [95, 156] and in Section 2.5.1), and if

their credentials are RSA signatures on these pseudonyms, then every member of a group can trivially be impersonated by a corrupt GA: the latter would just have to recompute the credential of targeted pseudonym. However, this would contradict our security model from Section 3.1. In our solution, we deal with this issue by replacing user pseudonyms by public keys of an independent signature scheme (cf. Section 2.2). User authentication can then be performed by checking signatures on session transcripts.

3.2.1. Protocol specification

Let $H_1 : \{0, 1\}^* \rightarrow \{0, 1\}^{3\ell}$ and, for any (RSA modulus) $n \in \mathbb{N}$, $H_n : \{0, 1\}^* \rightarrow \mathbb{Z}_n$ be hash functions, where $\ell = \ell(\kappa)$ is fixed. For instance, if $H : \{0, 1\}^* \rightarrow \mathbb{Z}_{2^{\kappa+\ell}}$ denotes an auxiliary hash function, then H_n can be constructed as $H_n(x) := H(n \| x) \bmod n$. Let $\Sigma = (\text{KGen}, \text{Sign}, \text{Verify})$ be an unforgeable signature scheme (cf. Definition 6), and let $T = 2^{\kappa+\ell}$.

Camenisch and Michels [42] show how to prove in zero-knowledge (ZK) the correct generation of an RSA modulus $n = pq$, for safe primes p and q , including the necessary primality tests and without revealing any further information about the factors. We deploy an extended version of these ZK proofs: In Section 3.2.4 we describe a technique based on [42] that constructs a ZK proof that, for given (n, g, e) , shows that (n, e) is well-formed according to Definition 3, and that $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$. By $\Pi_{n,g,e}$ we denote a non-interactive version of that proof, e.g., obtained via Fiat-Shamir transformation [74].

CreateGroup

To set up a new group, GA generates fresh RSA parameters $(n, e, d) \leftarrow_R \text{SRSA-GEN}(1^\kappa)$ and picks an element $g \in \mathbb{Z}_n^\times$ such that $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$ (cf. Section 2.5.1). Let $\Pi_{n,g,e}$ be a non-interactive ZK proof that establishes well-formedness of (n, g, e) (see Section 3.2.4 for details). The algorithm sets $G.\text{prl} \leftarrow \emptyset$ and outputs $G.\text{sk} = (n, g, e, d, \Pi_{n,g,e})$ and $G.\text{prl}$.

AddUser

Member admission is implemented using a protocol between U and GA, as specified in Figure 3.2. Communication between U and GA is assumed to be authentic, yet it does not need to be confidential as in [95]. In a first step, U obtains and examines the validity of group parameters (n, g, e) , by checking the NIZK proof $\Pi_{n,g,e}$. Then, U generates a fresh signature key pair (pk, sk) of signature scheme Σ . The verification key, pk , is thereafter used by U as its pseudonym id in group G , i.e., we set² $\text{id} \leftarrow \text{pk}$. Using

²In practice, one would typically set $\text{id} \leftarrow F(\text{pk})$, for a collision-resistant hash function F . Here, for simplicity, we assume that $\text{id} = \text{pk}$.

standard blind RSA signature scheme (cf. Section 2.2.2), U obtains an RSA signature $\sigma_{\text{id}} = H_n(\text{pk})^d$ on $H_n(\text{pk})$. Note that the blinding factor r^e effectively hides $\text{id} = \text{pk}$ and $H_n(\text{pk})$ from GA. The output of U is $(\text{id}, \text{sk}_G[\text{id}])$, where $\text{sk}_G[\text{id}]$ contains signing key sk , the RSA parameters, and the signature on pk .

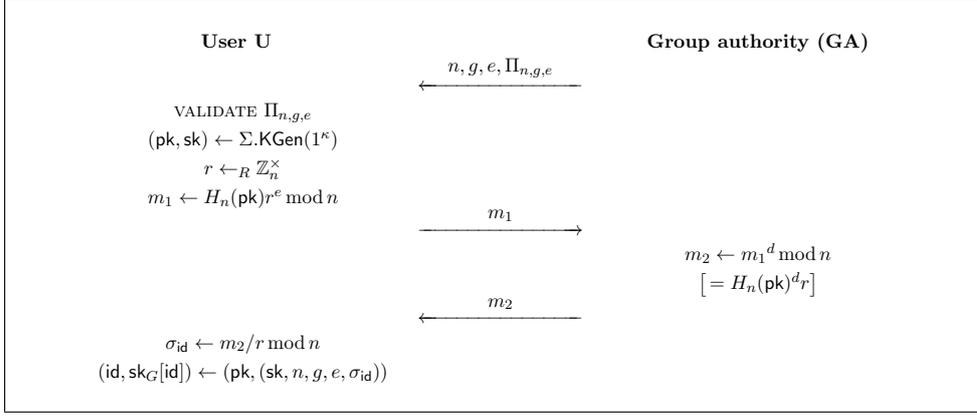


Figure 3.2.: RSA-based AddUser protocol

Handshake

The protocol is executed between two users, A and B , holding pseudonyms $\text{id}_A = \text{pk}_A$ and $\text{id}_B = \text{pk}_B$, and credentials $\text{sk}_{G_A}[\text{id}_A] = (\text{sk}_A, n_A, g_A, e_A, \sigma_{\text{id}_A})$ and $\text{sk}_{G_B}[\text{id}_B] = (\text{sk}_B, n_B, g_B, e_B, \sigma_{\text{id}_B})$, respectively. The protocol is specified in Figure 3.3. Padding function pad is specified in Section 2.5.1 and effectively hides moduli n_A and n_B from observers.

Correctness of the protocol follows by inspection. Observe that intermediate values r_A and r_B match in case $(n_A, g_A, e_A) = (n_B, g_B, e_B)$ (cf. equation (2.1) in Section 2.5.1), i.e., if users A and B are members of the same group.

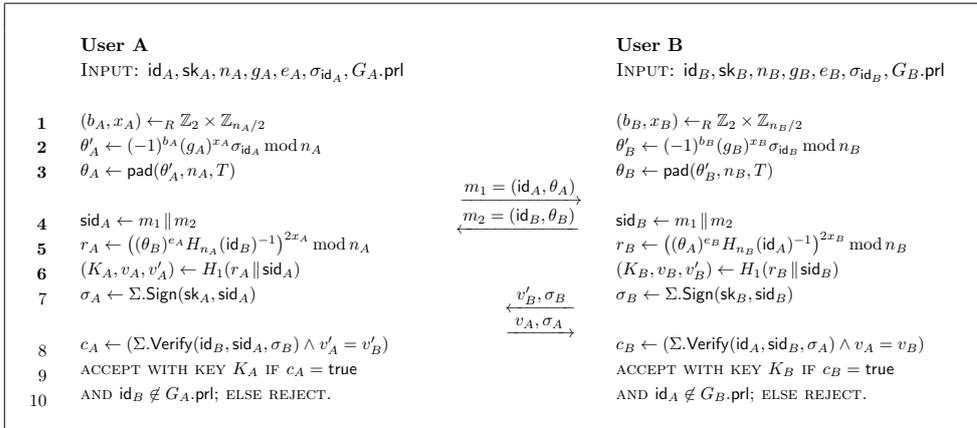


Figure 3.3.: RSA-based Handshake protocol

Revoke

The pseudonym id to be revoked is added to $G.pr$. It is assumed that this list is distributed authentically to all group members.

Remark 2 *Observe the important difference between our Handshake protocol and that from [95] (cf. Section 2.5.1): In addition to key confirmation messages v, v' , signatures σ_A, σ_B are sent in the second round of our protocol; their validity confirms not only the equality of computed session keys, but also serves as a proof of ownership of ids claimed by the respective participants (through knowledge of the corresponding secret keys). This thwarts active impersonation attacks where the adversary exploits the blinding property of the AddUser protocol to obtain credentials for pseudonyms of honest users. Note that the possibility of such pseudonym impersonation by insiders (group members) would violate key security as defined in Section 3.1.4.*

3.2.2. Efficiency and optimizations

The cost of our Handshake protocol is dominated by the computations of $\theta'_{A/B}$, $r_{A/B}$, generation of $\sigma_{A/B}$, and verification of $\sigma_{A/B}$. The first two involve exponentiations, while the cost of the latter depends on the balance between $\Sigma.\text{Sign}$ and $\Sigma.\text{Verify}$. Many current signature schemes involve either low verification and high generation costs (e.g., RSA), or vice versa (e.g., DSA). In any case, suffice it to say that, for each participant, the overall computation cost amounts to approximately 3–4 exponentiations. Considering the high degree of security offered by our scheme, this overhead is very low. Observe that the Handshake protocol can trivially be turned into a three-pass protocol, by sending messages m_2 and v'_B, σ_B jointly.

Verification of NIZK proof $\Pi_{n,g,e}$ in the AddUser protocol is a considerably more expensive operation. Indeed, the verifier would have to compute about $24\kappa t \log n$ (multi-)exponentiations, where 2^{-t} is the error-probability for the primality tests (see [42], Sections 4.3 and 5.1). Note that [42] suggests two optimizations of the protocol: the first one [42, Section 5.2] effectively removes factor t from the above equation; the second one [42, Section 2.2] is applicable only to interactive ZK proofs and eliminates factor κ . Nevertheless, the complexity of verifying well-formedness of group parameters remains relatively high. In practice, however, it is conceivable to completely omit the verification of $\Pi_{n,g,e}$, since the set of (public) RSA parameters of a group is fixed. Therefore, its verification by a single trusted auditing authority would suffice. An appropriate (weaker) security model is easily derived from that given in Section 3.1 by modifying the AddUserU query such that only group parameters are accepted that were previously established by a CreateGroup query. Note that, in this relaxed model, untraceability of our AHA scheme becomes unconditional (since there is no need to assume soundness of $\Pi_{n,g,e}$ in Theorem 5).

3.2.3. Security analysis

Our RSA-based AHA construction satisfies the security goals defined in Section 3.1. The proofs to corresponding Theorems 3–5 are given on pages 44, 48, and 49, respectively.

Theorem 3 *In the random oracle model, our RSA-based AHA scheme offers key security (with forward secrecy) under the RSA assumption on safe moduli, the SCDH assumption, given that Σ is EUF-CMA-secure, and given that $\Pi_{n,g,e}$ is sound and zero-knowledge.*

Theorem 4 *In the random oracle model, our RSA-based AHA scheme is affiliation-hiding under the RSA assumption on safe moduli, the SCDH assumption, given that Σ is EUF-CMA-secure, and given that $\Pi_{n,g,e}$ is sound and zero-knowledge.*

Theorem 5 *Our RSA-based AHA scheme is untraceable, given that $\Pi_{n,g,e}$ is sound.*

Remark 3 *It might seem that, due to the deployment of blind signatures in the AddUser protocol, key security of our AHA protocol cannot be shown without relying on the hardness of the ‘one-more RSA inversion’ problem [13]. However, careful examination of the constraints in our security model in Section 3.1 reveals that the AddUserG query (i.e., adversary’s access to the blind signature oracle) is available only in cases where the corresponding GA may be corrupted anyway. Hence, standard RSA assumption suffices to prove protocol’s security.*

Proof of Theorem 3. Besides to the experiments $\text{Expt}^{\text{ake},b}$ from Section 3.1.4, we will refer to a set of auxiliary games (experiments) that will help us to prove that our AHA scheme offers key security with forward secrecy. For each of these games \mathbf{G} , let $W = \Pr[\mathbf{G}(\kappa) = 1]$ denote the probability that \mathbf{G} ’s execution results in the output of 1. We will parametrize these games with a bit b and denote this with a superscript, e.g., \mathbf{G}^b .

Fix adversary \mathcal{A} and security parameter κ . We assume that, for any protocol session π , session variables $\pi.\text{partner}$ and $\pi.\text{sid}$ are set immediately after receiving the first message in the protocol (this is possible in our scheme, as opposed to Section 2.4 and Definition 15).

Observe that adversary cannot distinguish experiments $\text{Expt}^{\text{ake},0}$ and $\text{Expt}^{\text{ake},1}$ without posing exactly one `Test` query, on a fresh session π^* . In particular, we can assume that session π^* accepts during the simulation. We start by proving that this implies that there is also a session π' partnered to π^* :

Lemma 1 *In the simulation of $\text{Expt}_{\text{AHA},\mathcal{A}}^{\text{ake},b}(\kappa)$, there exists (with overwhelming probability) a session $\pi' \neq \pi^*$ such that π^* and π' compute the same session id in line 4 of the protocol, i.e., $\pi^*.\text{sid} = \pi'.\text{sid}$.*

Proof. As session π^* is fresh (cf. Definition 17), queries $\text{Corrupt}(\pi^*.\text{id})$ and $\text{Corrupt}(\pi^*.\text{partner})$ are not posed until π^* accepts. Instead of the lemma, we will prove the stronger statement that session π' exists (with overwhelming probability) already in the moment that session π^* accepts. In particular, observe that the following games need not be simulated after π^* accepts:

Game $\bar{\mathbf{G}}_0^b$. This game is identical to $\text{Exp}_{\text{AHA}, \mathcal{A}}^{\text{ake}, b}(\kappa)$.

Game $\bar{\mathbf{G}}_1^b$. Game $\bar{\mathbf{G}}_1^b$ is like Game $\bar{\mathbf{G}}_0^b$, except that the simulation is aborted if, for any pseudonym $\text{id} \in \text{IDLi}$ and any two sessions run by id , a collision of session ids occurs, i.e., if there exist sessions $\pi \neq \pi'$ with $(\pi.\text{id}, \pi.\text{sid}) = (\pi'.\text{id}, \pi'.\text{sid})$.

Observe that session ids, as assigned in line 4 of the protocol, contain value θ that is freshly and independently picked for each session and carries about $\log_2 T = \kappa + \ell$ bits of entropy. By the birthday paradox, the probability of collisions of session ids to occur is bounded by $q_s^2/T = q_s^2/2^{\kappa+\ell}$, where q_s denotes the total number of posed Handshake queries.

Game $\bar{\mathbf{G}}_2^b$. Game $\bar{\mathbf{G}}_2^b$ is like Game $\bar{\mathbf{G}}_1^b$, except that the simulator makes an a priori guess on the session that will be Test session π^* . The experiment aborts if, in the later simulation, this guess turns out to be incorrect.

Game $\bar{\mathbf{G}}_3^b$. Recall that Test session π^* is run by an honest user $\pi^*.\text{id} \in \text{IDLi}$. Game $\bar{\mathbf{G}}_3^b$ is like Game $\bar{\mathbf{G}}_2^b$, except that the simulator makes an a priori guess $\text{id}^* \in \text{IDLi}$ on the pseudonym that will be $\pi^*.\text{id}$. If this guess later turns out to be incorrect, i.e., if adversary demands Test session be run by another pseudonym, then the experiment outputs a random bit (i.e., the simulation aborts).

Game $\bar{\mathbf{G}}_4^b$. Let ID' denote the list that contains the pseudonyms of all honest users plus the pseudonyms that appear ‘on the wire’ in sessions simulated for pseudonym id^* , i.e., that appear in received first round messages $m = (\text{id}, \theta)$. We assume that ID' is initialized as $\text{ID}' \leftarrow \emptyset$, and during the simulation new entries (collected from Send queries) are appended at the end, unless they are already on the list. Clearly we have $|\text{ID}'| \leq q_a + q_s$, where q_a, q_s denote the total numbers of posed AddUserU and Handshake queries, respectively.

Game $\bar{\mathbf{G}}_4^b$ is like Game $\bar{\mathbf{G}}_3^b$, except that the simulator picks a random pointer $t \leftarrow_R \{1, \dots, |\text{ID}'|\}$ into this list. Denote by id' the t -th entry in ID' . Once the partner $\pi^*.\text{partner}$ of Test session is determined, the simulation aborts if $\pi^*.\text{partner} \neq \text{id}'$ (or id' is still undefined at that point).

As it is impossible to efficiently guess a priori pseudonym $\pi^*.\text{partner}$ that the adversary will use in Test session π^* (the adversary may send any arbitrary

string), in this game we instead guess its first occurrence in the simulation. The experiment will hence ‘learn’ id' *before* it is actually deployed.

Consider the case where pseudonym id' is honestly generated and hence $\text{Corrupt}(\text{id}')$ is not asked (cf. condition (1) in Definition 17). Assuming additionally that a session π' with $\pi^*.\text{sid} = \pi'.\text{sid}$ does not exist, then signature σ received and verified in line 8 is either invalid (and session π^* would not accept), or was forged by the adversary (as session ids sid do not repeat, cf. Game $\bar{\mathbf{G}}_1^b$). While the first case does not occur by assumption, by embedding an EUF-CMA challenge (cf. Definition 6) into pseudonym id' , we notice that the second case occurs only with a probability bounded by (negligible) $\text{Succ}_{\Sigma, \mathcal{A}'}^{\text{euf-cma}}(\kappa)$, for an adversary \mathcal{A}' , and does not need be considered in the following analysis. In particular, we may conclude that either pseudonym id' is not honestly generated or that session π' does exist.

Game $\bar{\mathbf{G}}_5^b$. Game $\bar{\mathbf{G}}_5^b$ is like Game $\bar{\mathbf{G}}_4^b$, except that the simulator makes an a priori guess on group $G \in \text{GLi}$ such that session π^* is executed in this group. If the guess on G later turns out to be incorrect, then the experiment outputs a random bit (i.e., the simulation aborts).

Game $\bar{\mathbf{G}}_6^b$. Let r^* be the value r computed in Test session π^* , in line 5. Game $\bar{\mathbf{G}}_6^b$ is like Game $\bar{\mathbf{G}}_5^b$, except that all confirmation messages v, v' and keys K (line 6) computed in session π^* and all sessions π' with $\pi^*.\text{sid} = \pi'.\text{sid}$, are consistently replaced by random values in the respective range.

Observe that all named confirmation tags and keys are computed from r^* by hashing this value, using hash function H_1 . By the random oracle model, adversary can detect the difference between Games $\bar{\mathbf{G}}_5^b$ and $\bar{\mathbf{G}}_6^b$ only by querying (a string that contains) r^* to this oracle. However, the probability of this to happen can be bounded by $\text{Succ}_{\text{SRSA-GEN}}^{\text{srsa}}$ (cf. Definition 3), as discussed in Section 2.5.1.

$$|\Pr[\bar{W}_6^b] - \Pr[\bar{W}_5^b]| \leq c \cdot \text{Succ}_{\text{SRSA-GEN}, \mathcal{A}'}^{\text{srsa}}(\kappa)$$

(for an adversary \mathcal{A}' and a constant c).

In particular, by embedding an SRSA challenge (n, e, z) into parameters n, g, e of group G and into pseudonym id' , a solution to the challenge can be computed from any hash query on r^* . Moreover, the actions of all (honest) users continue to be simulatable, with the exception that pseudonym id' cannot be ‘corrupted’. As, in our setting, group G is honest and $\text{AddUserG}(G, \text{id}')$ is not posed (cf. condition (2) in Definition 17), this is unproblematic. For further details on the reduction we refer to Section 2.5.1, Appendix A, and to the analysis by Gennaro, Krawczyk, and Rabin [82].

Observe that embedding an SRSA challenge into G 's parameters for which the factorization is not known requires the simulator to forge proof $\Pi_{n,g,e}$. However, in the random oracle model, it is possible to simulate (i.e., forge) non-interactive ZK proofs for arbitrary statements.

Now, in Game $\bar{\mathbf{G}}_6^b$, if no session π' exists such that $\pi^*.sid = \pi'.sid$, then verification tags v, v' computed by session π^* in line 6 are random and completely independent from the rest of the simulation (recall from Game $\bar{\mathbf{G}}_1^b$ that session ids sid do not repeat). Hence, an upper bound for the probability that π^* will accept is given by $1/T = 2^{-(\kappa+\ell)}$ (due to the equality check in line 8). However, this contradicts our assumption that π^* accepts with probability 1. We conclude that a session π' with $\pi^*.sid = \pi'.sid$ exists. \square

Given the result from Lemma 1, the proof for key security is straight forward. Consider the following games:

Game \mathbf{G}_0^b . This game is identical to $\text{Expt}_{\text{AHA}, \mathcal{A}}^{\text{ake}, b}(\kappa)$.

Our goal is to show that $|W_0^0 - W_0^1|$ is bounded by a negligible function.

Game \mathbf{G}_1^b . Due to Lemma 1, there exists a session $\pi' \neq \pi^*$ such that $\pi^*.sid = \pi'.sid$. Game \mathbf{G}_1^b is like Game \mathbf{G}_0^b , except that the simulator makes a priori guesses on these sessions. The experiment aborts if, in the later simulation, one of the guesses on π^*, π' turns out to be incorrect.

Game \mathbf{G}_2^b . Game \mathbf{G}_2^b is like Game \mathbf{G}_1^b , except that keys K in sessions π^*, π' are assigned via $K \leftarrow K'$, where $K' \in_R \{0, 1\}^\ell$ is a fixed but random string.

It can be generally assumed that the simulation is aware of the factorization of all SRSA moduli n corresponding to the groups $G \in \text{GLI}$, even if some moduli n are provided by the adversary. This follows from soundness of ZK proof $\Pi_{n,g,e}$ and a standard rewinding argument that extracts the factors from the proof. Let G denote the group corresponding to session π^* and (n, g, e) its parameters. In addition, let $n = pq$ be the factorization of modulus n , where p, q are safe primes. Without loss of generality, let $\langle g \rangle_p = QR(p)$ and $\langle g \rangle_q = \mathbb{Z}_q^\times$.

The modification introduced in Game \mathbf{G}_2^b can be detected by an adversary only by posing an H_1 query on (a string that contains) Diffie-Hellman value $g^{2ex_A x_B}$ (line 5), where $g^{x_A}, g^{x_B} \in \langle g \rangle_n$ are the values used for mounting $\pi^*.sid$'s values θ' (line 2, see also equation (2.1) in Section 2.5.1). Using CRT (cf. Section 2.1.1), we embed an SCDH challenge (cf. Definition 2) in group $QR(p) = \langle g \rangle_p$ into $g^{x_A}, g^{x_B} \in \langle g \rangle_n$, using, as counterparts in \mathbb{Z}_q^\times , values $g^{y_A}, g^{y_B} \in \langle g \rangle_q$,

for known $y_A, y_B \in \mathbb{Z}_{q-1}$. This technique bounds the probability of adversary \mathcal{A} asking a H_1 query on $g^{2ex_Ax_B}$ by a negligible function:

$$|\Pr[W_2^b] - \Pr[W_1^b]| \leq \text{Succ}_{\mathcal{A}'}^{\text{scdh}}(\kappa) \quad (\text{for an adversary } \mathcal{A}').$$

Note that exponent $2e \in \mathbb{Z}_{\varphi(n)}$ can readily be removed from $g^{2ex_Ax_B}$ if factorization $n = pq$ is known.

As key K of session π^* is randomly chosen in Game \mathbf{G}_2^b and the adversary is not allowed to pose **Reveal** queries to neither π^* nor π' (due to condition (a) in Definition 17), we have $\Pr[W_2^0] = \Pr[W_2^1]$. Putting everything together, we note that $\text{Adv}_{\text{AHA}, \mathcal{A}}^{\text{ake}}(\kappa) = |W_0^0 - W_0^1|$ is bounded by a negligible function, provided that the required assumptions hold. \square

Proof of Theorem 4. Recall that affiliation-hiding security (Definition 16) is defined in respect to a simulator \mathcal{SIM} that generates messages that are indistinguishable from real protocol messages. For the following proof we require \mathcal{SIM} to act as follows:

$\mathcal{SIM}.\text{Handshake}(1^\kappa, \text{id}, r)$

Recall that, according to the specification of **AddUser** protocol (cf. Figure 3.2), pseudonym id coincides with a verification key pk of signature scheme Σ . Let sk denote the signing key corresponding to id and observe that sk is known to the challenger in experiment $\text{Expt}^{\text{ah}, b}$. We consider sk as part of id , i.e., we assume that \mathcal{SIM} implicitly obtains this key in its parameters (note that knowledge of sk does not give \mathcal{SIM} a hint about the affiliation to simulate, as this key is generated independently of the group to which id is registered). Under this premise, algorithm $\mathcal{SIM}.\text{Handshake}$ proceeds as follows: it picks a random value $\theta \leftarrow_R [0, T - 1]$, sets $st \leftarrow (1^\kappa, \text{id}, \text{sk}, r, \theta)$ and $M \leftarrow (\text{id}, \theta)$, and outputs (st, M) .

$\mathcal{SIM}.\text{Send}(st, M)$

This algorithm parses $(1^\kappa, \text{id}, \text{sk}, r, \theta) \leftarrow st$, sets $\text{sid} \leftarrow (\text{id}, \theta) \parallel M$ or $\text{sid} \leftarrow M \parallel (\text{id}, \theta)$ (according to role r) and $\sigma \leftarrow \Sigma.\text{Sign}(\text{sk}, \text{sid})$, picks $v \leftarrow_R \{0, 1\}^\ell$, and outputs (v, σ) .

Given this specification of \mathcal{SIM} , the difference between experiments $\text{Expt}_{\text{AHA}, \mathcal{A}, \mathcal{SIM}}^{\text{ah}, 0}$ and $\text{Expt}_{\text{AHA}, \mathcal{A}, \mathcal{SIM}}^{\text{ah}, 1}$ from the point of view of the adversary manifests itself at at most three points: (a) the distributions of θ s in first protocol messages might differ (observe that the remaining part of first messages, id , is perfectly simulated); (b) the distributions of confirmation tags v in second protocol messages might differ (observe that the remaining part of second messages, σ , is perfectly simulated); (c) established

session keys $\pi.\text{key}$ might be inconsistent with real ones (cf. specification of `Reveal` query in Section 3.1.3).

In respect to (a), the difference between $\text{Expt}^{\text{ah},0}$ and $\text{Expt}^{\text{ah},1}$ is bounded by the (q_s -fold, where q_s denotes the total number of posed `Handshake` queries) statistical difference between the two methods to generate θ . As discussed in Section 2.5.1 and [95], this difference is negligible.

Case (c) is just an unusual verbalization of the requirement of secure key establishment: The latter states that any adversary is unable to distinguish real session keys from random ones, unless it actively takes part in protocol sessions as a member of the corresponding group (what is excluded by the model, as *STM* only simulates intact groups, see Definition 14 and Section 3.1.3). Key security of our AHA protocol is formally established in Theorem 3.

That the adversary cannot tell apart $\text{Expt}^{\text{ah},0}$ and $\text{Expt}^{\text{ah},1}$ from confirmation tags v (case (b)) follows as well from key security, and from the simple observation that keys and confirmation messages are computed using the same hash function query (cf. line 6 in Figure 3.3). \square

Proof of Theorem 5. It is well known that the RSA blind signature scheme offers unconditional privacy [49]. In fact, the deployed blinding method, also used in the `AddUser` protocol in Figure 3.2, is exactly the application of one-time pad encryption in \mathbb{Z}_n^\times and hence offers perfect secrecy (note that, if n, e are RSA parameters and $r \in_R \mathbb{Z}_n^\times$ is picked uniformly at random, then also $r^e \bmod n$ is uniformly distributed in \mathbb{Z}_n^\times). It follows that GAs cannot learn any information about registering users from `AddUser` protocol sessions. In particular, `id` and $H_n(\text{id})$ are kept completely hidden from GAs.

However, this line of argumentation holds only if n, e are indeed valid RSA parameters. More precisely, in order to guarantee untraceability of users, we have to ensure that (a) random elements in \mathbb{Z}_n are, with overwhelming probability, also elements in \mathbb{Z}_n^\times , and (b) exponentiation by e is a bijective operation, i.e., $e \in \mathbb{Z}_{\varphi(n)}^\times$. Both these requirements are ensured to hold by the proof of well-formedness of parameters (n, g, e) , i.e., by $\Pi_{n,g,e}$. We may conclude that

$$\text{Adv}_{\text{AHA}, \mathcal{A}}^{\text{trace}}(\kappa) \leq \text{Succ}_{\Pi_{n,g,e}, \mathcal{A}'}^{\text{snd}}(\kappa) \quad (\text{for an adversary } \mathcal{A}'),$$

where $\text{Succ}_{\Pi_{n,g,e}, \mathcal{A}'}^{\text{snd}}$ denotes the soundness error of ZK proof $\Pi_{n,g,e}$. \square

3.2.4. Proving well-formedness of RSA parameters

Let $n \in \mathbb{N}$ and $g, e \in [0, n-1]$. We briefly describe how to construct a (NI)ZK proof $\Pi_{n,g,e}$ that certifies in zero-knowledge that the following three conditions on n, g, e are satisfied:

- (a) $n = pq$ for $p = 2p' + 1$ and $q = 2q' + 1$ such that p, q, p', q' are primes (cf. Definition 3);
- (b) e is a valid RSA exponent for n , i.e., there exists $d \in \mathbb{N}$ such that $ed = 1 \pmod{\varphi(n)}$;
- (c) $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$, i.e., $\text{ord}_n(g) = \varphi(n)/2$ and $-1 \notin \langle g \rangle_n$.

We assume that the prover has access to the factorization of n (and, thus, can compute $\varphi(n)$ and d). In contrast, the verifier, on input $n, g, e, \Pi_{n,g,e}$, shall not learn any information besides this input.

A solution for condition (a) is given by Camenisch and Michels in [42] and builds on Pedersen's commitments [136]. We provide a high-level description of their techniques and describe how they can be extended to also address conditions (b) and (c).

The basic idea in [42] is to represent natural numbers $a \in \mathbb{N}$ by corresponding commitments $C(a) = h_1^a h_2^a$, where h_1, h_2 are independent generators of a cyclic group, and a is a random exponent. Remarkably, simple arithmetic computations on committed values can be performed given just the commitments. In particular, [42] describes how to do this for the operations listed below. In addition, it is shown how correctness of these operations can be efficiently established by dedicated ZK proofs. Note that the (perfect) hiding property of the commitment scheme guarantees for the secrecy of all committed values.

Basic operations (addition, subtraction, multiplication, exponentiation)

Let $C(a), C(b), C(n)$ be commitments on $a, b, n \in \mathbb{N}$, respectively. Operations ADD and MADD compute the addition and modular addition, respectively, on these committed values, i.e., for

$$C(c_1) \leftarrow \text{ADD}(C(a), C(b)) \quad \text{and} \quad C(c_2) \leftarrow \text{MADD}(C(a), C(b), C(n))$$

we have $c_1 = a + b$ and $c_2 = a + b \pmod{n}$. Operations for (modular) subtraction (M)SUB and multiplication (M)MUL are defined analogously. By composing several of these operations, we also achieve a protocol for modular exponentiation, defined by $C(c) \leftarrow \text{MPOW}(C(a), C(b), C(n))$, such that $C(c)$ is a commitment on $c = a^b \pmod{n}$.

Primality tests

(Pseudo)primality of a committed number $C(p)$ can be verified using Lehmann's primality test [115]. The corresponding ZK proof internally deploys multiple MADD, MMUL, and MPOW operations and is denoted by $\text{pseudoprime}(C(p))$.

Given these operations (and the respective ZK proofs that certify their correct execution), it is straight forward to construct a proof that a committed safe prime

$p = 2p' + 1$ has indeed this form. Indeed, it suffices to check the following relation:

$$\text{pseudoprime}(C(p)) \wedge C(p) = \text{ADD}(\text{MUL}(C(2), C(p')), C(1)) \wedge \text{pseudoprime}(C(p')) .$$

Let us denote this predicate by $\text{safeprime}(C(p))$. Putting everything together, the proof that a committed number n is an SRSA modulus in the sense of condition (a) can be mounted like this:

$$C(n) = \text{MUL}(C(p), C(q)) \wedge \text{safeprime}(C(p)) \wedge \text{safeprime}(C(q)) .$$

Condition (b) is easily handled by committing to $\varphi(n) = 2p'q'$ and to exponents e and $d = e^{-1} \bmod \varphi(n)$, and by opening the following commitment to the value 1:

$$\text{MMUL}(C(e), C(d), C(\varphi)) \equiv 1 .$$

In practice, however, one would presumably pick exponent $e = 3$. In this case, condition (b) is always fulfilled.

Let us now address condition (c). Note that a safe RSA modulus n is in particular a Blum integer, i.e., every quadratic residue $a \in QR(n) \subseteq \mathbb{Z}_n^\times$ has exactly four distinct square roots. This also holds for $a = 1$, and the set of elements b satisfying $b^2 = 1 \pmod{n}$ is given by $\{\pm 1, \pm \omega\}$, for some $\omega \in \mathbb{Z}_n^\times$. Let x, y be integers satisfying $px + qy = 1$, where $n = pq$. Note that these integers can be found using Euclid's algorithm. By the Chinese Remainder Theorem (cf. Section 2.1.1), it is easily seen that ω is given by $\omega = \pm(px - qy) \bmod n$. Now, the set of all $g \in \mathbb{Z}_n^\times$ that satisfy $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$ is identical to the set of all $g \in \mathbb{Z}_n^\times$ that satisfy $g^{p'q'} = \pm \omega$ (recall $\lambda(n) = \varphi(n)/2 = 2p'q'$). Summing up, a ZK proof for condition (c) can be constructed as follows:

- (1) Compute x, y such that $px + qy = 1$. Publish commitments $C(x), C(y)$ and prove

$$\text{ADD}(\text{MUL}(C(p), C(x)), \text{MUL}(C(q), C(y))) \equiv 1 .$$

- (2) Compute $\omega \leftarrow px - qy \bmod n$. Publish $C(\omega)$ and prove

$$C(\omega) = \text{MSUB}(\text{MUL}(C(p), C(x)), \text{MUL}(C(q), C(y)), C(n)) .$$

- (3) Compute $h \leftarrow g^{p'q'}$. Publish $C(h)$ and prove

$$C(h) = \text{MPOW}(C(g), \text{MUL}(C(p'), C(q')), C(n)) .$$

- (4) Show $h = \pm \omega$ by opening one of the following commitments:

$$\text{SUB}(C(h), C(\omega)) \equiv 0 \vee \text{ADD}(C(h), C(\omega)) \equiv 0 .$$

Protocol	Security & Privacy				Revocation of	Complexity		
	AKE ¹	FS ²	AH ³	UT ⁴		Transf. bits ⁵	# passes ⁶	# exps ⁷
JKT [95] Sect. 3.2	hGA	✓	hGA	✗	users,pseudonyms	1264	3	2 long
	cGA	✓	cGA	✓	pseudonyms	1504	3	2 long + 3 short
¹ key security; ² forward secrecy; ³ affiliation-hiding security; ⁴ untraceability; ⁵ total number of sent bits per protocol run; ⁶ number of message passes per protocol run; ⁷ number of exponentiations per protocol run (with short or long exponents)								

Table 3.1.: Security and performance comparison of AHA protocols

3.3. Comparison of protocols

Table 3.1 compares security, privacy, and efficiency of our RSA-based AHA protocol from Section 3.2 with the scheme [95] it is based on. We see that, in respect to key security, forward secrecy (FS) is provided by both protocols, presuming honest behavior of GA — denoted by hGA — for [95] (otherwise, small subgroup attacks [117] would be possible, within others). In contrast, our protocol offers key security with forward secrecy even in the presence of corrupt GAs — denoted by cGA. As the user registration process in [95] is not blinded, the protocol cannot provide affiliation-hiding security if GAs are corrupt (malicious GAs could record AddUser transcripts and later recognize affiliated pseudonyms). The same holds for untraceability of users. In our protocol, however, both properties are given even in the presence of corrupt GAs. As pointed out in the introduction to this chapter, in untraceable schemes, revocation can only be performed based on pseudonyms.

We also compare the message and bandwidth complexity of the specific Handshake protocols. We assume RSA moduli of length 1024 bits. Moreover, we assume that ECDSA signatures [101] are deployed in the scheme from Section 3.2. The lengths of pseudonyms and key confirmation messages are assumed to be 80 bits. Although we admit that the protocol from [95] slightly outperforms our proposal, we still conclude that our strengthened scheme is competitive with the established solution.

Strategies towards multigroup AHA

The affiliation-hiding property provided by AHA protocols is meaningful only if multiple groups are present in the system. Since users might possibly belong to several groups at the same time, the inherent problem in practice is not to decide whether two given users are members of the same single group, but rather whether there is a non-empty intersection between the two sets of groups to which the users belong. Current AHA protocols [9, 47, 95, 96, 99, 161] ignore the latter problem by design, i.e., the Handshake execution is typically performed with respect to only a *single* input group per participant and session. Little attention has been paid so far to possible solutions for the more general *group discovery problem* [95, p. 356]. A protocol that solves the problem of group discovery would take as input a *set of groups* per participant and session, output the intersection of these sets, and, in the case that this intersection is not empty, provide a session key to the users for protection of their subsequent communication. One of the main challenges here is to prevent that participants inadvertently reveal non-matching groups from their input sets to each other or to outsiders.

A simple approach to group discovery is to execute a single-group protocol for each possible combination of group memberships, and, whenever some session is successful, the corresponding group is added to the intersection set. Clearly, this solution is highly inefficient. A further challenge would be the computation of the session key in a way that ensures that leakage of this key does not reveal any information about groups in the intersection set. Motivated by the importance of group discovery for the practical deployment of AHA, we highlight in this section the main related challenges and explore various strategies to attack the group discovery problem. We will see that, in comparison to the already mentioned trivial approach, the ad-hoc application of more sophisticated but ‘standard’ tools can result in less secure schemes.

In order to illustrate different approaches, we briefly introduce the setting of group discovery in AHA protocols. Consider a total of N different groups, G_1, \dots, G_N ,

that are managed by distinct GAs. We assume that user U_1 with pseudonym id_1 is a registered member of n_1 groups, i.e., U_1 holds a set $\{\text{sk}_{G_i}[\text{id}_1]\}_i$ of n_1 different membership credentials. Similarly, U_2 is a registered member of n_2 groups holding own set $\{\text{sk}_{G_j}[\text{id}_2]\}_j$ of n_2 credentials. To simplify the exposition, let us further assume that $n_1 = n_2$ and use the notation $n = n_1 = n_2$.

At a high level, the goal of group discovery in AHA protocols is to execute a **Handshake** session between U_1 and U_2 such that at the end of the session (a) the users identify the subset of groups for which both have respective membership credentials (without disclosing information about any other credentials they possess), and (b) if this subset of groups is non-empty, then the two users agree on a secret key. Current AHA protocols admit exactly one input group per **Handshake** participant and session, basically allowing for privacy-preserving matching of input groups G_i and G_j used by U_1 and U_2 , respectively. In our description, we will utilize this ability of U_1 and U_2 to execute such single-group AHA protocols using any of their membership credentials.

4.1. The naïve approach

For completeness, we repeat the trivial solution proposed above. The idea is that U_1 and U_2 use a single-group AHA protocol for any possible combination of their group memberships. This requires n^2 different AHA sessions, what might be a too large overhead in practice (even if sessions are carried out in parallel).

4.2. (Authorized) Private Set Intersection

We investigate whether the group discovery problem can be solved in a generic way using *private set intersection* (PSI) protocols, such as [5, 59, 61, 63, 78, 87, 88, 98, 100, 107]. In the PSI setting, users have on input individual sets of elements, and the goal of the protocol is to allow users to learn the intersection of these sets without disclosing any information about further elements. One might attempt to design a group discovery protocol by simply letting group credentials be random nonces from a large domain (but identical values are assigned to all members of a group), and by using a PSI protocol to check if given two users have matching nonces. With this solution, however, a number of problems arise: (a) providing the same credential to all group members precludes member revocation since users can trivially create and admit new members to their groups by duplicating their nonces, without GA noticing it, (b) as consequence, the proposed technique leads to an AHA protocol which is not affiliation-hiding in the sense defined in our model in Section 5.3, and (c) although PSI protocols with linear computational overhead are known [63, 88],

our group discovery protocols presented in later sections can be implemented more efficiently as they rely on simpler building blocks.

A related class of protocols [58,63], called *authorized PSI* (APSI), strengthens the requirements of PSI protocols in that users' inputs must contain authorized elements only, i.e., elements that have been previously certified by some trusted authority. A technique for computing the intersection of certified sets has been introduced in [44]. One may think that authorization of elements in APSI corresponds to the registration process of users to groups in AHA protocols. However, the APSI setting assumes that the *same* authority certifies all elements in the input sets. In contrast, the AHA setting explicitly requires existence of *multiple* independent GAs providing users with membership certificates. In addition to that, problem (a) in the PSI setting (support of revocation) also applies here.

4.3. Reducing the overhead by using hashing

A possible improvement over the naïve approach from Section 4.1 in respect to reduction of overhead involves the usage of hashing. We describe here only the basic ideas of this solution, since our main focus is on a more efficient approach based on a new encoding technique we introduce in Section 4.5.

In the hashing-based approach, the parties U_1 and U_2 use a common random hash function h , which either is chosen in advance or is jointly defined by the two parties. The hash function maps arbitrary values to an output in the range $[1, B]$, $B \in \mathbb{N}$, namely into one of B bins. Each party then assigns its membership credential in group G_i into bin $h(i)$. Now, when U_1 and U_2 meet, they need not run the AHA protocol between each of the $\binom{n}{2}$ combinations of their potential input groups. Instead, the protocol needs only be run between the groups that were mapped by both parties to the same bin. Indeed, for every group G_i for which both U_1 and U_2 have membership credentials, both parties map these credentials to the same bin $h(i)$ and will run an AHA protocol with these credentials.

The basic idea described above succeeds in finding every match between membership credentials of the two parties. However, in order to protect privacy, a protocol which is based on this approach must hide from each party how many credentials were mapped by the other party to each of the bins (otherwise some data is leaked; for example, if the first bin of U_1 is empty then U_2 learns that U_1 is not a member of any group G_i for which $h(i) = 1$). Hiding the number of items in every bin can be done (following [78]) by finding a bound M such that the following property holds with high probability: when n items are mapped by a random hash function to B bins, then no more than M items are mapped to any single bin. Given this bound M , each party first maps its credentials to the B bins, and then adds to each bin

that has less than M credentials additional “dummy” values, which are indistinguishable from real credentials, so that the total number of items in the bin is M . The protocol now requires to run M^2 Handshake sessions of the single-group AHA protocol for every bin, resulting in the total of BM^2 sessions. More specifically, users operate in the following way:

- (a) Users U_1 and U_2 agree on a random hash function $h(\cdot)$ that maps items to a range of size B .
- (b) User U_1 maps its credentials to bins according to $h(\cdot)$ (if more than M items are mapped to a bin then U_1 aborts the protocol, but this event should only happen with negligible probability). U_1 adds dummy items to the bins until each bin has exactly M items. Party U_2 performs the same procedure independently of U_1 .
- (c) For each bin, the parties run a single-group Handshake protocol M^2 times, for each combination of the items in their respective copies of the bin.

In order to set the right parameters, we can use the following well known fact [139, Theorem 1]:

Fact 1 *If n items are mapped at random to $B = n/\log n$ bins, then the probability that there is a bin with more than $M = O(\log n)$ items is $o(1)$.*

Clearly, communication and computational overhead of the protocol is $O(BM^2)$. Plugging in the parameters $B = n/\log n$ and $M = O(\log n)$, we get a total overhead of $O(n \log n)$, for both cases.

4.4. An attempt to further improve the overhead

The overhead of a PSI protocol based on hashing into bins was reduced in [78] by using a better hashing method — the *balanced allocation hashing*, introduced by Azar et al. in [7]. In this hashing algorithm, the function $h(\cdot)$ chooses *two* distinct bins for each item, and the item is mapped into the bin which is less occupied at the time of placement. Theorem 1.1 of [7] states that, if the number of bins is $B = n/\log \log n$, it holds with probability $1 - o(1)$ that the maximum number of items in a bin is $M = O(\log \log n)$. (It was also shown in [31] that the probability of this event not happening is exponentially small.)

This result gives rise to the following protocol: User U_1 maps its credentials into bins using the balanced allocation hashing method described above, with parameters $B = n/\log \log n$ and $M = O(\log \log n)$. Following that step, each of U_1 's credentials can be in one of two bins. User U_1 then pads each of its bins with dummy credentials

so that it contains exactly M items. Party U_1 then begins the protocol by sending the first message of the AHA session for each of the BM items in its bins. User U_2 answers in the following way: for each credential it has, it assumes that an AHA session for that credential might have been started with any of the first messages sent in each of the *two* bins to which the corresponding group is mapped by $h(\cdot)$. User U_2 then replies to each of these messages with its matching answer in the AHA session, and user U_1 completes to protocol by sending the last message (assuming a three-pass AHA protocol, as the ones from Section 2.5). The communication and computation overhead of the entire protocol is reduced to $BM^2 = O(n \log \log n)$.

However, this variant of the protocol is *insecure* as it might leak some private information: At the end of the protocol, if the two parties detect a match for a group G_i , then U_2 learns to which of the two bins the credential of G_i was mapped by U_1 . This reveals to U_2 that the other potential bin of G_i was more occupied at the time the mapping of item G_i into bins took place. This fact might leak information about the other groups in which U_1 is a member. We do not know how to provably solve this issue, and we therefore refrain from using this protocol. To conclude, we note that a similar problem arises if one attempts to use a protocol based on *Cuckoo hashing* [134].

4.5. Index-hiding message encoding

A tool that will play a central role in the group discovering AHA protocols we propose in later sections is a new primitive called *index-hiding message encoding* (IHME). By the related concept of *index-based message encoding* we understand a technique that encodes a set of input messages $m_1, \dots, m_n \in \mathcal{M}$ (where \mathcal{M} is a message space) into a single data structure \mathcal{S} . Any of these messages can individually be recovered from \mathcal{S} by addressing it via its *index*, which is arbitrarily chosen from an index set \mathcal{I} and specified at encoding time. If it is impossible for an adversary to reveal information about the deployed indices by inspecting \mathcal{S} , then the scheme is *index-hiding*. These notions are now formalized, by first giving a syntactical definition of IBME, and then a game-based definition of IHME's index-hiding property. We will motivate in Section 4.5.2 on how IHME can be used to construct a group discovering AHA protocol.

Definition 20 (Index-based message encoding) *An index-based message encoding scheme over an index space \mathcal{I} and a message space \mathcal{M} is a set $\text{IBME} = \{\text{iEncode}, \text{iDecode}\}$ of two efficient algorithms:*

$\text{iEncode}(\mathcal{P})$

On input a set \mathcal{P} of n index/message pairs, i.e., $\mathcal{P} = \{(i_1, m_1), \dots, (i_n, m_n)\} \subseteq \mathcal{I} \times \mathcal{M}$, with distinct indices $i_j, j \in [1, n]$, this algorithm outputs an encoding \mathcal{S} .

$\text{iDecode}(\mathcal{S}, i)$

On input of an encoding \mathcal{S} and an index $i \in \mathcal{I}$, this algorithm outputs a message $m \in \mathcal{M}$.

An IBME scheme is correct if $\text{iDecode}(\text{iEncode}(\mathcal{P}), i_j) = m_j$ for all $j \in [1, n]$, for all sets $\mathcal{P} = \{(i_1, m_1), \dots, (i_n, m_n)\} \subseteq \mathcal{I} \times \mathcal{M}$ with distinct indices i_j .

Informally, an IBME scheme is *index-hiding* if it hides the indices in which the messages are encoded. That is, it ensures that an attacker, who sees an encoding \mathcal{S} and might even know some of the indices and corresponding messages, cannot identify any other indices in which messages are encoded. We formalize this property in Definition 21.

Definition 21 (Index-hiding message encoding) Let $\text{IHME} = \{\text{iEncode}, \text{iDecode}\}$ denote an IBME scheme over index space \mathcal{I} and message space \mathcal{M} . Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary that participates in the experiment of Figure 4.1. The advantage of \mathcal{A} is defined as

$$\text{Adv}_{\text{IHME}, \mathcal{A}}^{\text{ihide}}(\kappa) = \left| \Pr \left[\text{Expt}_{\text{IHME}, \mathcal{A}}^{\text{ihide}, 0}(\kappa) = 1 \right] - \Pr \left[\text{Expt}_{\text{IHME}, \mathcal{A}}^{\text{ihide}, 1}(\kappa) = 1 \right] \right| .$$

We say that IHME is index-hiding if this advantage is negligible for all efficient adversaries \mathcal{A} . Moreover, IHME is perfectly index-hiding if $\text{Adv}_{\text{IHME}, \mathcal{A}}^{\text{ihide}}(\kappa) = 0$ for all (unbounded) adversaries \mathcal{A} , for all κ .

$\text{Expt}_{\text{IHME}, \mathcal{A}}^{\text{ihide}, b}(\kappa)$:

- (a) $(I_0, I_1, M, \text{state}) \leftarrow \mathcal{A}_1(1^\kappa)$ such that $I_0, I_1 \subseteq \mathcal{I}$, $|I_0| = |I_1|$, and $M \in \mathcal{M}^k$, with $I_0 \cap I_1 = \{i_1, \dots, i_k\}$ and $M = (m_1, \dots, m_k)$, for some i_j, m_j , and k . Let $n = |I_0| = |I_1|$.
(the adversary chooses two sets of n indices each, as well as, for each index i_j in the intersection of these sets, a corresponding message m_j)
- (b) let $I_b \setminus I_{1-b} =: \{i_{k+1}, \dots, i_n\}$ and $m_{k+1}, \dots, m_n \leftarrow_R \mathcal{M}$
($n - k$ further messages are chosen uniformly at random)
- (c) $\mathcal{S} \leftarrow \text{iEncode}(\mathcal{P})$ for $\mathcal{P} = \{(i_1, m_1), \dots, (i_n, m_n)\} \subseteq \mathcal{I} \times \mathcal{M}$
(the messages are encoded for the indices in I_b)
- (d) $b' \leftarrow \mathcal{A}_2(\text{state}, \mathcal{S})$
- (e) return b'

Figure 4.1.: ihide experiment

4.5.1. A construction of IHME

We propose an efficient and perfectly index-hiding construction of IHME which is based on polynomial interpolation in finite fields. Let \mathbb{F} denote an arbitrary finite field, and let $\mathcal{I} = \mathcal{M} = \mathbb{F}$. An index-hiding message encoding scheme $\text{IHME} = \{\text{iEncode}, \text{iDecode}\}$ with index space \mathcal{I} and message space \mathcal{M} is given by the following algorithms:

iEncode(\mathcal{P})

The encoding of $\mathcal{P} = \{(i_1, m_1), \dots, (i_n, m_n)\} \subseteq \mathcal{I} \times \mathcal{M} = \mathbb{F}^2$ is defined as the list $\mathcal{S} = (c_{n-1}, \dots, c_0)$ of coefficients of the polynomial $p(x) = \sum_{k=0}^{n-1} c_k x^k \in \mathbb{F}[x]$ that interpolates all points in \mathcal{P} , i.e., $p(i_j) = m_j$ for all $(i_j, m_j) \in \mathcal{P}$. Note that this polynomial exists uniquely (cf. Theorem 11 on page 92), i.e., the **iEncode** algorithm is deterministic.

iDecode(\mathcal{S}, i)

On input $\mathcal{S} = (c_{n-1}, \dots, c_0) \in \mathbb{F}^n$ and index $i \in \mathcal{I}$, this algorithm outputs $m = \sum_{k=0}^{n-1} c_k i^k$, i.e., evaluation $p(i)$ of the polynomial $p(x) \in \mathbb{F}[x]$ induced by the coefficients in \mathcal{S} .

Observe that our IHME construction is size-preserving: The total number of field elements needed to represent messages $\{m_1, \dots, m_n\}$ on the one side, and the encoding \mathcal{S} of these messages on the other side, is the same. While correctness of the construction is obvious, its index-hiding property is assured by the following theorem.

Theorem 6 (Security of IHME construction) *The proposed IHME scheme provides perfect index-hiding.*

Proof. If $I_0 = I_1$ then \mathcal{A} obviously cannot find b . Assume therefore that $I_0 \neq I_1$. Since the messages encoded for indices in $I_b \setminus I_{1-b}$ are chosen randomly, then regardless of whether $b = 0$ or $b = 1$, the coefficients seen by \mathcal{A} are of a polynomial which is random subject to the constraint that for the indices in $I_0 \cap I_1$ its values are equal to the fixed messages provided by \mathcal{A} . The distribution of the coefficients seen by \mathcal{A} is therefore independent of b , and \mathcal{A} 's advantage in the **ihide** experiment is thus 0. \square

We anticipate that we give an alternative construction of IHME in Section 6.2. In particular, we will show how to generically compose IHME schemes from other IHME schemes. Our transformation is motivated by the efficiency gain achieved by such a construction. Moreover, in Sections 6.1 and 6.2, we present various optimizations and performance measurements obtained by profiling concrete implementations of our polynomial-based IHME scheme.

4.5.2. Multigroup AHA from IHME

We briefly describe the ideas behind a solution for the group discovery problem that combines IHME with a single-group AHA scheme. However, the full specification and analysis of this construction is postponed to Section 5.4. Observe that the sketched idea is generally applicable not only to linkable AHA schemes, as treated in this thesis, but also to unlinkable ones [6, 96, 99, 150].

In our protocol, index set \mathcal{I} is identified with the set of all possible groups. User U_1 starts many single-group AHA sessions in parallel, namely one for each of the groups that it is affiliated with. The vector (m_j) of the first protocol messages generated by these Handshake instances is IHME-encoded into a single structure \mathcal{S} , using for each message the group-specific index. Encoding \mathcal{S} is sent to U_2 , who extracts the Handshake messages for only the groups it is affiliated with. Note that, for all matching groups G_i (i.e., groups in which both U_1 and U_2 are members), the first message of all Handshake instances is correctly transferred from U_1 to U_2 . The IHME technique is then independently applied to all subsequently exchanged Handshake messages.

Observe that, for the secure deployment of IHME (as per Definition 21), it is essential that messages exchanged between users in the given single-group Handshake are indistinguishable from random in $\mathcal{M} = \mathbb{F}$. This property is satisfied by some protocols, in particular by the AHA protocol from Section 2.5.1. We combine IHME and this specific AHA scheme in Section 5.4, and achieve the first construction of an AHA protocol that offers group discovery.

4.6. Comparison of strategies

Our different strategies to achieve group discovery are compared in Table 4.1. As our IHME scheme from Section 4.5.1 has zero message expansion, the communication complexity of the multi-group AHA construction sketched in Section 4.5.2 is that of $O(n)$ single-group Handshake executions. Thus, in contrast to the proposals from Sections 4.3 and 4.4, our IHME-based construction solves the group discovery problem with *linear* communication and computation complexity, not counting the computations that IHME encoding and decoding takes. Although the overhead of polynomial interpolation (as used in IHME encoding) is $O(n^2)$ multiplications in field \mathbb{F} , with regard to Table 4.1 we assume that the overhead of these operations is negligible when compared with the overhead of, say, exponentiations needed in AHA protocols. We highlight that our IHME-based group discovery scheme (asymptotically) outperforms all other suggested solutions.

Technique	Computations (Handshake invocations)	Communication	Remarks
Naïve approach	$O(n^2)$	$O(n^2)$	
Hashing into bins	$O(n \log n)$	$O(n \log n)$	
Balanced allocation hashing	$O(n \log \log n)$	$O(n \log \log n)$	not privacy preserving
AHA + IHME	$O(n)$	$O(n)$	

Table 4.1.: Solutions for group discovery with n groups per participant

In Chapter 4, we motivated the need for *group discovery* in affiliation-hiding authentication protocols: In the mAHA (multi-affiliation AHA) setting, users are envisioned to register their pseudonym to a multitude of available groups/GAs. In subsequent Handshake sessions, they provide all their credentials simultaneously, and the protocol will not only establish a secure session key, but also compute the intersection of participants' affiliations in a secure way, i.e., users learn partner's group memberships only for those groups to which they are affiliated themselves. In this section, we formalize this approach by proposing a corresponding syntax and a security model. Moreover, building on design strategies discussed in Chapter 4, we propose two efficient constructions that are secure in respect to our model.

We are not the first ones to propose a solution to the problem of group discovery; at least one other mAHA scheme supporting multiple credentials can be found in the literature. In particular, Jarecki and Liu [97] construct an *affiliation-hiding envelope* scheme (AHE) that supports multiple credentials. Additionally, they briefly describe how to construct a mAHA protocol from this primitive. We summarize their approach in the next section, but we anticipate that their scheme does not satisfy our stronger security notion of affiliation-hiding, that we will define in Section 5.3.

5.1. The mAHA scheme by Jarecki and Liu

We describe the mAHA construction by Jarecki and Liu [97], starting with a sketch of their AHE scheme. In a setting with multiple groups $\overline{G} = \{G_1, \dots, G_n\}$ that are managed by their particular group authorities (GA), an AHE scheme allows a sender S to transmit an encrypted message m to a receiver R , such that R can only decrypt the ciphertext when satisfying a ciphertext-specific authorization policy $\mathcal{P} \subseteq \overline{G}$. This policy \mathcal{P} is specified by S independently for any new encryption, and receiver R is compliant with this policy if R is affiliated to at least one group $G \in \mathcal{P}$,

i.e., if R received a credential from at least one of the corresponding GAs.

Concretely, the construction proposed in [97] is as follows: In a cyclic group setting (\mathcal{G}, g, q) (cf. Section 2.1.2), all users create their individual public/secret key pair as $\text{pk}_U = g^{x_U}$ and $\text{sk}_U = x_U$, for random $x_U \leftarrow_R \mathbb{Z}_q$. GAs have key pairs of the same form, i.e., $\text{pk}_G = g^{x_G}$, $\text{sk}_G = x_G$, for $x_G \leftarrow_R \mathbb{Z}_q$. Users U register to groups G by sending their respective secret key sk_U to the respective GA, which raises its public key pk_G to the power of sk_U , and returns this value, $r_{U,G} = g^{x_U x_G}$, together with $\Pi_{U,G}$, a NIZK proof of knowledge of discrete logarithm of this value, i.e., $\Pi_{U,G} = \text{PoK}[(\alpha) : r_{U,G} = g^\alpha]$ in the notation by Camenisch and Stadler [43]. Note that, under the DDH assumption, the elements $r_{U,G} \in \mathcal{G}$ reveal neither the identity of the user nor the identity of the issuing GA.

Given this setup, messages are sent from S to R by letting receiver R provide its $(r_{R,G}, \Pi_{R,G})$ pairs for all the groups it got credentials for, and by letting sender S , after checking validity of the $\Pi_{R,G}$, perform an ElGamal encryption to ‘public keys’ $r_{R,G}$ (actually, [97] describes a more sophisticated way of doing this, by reusing the ephemeral encryption exponent in order to bring down decryption complexity from $O(n^2)$ to $O(n)$ exponentiations, an idea pioneered in [11, 29]). The envelope is opened by receiver R by using its secret exponent x_R for ElGamal decryption. Note that the same exponent x_R serves as decryption key for all affiliations of R .

In their paper, Jarecki and Liu define multiple security notions for AHE schemes. Besides the requirement of CCA security that protects message’s secrecy under presence of a decryption oracle, privacy of sender and receiver are captured by the notions of *sender privacy* and *receiver privacy*, respectively. Basically, an AHE scheme is *sender private against outsiders* if a (malicious) receiver R cannot learn anything about the policy \mathcal{P} specified by the sender, given that R is not affiliated to *any* group in \mathcal{P} . In contrast, *sender privacy against insiders* encompasses the case where R is affiliated to some groups $\mathcal{R} \subsetneq \mathcal{P}$, but all commitments of S to groups in $\mathcal{P} \setminus \mathcal{R}$ would still remain hidden from R . The notion of *receiver privacy* assures that the affiliations of the receiver cannot be revealed by a malicious sender. The authors of [97] claim¹ that their AHE scheme is sender private against outsiders under DDH and GapDH assumption, and that it is receiver private under DDH assumption. However, they leave open the case of sender privacy against insiders (cf. [97, Section 4.1]).

Given such an AHE scheme, Jarecki and Liu construct a simple mAHA protocol by letting participants transmit encrypted nonces to each other using envelopes (where policies \mathcal{P} are set to the own affiliations), decrypt these nonces, and derive session keys from them. Clearly, this scheme does not offer forward secrecy of the key (where we admit that this issue could be fixed). Worse, as we discuss in Section 5.6,

¹The proofs for sender privacy against outsiders and receiver privacy do actually not appear in the paper [97], but are announced to be published in the full version. This version, however, does not yet seem to be available, not even on direct request to the authors.

the constructed mAHA scheme does not seem to provide the strong property of affiliation-hiding security that we define in the following sections. Moreover, we stress that the mAHA scheme in [97] is only vaguely specified, and corresponding security models and proofs are not provided at all.

5.2. Syntax of mAHA

We proceed by defining the syntax of mAHA schemes. As the general definition of (single-group) AHA from Section 2.4 assumes that users provide exactly one credential per Handshake session, in order to also cover the multi-group setting, we have to slightly adjust both the syntax of Handshake protocol invocation and the definition of mAHA correctness (cf. Definitions 11 and 12).

Handshake($U_1 \leftrightarrow U_2$)

This protocol is executed between two users, U_1 and U_2 . User U_i , $i \in \{1, 2\}$, provides as input parameters $\text{params}_i = (\text{id}_i, \mathcal{G}_i, r_i)$ and executes its individual part $\text{Handshake}'(\text{params}_i)$. It is expected that id_i is user U_i 's pseudonym, \mathcal{G}_i is a set of pairs of the form $(\text{sk}_G[\text{id}_i], G.\text{prl})$ for some groups G , and $r_i \in \{\text{init}, \text{resp}\}$. For all groups G in \mathcal{G}_i we require that $(\text{id}_i, \text{sk}_G[\text{id}_i])$ is a valid pseudonym/credential pair, obtained via **AddUser** algorithm (in particular, we require that U_i has registered the same pseudonym in all groups listed in \mathcal{G}_i). By $G.\text{prl}$ we denote the pseudonym revocation list of respective group G .

The protocol shall detect the set of groups G that both participants are member of (i.e., G is listed in both \mathcal{G}_1 and \mathcal{G}_2 , together with valid membership credentials). If there is any such group, the protocol shall accept with an established shared session key. Otherwise, it shall reject.

Users keep track of the state of created Handshake protocol sessions π through session variables that are initialized as follows: $\pi.\text{state} \leftarrow \text{running}$, $(\pi.\text{id}, \pi.\mathcal{G}) \leftarrow (\text{id}, \mathcal{G})$, (where id and \mathcal{G} are taken from params_i), $\pi.\text{key} \leftarrow \perp$, $\pi.\text{partner} \leftarrow \perp$, $\pi.\text{groups} \leftarrow \emptyset$. At some point, the protocol completes and $\pi.\text{state}$ is updated to either **rejected** or **accepted**. In the latter case, $\pi.\text{key}$ is set to the established session key (of length κ), the pseudonym of the Handshake partner is assigned to $\pi.\text{partner}$, and $\pi.\text{groups}$ holds a non-empty set of group identifiers. State **accepted** cannot be reached if the protocol partner is revoked (i.e., $\pi.\text{partner} \in G.\text{prl}$ for all groups G in \mathcal{G}).

Definition 22 (Correctness of mAHA) *Suppose that two users, U_1 and U_2 , participate in a Handshake protocol on inputs $(\text{id}_1, \mathcal{G}_1, r_1)$ and $(\text{id}_2, \mathcal{G}_2, r_2)$, respectively, and let π_1 and π_2 denote the corresponding sessions. (We assume that all credentials in \mathcal{G}_1 and \mathcal{G}_2 have been generated by appropriate **AddUser** executions). By \mathcal{G}_\cap we*

denote the set of groups that appear in both \mathcal{G}_1 and \mathcal{G}_2 with the restriction that neither id_1 nor id_2 are contained in the respective groups' revocation lists. The mAHA scheme is correct if (a) π_1 and π_2 complete in the same state which is accepted iff $\mathcal{G}_\cap \neq \emptyset$ and $r_1 \neq r_2$, and (b) if both sessions accept, then $(\pi_1.\text{key}, \pi_1.\text{partner}, \pi_1.\text{id}) = (\pi_2.\text{key}, \pi_2.\text{id}, \pi_2.\text{partner})$ and $\pi_1.\text{groups} = \pi_2.\text{groups} = \mathcal{G}_\cap$.

5.3. A security model for mAHA

We present a security model for mAHA schemes that takes into account the main challenges implied by affiliation-hiding authentication and the group discovery problem. In particular, we cover the two central security properties of mAHA: affiliation-hiding security and key security (see also Chapter 1). Both requirements are defined with regard to multiple input groups per participant and session. While the definition of the latter goal is similar to standard definitions of key security [14, 45, 95], and only minor modifications are necessary to fit the mAHA setting, the definition of affiliation-hiding security in the multi-group environment is non-standard and first introduced here.

5.3.1. Adversarial queries

In the security experiments defined below, adversary \mathcal{A} is modeled as a probabilistic algorithm that runs in polynomial time and interacts with the experiments via the following set of queries. Observe that, in contrast to Section 3.1, our mAHA model deals only with trusted GAs, i.e., queries `CreateGroup`, `AddUser`, `Corrupt(G)` for group management are not available.

`Handshake(id, \mathcal{G} , r)`

This query lets pseudonym id start a new session π of the Handshake protocol. It receives as input a set \mathcal{G} of groups G wherein the Handshake shall take place and a role identifier $r \in \{\text{init}, \text{resp}\}$ that determines whether the session will act as protocol initiator or responder. If there is a group G listed in \mathcal{G} for which id does not have a credential $\text{sk}_G[\text{id}]$ then this query is ignored. Session variable $\pi.\text{revealed}$ is initialized to false. Optionally, this query returns a first protocol message M .

`Send(π , M)`

Message M is delivered to session π . After processing M , the eventual output is given to \mathcal{A} . This query is ignored if π is not waiting for input.

`Reveal(π)`

If $\pi.\text{state} \in \{\text{accepted}, \text{rejected}\}$, this query returns $(\pi.\text{state}, \pi.\text{key}, \pi.\text{groups})$ and sets $\pi.\text{revealed} \leftarrow \text{true}$; otherwise, if $\pi.\text{state} = \text{running}$, the query is ignored.

Corrupt(id, G)

Credential $\text{sk}_G[\text{id}]$ of pseudonym id in group G is given to the adversary. Note that this query models the possibility of selective corruptions where pseudonyms id are corrupted only for specific groups.

Revoke(G , id)

This query lets GA of G include pseudonym id in its revocation list $G.\text{prl}$.

5.3.2. Affiliation-hiding security

We now define the notion of (linkable) affiliation-hiding security (AH). At a high level, the objective is to protect users from disclosing non-shared affiliations to Handshake partners. We model AH security using the indistinguishability approach: The goal of the adversary is to decide about which of two sets of affiliations, \mathcal{G}_0^* or \mathcal{G}_1^* , a specific challenge Handshake session π^* is running on. Let $\mathcal{D}^* := \Delta(\mathcal{G}_0^*, \mathcal{G}_1^*) = (\mathcal{G}_0^* \setminus \mathcal{G}_1^*) \cup (\mathcal{G}_1^* \setminus \mathcal{G}_0^*) = (\mathcal{G}_0^* \cup \mathcal{G}_1^*) \setminus (\mathcal{G}_0^* \cap \mathcal{G}_1^*)$ denote the symmetric difference between \mathcal{G}_0^* and \mathcal{G}_1^* . The adversary specifies \mathcal{G}_0^* and \mathcal{G}_1^* himself, and is allowed to invoke any number of Handshake sessions and to ask Reveal and Corrupt queries at will, provided that the pseudonyms in the groups from $\mathcal{D}^* = \Delta(\mathcal{G}_0^*, \mathcal{G}_1^*)$ are not impersonated (as this would lead to trivial attacks). This intuition is formalized in Definition 23 and Figure 5.1. Note that, to define AH security, we dropped the simulation based paradigm (cf. Section 3.1.3, and also [95]) in favor of a (more comprehensible) purely game-based approach.

Definition 23 (Affiliation-hiding security) *Let $\text{mAHA} = \{\text{CreateGroup}, \text{AddUser}, \text{Handshake}, \text{Revoke}\}$ and let $\text{Expt}^{\text{ah},0}$ and $\text{Expt}^{\text{ah},1}$ be the experiments specified in Figure 5.1. The advantage of adversary \mathcal{A} is defined as*

$$\text{Adv}_{\text{mAHA}, \mathcal{A}}^{\text{ah}}(\kappa, n, m) = \left| \Pr \left[\text{Expt}_{\text{mAHA}, \mathcal{A}}^{\text{ah},0}(\kappa, n, m) = 1 \right] - \Pr \left[\text{Expt}_{\text{mAHA}, \mathcal{A}}^{\text{ah},1}(\kappa, n, m) = 1 \right] \right| .$$

We say that mAHA is affiliation-hiding if $\text{Adv}_{\text{mAHA}, \mathcal{A}}^{\text{ah}}$ is negligible in κ (for all n, m polynomially dependent on κ), for all efficient adversaries \mathcal{A} .

In experiment $\text{Expt}^{\text{ah},b}$, conditions (1)–(3) exclude some trivial attacks on AH security. In particular, condition (1) thwarts the attack where \mathcal{A} starts a Handshake(id', \mathcal{G}' , r') session π' with $\mathcal{G}' \cap \mathcal{D}^* \neq \emptyset$, relays all messages between π^* and π' and finally asks Reveal(π^*). By protocol correctness, $\pi^*.\text{groups}$ would contain elements from \mathcal{D}^* and it would be trivial to correctly decide about bit b . Condition (2) handles the same attack, but from the point of view of π' . Condition (3) prevents \mathcal{A} from corrupting a pseudonym in a group in \mathcal{D}^* , to impersonate that pseudonym, and to decide about bit b in dependence of acceptance in a corresponding protocol execution with π^* .

$\text{Expt}_{\text{mAHA}, \mathcal{A}}^{\text{ah}, b}(\kappa, n, m)$:

- (a) the experiment creates a set of users $\{U_1, \dots, U_n\}$ and a set of corresponding pseudonyms $\text{ID} = \{\text{id}_1, \dots, \text{id}_n\}$
- (b) the experiment creates m groups $\mathcal{G} = \{G_1, \dots, G_m\}$ and registers user U_i with pseudonym id_i in group G_j , for all $(i, j) \in [1, n] \times [1, m]$
- (c) $\mathcal{A}(1^\kappa)$ interacts with all participants using the queries from Section 5.3.1; at some point, \mathcal{A} outputs a tuple $(\text{id}^*, \mathcal{G}_0^*, \mathcal{G}_1^*, r^*)$ where $\text{id}^* \in \text{ID}$, $\mathcal{G}_0^*, \mathcal{G}_1^* \subseteq \mathcal{G}$ with $|\mathcal{G}_0^*| = |\mathcal{G}_1^*|$, and $r^* \in \{\text{init}, \text{resp}\}$. Let $\mathcal{D}^* = \Delta(\mathcal{G}_0^*, \mathcal{G}_1^*)$.
- (d) the experiment invokes a $\text{Handshake}(\text{id}^*, \mathcal{G}_b^*, r^*)$ session π^* (and provides all needed credentials)
- (e) \mathcal{A} continues to interact via queries (including on session π^*) until it terminates and outputs bit b'
- (f) the output of the experiment is b' if all of the following hold; otherwise the output is 0:
 - (1) if π^* accepted and there is a Handshake session π' with $\mathcal{D}^* \cap \pi'.\mathcal{G} \neq \emptyset$ which was in state *running* while π^* was in state *running*, then no $\text{Reveal}(\pi^*)$ query was asked;
 - (2) no $\text{Reveal}(\pi')$ query was asked for any Handshake session π' with $\mathcal{D}^* \cap \pi'.\mathcal{G} \neq \emptyset$ and $\pi'.\text{partner} = \text{id}^*$ that was in state *running* while π^* was in state *running*;
 - (3) no $\text{Corrupt}(\text{id}, G)$ query with $(\text{id}, G) \in \text{ID} \times \mathcal{D}^*$ was asked.

Figure 5.1.: ah experiment

Remark 4 (Variant of affiliation-hiding security) *Observe that, in experiment $\text{Expt}^{\text{ah}, b}$, we do not pose requirements on sets $\mathcal{G}_0^*, \mathcal{G}_1^*$, except that we demand $|\mathcal{G}_0^*| = |\mathcal{G}_1^*|$. It is easily seen by a hybrid argument that a modified definition of affiliation-hiding security with the additional constraint $|\mathcal{G}_0^* \setminus \mathcal{G}_1^*| = 1 = |\mathcal{G}_1^* \setminus \mathcal{G}_0^*|$ would be equivalent to the one from Definition 23. In this case we would always have $|\mathcal{D}^*| = 2$.*

5.3.3. Key security

Key security of mAHA schemes is modeled similarly to [14, 45], where the goal of adversary \mathcal{A} is to distinguish the session key computed for a specific challenge session π^* from a random value of the same length. Adversary \mathcal{A} is allowed to freely invoke any number of Handshake sessions, to corrupt pseudonyms, and to reveal es-

tablished session keys, as long as it does not obtain the session key computed by π^* in some trivial way. For the formal definition of key security, we slightly modify the **Reveal** query from Section 5.3.1 and introduce the auxiliary **Test** query (that is dependent on a bit $b \in \{0, 1\}$). We also introduce a new session variable, $\pi.\text{tested}$, which is set to **false** upon session creation. Here is the specification of the new queries:

Reveal(π)

This query is processed as specified in Section 5.3.1 (in particular, $\pi.\text{revealed} \leftarrow \text{true}$ is assigned), unless $\pi.\text{tested} = \text{true}$ or $\pi'.\text{tested} = \text{true}$, for any session π' that is *partnered* with π (cf. Definition 24). In the latter case, the query is ignored.

Test(π)

This query is ignored if π is not *fresh* (cf. Definition 25). Otherwise, $\pi.\text{tested}$ is set to **true** and a key is returned, according to the following rule: If $b = 1$, $\pi.\text{key}$ is returned. If $b = 0$, a random element drawn uniformly from $\{0, 1\}^k$ is returned. The **Test** query may be invoked at most once.

The notions of session id, session partnering, and session freshness have proven to be indispensable in sound definitions of security of key agreement schemes:

Definition 24 (Session id, partnered session) *The session id $\pi.\text{sid}$ of a Handshake session π with $\pi.\text{state} = \text{accepted}$ is a value that uniquely identifies π in the set of all protocol instances run by $\pi.\text{id}$. Two Handshake sessions π, π' are partnered if $\pi.\text{state} = \pi'.\text{state} = \text{accepted}$ and $(\pi.\text{sid}, \pi.\text{id}, \pi.\text{partner}) = (\pi'.\text{sid}, \pi'.\text{partner}, \pi'.\text{id})$.*

Definition 25 (Session freshness) *A session π that is invoked in response to a **Handshake**($\text{id}, \mathcal{G}, r$) query is fresh if the following conditions are satisfied:*

- (a) $\pi.\text{state} = \text{accepted}$ and $\pi.\text{revealed} = \text{false}$ and $\pi'.\text{revealed} = \text{false}$ for all sessions π' that are partnered with π ;
- (b) there exists a group $G \in \pi.\text{groups}$ such that, in the moment that $\pi.\text{state} \leftarrow \text{accepted}$ was assigned, all of the following did hold:
 - (1) no **Corrupt**($\pi.\text{partner}, G$) query has been asked;
 - (2) no **Corrupt**($\pi.\text{id}, G$) query has been asked.

Conditions (a) and (b) are the usual constraints found in key secrecy models that include forward secrecy [45]. Observe that it suffices, for a session π to be considered fresh, that $\pi.\text{id}$ and $\pi.\text{partner}$ remain uncorrupted for a single common group G .

Condition (2) is not really mandatory, but permits the consideration of protocols that are not resilient to key compromise impersonation (KCI) attacks [23, 113]. We are now ready to formally define key security of mAHA schemes:

Definition 26 (Key security with forward secrecy) *Let $\text{mAHA} = \{\text{CreateGroup}, \text{AddUser}, \text{Handshake}, \text{Revoke}\}$ and let $\text{Expt}^{\text{ake},0}$ and $\text{Expt}^{\text{ake},1}$ be the experiments specified in Figure 5.2. The advantage of adversary \mathcal{A} is defined as*

$$\text{Adv}_{\text{mAHA},\mathcal{A}}^{\text{ake}}(\kappa, n, m) = \left| \Pr \left[\text{Expt}_{\text{mAHA},\mathcal{A}}^{\text{ake},0}(\kappa, n, m) = 1 \right] - \Pr \left[\text{Expt}_{\text{mAHA},\mathcal{A}}^{\text{ake},1}(\kappa, n, m) = 1 \right] \right| .$$

We say that mAHA offers key security with forward secrecy if $\text{Adv}_{\text{mAHA},\mathcal{A}}^{\text{ake}}$ is negligible in κ (for all n, m polynomially dependent on κ), for all efficient adversaries \mathcal{A} .

$\text{Expt}_{\text{mAHA},\mathcal{A}}^{\text{ake},b}(\kappa, n, m)$:

- (a) the experiment creates users U_1, \dots, U_n and corresponding pseudonyms $\text{id}_1, \dots, \text{id}_n$
- (b) the experiment creates m groups G_1, \dots, G_m and registers user U_i with pseudonym id_i in group G_j , for all $(i, j) \in [1, n] \times [1, m]$
- (c) $\mathcal{A}(1^\kappa)$ interacts with all participants using the queries from Section 5.3.1; at some point, \mathcal{A} asks $\text{Test}(\pi^*)$ to a fresh session π^* .
- (d) \mathcal{A} continues to interact via queries (including on session π^*) until it terminates and outputs bit b'
- (e) the output of the experiment is b'

Figure 5.2.: ake experiment

5.4. A mAHA construction based on RSA

We give an efficient mAHA construction that solves the group discovery problem. Our protocol builds on the RSA-based single-group AHA scheme sketched in Section 2.5.1, that allows to check for correspondence of only one group per participant and Handshake session. However, this scheme brings along the nice property that the transferred messages are indistinguishable from random to any observer that is not a group member. We exploit this fact and, by combining the protocol with IHME from Section 4.5, we achieve group discovery with (almost) linear complexity, without assuming any further building blocks.

5.4.1. Protocol specification

Let $\ell = \ell(\kappa)$ be polynomially dependent on security parameter κ . The innovative building block of our mAHA protocol is the perfect IHME scheme constructed in Section 4.5.1 and defined over an arbitrary finite field $\mathbb{F} = GF(q)$. Although, generally, finite fields exist for any prime power $q = p^n$, for concreteness and ease of exposition we will use $\mathbb{F} = GF(T)$, where T is the smallest prime number satisfying $T > 2^{\kappa+\ell}$. We admit that, in some scenarios, finite fields of the form $GF(2^{\kappa+\ell})$ could offer benefits in efficiency. The security of our scheme, however, is not dependent on the particular choice of \mathbb{F} .

Our mAHA scheme is defined in respect to a set of hash functions: For any (RSA modulus) $n \in \mathbb{N}$, let $H_n : \{0, 1\}^* \rightarrow \mathbb{Z}_n$ be a hash function. For instance, H_n can be constructed as $H_n(x) := H'(n \| x) \bmod n$, where $H' : \{0, 1\}^* \rightarrow \mathbb{Z}_{2^{\kappa+\ell}}$ denotes an auxiliary hash function. We further assume hash functions $H : \{0, 1\}^* \rightarrow [0, T - 1]$ and $H_1 : \{0, 1\}^* \rightarrow \{0, 1\}^\ell$.

We specify the different algorithms of our mAHA protocol as follows:

CreateGroup

To set up a new group, GA generates fresh RSA parameters $(n, e, d) \leftarrow_{\text{R}} \text{SRSA-GEN}(1^\kappa)$ and picks an element $g \in \mathbb{Z}_n^\times$ such that $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$ (cf. Section 2.5.1). GA sets $G.\text{prl} \leftarrow \emptyset$, and outputs $G.\text{sk} = (n, g, e, d)$ and $G.\text{prl}$.

AddUser

The credential $\text{sk}_G[\text{id}] = (n, g, e, \sigma_{\text{id}})$ corresponding to pseudonym $\text{id} \in \{0, 1\}^*$ in group G consists of parameters (n, g, e) and the RSA signature $\sigma_{\text{id}} = H_n(\text{id})^d \bmod n$ on the full-domain hash of id (cf. Section 2.2.2).

Handshake

The protocol is executed between two users, U_A and U_B , holding pseudonyms id_A, id_B and lists $\mathcal{G}_A, \mathcal{G}_B$ of pairs $(\text{sk}_G[\text{id}], G.\text{prl})$, respectively. The full specification is given in Figure 5.3, where padding function pad is defined as in Section 2.5.1. The aim of the latter is to hide moduli n_A and n_B from observers.

The lines where the numbering is formatted in bold face coincide with Figure 2.6; in particular, this includes the calculation of the $\theta = \text{pad}((-1)^b g^x \sigma_{\text{id}}, n, T)$ values (lines 3–6), the intermediate keys $r = (\theta^e H_n(\text{id})^{-1})^{2x}$ (line 16), and the confirmation messages v (lines 17 and 25). Lines 14 and 19 effectively implement user revocation. Innovative in this protocol is the parallel transmission of multiple θ and v values encoded as IHME structures \mathcal{S} and \mathcal{S}' , respectively (lines 9, 15, 21, and 24). Note the usage of RSA moduli n as group specific indices (lines 7, 15, 20, and 24). Lists \mathcal{T} and \mathcal{R} are not transmitted, but hold the inner state of the protocol.

Correctness of the protocol follows by inspection (see also equation (2.1) in Section 2.5.1), given that string X is mounted in the same order by both U_A and U_B (lines 22–27). This can be achieved by letting the corresponding FOR loop iterate in order of ascending n . Recall from Section 2.5.1 that, for any group G for which both participants provide valid credentials, the intermediate keys r_A, r_B will match.

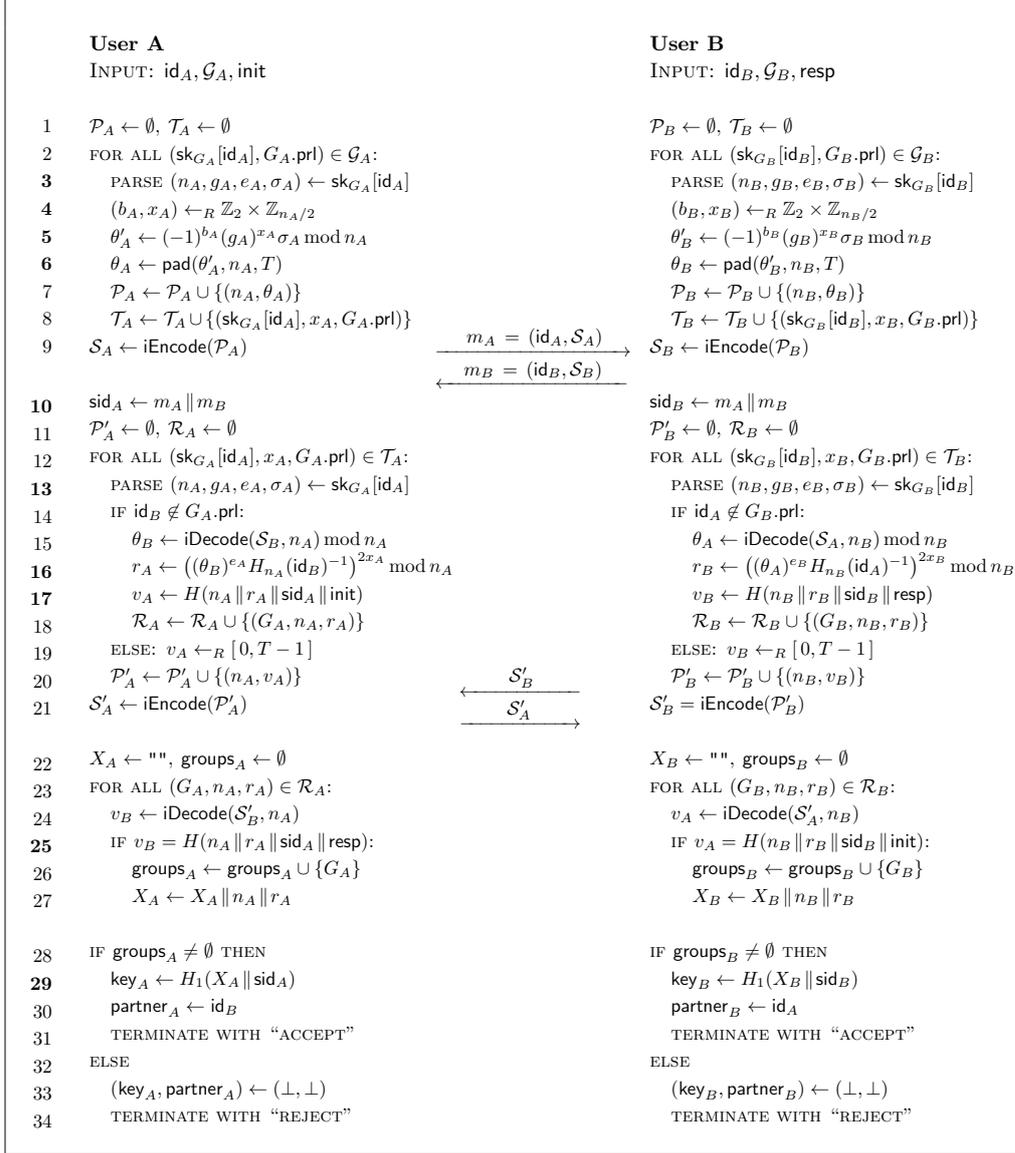


Figure 5.3.: RSA-based Handshake protocol with group discovery

Revoke

The pseudonym id to be revoked is added to G, prl . It is assumed that this list is distributed authentically to all group members.

5.4.2. Efficiency and optimizations

The computational costs of the Handshake protocol mainly consist of the exponentiations by x , which are executed twice per credential, in lines 5 and 16. More precisely, the computational effort a user executing a Handshake session has to stem can be estimated by $2n$ exponentiations (with modulus and exponent size κ), where $n = |\mathcal{G}|$ denotes the number of credentials the user provides. Observe that the revocation checks (line 14) can be implemented with logarithmic complexity, i.e., we assume the overhead is negligible. The exact performance penalty introduced by deployed IHME scheme will be investigated in full detail in Sections 6.1–6.3. However, we anticipate that the computational overhead of the protocol scales roughly linearly with the number n of provided credentials (at least for reasonable sizes of n). The size of IHME encodings $\mathcal{S}, \mathcal{S}'$ grows linearly with the number of affiliations as well. Note that the scheme is displayed as a four-message protocol for reasons of better readability. By combining messages m_B and \mathcal{S}'_B into a single datagram, the scheme can be relieved by one message transmission. For further discussions on the scheme's efficiency, including suggestions for further improvements, we refer to Sections 5.6 and 6.3.

5.4.3. Security analysis

Our mAHA construction satisfies all security goals formalized in Section 5.3. In particular, it is affiliation-hiding and offers key security with forward secrecy. The proofs to corresponding Theorems 7 and 8 are given on pages 73 and 76, respectively.

Theorem 7 *Our RSA-based mAHA scheme from Section 5.4.1 is affiliation-hiding under the RSA assumption on safe moduli, in the random oracle model.*

Theorem 8 *Our RSA-based mAHA scheme from Section 5.4.1 offers key security (with forward secrecy) under the RSA assumption on safe moduli, in the random oracle model.*

Proof of Theorem 7. Besides to the experiments $\text{Expt}^{\text{ah},b}$ from Figure 5.1 (including the modification proposed in Remark 4), we will refer to a set of auxiliary games (experiments) that will help us to prove that our mAHA scheme is affiliation-hiding. For each of these games \mathbf{G} , let $W = \Pr[\mathbf{G}(\kappa, n, m) = 1]$ denote the probability that \mathbf{G} 's execution results in the output of 1. We will parametrize these games with a bit b and denote this with a superscript, e.g., \mathbf{G}^b .

Fix adversary \mathcal{A} and parameters κ , $n = n(\kappa)$ and $m = m(\kappa)$. We assume that, for any protocol session π , session variables $\pi.\text{partner}$ and $\pi.\text{sid}$ are set immediately after receiving the first message in the protocol (this is possible in our scheme, as opposed to Section 5.2 and Definition 24). Consider the following games:

Game \mathbf{G}_0^b . This game is identical to $\text{Expt}_{\text{mAHA}, \mathcal{A}}^{\text{ah}, b}(\kappa, n, m)$.

Our goal is to show that $|W_0^0 - W_0^1|$ is bounded by a negligible function. This holds trivially if the adversary violates any of the conditions (1)–(3) in Figure 5.1, as this would directly imply $W_0^0 = W_0^1 = 0$. We hence assume in the following that adversary complies with the named conditions.

Game \mathbf{G}_1^b . Game \mathbf{G}_1^b is like Game \mathbf{G}_0^b , except that the simulation is aborted if, for any pseudonym $\text{id} \in \text{ID}$ and any two sessions run by id , a collision of session ids occurs, i.e., if there exist sessions $\pi \neq \pi'$ with $(\pi.\text{id}, \pi.\text{sid}) = (\pi'.\text{id}, \pi'.\text{sid})$.

Observe that session ids, as assigned in line 10 of the protocol, contain values θ that are freshly and independently picked for each session and carry about $\log_2 T > \kappa + \ell$ bits of entropy each. By the birthday paradox, the probability of collisions of session ids to occur is bounded by $q_s^2/T < q_s^2/2^{\kappa+\ell}$, where q_s denotes the total number of posed Handshake queries.

Game \mathbf{G}_2^b . Recall that challenge session π^* is run by an honest user $\pi^*.\text{id} \in \text{ID}$, where $|\text{ID}| = n$. Game \mathbf{G}_2^b is like Game \mathbf{G}_1^b , except that the simulator makes an a priori guess $\text{id}^* \in \text{ID}$ on the pseudonym that will be $\pi^*.\text{id}$. If this guess later turns out to be incorrect, i.e., if adversary demands challenge session be run by another pseudonym, then the experiment outputs a random bit (i.e., the simulation aborts).

Game \mathbf{G}_3^b . Let ID' denote the list that contains the pseudonyms of all honest users plus the pseudonyms that appear ‘on the wire’ in sessions simulated for pseudonym id^* , i.e., that appear in received first round messages $m = (\text{id}, \mathcal{S})$. We assume that ID' is initialized as $\text{ID}' \leftarrow \text{ID}$, and during the simulation new entries are appended at the end, unless they are already on the list. Clearly we have $|\text{ID}'| \leq n + q_s$, where q_s denotes the total number of posed Handshake queries.

Game \mathbf{G}_3^b is like Game \mathbf{G}_2^b , except that the simulator picks a random pointer $t \leftarrow_R \{1, \dots, |\text{ID}'|\}$ into this list. Denote by id' the t -th entry in ID' . Once the partner $\pi^*.\text{partner}$ of the challenge session is determined, the simulation aborts if $\pi^*.\text{partner} \neq \text{id}'$ (or id' is still undefined at that point).

As it is impossible to efficiently guess a priori pseudonym $\pi^*.\text{partner}$ that the adversary will use in challenge session π^* (the adversary may send any arbitrary string), in this game we instead guess its first occurrence in the simulation. The experiment will hence ‘learn’ id' *before* it is actually deployed.

Game \mathbf{G}_4^b . Game \mathbf{G}_4^b is like Game \mathbf{G}_3^b , except that the simulator makes an a priori guess on group G^b such that $\{G^b\} = \mathcal{G}_b^* \setminus \mathcal{G}_{1-b}^*$, out of a set of size $|\mathcal{G}| = m$.

Note that we assume the modification to experiment $\text{Expt}^{\text{ah},b}$ that is proposed in Remark 4. If the guess on G^b later turns out to be incorrect, then the experiment outputs a random bit (i.e., the simulation aborts).

Game \mathbf{G}_5^b . Let r^* be the value r computed in challenge session π^* for group G^b (line 16). Game \mathbf{G}_5^b is like Game \mathbf{G}_4^b , except that all confirmation messages v (lines 17 and 25) and keys key (line 29), that are computed in session π^* and all sessions π' with $\pi^*.\text{sid} = \pi'.\text{sid}$ in dependence on r^* , are consistently replaced by random values in the respective range.

Observe that all named confirmation tags and keys are computed from r^* by hashing this value, using hash functions H and H_1 . By the random oracle model, adversary can detect the difference between Games \mathbf{G}_4^b and \mathbf{G}_5^b only by querying (a string that contains) r^* to these oracles. However, the probability of this to happen can be bounded by $\text{Succ}_{\text{SRSA-GEN}}^{\text{srsa}}$ (cf. Definition 3), as discussed in Section 2.5.1.

In particular, by embedding an SRSA challenge (n, e, z) into parameters n, g, e of group G^b and into pseudonym id' , a solution to the challenge can be computed from any hash query on r^* . Moreover, the actions of all (honest) users continue to be simulatable, with the exception that pseudonym id' cannot be corrupted in group G^b . This behavior, however, is compliant with the rules in $\text{Expt}^{\text{ah},b}$. For further details on the reduction we refer to Section 2.5.1, Appendix A, and to the analysis by Gennaro, Krawczyk, and Rabin [82]. We conclude that, for a constant c ,

$$|\Pr[W_5^b] - \Pr[W_4^b]| \leq c \cdot \text{Succ}_{\text{SRSA-GEN}, \mathcal{A}'}^{\text{srsa}}(\kappa) \quad (\text{for an adversary } \mathcal{A}').$$

Game \mathbf{G}_6^b . Game \mathbf{G}_6^b is like Game \mathbf{G}_5^b , except that value θ for group G^b , as computed by session π^* in line 6, is replaced by a random element: $\theta \leftarrow_R [0, T - 1]$.

Observe that, in the protocol, θ is exclusively used to compute r^* in line 16 (and, correspondingly, in sessions π' with $\pi^*.\text{sid} = \pi'.\text{sid}$). As we decoupled this value from the remaining simulation in Game \mathbf{G}_5^b , the difference between W_5^b and W_6^b is bounded by the statistical difference of the two methods to generate θ . As discussed in Section 2.5.1 and [95], this difference is negligible.

Game \mathbf{G}_7^b . Game \mathbf{G}_7^b is like Game \mathbf{G}_6^b , except that, in session π^* , we replace index n , used for IHME encoding value θ in group G^b , by a fixed (unused) index, e.g., $n = 0$ (cf. lines 7 and 9).

The change introduced in Game \mathbf{G}_7^b corresponds to the security experiment of IHME's index-hiding property (cf. Figure 4.1): As θ is chosen uniformly from $[0, T - 1]$, which coincides with IHME's message space \mathcal{M} , we can readily

construct an IHME adversary \mathcal{A}' from any distinguisher between Games \mathbf{G}_6^b and \mathbf{G}_7^b . In the reduction, the set of moduli of the groups in \mathcal{G}_b^* is assigned to index set I_0 , while the set of moduli of the groups in $\mathcal{G}_b^* \setminus \{G^b\}$ together with index $n = 0$ is assigned to I_1 . As messages M corresponding to the indices in $I_0 \cap I_1$ the θ -values for the groups in $\mathcal{G}_b^* \setminus \{G^b\}$ are taken without modification. We conclude that

$$|\Pr[W_7^b] - \Pr[W_6^b]| \leq \text{Adv}_{\text{IHME}, \mathcal{A}'}^{\text{ihide}}(\kappa) \quad (\text{for an adversary } \mathcal{A}').$$

Consider, in Game \mathbf{G}_7^b , the existence of a session π' such that $\pi^*. \text{sid} = \pi'. \text{sid}$ and $\mathcal{D}^* \cap \pi'. \mathcal{G} \neq \emptyset$.

If such a session does not exist, then verification tags v assigned by session π^* for group G^b are random and completely independent from G^b and the rest of the simulation (recall the changes introduced in Game \mathbf{G}_5^b). In particular, (a) the tag v that π^* sends in lines 20 and 21 contains no information about group G^b , (b) IHME structure \mathcal{S}' that π^* sends in line 21 leaks no information about G^b (by an argument similar to the one in the hop to Game \mathbf{G}_7^b), and (c) a $\text{Reveal}(\pi^*)$ query unveils no information about G^b , as the test in line 25 corresponding to group G^b will pass only with negligible probability $1/T < 2^{-(\kappa+\ell)}$. Recall that the protocol's first message, m , sent in line 9, does not leak information about group G^b since the hop to Game \mathbf{G}_7^b .

If such a session π' does exist, then this can only be if the lifetimes of π^* and π' overlap. In this case, posing $\text{Reveal}(\pi')$ or $\text{Reveal}(\pi^*)$ queries is not allowed (cf. conditions (1) and (2) in Figure 5.1). Although the verification tag for group G^b that π^* sends in line 21 is not independent from G^b in the simulation (it is potentially also computed and expected by session π'), it is so from the point of view of the adversary, as the latter has no means to learn how this tag is processed within π' .

In any case, we observe that the adversary cannot efficiently distinguish experiments \mathbf{G}_7^0 and \mathbf{G}_7^1 , i.e., we have $W_7^0 \approx W_7^1$. Putting everything together, we note that $\text{Adv}_{\text{mAHA}, \mathcal{A}}^{\text{ah}}(\kappa, n, m) = |W_0^0 - W_0^1|$ is bounded by a negligible function, provided that the required assumptions hold. \square

Proof of Theorem 8. Besides to the experiments $\text{Expt}^{\text{ake}, b}$ from Figure 5.2, we will refer to a set of auxiliary games (experiments) that will help us to prove that our mAHA scheme offers key security with forward secrecy. For each of these games \mathbf{G} , let $W = \Pr[\mathbf{G}(\kappa, n, m) = 1]$ denote the probability that \mathbf{G} 's execution results in the output of 1. We will parametrize these games with a bit b and denote this with a superscript, e.g., \mathbf{G}^b .

Fix adversary \mathcal{A} and parameters κ , $n = n(\kappa)$ and $m = m(\kappa)$. We assume that, for any protocol session π , session variables $\pi. \text{partner}$ and $\pi. \text{sid}$ are set immediately after receiving the first message in the protocol.

Recall that, in experiment $\text{Expt}^{\text{ake},b}$, the adversary poses exactly one **Test** query, on a fresh session π^* . In particular, session π^* accepts during the simulation. We start by proving that this implies that there is also a session π' matching π^* :

Lemma 2 *In the simulation of $\text{Expt}_{\text{mAHA},\mathcal{A}}^{\text{ake},b}(\kappa, n, m)$, there exists (with overwhelming probability) a session $\pi' \neq \pi^*$ such that π^* and π' compute the same session id in line 10 of the protocol, i.e., $\pi^*.\text{sid} = \pi'.\text{sid}$.*

Proof. As session π^* is fresh (cf. Definition 25), there exists a group $G^* \in \pi^*.\text{groups}$ such that queries $\text{Corrupt}(\pi^*.\text{id}, G^*)$ and $\text{Corrupt}(\pi^*.\text{partner}, G^*)$ are not posed until π^* accepts. Instead of the lemma, we will prove the stronger statement that session π' exists (with overwhelming probability) already in the moment that session π^* executes line 26 of the protocol for group G^* (what has to occur by definition of G^*). Observe that the following games need not be simulated after π^* accepts:

Game $\bar{\mathbf{G}}_0^b$. This game is identical to $\text{Expt}_{\text{mAHA},\mathcal{A}}^{\text{ake},b}(\kappa, n, m)$.

Game $\bar{\mathbf{G}}_1^b$. Game $\bar{\mathbf{G}}_1^b$ is like Game $\bar{\mathbf{G}}_0^b$, except that the simulation is aborted if, for any pseudonym $\text{id} \in \text{ID}$ and any two sessions run by id , a collision of session ids occurs, i.e., if there exist sessions $\pi \neq \pi'$ with $(\pi.\text{id}, \pi.\text{sid}) = (\pi'.\text{id}, \pi'.\text{sid})$.

Observe that session ids, as assigned in line 10 of the protocol, contain values θ that are freshly and independently picked for each session and carry about $\log_2 T > \kappa + \ell$ bits of entropy each. By the birthday paradox, the probability of collisions of session ids to occur is bounded by $q_s^2/T < q_s^2/2^{\kappa+\ell}$, where q_s denotes the total number of posed **Handshake** queries.

Game $\bar{\mathbf{G}}_2^b$. Game $\bar{\mathbf{G}}_2^b$ is like Game $\bar{\mathbf{G}}_1^b$, except that the simulator makes an a priori guess on the session that will be **Test** session π^* . The experiment aborts if, in the later simulation, this guess turns out to be incorrect.

Game $\bar{\mathbf{G}}_3^b$. Recall that **Test** session π^* is run by an honest user $\pi^*.\text{id} \in \text{ID}$, where $|\text{ID}| = n$. Game $\bar{\mathbf{G}}_3^b$ is like Game $\bar{\mathbf{G}}_2^b$, except that the simulator makes an a priori guess $\text{id}^* \in \text{ID}$ on the pseudonym that will be $\pi^*.\text{id}$. If this guess later turns out to be incorrect, i.e., if adversary demands **Test** session be run by another pseudonym, then the experiment outputs a random bit (i.e., the simulation aborts).

Game $\bar{\mathbf{G}}_4^b$. Let ID' denote the list that contains the pseudonyms of all honest users plus the pseudonyms that appear ‘on the wire’ in sessions simulated for pseudonym id^* , i.e., that appear in received first round messages $m = (\text{id}, \mathcal{S})$. We assume that ID' is initialized as $\text{ID}' \leftarrow \text{ID}$, and during the simulation new entries are appended at the end, unless they are already on the list. Clearly we

have $|\text{ID}'| \leq n + q_s$, where q_s denotes the total number of posed Handshake queries.

Game $\bar{\mathbf{G}}_4^b$ is like Game $\bar{\mathbf{G}}_3^b$, except that the simulator picks a random pointer $t \leftarrow_R \{1, \dots, |\text{ID}'|\}$ into this list. Denote by id' the t -th entry in ID' . Once the partner $\pi^*.\text{partner}$ of Test session is determined, the simulation aborts if $\pi^*.\text{partner} \neq \text{id}'$ (or id' is still undefined at that point).

As it is impossible to efficiently guess a priori pseudonym $\pi^*.\text{partner}$ that the adversary will use in Test session π^* (the adversary may send any arbitrary string), in this game we instead guess its first occurrence in the simulation. The experiment will hence ‘learn’ id' *before* it is actually deployed.

Game $\bar{\mathbf{G}}_5^b$. Let r^* be the value r computed in Test session π^* for group G^* (line 16). Game $\bar{\mathbf{G}}_5^b$ is like Game $\bar{\mathbf{G}}_4^b$, except that all confirmation messages v (lines 17 and 25) and keys key (line 29), that are computed in session π^* and all sessions π' with $\pi^*.\text{sid} = \pi'.\text{sid}$ in dependence on r^* , are consistently replaced by random values in the respective range.

Observe that all named confirmation tags and keys are computed from r^* by hashing this value, using hash functions H and H_1 . By the random oracle model, adversary can detect the difference between Games $\bar{\mathbf{G}}_4^b$ and $\bar{\mathbf{G}}_5^b$ only by querying (a string that contains) r^* to these oracles. However, the probability of this to happen can be bounded by $\text{Succ}_{\text{SRSA-GEN}}^{\text{srsa}}$ (cf. Definition 3), as discussed in Section 2.5.1.

$$|\Pr[\bar{W}_5^b] - \Pr[\bar{W}_4^b]| \leq c \cdot \text{Succ}_{\text{SRSA-GEN}, \mathcal{A}'}^{\text{srsa}}(\kappa)$$

(for an adversary \mathcal{A}' and a constant c).

In particular, by embedding an SRSA challenge (n, e, z) into parameters n, g, e of group G^* and into pseudonym id' , a solution to the challenge can be computed from any hash query on r^* . Moreover, the actions of all (honest) users continue to be simulatable, with the exception that pseudonym id' cannot be corrupted in group G^* , what is, in our setting, unproblematic. For further details on the reduction we refer to Section 2.5.1, Appendix A, and to the analysis by Gennaro, Krawczyk, and Rabin [82].

Now, in Game $\bar{\mathbf{G}}_5^b$, if no session π' exists such that $\pi^*.\text{sid} = \pi'.\text{sid}$, then verification tags v computed by session π^* in lines 17 and 25 for group G^* are random and completely independent from the rest of the simulation (recall from Game $\bar{\mathbf{G}}_1^b$ that session ids sid do not repeat). Hence, an upper bound for the probability that π^* will execute line 26 for group G^* is given by $1/T \leq 2^{-(\kappa+\ell)}$. However, we have $G^* \in \pi^*.\text{groups}$ by assumption, i.e., line 26 is executed with probability 1, a contradiction. We conclude that a session π' with $\pi^*.\text{sid} = \pi'.\text{sid}$ exists. \square

Given the result from Lemma 2, the proof for key security is straight forward. Consider the following games:

Game \mathbf{G}_0^b . This game is identical to $\text{Expt}_{\text{mAHA}, \mathcal{A}}^{\text{ake}, b}(\kappa, n, m)$.

Our goal is to show that $|W_0^0 - W_0^1|$ is bounded by a negligible function.

Game \mathbf{G}_1^b . Due to Lemma 2, there exists a session $\pi' \neq \pi^*$ such that $\pi^*.\text{sid} = \pi'.\text{sid}$. Game \mathbf{G}_1^b is like Game \mathbf{G}_0^b , except that the simulator makes a priori guesses on these sessions. The experiment aborts if, in the later simulation, one of the guesses on π^*, π' turns out to be incorrect.

Game \mathbf{G}_2^b . Game \mathbf{G}_2^b is like Game \mathbf{G}_1^b , except that keys $\pi^*.\text{key}, \pi'.\text{key}$ of sessions π^*, π' are assigned via $\text{key} \leftarrow K$, where $K \in_R \{0, 1\}^\ell$ is a fixed but random string.

This modification can be detected by an adversary only by posing an H_1 query on (a string that contains) Diffie-Hellman value g^{2exAxB} , where g^{xA}, g^{xB} are the values contained in $\pi^*.\text{sid}$'s values θ' for group G^* , as computed in line 5 of the protocol (see also equation (2.1) in Section 2.5.1). However, by embedding a CDH challenge in group $\langle g \rangle_n$ into g^{xA}, g^{xB} , the probability of this to happen can be bounded by a negligible function:

$$|\Pr[W_2^b] - \Pr[W_1^b]| \leq c \cdot \text{Succ}_{\text{SRSA-GEN}, \mathcal{A}'}^{\text{srsa}}(\kappa)$$

(for an adversary \mathcal{A}' and a constant c).

Roughly speaking, this bound holds as the (S)RSA assumption implies hardness of factoring, which, then again, implies hardness of CDH problem in \mathbb{Z}_n^\times [8, 18, 126, 148]. The adaption to the non-classical CDH setting where g^{2exAxB} (instead of g^{xAxB}) is provided by the adversary is detailed out in [95, p. 367].

As session key $\pi^*.\text{key}$ is randomly chosen in Game \mathbf{G}_2^b and the adversary is not allowed to pose **Reveal** queries to neither π^* nor π' (due to condition (a) in Definition 25), we have $\Pr[W_2^0] = \Pr[W_2^1]$. Putting everything together, we note that $\text{Adv}_{\text{mAHA}, \mathcal{A}}^{\text{ake}}(\kappa, n, m) = |W_0^0 - W_0^1|$ is bounded by a negligible function, provided that the required assumptions hold. \square

5.5. A mAHA construction based on NIKDS

The mAHA protocol that we present in the following builds on ideas of the NIKDS-based AHA scheme sketched in Section 2.5.3. As we will see, its bandwidth consumption and its (asymptotic) computational efficiency improve on our mAHA scheme from Section 5.4.

5.5.1. Protocol specification

The central building block of our construction is a generic NIKDS scheme (cf. Definition 7). In particular, the pairing-based construction from Section 2.3.1 is suitable. Recall that the algorithms of a NIKDS are denoted NSetup , NRegister , and NGetKey . We specify our mAHA scheme as follows:

CreateGroup

To initialize a group G , GA sets up a new KGC of a NIKDS by running $\text{msk} \leftarrow \text{NSetup}(1^\kappa)$. GA sets $G.\text{prl} \leftarrow \emptyset$, and outputs $G.\text{sk} = \text{msk}$ and $G.\text{prl}$.

AddUser

Admission of a user with pseudonym $\text{id} \in \{0, 1\}^*$ is done by computing id 's credential as $\text{sk}_G[\text{id}] \leftarrow \text{NRegister}(\text{msk}, \text{id})$, where $\text{msk} = G.\text{sk}$.

Handshake

The specification of our Handshake protocol is given in Figure 5.4. Besides the NIKDS, the protocol makes use of the following additional building blocks:

- To achieve forward secrecy of the established session key, a standard Diffie-Hellman key agreement [69] is incorporated into the protocol (cf. lines 1 and 17). Hence, we require existence of a cyclic group $\mathcal{G} = \langle g \rangle$ in which the CDH problem is hard (cf. Definition 1). Let (\mathcal{G}, g, q) be such a group, generated by GGen from Section 2.1.2.
- By $H_1 : \{0, 1\}^* \rightarrow \{0, 1\}^\ell$ we denote a hash function, where $\ell = \ell(\kappa)$ is polynomially dependent on security parameter κ . It will be modeled as random oracle in the security analysis of the protocol.
- By $\text{Sort}(\mathcal{M})$, for a set $\mathcal{M} \subseteq \{0, 1\}^\ell$ of strings of length ℓ , we denote the lexicographic ordering of \mathcal{M} . It is well-known that $\text{Sort}()$ can be implemented as an $O(n \log n)$ algorithm (e.g., using ‘Heapsort’), and that look-up in an ordered set is an $O(\log n)$ operation.

Revoke(G, id)

The pseudonym id to be revoked is added to $G.\text{prl}$. It is assumed that this list is distributed authentically to all group members.

We briefly explain the design principles of the protocol from the point of view of user U_A . Note that the lines in Figure 5.4 where the numbering is formatted in bold face coincide with Figure 2.8. For all groups G in which id_A is registered (line 4) and in which id_B is not revoked (line 5), the NIKDS key K'_A shared by id_A and id_B is computed (line 6) and used to derive two authentication tags, $v_{A,0}$ and $v_{A,1}$, in lines 7 and 8 (these tags also serve for key confirmation). One of the tags is sent

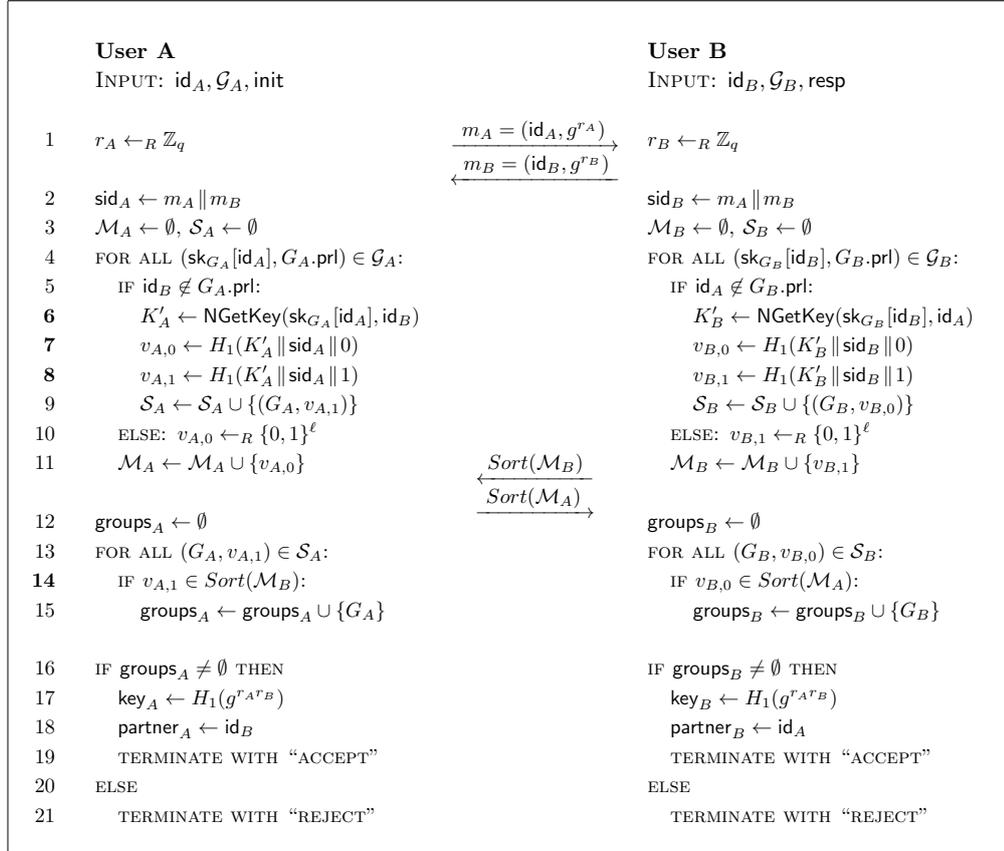


Figure 5.4.: NIKDS-based Handshake protocol with group discovery

to user U_B (line 11), while the other one is stored in state variable \mathcal{S}_A for later use (line 9). Note that user U_B computes the same tags for all groups that both users are member of. This intersection (named *groups*) is determined in lines 12–15, by recording all matches of group-specific authentication tags v . If users U_A and U_B have at least one group in common (line 16), then the protocol accepts with a secure session key (lines 1 and 17). Observe that the purpose of the sorting step (line 11) is not only to enable an $O(\log n)$ look-up of authentication tags in line 14, but also to hide the order in which these tags have been computed. This is an important prerequisite to make the scheme affiliation-hiding.

Correctness of the mAHA protocol follows from correctness of deployed NIKDS and inspection of Figure 5.4. Recall also the exposition of design rationale in Section 2.5.3.

5.5.2. Efficiency analysis

Asymptotically, the Handshake protocol in Figure 5.4 is an $O(n \log n)$ protocol, where $n = |\mathcal{G}|$ denotes the number of credentials per user. This is due to the fact that both the sorting step (line 11) and the tag-matching step (lines 13–15) are $O(n \log n)$, and

the revocation checks (line 5) can be implemented with logarithmic complexity as well (although in respect to the size of the revocation list `prl`, to be precise). However, the number of expensive operations (e.g., pairing evaluations in the NIKDS) grows only linearly in the number of affiliations. More precisely: A user that provides credentials for n groups has to evaluate n pairings to complete the protocol (or even less, when considering the possibility of revoked users), plus the two Diffie-Hellman exponentiations. Observe, however, that all NIKDS computations are session-independent and can be cached: If the same two users run the Handshake protocol multiple times, they can fall back to previously established keys K' in order to considerably save computation time. The protocol's bandwidth consumption also grows linearly in $|\mathcal{G}|$. As mostly short confirmation tags have to be transmitted, the total number of transferred bits is relatively small. Note that the scheme is displayed as a four-message protocol for reasons of better readability: To obtain a three-pass protocol, messages m_B and $Sort(\mathcal{M}_B)$ can be combined into a single datagram. We further analyze efficiency of this scheme in Sections 5.6 and 6.4.

5.5.3. Security analysis

Our mAHA construction satisfies all security goals formalized in Section 5.3. In particular, it is affiliation-hiding and offers key security with forward secrecy. The proofs to corresponding Theorems 9 and 10 are given on pages 82 and 85, respectively.

Theorem 9 *Our NIKDS-based mAHA scheme from Section 5.5.1 is affiliation-hiding given that NIKDS is OW-CIA secure, in the random oracle model.*

Theorem 10 *Our NIKDS-based mAHA scheme from Section 5.5.1 offers key security (with forward secrecy) under the CDH assumption and given that NIKDS is OW-CIA secure, in the random oracle model.*

Proof of Theorem 9. Besides to the experiments $\text{Expt}^{\text{ah},b}$ from Figure 5.1 (including the modification proposed in Remark 4), we will refer to a set of auxiliary games (experiments) that will help us to prove that our mAHA scheme is affiliation-hiding. For each of these games \mathbf{G} , let $W = \Pr[\mathbf{G}(\kappa, n, m) = 1]$ denote the probability that \mathbf{G} 's execution results in the output of 1. We will parametrize these games with a bit b and denote this with a superscript, e.g., \mathbf{G}^b .

Fix adversary \mathcal{A} and parameters κ , $n = n(\kappa)$ and $m = m(\kappa)$. We assume that, for any protocol session π , session variables $\pi.\text{partner}$ and $\pi.\text{sid}$ are set immediately after receiving the first message in the protocol (this is possible in our scheme, as opposed to Section 5.2 and Definition 24). Consider the following games:

Game \mathbf{G}_0^b . This game is identical to $\text{Expt}_{\text{mAHA},\mathcal{A}}^{\text{ah},b}(\kappa, n, m)$.

Our goal is to show that $|W_0^0 - W_0^1|$ is bounded by a negligible function. This holds trivially if the adversary violates any of the conditions (1)–(3) in Figure 5.1, as this would directly imply $W_0^0 = W_0^1 = 0$. We hence assume in the following that adversary complies with the named conditions.

Game \mathbf{G}_1^b . Game \mathbf{G}_1^b is like Game \mathbf{G}_0^b , except that the simulation is aborted if, for any pseudonym $\text{id} \in \text{ID}$ and any two sessions run by id , a collision of session ids occurs, i.e., if there exist sessions $\pi \neq \pi'$ with $(\pi.\text{id}, \pi.\text{sid}) = (\pi'.\text{id}, \pi'.\text{sid})$.

Observe that session ids, as assigned in line 2 of the protocol, contain value g^r which is freshly and independently picked for each session. Hence, a collision of session ids implies that the user, by coincidence, picked the same $r \leftarrow_R \mathbb{Z}_q$ twice, in line 1. By the birthday paradox, this happens with probability smaller than $q_s^2/|\mathbb{Z}_q| = q_s^2/q$, where q_s denotes the total number of posed Handshake queries. Note that $q = q(\kappa)$ grows super-polynomially in κ (cf. Section 2.1.2), so that this probability is negligible.

Game \mathbf{G}_2^b . Recall that challenge session π^* is run by an honest user $\pi^*.\text{id} \in \text{ID}$, where $|\text{ID}| = n$. Game \mathbf{G}_2^b is like Game \mathbf{G}_1^b , except that the simulator makes an a priori guess $\text{id}^* \in \text{ID}$ on the pseudonym that will be $\pi^*.\text{id}$. If this guess later turns out to be incorrect, i.e., if adversary demands challenge session be run by another pseudonym, then the experiment outputs a random bit (i.e., the simulation aborts).

Game \mathbf{G}_3^b . Let ID' denote the list that contains the pseudonyms of all honest users plus the pseudonyms that appear ‘on the wire’ in sessions simulated for pseudonym id^* , i.e., that appear in received first round messages $m = (\text{id}, g^r)$. We assume that ID' is initialized as $\text{ID}' \leftarrow \text{ID}$, and during the simulation new entries are appended at the end, unless they are already on the list. Clearly we have $|\text{ID}'| \leq n + q_s$, where q_s denotes the total number of posed Handshake queries.

Game \mathbf{G}_3^b is like Game \mathbf{G}_2^b , except that the simulator picks a random pointer $t \leftarrow_R \{1, \dots, |\text{ID}'|\}$ into this list. Denote by id' the t -th entry in ID' . Once the partner $\pi^*.\text{partner}$ of the challenge session is determined, the simulation aborts if $\pi^*.\text{partner} \neq \text{id}'$ (or id' is still undefined at that point).

As it is impossible to efficiently guess a priori pseudonym $\pi^*.\text{partner}$ that the adversary will use in challenge session π^* (the adversary may send any arbitrary string), in this game we instead guess its first occurrence in the simulation. The experiment will hence ‘learn’ id' *before* it is actually deployed.

Game \mathbf{G}_4^b . Game \mathbf{G}_4^b is like Game \mathbf{G}_3^b , except that the simulator makes an a priori guess on group G^b such that $\{G^b\} = \mathcal{G}_b^* \setminus \mathcal{G}_{1-b}^*$, out of a set of size $|\mathcal{G}| = m$.

Note that we assume the modification to experiment $\text{Expt}^{\text{ah},b}$ that is proposed in Remark 4. If the guess on G^b later turns out to be incorrect, then the experiment outputs a random bit (i.e., the simulation aborts).

Game \mathbf{G}_5^b . Game \mathbf{G}_5^b is like Game \mathbf{G}_4^b , except that, for all sessions π with $\{\pi.\text{id}, \pi.\text{partner}\} = \{\text{id}^*, \text{id}'\}$, confirmation messages v_0, v_1 for group G^b , as computed in lines 7 and 8 of the protocol, are assigned via $v_d \leftarrow H'_1(\text{sid} \parallel d)$, where H'_1 is a private random oracle. In particular, the v_d are assigned independently of NIKDS key K' .

As seen in Figure 5.4, the simulation of Game \mathbf{G}_4^b uses key $K' = \text{NSharedKey}(G^b.\text{sk}; \text{id}^*, \text{id}')$ exactly for the computation of the v_0, v_1 specified above, and nowhere else. The modification introduced in this game can be detected by adversary \mathcal{A} only by posing an H_1 query on (a string that contains) respective NIKDS's key K' . By embedding an OW-CIA challenge (cf. Definition 9) into group G^b and pseudonyms id^*, id' , the probability of this to happen can be bounded by a negligible function (where q_{H_1} denotes the total number of posed H_1 queries):

$$|\Pr[W_5^b] - \Pr[W_4^b]| = q_{H_1} \text{Succ}_{\text{NIKDS}, \mathcal{A}'}^{\text{ow-cia}}(\kappa) \quad (\text{for an adversary } \mathcal{A}').$$

Observe that, in this step, we exploited condition (3) from experiment $\text{Expt}^{\text{ah},b}$: Pseudonyms id^* and id' may not be corrupted in group G^b .

Consider, in Game \mathbf{G}_5^b , the existence of a session π' such that $\pi^*.\text{sid} = \pi'.\text{sid}$ and $\mathcal{D}^* \cap \pi'.\mathcal{G} \neq \emptyset$.

If such a session does not exist, then verification tags $v_d^* = H'_1(\text{sid} \parallel d)$ assigned by session π^* for group G^b are random and completely independent from G^b and the rest of the simulation (recall from Game \mathbf{G}_1^b that session ids sid do not repeat, so H'_1 is never queried on the same inputs again). In particular, (a) the messages that π^* sends contain no information about group G^b , and (b) a $\text{Reveal}(\pi^*)$ query unveils no information about G^b , as the test in line 14 corresponding to group G^b will pass only with negligible probability $|\text{Sort}(\mathcal{M})|/2^\ell \leq m/2^\ell$.

If such a session π' does exist, then this can only be if the lifetimes of π^* and π' overlap. In this case, posing $\text{Reveal}(\pi')$ or $\text{Reveal}(\pi^*)$ queries is not allowed (cf. conditions (1) and (2) in Figure 5.1). Although the verification tag for group G^b that π^* sends in line 11 is not independent from G^b in the simulation (it is potentially also computed and expected by session π'), it is so from the point of view of the adversary, as the latter has no means to learn how this tag is processed within π' .

In any case, we observe that the adversary cannot efficiently distinguish experiments \mathbf{G}_5^0 and \mathbf{G}_5^1 , i.e., we have $W_5^0 \approx W_5^1$. Putting everything together, we note that $\text{Adv}_{\text{mAHA}, \mathcal{A}}^{\text{ah}}(\kappa, n, m) = |W_0^0 - W_0^1|$ is bounded by a negligible function, provided that the required assumption on NIKDS holds. \square

Proof of Theorem 10. Besides to the experiments $\text{Expt}^{\text{ake},b}$ from Figure 5.2, we will refer to a set of auxiliary games (experiments) that will help us to prove that our mAHA scheme offers key security with forward secrecy. For each of these games \mathbf{G} , let $W = \Pr[\mathbf{G}(\kappa, n, m) = 1]$ denote the probability that \mathbf{G} 's execution results in the output of 1. We will parametrize these games with a bit b and denote this with a superscript, e.g., \mathbf{G}^b .

Fix adversary \mathcal{A} and parameters κ , $n = n(\kappa)$ and $m = m(\kappa)$. We assume that, for any protocol session π , session variables $\pi.\text{partner}$ and $\pi.\text{sid}$ are set immediately after receiving the first message in the protocol.

Recall that, in experiment $\text{Expt}^{\text{ake},b}$, the adversary poses exactly one **Test** query, on a fresh session π^* . In particular, session π^* accepts during the simulation. We start by proving that this implies that there is also a session π' matching π^* :

Lemma 3 *In the simulation of $\text{Expt}_{\text{mAHA},\mathcal{A}}^{\text{ake},b}(\kappa, n, m)$, there exists (with overwhelming probability) a session $\pi' \neq \pi^*$ such that π^* and π' compute the same session id in line 2 of the protocol, i.e., $\pi^*.\text{sid} = \pi'.\text{sid}$.*

Proof. As session π^* is fresh (cf. Definition 25), there exists a group $G^* \in \pi^*.\text{groups}$ such that queries $\text{Corrupt}(\pi^*.\text{id}, G^*)$ and $\text{Corrupt}(\pi^*.\text{partner}, G^*)$ are not posed until π^* accepts. Instead of the lemma, we will prove the stronger statement that session π' exists (with overwhelming probability) already in the moment that session π^* executes line 15 of the protocol for group G^* (what has to occur by definition of G^*). Observe that the following games need not be simulated after π^* accepts:

Game $\bar{\mathbf{G}}_0^b$. This game is identical to $\text{Expt}_{\text{mAHA},\mathcal{A}}^{\text{ake},b}(\kappa, n, m)$.

Game $\bar{\mathbf{G}}_1^b$. Game $\bar{\mathbf{G}}_1^b$ is like Game $\bar{\mathbf{G}}_0^b$, except that the simulation is aborted if, for any pseudonym $\text{id} \in \text{ID}$ and any two sessions run by id , a collision of session ids occurs, i.e., if there exist sessions $\pi \neq \pi'$ with $(\pi.\text{id}, \pi.\text{sid}) = (\pi'.\text{id}, \pi'.\text{sid})$.

Observe that session ids, as assigned in line 2 of the protocol, contain value g^r which is freshly and independently picked for each session. Hence, a collision of session ids implies that the user, by coincidence, picked the same $r \leftarrow_R \mathbb{Z}_q$ twice, in line 1. By the birthday paradox, this happens with probability smaller than $q_s^2/|\mathbb{Z}_q| = q_s^2/q$, where q_s denotes the total number of posed **Handshake** queries. Note that $q = q(\kappa)$ grows super-polynomially in κ (cf. Section 2.1.2), so that this probability is negligible.

Game $\bar{\mathbf{G}}_2^b$. Game $\bar{\mathbf{G}}_2^b$ is like Game $\bar{\mathbf{G}}_1^b$, except that the simulator makes an a priori guess on the session that will be **Test** session π^* . The experiment aborts if, in the later simulation, this guess turns out to be incorrect.

Game $\bar{\mathbf{G}}_3^b$. Recall that **Test** session π^* is run by an honest user $\pi^*.id \in \text{ID}$, where $|\text{ID}| = n$. Game $\bar{\mathbf{G}}_3^b$ is like Game $\bar{\mathbf{G}}_2^b$, except that the simulator makes an a priori guess $\text{id}^* \in \text{ID}$ on the pseudonym that will be $\pi^*.id$. If this guess later turns out to be incorrect, i.e., if adversary demands **Test** session be run by another pseudonym, then the experiment outputs a random bit (i.e., the simulation aborts).

Game $\bar{\mathbf{G}}_4^b$. Let ID' denote the list that contains the pseudonyms of all honest users plus the pseudonyms that appear ‘on the wire’ in sessions simulated for pseudonym id^* , i.e., that appear in received first round messages $m = (\text{id}, g^r)$. We assume that ID' is initialized as $\text{ID}' \leftarrow \text{ID}$, and during the simulation new entries are appended at the end, unless they are already on the list. Clearly we have $|\text{ID}'| \leq n + q_s$, where q_s denotes the total number of posed **Handshake** queries.

Game $\bar{\mathbf{G}}_4^b$ is like Game $\bar{\mathbf{G}}_3^b$, except that the simulator picks a random pointer $t \leftarrow_R \{1, \dots, |\text{ID}'|\}$ into this list. Denote by id' the t -th entry in ID' . Once the partner $\pi^*.partner$ of the **Test** session is determined, the simulation aborts if $\pi^*.partner \neq \text{id}'$ (or id' is still undefined at that point).

As it is impossible to efficiently guess a priori pseudonym $\pi^*.partner$ that the adversary will use in **Test** session π^* (the adversary may send any arbitrary string), in this game we instead guess its first occurrence in the simulation. The experiment will hence ‘learn’ id' *before* it is actually deployed.

Game $\bar{\mathbf{G}}_5^b$. Game $\bar{\mathbf{G}}_5^b$ is like Game $\bar{\mathbf{G}}_4^b$, except that, in all sessions π with $\{\pi.id, \pi.partner\} = \{\text{id}^*, \text{id}'\}$, confirmation messages v_0, v_1 for group G^* , as computed in lines 7 and 8 of the protocol, are assigned via $v_d \leftarrow H'_1(\text{sid} \| d)$, where H'_1 is a private random oracle.

As seen in Figure 5.4, the simulation of Game $\bar{\mathbf{G}}_4^b$ uses key $K' = \text{NSharedKey}(G^*.sk; \text{id}^*, \text{id}')$ exactly for the computation of the v_0, v_1 specified above, and nowhere else. The modification introduced in this game can be detected by adversary \mathcal{A} only by posing an H_1 query on (a string that contains) respective NIKDS’s key K' . By embedding an OW-CIA challenge (cf. Definition 9) into group G^* and pseudonyms id^*, id' , the probability of this to happen can be bounded by a negligible function (where q_{H_1} denotes the total number of posed H_1 queries):

$$|\Pr[\bar{W}_5^b] - \Pr[\bar{W}_4^b]| \leq q_{H_1} \text{Succ}_{\text{NIKDS}, \mathcal{A}'}^{\text{ow-cia}}(\kappa) \quad (\text{for an adversary } \mathcal{A}').$$

Now, in Game $\bar{\mathbf{G}}_5^b$, if no session π' exists such that $\pi^*.sid = \pi'.sid$, then verification tags $v_d^* = H'_1(\text{sid} \| d)$ computed by session π^* in lines 7 and 8 for group G^* are random

and completely independent from the rest of the simulation (recall from Game $\bar{\mathbf{G}}_1^b$ that session ids sid do not repeat, so H_1' is never queried on the same inputs again). Hence, an upper bound for the probability that π^* will execute line 15 for group G^* is given by $|\text{Sort}(\mathcal{M})|/2^\ell \leq m2^{-\ell}$. However, by definition of G^* , we know that line 15 is executed with probability 1, a contradiction. We conclude that a session π' with $\pi^*.\text{sid} = \pi'.\text{sid}$ exists. \square

Given the result from Lemma 3, the proof for key security is straight forward. Consider the following games:

Game \mathbf{G}_0^b . This game is identical to $\text{Expt}_{\text{mAHA}, \mathcal{A}}^{\text{ake}, b}(\kappa, n, m)$.

Our goal is to show that $|W_0^0 - W_0^1|$ is bounded by a negligible function.

Game \mathbf{G}_1^b . Due to Lemma 3, there exists a session $\pi' \neq \pi^*$ such that $\pi^*.\text{sid} = \pi'.\text{sid}$. Game \mathbf{G}_1^b is like Game \mathbf{G}_0^b , except that the simulator makes a priori guesses on these sessions. The experiment aborts if, in the later simulation, one of the guesses on π^*, π' turns out to be incorrect.

Game \mathbf{G}_2^b . Game \mathbf{G}_2^b is like Game \mathbf{G}_1^b , except that keys $\pi^*.\text{key}, \pi'.\text{key}$ of sessions π^*, π' are assigned via $\text{key} \leftarrow K$, where $K \in_R \{0, 1\}^\ell$ is a fixed but random string.

This modification can be detected by an adversary only by posing an H_1 query on Diffie-Hellman value $g^{r^A r^B}$, where g^{r^A}, g^{r^B} are the values contained in $\pi^*.\text{sid}$. However, by embedding a CDH challenge (cf. Definition 1) into g^{r^A}, g^{r^B} , the probability of this to happen can be bounded by a negligible function (where q_{H_1} denotes the total number of posed H_1 queries):

$$|\Pr[W_2^b] - \Pr[W_1^b]| \leq q_{H_1} \text{Succ}_{\text{Gen}, \mathcal{A}'}^{\text{cdh}}(\kappa) \quad (\text{for an adversary } \mathcal{A}').$$

Note that factor q_{H_1} disappears in the GapDH setting.

As session key $\pi^*.\text{key}$ is randomly chosen in Game \mathbf{G}_2^b and the adversary is not allowed to pose **Reveal** queries to neither π^* nor π' (due to condition (a) in Definition 25), we have $\Pr[W_2^0] = \Pr[W_2^1]$. Putting everything together, we note that $\text{Adv}_{\text{mAHA}, \mathcal{A}}^{\text{ake}}(\kappa, n, m) = |W_0^0 - W_0^1|$ is bounded by a negligible function, provided that the required assumptions hold. \square

5.6. Comparison of our mAHA solutions

We compare security and complexity of known mAHA schemes in Table 5.1, namely our schemes from Sections 5.4 and 5.5, the naïve approach from Section 4.1, and the

Protocol	Security & Privacy			Setting	Complexity			
	AKE ¹	FS ²	AH ³		# PK ⁴	complexity ⁵	# passes ⁶	Transf. bits ⁷
Sect. 4.1	✓	✓	✓	generic	$\geq O(n^2)$	$\geq O(n^2)$	constant	$O(n^2)$
Sect. 5.1	✓	✗	?	DDH, GapDH	$5n$	$O(n^2)$	3	$800n$
Sect. 5.4	✓	✓	✓	RSA	$2n$	$O(n^2)$	3	$2208n + 80$
Sect. 5.5	✓	✓	✓	BCDH, CDH	$n + 2$	$O(n \log n)$	3	$80n + 240$

¹key security; ²forward secrecy; ³affiliation-hiding security; ⁴number of basic public key operations (e.g., exponentiations, pairing evaluations); ⁵overall computational complexity; ⁶number of message passes per protocol execution; ⁷total number of bits sent per protocol execution

Table 5.1.: Security and performance comparison of mAHA protocols

scheme from [97] (see Section 5.1). Focusing on those schemes where the number of expensive operations (e.g., exponentiations and pairing evaluations) grows at most linearly in the number $n = |\mathcal{G}|$ of provided credentials, we notice that only the schemes from Sections 5.4 and 5.5 achieve key security with forward secrecy. It remains unclear whether the protocol by Jarecki and Liu [97] is affiliation-hiding in the strong sense of our model (cf. Section 5.3.2). This is due to the fact that sender privacy of the AHE scheme underlying their construction is only claimed to hold against outsider attacks, in contrast to the more powerful insider attacks that are allowed in our model. Moreover, all privacy-related properties of both their AHE and mAHA schemes remain unproven in their paper (see footnote in Section 5.1).

In regards to computational performance, the protocols from [97] and Section 5.4 have quadratic complexity. While in [97] the envelope’s receiver has to perform $O(n^2)$ decryptions of a symmetric CCA-secure encryption scheme, in our RSA-based scheme from Section 5.4 it is the IHME encoding that takes $O(n^2)$ field multiplications. Note that, in Chapter 6, we will investigate the practical impact of IHME’s performance in full detail. Best efficiency for large n we clearly expect from the NIKDS-based protocol from Section 5.5. Especially in respect to bandwidth consumption, our protocol from Section 5.5 impressively outperforms its competitors, as in its execution mainly (short) authentication tags have to be transferred. Observe that our NIKDS-based scheme consumes only 3.6% (respectively, 10%) of the bandwidth, when compared to our RSA-based protocol (resp., the protocol from [97]). When estimating bandwidth complexity of all protocols, we assume RSA moduli of length 1024 bit, DLP groups with 160 bit element representation, authentication tags with a length of 80 bits, and symmetric CCA-secure ciphertexts of length 160 bits.

5.7. On the feasibility of DLP-based mAHA

We conclude this section by giving intuition about the reason why the DLP-based AHA scheme from Section 2.5.2 cannot be transformed into a (DLP-based) mAHA

scheme by applying the IHME primitive from Section 4.5. Regarding the discussion in Section 4.5.2, at first sight, such a generic conversion could be expected to be straight forward. However, as we will justify in the following, this transformation seems not to be possible at all. Recall that, in the first protocol messages of the DLP-based scheme from Figure 2.7, participants exchange elements $\omega \in \mathcal{G}$ that allow recovery of ElGamal encryption keys z , by computing $z \leftarrow \omega y^{H(\omega)}$. Observe that values ω serve as individual identifiers for group members, and are generated and assigned in a randomized process in `AddUser` procedure. In particular, these tokens cannot be freely specified by the users, and users would, with overwhelming probability, receive two different and independent elements ω when registering to two different GAs. On the other hand, the IHME primitive hides indices only for random (looking) messages (cf. Definition 21). It follows that, if a single-group `Handshake` protocol comprises the transmission of any affiliation-dependent constant element (e.g., identifier ω), then a generic IHME-based conversion from AHA to mAHA will generally not result in a secure scheme. Observe that our mAHA solution from Section 5.4 exploits the fact that users may freely pick their pseudonyms in the underlying AHA scheme, i.e., users may use the same pseudonym in all groups, so that no constant data has to be IHME-transmitted.

Multigroup AHA in practice

Only little work on practical aspects of affiliation-hiding authentication schemes, pursuing optimized implementations and deployment, has been reported in the literature so far, and the main question a practitioner might ask — whether AHA schemes are truly practical today — remains widely unanswered. In this section, we analyze and optimize practical performance of our mAHA schemes from Chapter 5.

Regarding our RSA-based scheme from Section 5.4, we propose numerous algorithmic optimizations that remarkably speed up the most important operations. Results are demonstrated not only at theoretical level, but we also offer implementations, performance measurements, and comparisons. In respect to the NIKDS-based mAHA scheme from Section 5.5, we observe that the by far most time consuming operation during protocol execution is the evaluation of NIKDS’s bilinear maps (cf. Section 2.3.1). However, the improvement of efficiency of this operation is clearly out of the scope of this thesis. Hence, we abstain from optimizing the scheme itself and restrict ourselves to give expected performance values for the protocol execution.

6.1. Optimizing IHME

The central building block that we used in the construction of RSA-based mAHA is a generic IHME scheme (cf. Section 4.5). A possible instantiation of this primitive, based on polynomial interpolation in finite fields, was proposed in Section 4.5.1. Note that corresponding IHME encoding algorithm takes $O(n^2)$ field operations. Hence, the same bound applies to the full Handshake protocol. However, as field operations are rather cheap (IHME mainly uses multiplications), one may ask which role this asymptotic bound plays *in practice* for the overall performance of the protocol, where one might expect that running time is dominated by RSA exponentiations. In the following, we give several optimized algorithms for the implementation of IHME and compare them in respect to computational efficiency and memory consumption. These implementational aspects were left unconsidered in Sections 4.5 and 5.4.

The algorithms presented in the following sections make computations in a given finite field \mathbb{F} . Efficient implementation of field arithmetic and element representation is a wide field of research and out of the scope of this thesis; see Hankerson *et al.* [86, Chapter 2] for a comprehensive overview. Generally speaking, fields of small characteristic (e.g., $\mathbb{F} = GF(2^k)$ for some k) offer speed advantages on SIMD machines and dedicated hardware, while modern PCs with 32/64 bit ALUs benefit from fields of large characteristic (e.g., $\mathbb{F} = GF(p)$ for a large prime p).

In the analysis of the following algorithms, we measure computational performance by counting the number of expensive field operations, i.e., multiplications ($c \leftarrow ab$), inversions ($c \leftarrow a^{-1}$), and divisions ($c \leftarrow a/b$). As $a/b = ab^{-1}$, divisions can always be implemented at the cost of one inversion and one multiplication. However, often both operations can be conflated into a single operation of the cost of one inversion [86, Section 2.3.6]. In the following, we denote the time needed to perform a multiplication, an inversion, or a division, by M , I , and D , respectively. In practice, it is reasonable to assume $I \approx D \approx 60M$ (cf. [86, Section 5.1.5]).

6.1.1. Polynomial interpolation¹

The following well-known theorem [142] ensures existence and uniqueness of interpolation polynomials.

Theorem 11 (Polynomial interpolation) *In a field \mathbb{F} , let $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{F} \times \mathbb{F}$ be n pairs of elements satisfying $i \neq j \Rightarrow x_i \neq x_j$. Then there exists a polynomial $p \in \mathbb{F}[x]$ of degree $\deg(p) < n$ that interpolates all points (x_i, y_i) , i.e., $y_i = p(x_i)$ for all $1 \leq i \leq n$. Moreover, this polynomial exists uniquely.*

For fixed $n \in \mathbb{N}$, the set Π^n consisting of all polynomials $p \in \mathbb{F}[x]$ of degree $\deg(p) \leq n$ naturally constitutes a vector space over \mathbb{F} . Algorithms for polynomial interpolation [142] usually represent computed polynomials in Π^n by the coefficients of the corresponding linear combination of some basis elements of Π^n . While the *monomial basis* $\{1, x, x^2, \dots, x^n\}$ seems to be the most versatile one, popular interpolation algorithms do not refer to it, but instead compute coefficients in respect to specially crafted bases, that often depend on the specific problem instance. We stress that such algorithms might not serve for secure IHME implementations. For example, the bases of two well-known interpolation algorithms, namely Lagrange and Newton Interpolation, are directly dependent on deployed x -abscissas. This behavior contradicts the desired index-hiding property of IHME, as x -values (i.e., indices) would have to be included in IHME structures \mathcal{S} .

¹We clarify that by ‘polynomial interpolation’ we comprehend the determination of a set of coefficients that fully describe the sought for polynomial. In the literature, however, often the evaluation of this polynomial at given points is subsumed under the same term, possibly without explicit computation of the coefficients.

6.1.2. Lagrange interpolation

In the terms of Theorem 11, a polynomial that interpolates $(x_1, y_1), \dots, (x_n, y_n)$ is given by

$$p(x) = \sum_{k=1}^n \left(y_k \prod_{\substack{j=1 \\ j \neq k}}^n \frac{x - x_j}{x_k - x_j} \right) .$$

Correctness of this approach can be seen as follows: For all $k \in [1, n]$, function

$$L_k(x) = \prod_{j=1, j \neq k}^n \frac{x - x_j}{x_k - x_j}$$

is a polynomial of degree $n - 1$ which evaluates to 1 at position x_k , and evaluates to 0 at positions x_l for all $1 \leq l \leq n$, $l \neq k$. It follows that $p(x) = \sum_{k=1}^n y_k L_k(x) \in \Pi^{n-1}$ interpolates $(x_1, y_1), \dots, (x_n, y_n)$.

Relating this to the said above, Lagrange's method for polynomial interpolation *does not* look for coefficients of a linear combination of fixed basis elements of Π^{n-1} , but rather outputs vectors L_1, \dots, L_n in Π^{n-1} such that p is their 'trivial' linear combination with coefficients y_i .

In practice, Lagrange's method is rarely used for polynomial interpolation for being awkward and inefficient. Its importance is more on the theoretical side, e.g., for proving 'existence' in Theorem 11.

6.1.3. Newton interpolation

A far more efficient (and popular) way to perform polynomial interpolation in the context of scientific computing is due to Newton. We refer to [142, Section 4.2] for a detailed exposition, but stress that this method outputs coefficients a_k in respect to *Newton bases* $\{N_1, \dots, N_n\}$ of Π^n , which are instance-specific as well. In particular, if the points $(x_1, y_1), \dots, (x_n, y_n)$ are to be interpolated, the corresponding basis consists of the polynomials

$$N_k(x) = \prod_{j=1}^{k-1} (x - x_j) .$$

Corresponding coefficients $a_k = [y_1, \dots, y_k]$ can be efficiently computed via *divided differences* [142], to obtain the interpolating polynomial as $p(x) = \sum_{k=1}^n a_k N_k(x)$.

6.1.4. Interpolation without precomputation

An algorithm for polynomial interpolation that outputs coefficients in respect to monomial basis $\{1, x, x^2, \dots, x^n\}$ of Π^n is due to Björck and Pereyra [20, 84], and

portrayed below as Algorithm 1. It has (quadratic) running time

$$\frac{n(n-1)}{2}(D+M)$$

and, as most of the algorithms proposed in this section, needs no extra storage, as all calculations can be implemented ‘in place’.

Algorithm 1 Polynomial interpolation (Björck and Pereyra)

Input: Pairs $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{F} \times \mathbb{F}$ with $i \neq j \Rightarrow x_i \neq x_j$

Output: Coefficients $c_0, \dots, c_{n-1} \in \mathbb{F}$ such that $y_i = \sum_{k=0}^{n-1} c_k(x_i)^k \forall i$

$(c_0, \dots, c_{n-1}) \leftarrow (y_1, \dots, y_n)$

for $k = 0$ **to** $n - 2$ **do**

for $j = n - 1$ **downto** $k + 1$ **do**

$c_j \leftarrow (c_j - c_{j-1}) / (x_{j+1} - x_{j-k})$

end for

end for

for $k = n - 2$ **downto** 0 **do**

for $j = k$ **to** $n - 2$ **do**

$c_j \leftarrow c_j - x_{k+1}c_{j+1}$

end for

end for

Algorithm 1 already solves the problem of polynomial interpolation in reasonable time. However, we develop a technique to further improve computational efficiency by reducing the number of field divisions from $O(n^2)$ to 1, while at the same time increasing the number of multiplications only moderately. The trick is to represent intermediate variables c not as field elements, but as *fractions*²

$$c/d \hat{=} (c, d) \in \mathbb{F} \times \mathbb{F}^\times,$$

where we identify fraction $(c, 1)$ with field element c . Note that the field operations translate to operations on fractions as expected, e.g., $(c, d) + (c', d') = (cd' + c'd, dd')$ and $(c, d)(c', d') = (cc', dd')$. The benefit achieved by the redundancy introduced by the ‘computing with fractions’ technique is that most divisions can be replaced by multiplications, as the example $(c, d)/(c', d') = (cd', dc')$ illustrates. It is quite natural to consider two fractions $(c, d), (c', d') \in \mathbb{F} \times \mathbb{F}^\times$ *equivalent* (or *equal*) if $cd' = c'd$. Fractions are normalized to equivalent field elements by the *reduction mapping* $(c, d) \mapsto cd^{-1}$. If n of these reductions are to be computed in batch, the required n divisions can be conflated into a single inversion, at the cost of some additional multiplications (see [123, Appendix B], or [56, Algorithm 10.3.4], or [128]).

²Note that this idea is somewhat similar to the use of projective coordinates in efficient implementations of elliptic curve cryptography [86, Section 3.2.1].

This technique, applied to Algorithm 1, results in our Algorithm 2, which has computational performance

$$\left(\frac{5n(n-1)}{2} + 1\right)M + 1I .$$

Its speed advantage over Algorithm 1 is obvious for $D \gg M$. We note that Algorithm 2 needs extra storage for $n - 1$ auxiliary variables d_1, \dots, d_{n-1} .

Algorithm 2 Interpolation with deferred inversion

Input: Pairs $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{F} \times \mathbb{F}$ with $i \neq j \Rightarrow x_i \neq x_j$

Output: Coefficients $c_0, \dots, c_{n-1} \in \mathbb{F}$ such that $y_i = \sum_{k=0}^{n-1} c_k (x_i)^k \forall i$

$(c_0, \dots, c_{n-1}) \leftarrow (y_1, \dots, y_n)$

for $j = n - 1$ **downto** 1 **do**

$c_j \leftarrow c_j - c_{j-1}$

$d_j \leftarrow x_{j+1} - x_j$

end for

for $k = 1$ **to** $n - 2$ **do**

for $j = n - 1$ **downto** $k + 1$ **do**

$c_j \leftarrow c_j d_{j-1} - c_{j-1} d_j$

$d_j \leftarrow d_j d_{j-1} (x_{j+1} - x_{j-k})$

end for

end for

$c_j \leftarrow c_j d_j^{-1}$ **for all** $1 \leq j \leq n - 1$ (see note on batched reduction)

for $k = n - 2$ **downto** 0 **do**

for $j = k$ **to** $n - 2$ **do**

$c_j \leftarrow c_j - x_{k+1} c_{j+1}$

end for

end for

6.1.5. Interpolation with precomputation

In some occasions, polynomial interpolations have to be computed many times in succession, with fixed inputs x_1, \dots, x_n but variable inputs y_1, \dots, y_n . These cases are susceptible for improvements in efficiency by splitting calculations into a pre-computation phase (on input the x_i), and a computation phase (on input the y_i and the precomputed state). The costs of polynomial interpolation are then determined by the costs of the second step, which might be more efficient than a regular interpolation by Algorithms 1 or 2.

Observe that, for the coefficients c_k of the polynomial $p(x) = \sum_{k=0}^{n-1} c_k x^k \in \mathbb{F}[x]$ that passes through the set of points $\{(x_1, y_1), \dots, (x_n, y_n)\}$, the following linear system of equations holds:

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & & & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad (6.1)$$

The $(n \times n)$ -matrix $V = V(x_1, \dots, x_n)$ on the left is called *Vandermonde matrix* [142]. It is known from Numerical Analysis that V is invertible iff $i \neq j \Rightarrow x_i \neq x_j$. After determining V^{-1} in a precomputation step, one can solve equation (6.1) for c_0, \dots, c_{n-1} by computing $(c_0, \dots, c_{n-1})^T \leftarrow V^{-1}(y_1, \dots, y_n)^T$, essentially performing a matrix by vector multiplication with n^2M costs (cf. Algorithm 3). Explicit formulae for V^{-1} are developed in [64, 159], see also [123, Appendix A] for a clean presentation.

Algorithm 3 Interpolation after precomputation

Input: Inverted Vandermonde matrix $V^{-1} = (m_{1,1}, \dots, m_{n,n}) \in \mathbb{F}^{n \times n}$ as output by [123, Algorithm 7], elements $y_1, \dots, y_n \in \mathbb{F}$

Output: Coefficients $c_0, \dots, c_{n-1} \in \mathbb{F}$ such that $y_i = \sum_{k=0}^{n-1} c_k(x_i)^k \forall i$

for $i = 1$ **to** n **do**

$c_{i-1} \leftarrow 0$

for $j = 1$ **to** n **do**

$c_{i-1} \leftarrow c_{i-1} + m_{i,j}y_j$

end for

end for

6.1.6. Performance comparison of interpolation algorithms

We compare practical efficiency of Algorithms 1, 2, and 3 in Figure 6.1. It becomes obvious that Algorithm 1 by Björck and Pereyra is actually not competitive with our optimized variant (Algorithm 2), and that precomputations can, moreover, roughly halve running time of polynomial interpolation. Time consumption, on the right axis, is estimated by assuming $M = 0.44\mu s$, as measured in our test implementation (see also notes in Figure 6.4).

6.1.7. Polynomial evaluation

Recall that IHME decoding is defined through polynomial evaluation. For a set $c_0, \dots, c_{n-1} \in \mathbb{F}$ of coefficients, the naïve way of evaluating polynomial $p(x) = \sum_{k=0}^{n-1} c_k x^k \in \mathbb{F}[x]$ at a given point $x \in \mathbb{F}$ would have $O(n^2)$ performance. Deployment of Horner's scheme (Algorithm 4), however, reduces the running time to $(n-1)M$.

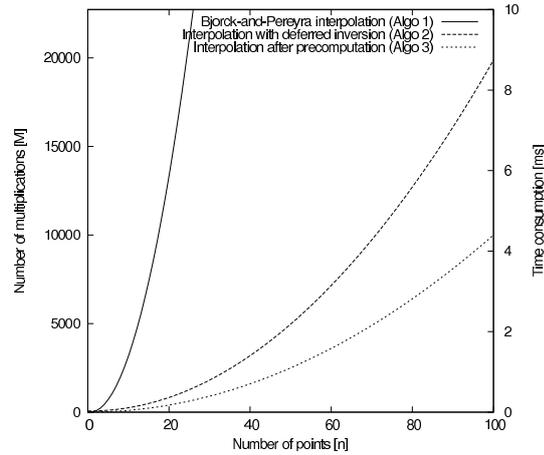


Figure 6.1.: Efficiency comparison of interpolation algorithms 1, 2, and 3. The axis on the left reflects the number of computed field multiplications (we assume $D = I = 60M$), the axis on the right indicates time consumption for a finite field \mathbb{F} of about 2^{80} elements.

Algorithm 4 Polynomial evaluation

Input: Coefficients $c_0, \dots, c_{n-1} \in \mathbb{F}$ and $x \in \mathbb{F}$

Output: Element $y \in \mathbb{F}$ with $y = \sum_{k=0}^{n-1} c_k x^k$

```

y ← cn-1
for k = n - 2 downto 0 do
    y ← ck + xy
end for

```

6.2. Interleaved IHME

In Sections 6.1.4, 6.1.5, and 6.1.7 we have seen implementations of IHME’s iEncode and iDecode routines. Their computational complexity is $O(n^2)$ and $O(n)$, respectively, with regards to a fixed finite field \mathbb{F} . In practice (e.g., in the Handshake protocol in Figure 5.3), these fields may become rather large, e.g., $|\mathbb{F}| \approx 2^{1108}$, and IHME will perform accordingly slow (although still in $O(n^2)$). In this section, we present an *interleaving technique* which allows to (generically) speedup IHME computations. Note that the algorithms remain in $O(n^2)$ and $O(n)$, respectively; it is rather the constant that is considerably reduced.

Consider, for instance, an IHME setting with $\mathbb{F} = GF(2^{1024})$ and $\mathcal{M} = \mathcal{I} = \mathbb{F} \cong \{0, 1\}^{1024}$. Instead of encoding messages $m_1, m_2, \dots \in \mathcal{M}$ over this field, one could split all messages m_i into, say, 8 chunks $m_{i,1}, \dots, m_{i,8}$, each of length $1024/8 = 128$. Now, using IHME over field $\mathbb{F}' = GF(2^{128})$, all $m_{i,1}$ can be IHME-encoded into a structure \mathcal{S}_1 , all $m_{i,2}$ can be independently encoded into a structure \mathcal{S}_2 , and so on. The overall encoding is then $\mathcal{S} = (\mathcal{S}_1, \dots, \mathcal{S}_8)$. A gain in efficiency is caused by the

trade of super-linear costs of finite field arithmetics for linear costs of splitting the field elements.

We formalize the ideas of the preceding paragraph in a more general setting: We show how to generically compose IHME schemes from IHME schemes with smaller message sets.

Definition 27 (Interleaved IHME) *Let $\text{IHME}' = \{\text{iEncode}', \text{iDecode}'\}$ be an index-hiding message encoding scheme over index set \mathcal{I}' and message set \mathcal{M}' . For any $\nu \in \mathbb{N}$, the ν -interleaved index-hiding message encoding scheme $\text{IHME} = \{\text{iEncode}, \text{iDecode}\}$ with index space $\mathcal{I} = \mathcal{I}'$ and message space $\mathcal{M} = (\mathcal{M}')^\nu$ is constructed from IHME' as follows:*

$\text{iEncode}(\mathcal{P})$

On input of $\mathcal{P} = \{(i_1, (m_{1,1}, \dots, m_{1,\nu})), \dots, (i_n, (m_{n,1}, \dots, m_{n,\nu}))\} \subseteq \mathcal{I} \times \mathcal{M} = \mathcal{I}' \times (\mathcal{M}')^\nu$, the resulting encoding is the list $\mathcal{S} = (\mathcal{S}_1, \dots, \mathcal{S}_\nu)$ of IHME' encodings

$$\mathcal{S}_k = \text{iEncode}'(\{(i_j, m_{j,k})\}_{1 \leq j \leq n}) \quad \text{for } 1 \leq k \leq \nu .$$

$\text{iDecode}(\mathcal{S}, i)$

On input of $\mathcal{S} = (\mathcal{S}_1, \dots, \mathcal{S}_\nu)$ and index $i \in \mathcal{I}$, this algorithm outputs $m = (m_1, \dots, m_\nu)$, where

$$m_k = \text{iDecode}'(\mathcal{S}_k, i) \quad \text{for } 1 \leq k \leq \nu .$$

Index-hiding security of interleaved IHME (cf. Definition 21) is established via a standard hybrid argument (with $\nu - 1$ intermediate steps), where in the i -th hybrid experiment index set I_1 is used for structures $\mathcal{S}_1, \dots, \mathcal{S}_i$, and index set I_0 is used for structures $\mathcal{S}_{i+1}, \dots, \mathcal{S}_\nu$ (cf. experiment $\text{Expt}^{\text{hide}}$ in Figure 4.1). The tightness factor obtained in the corresponding reduction is ν .

Theorem 12 (Security of interleaved IHME) *For any given index-hiding IHME' scheme and any $\nu \in \mathbb{N}$, the ν -interleaved scheme IHME constructed in Definition 27 is index-hiding as well. If IHME' is perfectly-index hiding, then so is IHME .*

6.2.1. Efficiency analysis

The gain in efficiency over standard IHME, achieved by deployment of ν -interleaved IHME, is illustrated in Figure 6.2. We use 1104 bit fields in the test case as fields of this size naturally emerge in the setting of affiliation-hiding protocols (cf. Section 5.4). We observe that ν -interleaved IHME outperforms standard IHME by about 30%, for both underlying Algorithms 2 and 3.

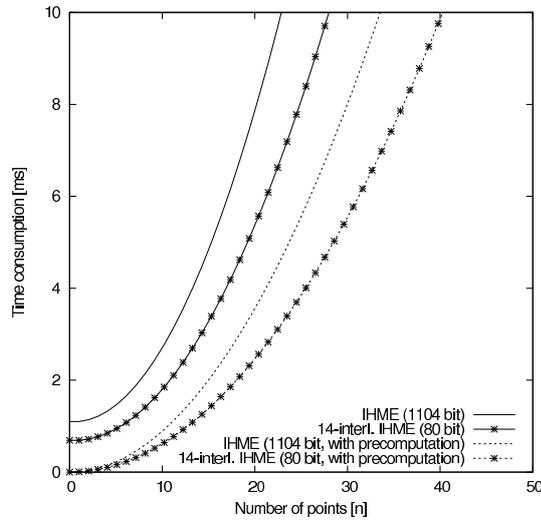


Figure 6.2.: Efficiency comparison of standard IHME (Section 4.5.1) and ν -interleaved IHME (Section 6.2). More precisely, we compare standard IHME over a 1104 bit prime field with (the more or less equivalent) 14-interleaved IHME over a 80 bit prime field (note that $80 \cdot 14 = 1120$), both without and with precomputations (Algorithms 2 and 3, respectively). The offset on the y -axis for interpolations without precomputation is due to the (relatively high) cost of the inversion in Algorithm 2. See Figure 6.4 for further implementation details.

6.2.2. Interleaved IHME over the integers

Let Π be a prime and $\nu \in \mathbb{N}$. By $\text{iEncode}(\mathcal{P}, \Pi, \nu)$ we denote the IHME encoding with index space $\mathcal{I} = [0, \Pi - 1]$ and message space $\mathcal{M} = [0, \Pi^\nu - 1]$, i.e., $\mathcal{P} \subseteq \mathcal{I} \times \mathcal{M}$. This scheme is obtained by combining the interpolation-based construction from Section 4.5.1 with Definition 27, and by exploiting existence of finite field $\mathbb{F} = GF(\Pi)$ and the natural and efficient bijections $[0, \Pi - 1] \rightarrow \mathbb{F}$ and $[0, \Pi^\nu - 1] \rightarrow \mathbb{F}^\nu$ (e.g., for the latter, the representation to base Π , i.e., $a \mapsto (a_0, \dots, a_{\nu-1})$ such that $a = \sum_{k=0}^{\nu-1} a_k \Pi^k$). Analogously, by $\text{iDecode}(\mathcal{S}, \Pi, \nu, i)$ we denote the corresponding IHME decoding at index $i \in [0, \Pi - 1]$.

6.3. An optimized RSA-based mAHA protocol

In this section, we propose a set of optimizations to different algorithms of RSA-based mAHA protocol from Section 5.4 that lead towards a truly practical implementation. Within others, we replace the IHME scheme used in the original Handshake specification by the optimized version from Section 6.2.2, but our improvements also cover a multitude of unrelated aspects and optimize computational performance, bandwidth consumption, and key sizes. The verbosity of the presentation of all al-

gorithms and protocols should be sufficient for an immediate implementation of the protocol.

6.3.1. Optimized CreateGroup algorithm

The CreateGroup algorithm is run once per group authority and, hence, computational efficiency is not too important for its implementation. Instead, we decide to improve on storage size of group parameters. Recall that part of users' credentials in Section 5.4.1 are triples (n, g, e) , where n is a safe RSA modulus, e is an RSA exponent suitable for modulus n , and g is such that $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$. Our improved algorithm, denoted as Algorithm 5, outputs RSA moduli n that are crafted in such a way that $g = 2$ and $e = 3$ are valid group generators and RSA exponents, respectively. As consequence, it suffices to store just modulus n in the credentials. In addition, in Section 6.3.3, we will see that choosing $g = 2$ offers an attractive opportunity to implement the exponentiations in the Handshake protocol very efficiently. We start by giving some convenient lemmas:

Lemma 4 *Let $p = 2p' + 1$ be a safe prime. Then $p = 11 \pmod{12}$.*

Proof. As p' is a prime number we have $p' \in \{1, 5\} \pmod{6}$. Case $p' = 1 \pmod{6}$ leads to $p = 2p' + 1 = 3 \pmod{12}$, a contradiction. Hence $p' = 5 \pmod{6}$ and $p = 11 \pmod{12}$. \square

Lemma 5 *Let $n = pq$ for safe primes $p = 2p' + 1, q = 2q' + 1$ with $p = 3 \pmod{8}$ and $q = 7 \pmod{8}$. Then for $g = 2$ we have $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$.*

Proof. Under the named conditions we have $\left(\frac{2}{p}\right) = -1$ and $\left(\frac{2}{q}\right) = 1$ (cf. Section 2.1.1). Hence $g = 2$, considered as element $g \in \mathbb{Z}_p^\times$, generates $\langle g \rangle_p = \mathbb{Z}_p^\times$, while $g \in \mathbb{Z}_q^\times$ generates $\langle g \rangle_q = QR(q)$. By applying CRT, we see that $\langle g \rangle_n \cong \langle g \rangle_p \times \langle g \rangle_q$ and that the order of $g \in \mathbb{Z}_n^\times$ is $\text{lcm}(\text{ord}_p g, \text{ord}_q g) = \text{lcm}(2p', q') = 2p'q' = \lambda(n)$. In addition, as $q = 3 \pmod{4}$, we have $\left(\frac{-1}{q}\right) = -1$, i.e., $-1 \notin QR(q) = \langle g \rangle_q$, and hence $-1 \notin \langle g \rangle_n$. This proves $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$. \square

Algorithm 5 outputs tuples (n, d, p, q) such that $n = pq$ is a safe RSA modulus, $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle 2 \rangle_n$, and a public/privat RSA key pair is given by $(3, d)$, i.e., $3d = 1 \pmod{\lambda(n)}$. Factor $p = 2p' + 1$ of $n = pq$ is chosen such that $p = 11 \pmod{12}$ and $p = 3 \pmod{8}$, and hence $p = 11 \pmod{24}$, while for $q = 2q' + 1$ we require $q = 11 \pmod{12}$ and $q = 7 \pmod{8}$, i.e., $q = 23 \pmod{24}$ (cf. Lemmas 4 and 5). The search for safe primes for which these congruences hold is performed by the two `while` loops. By `RandNum`([a, b]) we denote the uniformly random choice of an integer x with $a \leq x \leq b$. By `IsPrime`(x) we denote the application of a (probabilistic) primality test, e.g., the Miller-Rabin test, to integer x . Note that $\lambda(n) = 2p'q'$, and hence $e = 3$ is always invertible mod $\lambda(n)$.

Algorithm 5 Implementing CreateGroup**Input:** Security parameter κ (typically $1024 \leq \kappa \leq 2048$)**Output:** Secret key $\text{sk} = (n, d, p, q)$

```

 $\ell \leftarrow \lfloor \kappa'/2 \rfloor$ 
 $p \leftarrow 24 \cdot \lfloor \text{RandNum}([2^{\ell-1}, 2^\ell - 1])/24 \rfloor + 11$ 
while  $\neg \text{IsPrime}(p) \vee \neg \text{IsPrime}((p-1)/2)$  do
   $p \leftarrow p + 24$ 
end while
 $q \leftarrow 24 \cdot \lfloor \text{RandNum}([2^{\ell-1}, 2^\ell - 1])/24 \rfloor + 23$ 
while  $\neg \text{IsPrime}(q) \vee \neg \text{IsPrime}((q-1)/2)$  do
   $q \leftarrow q + 24$ 
end while
 $\lambda = (p-1)(q-1)/2$ 
 $(n, d, p, q) \leftarrow (pq, 3^{-1} \bmod \lambda, p, q)$ 

```

6.3.2. Optimized AddUser algorithm

In the AddUser protocol from Section 5.4.1, users obtain credentials $\text{sk}[\text{id}]$ of the form $(n, g, e, \sigma_{\text{id}})$, where $\sigma_{\text{id}} = H_n(\text{id})^d \bmod n$, i.e., σ_{id} is the full-domain hash RSA signature on the respective pseudonym id (cf. Section 2.2.2). (A concrete instantiation of hash functions $H_n : \{0, 1\}^* \rightarrow \mathbb{Z}_n$ is proposed in [123, Section 5]). We observe that, in the actual Handshake protocol, term $H_n(\text{id})$ only occurs in contexts where values are *divided by* $H_n(\text{id})$. For the sake of efficiency, and without influencing the scheme's security, we move these necessary inversions into the registration process, by altering the generation of user credentials to

$$\text{sk}[\text{id}] = (n, \sigma_{\text{id}}) \quad \text{where} \quad \sigma_{\text{id}} = H_n(\text{id})^{-d} \bmod n .$$

A standard trick [102] to speed up private RSA operations is to apply CRT before computing exponentiations by d . More concretely, if the factorization of RSA modulus $n = pq$ is known, then $y = x^d \bmod n$ can be computed by CRT-decomposing x into $x_p = x \bmod p$ and $x_q = x \bmod q$, by computing $y_p = x_p^d = x_p^{d \bmod \varphi(p)} \pmod{p}$ and $y_q = x_q^d = x_q^{d \bmod \varphi(q)} \pmod{q}$, and by mapping (y_p, y_q) back to \mathbb{Z}_n , by applying CRT a second time. Besides the fact that exponentiations mod p and mod q can be computed substantially faster than exponentiations mod n , many intermediate values of this alternative signing method can be precomputed. An implementation of the AddUser algorithm that includes these optimizations is given in Algorithm 6.

6.3.3. Optimized Handshake protocol

Our optimized version of the mAHA protocol from Figure 5.3 is presented in Figure 6.3. By $H_1 : \{0, 1\}^* \rightarrow \{0, 1\}^\ell$ we denote a hash function. Let Π be a prime

Algorithm 6 Implementing AddUser**Input:** Secret key $\mathbf{sk} = (n, d, p, q)$, pseudonym $\text{id} \in \{0, 1\}^*$ **Output:** Credential $\mathbf{sk}[\text{id}] = (n, \sigma_{\text{id}})$ **Precompute:** $d_p \leftarrow -3^{-1} \pmod{p-1}$, $d_q \leftarrow -3^{-1} \pmod{q-1}$, $u \leftarrow p^{-1} \pmod{q}$ $h \leftarrow H_n(\text{id})$ $(h_p, h_q) \leftarrow (h \pmod{p}, h \pmod{q})$ $(\sigma_p, \sigma_q) \leftarrow (h_p^{d_p} \pmod{p}, h_q^{d_q} \pmod{q})$ $a \leftarrow u \cdot (\sigma_q - \sigma_p) \pmod{q}$ $\sigma_{\text{id}} \leftarrow \sigma_p + pa$ $\mathbf{sk}[\text{id}] \leftarrow (n, \sigma_{\text{id}})$

slightly greater than 2^ℓ and let $\nu \in \mathbb{N}$ be minimal such that $\Pi^\nu > 2^{\kappa+\ell}$. A typical configuration in practice would be $(\kappa, \ell, \nu) = (1024, 80, 14)$.

The principal enhancement of the new design over the scheme from Section 5.4.1 is the deployment of the more efficient interleaved IHME scheme introduced in Section 6.2.2 (see lines 10, 15, 23, and 26). In the original protocol, all messages that are exchanged in the two communication rounds are, when IHME-encoded, considered as elements of a certain finite field \mathbb{F} . In particular, in the first round, padded RSA values of length $\kappa + \ell$ are exchanged. Hence, \mathbb{F} has to be chosen accordingly large ($|\mathbb{F}| \geq 2^{1104}$ at least) and field arithmetic performs rather slow. In contrast, in Figure 6.3, first-round messages $\theta \in [0, \Pi^\nu - 1]$ are encoded over a (much smaller) field of $\Pi \approx 2^\ell$ elements, using the ν -interleaved technique. Note that careful choice of Π , e.g., of low Hamming weight, allows impressively fast implementations of field arithmetics [86, Section 2.2.6]. Considering the second round messages, in the protocol from Section 5.4.1, the per-group key confirmation messages are also of length $\kappa + \ell$, but actually ℓ bits would suffice for a secure scheme. In our new protocol, confirmation messages are shortened to this more reasonable level and encoded using IHME, again over the field of $\Pi \approx 2^\ell$ elements. Both these optimizations lead to a considerable boost of computational efficiency and bandwidth consumption, when compared to a naïve implementation of the protocol.

A consequence of the switch to a smaller field is that also deployed IHME indices have to be chosen from a smaller set (see Section 6.2.2). While, in Section 5.4.1, RSA moduli n serve directly as group index, in our protocol the set of possible indices is reduced to the elements of $[0, \Pi - 1]$, which is much too small for allowing a direct embedding of moduli n . We solve this problem by hashing public group parameters into \mathbb{F} (line 4). These hash values, $h_n = H_1(n)$, can further on be considered as convenient ‘handles’ for groups, and are therefore designated as the elements of the ‘shared group’ set **groups** that is part of the output of the protocol (see line 27).

A further improvement over the protocol from Figure 5.3 is the more straight

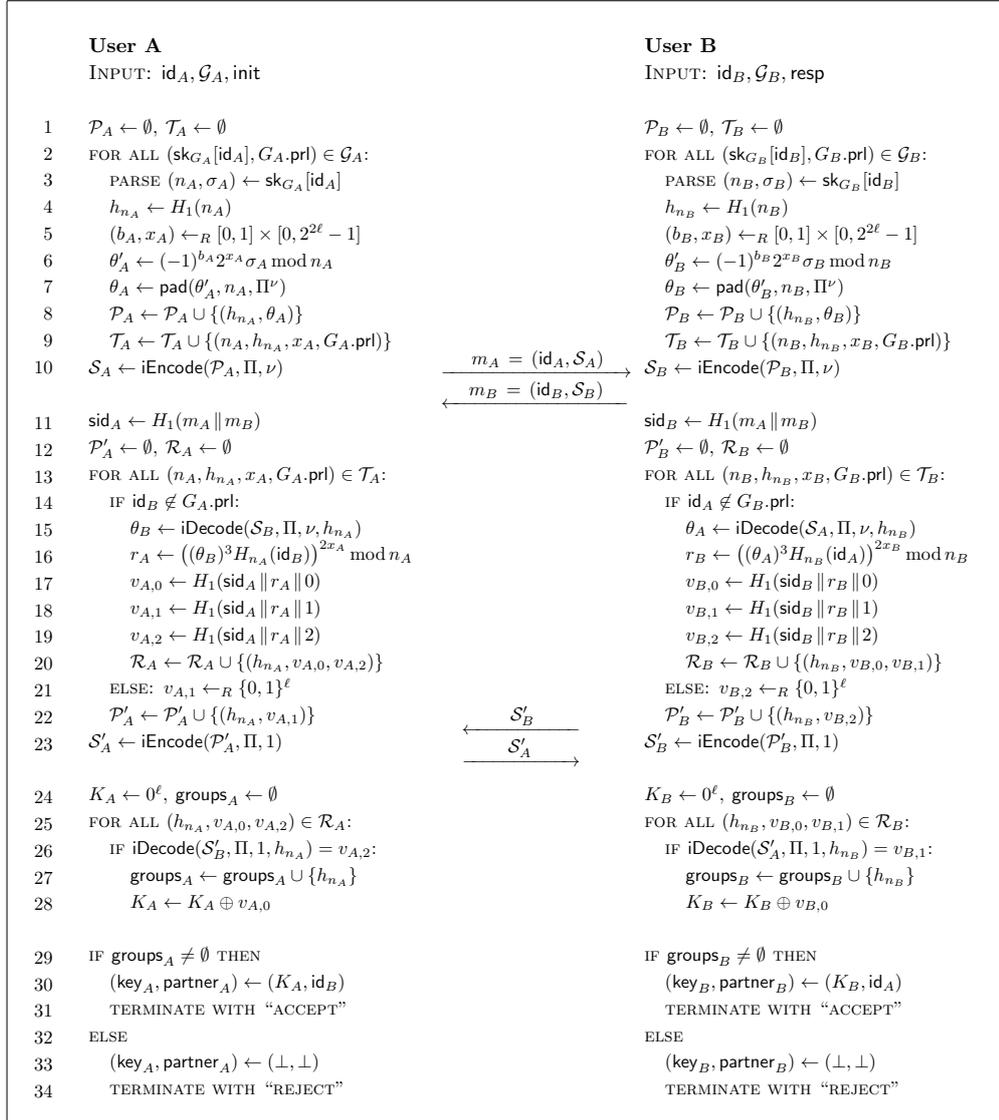


Figure 6.3.: Optimized RSA-based Handshake protocol with group discovery

forward way of session key derivation. In Section 5.4.1, the key is computed as the hash value of a string that is composed of several per-group secrets. To achieve correctness of the protocol, both protocol participants have to mount this string in the same order. To do this, a canonical ordering of groups has to be assumed. In our new protocol, however, the key is computed as XOR-sum of per-group secrets (line 28), which, without loss of security, now can be computed in any order.

In contrast to Section 5.4, in our optimized protocol, ephemeral exponents x (see lines 5, 6 and 16) are not chosen from $\mathbb{Z}_{n/2}$ (where n is an RSA modulus of length κ), but from much smaller range $[0, 2^{2\ell} - 1]$. This, again, leads to a notable gain in efficiency in the modular exponentiations. Under the common assumption [81, 83] that Discrete Logarithm Problem (DLP) in \mathbb{Z}_n^\times is hard even when exponents are short,

distributions of ephemeral keys with short and long exponents, respectively, are computationally indistinguishable from each other (see Lemma 3.6 in [83]). Hence, shortening ephemeral keys in the described way does not result in a considerably weaker security of the protocol.

Observe that the exponentiation in line 6 has an a-priori known basis, namely $g = 2$. In general, diverse fast algorithms for fixed-basis exponentiations are known [127, Section 14.6]. However, we stress that in our case exponentiations' performance can be even further improved by exploiting the structure of 'square and multiply' algorithms [127], where an accumulator is repeatedly multiplied by the base element. When $g = 2$, this multiplication becomes a doubling of the accumulator, which can be implemented by a simple left-shift. The overall performance then depends solely on the cost of the squaring operation. We remark that term $(-1)^b$ for $b \in \{0, 1\}$ in line 6 should, of course, never be computed by calling an exponentiation subroutine, but by doing a distinction of cases. We conclude this paragraph by noting that messages θ (or even whole IHME structures $\mathcal{S}_A/\mathcal{S}_B$) can be precomputed before the protocol session starts in order to further reduce run-time computations.

Note that, even though the protocol in Figure 6.3 is displayed as four-message protocol, by concatenating messages m_B and \mathcal{S}'_B into a single message, the protocol is trivially turned into a three-message protocol.

6.3.4. Performance analysis and discussion

We discuss performance results obtained from a concrete implementation of our optimized Handshake protocol from Figure 6.3. In particular, we present performance measurements and investigate the scalability of our Handshake protocol for the security level $(\kappa, \ell) = (1024, 80)$, i.e., 1024 bit RSA combined with 80 bit 'symmetric security', and varying numbers of credentials $n = |\mathcal{G}|$. Note that, in the implementation, we use ν -interleaved IHME with $\nu = 14$.

In Figure 6.4, by plotting separately the amount of time spent in exponentiations and IHME routines (T_e and T_i , respectively), we expose the relevance of IHME's optimization. For instance, while in a Handshake with precomputation (Algorithm 3) the balance between T_e and T_i is 50:50 for about $n = 200$ affiliations, this bound is reached for Handshake executions without precomputations (Algorithm 2) already for about $n = 120$ groups. Note that, in typical application scenarios of mAHA protocols, e.g., in social networks, the average number of affiliations (social groups) is about 80 [154].

The charts cannot belie that the IHME primitive with its quadratic complexity introduces a noticeable weight into protocol's performance, especially for large n . Noticing, however, that a full Handshake without precomputation takes only 2 seconds to complete, even for 250 affiliations, we argue that this quadratic overhead is still acceptable for practical deployment.

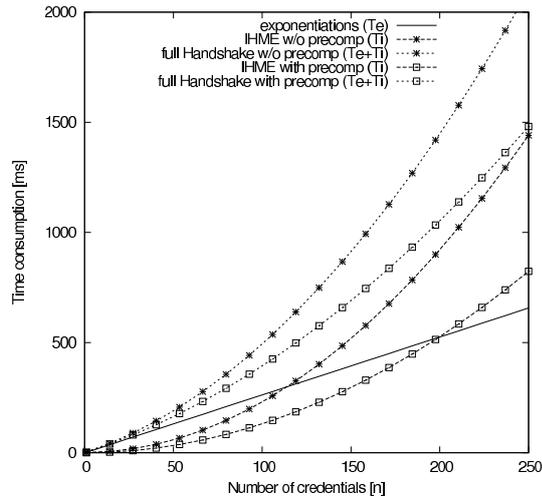


Figure 6.4.: Timing values for our optimized protocol from Figure 6.3. All measurements were made on a single core of an ‘Intel XEON 2.66 GHz’ machine, using the `gcrypt` library [108] for bigint arithmetic.

6.4. Efficiency of NIKDS-based mAHA protocol

We highlighted in Section 5.6 that one of the strengths of our NIKDS-based mAHA scheme is its small bandwidth requirements. However, generally speaking, the evaluation of a pairing is a more time-consuming operation than an RSA exponentiation (at a comparable security level). Hence, in terms of computational costs, the NIKDS-based scheme might be more expensive than its competitor, the optimized RSA-based protocol from Figure 6.3.

We did not actually implement the protocol, but we still try to give a performance estimation based on measurements of bilinear map evaluations found in the literature. Although many speed records on different architectures have been recently reported, care has to be taken with the interpretation of the results, as not all pairing parameters necessarily match our needs. In particular, the NIKDS construction from Section 2.3.1 assumes a *symmetric* pairing $\mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}_T$, in contrast to an *asymmetric* pairing $\mathcal{G}_1 \times \mathcal{G}_2 \rightarrow \mathcal{G}_T$, for different groups $\mathcal{G}_1, \mathcal{G}_2$. Generally, the NIKDS’s specification can also be adapted to the asymmetric setting, but still there would remain the need to hash to both input groups, \mathcal{G}_1 and \mathcal{G}_2 , which might be problematic [79]. As most recent implementations [2, 3, 129] focus only on asymmetric pairings (as these, within others, offer large embedding degrees), we will use timings reported by Scott in 2007 [146] for our estimations. The (Tate) pairing implemented in [146] is defined over a 512 bit prime field and has embedding degree $k = 2$. One pairing evaluation is claimed to be computable in less than 3 ms, on a 3 GHz Pentium IV.

In respect to our estimations, assuming that the computational overhead of a full protocol execution is almost exclusively determined by the time spent in the $n = |\mathcal{G}|$ pairing evaluations, the total running time of one Handshake can be estimated with $3|\mathcal{G}|$ ms.

6.5. Practical comparison of our mAHA solutions

We conclude this chapter by comparing practical performance of our RSA-based and NIKDS-based mAHA schemes. While, in Figure 6.5, the RSA-related plots are identical with those from Figure 6.4, we estimate the performance of the NIKDS-based protocol via the linear prediction from Section 6.4. We see that the scheme from Figure 5.4 mostly outperforms our optimized Handshake scheme from Figure 6.3. Deployment of the latter, however, may be advantageous for a small number n of credentials, e.g., for $n \leq 30$ affiliations — presuming a setting in which it is feasible to store precomputed values. On the other hand, if storage place for precomputed data is not available, then, efficiency-wise, there is no point in using the RSA-based scheme. We hence propose, as a general recommendation, deployment of the NIKDS-based scheme in all cases, for offering more flexibility and efficiency for a wide range of affiliations. Note that, although execution times of RSA-based and NIKDS-based schemes were measured on different machines, we expect that more consistent measurements would have led to the same conclusion.

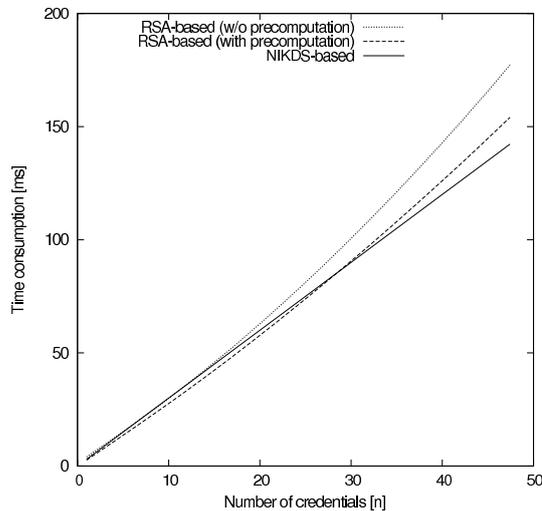


Figure 6.5.: Comparison of measured performance of optimized RSA-based protocol (cf. Figure 6.3) and expected performance of NIKDS-based protocol (cf. Figure 5.4).

Applications of AHA: private contact discovery

One of today's most popular internet applications is given by online social networks (OSN). These provide services to share interests, activities, and pictures, and have proved valuable to help participants to build and reflect social relations to other participants. In late 2011, popular social network websites, such as Facebook, LinkedIn, or MySpace, involve billions of active users [118, 154]. Moreover, a clear tendency of users accessing OSN services *ubiquitously* has to be noted: 350 of today's 800 million Facebook users access the site from their mobile devices [154].

One of the first steps towards establishing social network relationships is to verify the existence of common contacts or friends. An interesting problem occurs when two unfamiliar users want to do this privately: Consider two users with their mobile devices that are connected via a mobile ad-hoc network (MANET) and want to assess their social proximity by discovering mutual contacts. A naïve solution would require participants to reveal their friends to each other. Clearly, this would not preserve users' privacy, since their complete lists would be exposed. Another trivial solution would employ a central server to find and output the common friends. However, such a server is not always considered trustworthy, as it would learn not only participants' friends, but also which users become friends, how, when, and where. Moreover, central servers are not necessarily reachable in MANET settings, e.g., if users meet in a bar or on the subway.

We introduce the concept of *private contact discovery*, a novel general construct geared to preserve user privacy, not only in social network interactions, but also in any other application that uses personal contact lists. The corresponding cryptographic primitive, termed *contact discovery scheme* (CDS), lets two users, on input their respective contact lists, learn their mutual contacts (if any), and nothing else. Following a rigorous cryptographic treatment of the problem, we define the privacy-protecting *contact-hiding* property and construct corresponding provably secure protocols. Note that our schemes neither rely on any (trusted) third party nor are they

bound to a specific network infrastructure. They are hence suitable for users that may want to operate outside the social network website to establish new relationships. Moreover, the efficiency of our protocols allows deployment even on mobile devices, as corresponding measurements attest.

In order to protect privacy in the described setting, within others, it is necessary to prevent malicious users from arbitrarily manipulating their lists of friends, e.g., by populating them with their best guesses of other users' lists to maximize the amount of information learned. Our model and constructions prevent users from claiming unwarranted friendships by introducing *contact certification*. For instance, in order to include Carol in her contact list, Alice needs to obtain an individual certificate from Carol attesting this friendship. Then, when Alice interacts with Bob, not only the entries in her contact list are hidden from Bob, but also the possession of corresponding certificates with respect to non-common friends.

7.1. Approaches towards private contact discovery

We briefly discuss some private contact discovery schemes that were proposed in the literature so far. We also analyze whether the CDS challenge can be solved using existing cryptographic building blocks. However, we will see that all these approaches do not reach a satisfactory level of security. In particular, none of the schemes fulfills our strong privacy requirements formalized in Section 7.4.

Von Arb *et al.* [157] present a mobile social networking platform that enables *Friend-of-Friend* (FoF) detection in physical proximity. Their solution compares friend lists through PSI techniques [92,100]. For the reasons discussed in Section 4.2, we argue that such protocols, constructed to privately compute the intersection of input sets, do not yield satisfying CDS solutions: PSI schemes do not prevent parties from arbitrarily manipulating their input lists. Observe that also the authenticated variant, APSI, does not solve the CDS problem accurately: the primitive assumes a single authority, in contrast to contact discovery, where all users act as their own CA and issue certificates independently of each other.

Freedman and Nicolosi [77] propose two solutions for the FoF problem, in the context of trust establishment in email whitelisting. One solution is based on hash functions and symmetric encryptions, the other on bilinear maps. Both solutions leverage friendship attestation, but basically implement an (optimized) variant of the naïve matching approach from Section 4.1. Moreover, their solution based on symmetric encryption allows users to maliciously transfer attestations to other users (what would contradict our security model), while their pairing-based technique is inefficient for involving a quadratic number of bilinear map operations. Furthermore, the paper lacks a rigorous security analysis.

Huang, Chapman, and Evans [48,91] recently described their ready-to-use contact discovery application for the Android platform. Using garbled circuits [165] to solve CDS as a generic instance of secure multi-party computation, they report timing values of 150 seconds to match 128 contacts. Their construction suffers from the security issues discussed above and in Section 4.2, i.e., adversaries are not hindered from arbitrarily populating their contact lists. Moreover, security is claimed only in a model with semi-honest adversaries.

Further (non-cryptographic) treatments of Friend-of-Friend detection with no or unclear privacy properties are given in [54,109,110].

The cryptographic tools discussed in Section 1.4, e.g., group signatures or anonymous credentials, do also not induce appropriate solutions to CDS, as hiding the respective certifying party (in our case: the user that issues contact certificates) is generally neither an explicit security goal of such schemes, nor is it implicitly achieved.

Some works analyze, to a higher extent, social relationships, without focusing on privacy. For instance, [138] uses random walks to discover *communities* in large social-network graphs, [166, Chapter 12] formalizes the problem of dynamically identifying core communities (i.e., sets of entities with frequent and consistent interactions), [167] builds a prediction model to identify certain social structures, e.g., friendship ties and family circles, while [68] attempts to identify communications that substantiate social relationship types.

7.2. Contact discovery from mAHA?

Intuitively, mAHA schemes could provide a generic solution to the privacy challenge imposed by the CDS problem. Consider a setting where each CDS user controls its own group authority (GA) of a mAHA scheme, independently of all other users. Contact certificates issued by the user would be credentials obtained from mAHA's `AddUser` operation. The discovery process itself would be implemented through mAHA's `Handshake` protocol: Whenever two users want to discover which contacts they have in common, they execute `Handshake` on input their credentials (i.e., contact certificates): the set of matching affiliations would correspond to the list of common contacts, i.e., the list of users that both participants got certificates from. Moreover, the security notion of affiliation-hiding (cf. Definition 23) would hopefully translate to a meaningful notion of 'contact-hiding'.

We argue, however, that this 'CDS from mAHA' construction is not secure in general. The main reason for this is that, in our mAHA model from Section 5.3, group authorities are unconditionally trusted and are assumed to always follow the protocol specification. While this assumption might be realistic in classic mAHA scenarios

(where GAs are notaries or otherwise trusted agencies), it is not reasonable, in the context of contact discovery, to trust all users, e.g., of a social network. For concreteness, we show that the CDS protocol obtained by applying the transformation described above to the mAHA scheme by Jarecki and Liu [97] is not secure.

Recall from Section 5.1 that, in the protocol from [97], users U register to groups G by sending their secret key x_U to the respective GA, which returns $r_{U,G} = g^{x_U x_G}$ and $\Pi_{U,G} = \text{PoK}[(\alpha) : r_{U,G} = g^\alpha]$ as credentials, where x_G is GA's private key. Now, in the context of CDS, this means that users reveal their secret keys to all their contacts. However, in discovery sessions with these contacts, users would provide their respective $r_{U,G} = (g^{x_G})^{x_U}$ values to parties knowing x_U (cf. Section 5.1), i.e., the public keys g^{x_G} , and hence the identities, of all their contacts would be directly revealed.

This example illustrates that it is not clear how to generically convert mAHA schemes into satisfyingly secure CDS. Apparently, privacy of CDS needs to be modelled specifically, and corresponding schemes deserve individual evaluation and security proofs. Nevertheless, we admit that the CDSs we propose in Sections 7.5 and 7.6 build on ideas that also drive our mAHA constructions from Chapter 5.

7.3. Syntax of CDS

Formally, a CDS consists of algorithms to initialize users, to issue contact certificates to other users, and to run the actual contact discovery protocol. Note that our definition does not comprehend the possibility of user revocation, that, as we argue, is mostly meaningless in the context of social relationship management. However, should revocation of contacts be considered indispensable, then our syntax and protocols for CDS can be easily retrofitted (in a way similar to revocation handling in our mAHA schemes in Sections 5.4 and 5.5).

Definition 28 (Contact discovery scheme) *A contact discovery scheme is defined as a set $\text{CDS} = \{\text{InitUser}, \text{AddContact}, \text{Discover}\}$ of three algorithms and protocols:*

$\text{InitUser}(1^\kappa)$

This algorithm is executed once by each user U . On input of security parameter 1^κ , it initializes U 's private key $U.\text{sk}$.

$\text{AddContact}(U, V)$

This algorithm is executed by user U , on input the identity of a user V . User U certifies a given social relation to V by issuing V a corresponding contact certificate $cc_{U \rightarrow V}$.

Note that we model contact certification as an unidirectional process: A mutual certification requires two executions of AddContact algorithm.

Discover($U \leftrightarrow U'$)

This protocol is executed between two users, U and U' , to discover common contacts. User U 's private input is $(\text{CL}_U, \text{partner}_U, r_U)$, where contact list CL_U is a set of pairs of the form $(V, \text{cc}_{V \rightarrow U})$, for some users V , partner_U is the name/id of the supposed protocol partner, and $r_U \in \{\text{init}, \text{resp}\}$ specifies the role of the session as initializer or responder. All values $\text{cc}_{V \rightarrow U}$ are assumed to be contact certificates previously obtained as output of AddContact(V, U). Private input of user U' is $(\text{CL}_{U'}, \text{partner}_{U'}, r_{U'})$, defined analogously.

The protocol shall detect the set of users V for which both participants provide corresponding contact certificates, $\text{cc}_{V \rightarrow U}$ and $\text{cc}_{V \rightarrow U'}$, respectively. This shared contact list is denoted by SCL.

Users keep track of the state of created Discover protocol sessions π through session variables that are initialized by setting $\pi.\text{state} \leftarrow \text{running}$ and $\pi.\text{SCL} \leftarrow \emptyset$, and by initializing $\pi.\text{CL}$ and $\pi.\text{partner}$ from the session parameters. In addition, $\pi.\text{id}$ is set to the own identity. After the protocol completes, $\pi.\text{state}$ is updated to `accepted` and $\pi.\text{SCL}$ holds a (possibly empty) set of user identifiers.

Observe that, in contrast to mAHA schemes (cf. Section 5.2), CDSs are pure authentication protocols: Besides the set SCL of contacts shared between the two participants, protocol executions do not output additional session keys.

Definition 29 (Correctness of CDS) *Suppose that users U and U' interact in a Discover protocol on input $(\text{CL}_U, U', \text{init})$ and $(\text{CL}_{U'}, U, \text{resp})$, respectively. Let π and π' denote the corresponding sessions. Let CL_\cap denote the set of users (contacts) V that appear in both CL_U and $\text{CL}_{U'}$. The CDS is correct if both sessions accept and $\pi.\text{SCL} = \pi'.\text{SCL} = \text{CL}_\cap$.*

7.4. Security model for CDS

We introduce a security model for private contact discovery schemes by describing the capabilities of the adversary and by defining appropriate experiments for the security goal of *contact-hiding*. Note that, as we deal with a pure authentication protocol, there is no need to model key security.

Moreover, many applications that would deploy CDSs, such as social networks and other group applications, already provide an independent user-based authentication infrastructure, e.g., they deploy a PKI or use password-based techniques. Such an authentication infrastructure can be used for various types of communication,

including the execution of CDS protocols within a secure channel [28, 111]. With this assumption in mind, we can now focus on the core functionality of the CDS, namely the private discovery of shared contacts, for which potential attacks may be mounted by other users of the application, i.e., from the inside.

7.4.1. Adversarial queries

We model adversary \mathcal{A} as polynomially-bounded probabilistic algorithm that has the following queries at its disposal to interact with protocol participants. By \mathcal{U}^h we denote the set of honest users in the system.

RequestCC(U, V)

Contact certificate $cc_{U \rightarrow V}$ issued by user $U \in \mathcal{U}^h$ for user V is given to the adversary. Note that this query corresponds to **AddContact** algorithm and models the possibility of selective contact corruptions.

Discover($U, \text{CL}, \text{partner}, r$)

User $U \in \mathcal{U}^h$ initiates a new session π of the **Discover** protocol, using all available certificates received from users listed in $\text{CL} \subseteq \mathcal{U}^h$, and using **partner** and r as further session parameters, where **partner** may not be an honest user, i.e., **partner** $\notin \mathcal{U}^h$. This query returns a first protocol message M (if available).

Note that restriction **partner** $\notin \mathcal{U}^h$ models assumed deployment of secure channels between users that execute the **Discover** protocol.

Send(π, M)

Message M is delivered to session π . After processing M , the output (if any) is given to \mathcal{A} . This query is ignored if π is not waiting for input.

Reveal(π)

This query returns $(\pi.\text{state}, \pi.\text{SCL})$.

Note that, as opposed to the model of **mAHA** in Section 5.3.1, we do not provide a query for user corruption. If defined, it would reveal user's secret key $U.\text{sk}$ and the set of stored certificates $cc_{V \rightarrow U}$. However, we argue that **Corrupt** queries are mainly needed to model forward secrecy in key establishment protocols. In particular, as the corruption of users in CDS would reveal all their contacts anyway, there is close to nothing left to protect against, in the CDS setting.

7.4.2. Contact-hiding security

Informally, the property of contact-hiding (CH) protects users from disclosing non-matching contacts to other participants. We model CH security by means of an experiment, following the indistinguishability approach. The goal of the adversary

is to decide which of two contact lists, CL_0^* or CL_1^* , is used in a challenge Discover session π^* . The adversary can invoke any number of independent Discover sessions, and perform Reveal and RequestCC queries at will. Observe the similarity between the CH experiment and affiliation-hiding security in Definition 23.

Definition 30 (Contact-hiding security) *Let $\text{CDS} = \{\text{InitUser}, \text{AddContact}, \text{Discover}\}$ and let $\text{Expt}_{\text{CDS}, \mathcal{A}}^{\text{ch}, 0}$ and $\text{Expt}_{\text{CDS}, \mathcal{A}}^{\text{ch}, 1}$ be the experiments specified in Figure 7.1. The advantage of adversary \mathcal{A} is defined as*

$$\text{Adv}_{\text{CDS}, \mathcal{A}}^{\text{ch}}(\kappa, n) = \left| \Pr \left[\text{Expt}_{\text{CDS}, \mathcal{A}}^{\text{ch}, 0}(\kappa, n) = 1 \right] - \Pr \left[\text{Expt}_{\text{CDS}, \mathcal{A}}^{\text{ch}, 1}(\kappa, n) = 1 \right] \right| .$$

We say that CDS is contact-hiding if $\text{Adv}_{\text{CDS}, \mathcal{A}}^{\text{ch}}$ is negligible in κ (for all n polynomially dependent on κ), for all efficient adversaries \mathcal{A} .

In Figure 7.1, conditions (1) and (2) exclude some trivial attacks on contact-hiding. In particular, condition (1) thwarts the attack where \mathcal{A} starts a $\text{Discover}(U', \text{CL}', \text{partner}', r')$ session π' with $\text{CL}' \cap \mathcal{D}^* \neq \emptyset$ and $(\pi'.\text{id}, \pi'.\text{partner}) = (\pi^*. \text{partner}, \pi^*. \text{id})$, relays all messages between π^* and π' , and finally asks $\text{Reveal}(\pi^*)$ or $\text{Reveal}(\pi')$. By protocol correctness, $\pi^*. \text{SCL} = \pi'. \text{SCL}$ would contain elements from \mathcal{D}^* , and it would be trivial to correctly decide about b . Condition (2) prevents \mathcal{A} from asking for contact certificates issued by users $V \in \mathcal{D}^*$ for a user $U' \in \mathcal{U}$, to simulate a protocol session on behalf of U' with challenge session π^* , and to decide about bit b from resulting SCL.

Observe that the ‘winning conditions’ in $\text{Expt}_{\text{CDS}, \mathcal{A}}^{\text{ch}, b}$ are simpler than those of $\text{Expt}_{\text{CDS}, \mathcal{A}}^{\text{ah}, b}$ from Definition 23. The reason for this is that (a) the model does not have to deal with user corruptions, and (b) due to a change in the syntax (cf. Definition 28), protocol partner $\pi^*. \text{partner}$ of the challenge session is known before the session is actually starts. An additional novelty is that the new security definition allows deterministic protocols, like the one we present in Section 7.6, to be proven secure.

Remark 5 (Variant of contact-hiding security) *Observe that, in experiment $\text{Expt}_{\text{CDS}, \mathcal{A}}^{\text{ch}, b}$, we do not pose requirements on sets $\text{CL}_0^*, \text{CL}_1^*$, except that we demand $|\text{CL}_0^*| = |\text{CL}_1^*|$. It is easily seen by a hybrid argument that a modified definition of contact-hiding security with the additional constraint $|\text{CL}_0^* \setminus \text{CL}_1^*| = 1 = |\text{CL}_1^* \setminus \text{CL}_0^*|$ would be equivalent to the one from Definition 30. In this case we would always have $|\mathcal{D}^*| = 2$.*

7.5. A CDS construction based on RSA

We present an RSA-based construction of CDS that is secure in the model presented in Section 7.4. It is quite similar to our mAHA scheme from Section 5.4.1. We

$\text{Expt}_{\text{CDS}, \mathcal{A}}^{\text{ch}, b}(\kappa, n)$:

- (a) the experiment creates a set of n users, denoted by $\mathcal{U} = \{U_1, \dots, U_n\}$. The adversary \mathcal{A} specifies a set $\mathcal{U}^c \subseteq \mathcal{U}$ of initially corrupted users. Let $\mathcal{U}^h = \mathcal{U} \setminus \mathcal{U}^c$. $\text{InitUser}(1^\kappa)$ is run for all $U \in \mathcal{U}^h$, and, for all combinations $(U, V) \in \mathcal{U}^h \times \mathcal{U}^h$, contact certificates $\text{cc}_{U \rightarrow V}$ are created by respective user U and given to V , each time by running the $\text{AddContact}(U, V)$ algorithm. For all $U \in \mathcal{U}^c$, the adversary sets up all parameters himself. He then specifies a list $\mathcal{L} \subseteq \mathcal{U}^h \times \mathcal{U}^c$, and for all $(U, V) \in \mathcal{L}$, algorithm $\text{AddContact}(U, V)$ is run, and the respective certificate $\text{cc}_{U \rightarrow V}$ is given to \mathcal{A} .
- (b) $\mathcal{A}(1^\kappa)$ interacts with all (honest) users using the queries from Section 7.4.1; at some point, \mathcal{A} outputs a tuple $(U^*, \text{CL}_0^*, \text{CL}_1^*, \text{partner}^*, r^*)$, where $U^* \in \mathcal{U}^h$, $\text{CL}_0^*, \text{CL}_1^* \subseteq \mathcal{U}^h$ with $|\text{CL}_0^*| = |\text{CL}_1^*|$, partner^* is any user id (in \mathcal{U}), and $r^* \in \{\text{init}, \text{resp}\}$. Let $\mathcal{D}^* = \Delta(\text{CL}_0^*, \text{CL}_1^*)$ denote the symmetric difference of the sets CL_0^* and CL_1^* (cf. Section 5.3.2).
- (c) the experiment invokes a $\text{Discover}(U^*, \text{CL}_b^*, \text{partner}^*, r^*)$ session π^* (and provides all needed certificates)
- (d) \mathcal{A} continues to interact via queries (including on session π^*), until it terminates and outputs bit b'
- (e) the output of the experiment is b' if all of the following hold; otherwise the output is 0:
 - (1) if there is a Discover session π' with $\mathcal{D}^* \cap \pi'.\text{CL} \neq \emptyset$ and $(\pi'.\text{id}, \pi'.\text{partner}) = (\pi^*.\text{partner}, \pi^*.\text{id})$, then neither $\text{Reveal}(\pi^*)$ nor $\text{Reveal}(\pi')$ was asked;
 - (2) for no user $V \in \mathcal{D}^*$, a $\text{RequestCC}(V, \pi^*.\text{id})$ or $\text{RequestCC}(V, \pi^*.\text{partner})$ query has been posed, or a pair $(V, \pi^*.\text{partner})$ is contained in \mathcal{L} , i.e., the adversary did not request a contact certificate for id^* or partner^* issued by any user in set \mathcal{D}^* .

Figure 7.1.: ch experiment

assume that all interactions between users during AddContact and Discover sessions are protected by secure channels, as motivated in Section 7.4. However, we claim that our scheme would also be secure if Discover sessions would be run over a channel that guarantees authenticity but not privacy.

7.5.1. Protocol specification

Let $\ell = \ell(\kappa)$ be polynomially dependent on security parameter κ . As in our mAHA protocol from Section 5.4.1, we use the perfect IHME scheme from Section 4.5.1 as a building block, where IHME is defined over field $\mathbb{F} = GF(T)$, for the smallest prime number T satisfying $T > 2^{\kappa+\ell}$. Moreover, let $H : \{0, 1\}^* \rightarrow [0, T - 1]$ and $H_n : \{0, 1\}^* \rightarrow \mathbb{Z}_n$ be hash functions, for any (RSA modulus) $n \in \mathbb{N}$.

The three algorithms of our CDS protocol are defined as follows:

InitUser

The setup routine run by each user generates safe RSA parameters $(n, e, d) \leftarrow_R \text{SRSA-GEN}(1^\kappa)$ (cf. Definition 3) and picks an element $g \in \mathbb{Z}_n^\times$ such that $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$ (cf. Section 2.5.1). User U keeps key $U.\text{sk} \leftarrow (n, g, e, d)$ secret.

AddContact

This routine is executed by user U , on input $U.\text{sk} = (n, g, e, d)$ and identifier $\text{id}_V \in \{0, 1\}^*$ of a user V . User V receives contact certificate $\text{cc}_{U \rightarrow V} = (n, g, e, \sigma_V)$, where σ_V is the RSA signature $\sigma_V = H_n(\text{id}_V)^d \bmod n$ on the full-domain hash of id_V (cf. Section 2.2.2).

Discover

The contact discovery protocol is executed between two users, U and U' , on inputs $(\text{CL}_U, \text{partner}_U, r_U)$ and $(\text{CL}_{U'}, \text{partner}_{U'}, r_{U'})$, respectively. The protocol is specified in detail in Figure 7.2, where padding function pad is defined as in Section 2.5.1.

As the protocol is rather similar to the RSA-based AHA protocols from Sections 2.5.1 and 5.4.1, we refer to these sections for an exposition of the working mechanisms of our CDS design. The main difference to the AHA schemes is that, in the CDS protocol, the key computation step has been removed: the scheme's output is just the list SCL of shared contacts. Correctness of the protocol follows as shown in Section 5.4.

7.5.2. Efficiency analysis

The similarity of our CDS and the mAHA scheme from Section 5.4 allows us to apply all efficiency considerations given in Section 5.4.2 also to the new protocol. In particular, we can apply all tweaks and improvements studied in Section 6.3, and would obtain the same timing measurements for protocol execution (see Section 6.3.4). However, as we argued in the introduction to this section that privacy-preserving contact discovery is important particularly on mobile devices, we made additional timing measurements on CPUs with reduced computational power. Figure 7.3 presents running times of our Discover protocol on different CPUs: a single

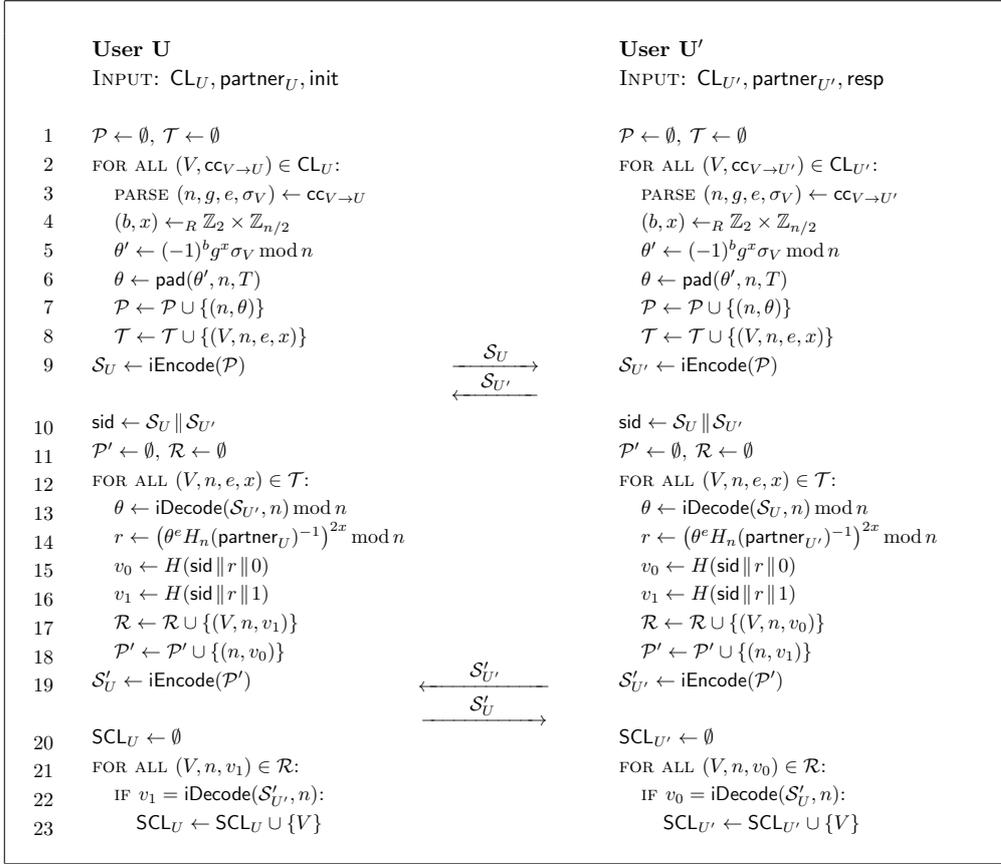


Figure 7.2.: RSA-based Discover protocol for private contact discovery

core of an Intel XEON 2.6 GHz CPU, an AMD NEO 1.6 GHz processor (often found in Netbook computers), and an ARMv7 600 MHz CPU (installed on many today's smartphones). All measurements were performed using the GMP library [76], thus, execution on smartphones presumably can even be speeded up by choosing a different cryptographic library, optimized for mobile environments. Note that, in our prototypes, we applied all optimizations from Section 6.3.

We observe that our protocol for private contact discovery scales fairly well. For security level $(\kappa, \ell) = (1024, 80)$, i.e., 1024-bit RSA moduli and 80-bit symmetric security, on laptops and server machines, a full protocol execution requires less than a second, even for 100 and more contacts per user. On cores with smaller footprint, e.g., on recent smartphones like Nokia's N900 (equipped with the ARMv7 600 MHz processor), protocol execution with 100 contacts requires about 5 seconds, which is an acceptable overhead. Note that smartphones' CPU speeds are envisioned to increase rapidly in the near future (e.g., the iPhone 4G is already equipped with a 1 GHz processor). Finally, we computed that each user sends and receives around 300 Bytes per user of his contact list, where we assume $|\text{CL}_U| = |\text{CL}_{U'}|$ for simplicity. That is, in the protocol execution with 100 contacts, a total of 30 KB is transmitted.

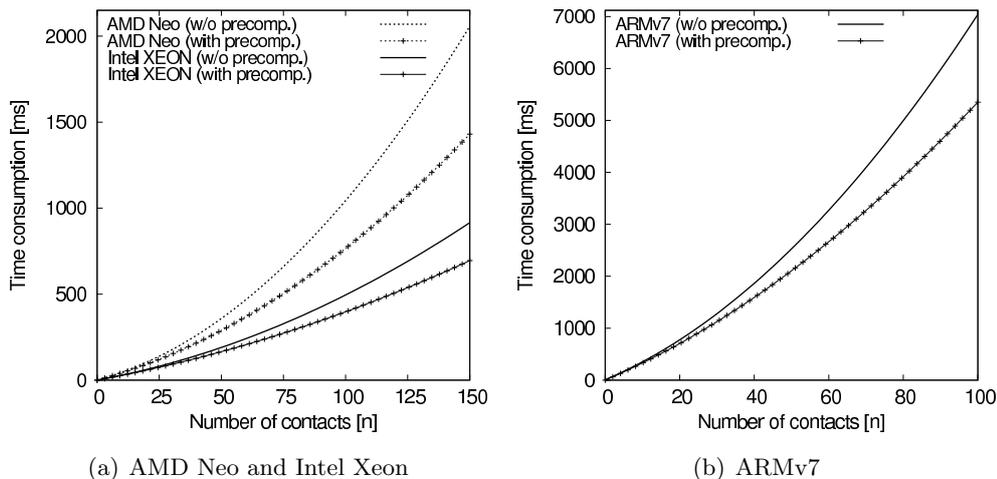


Figure 7.3.: Running times of our Discover protocol on different CPUs, in dependence of the number of contacts. For each CPU, we also consider session-independent (off-line) precomputations (cf. Section 6.1.5). All measurements are performed for 80 bit (symmetric) security and 1024 bit RSA moduli.

We conclude that our solution for private contact discovery is efficient and practical enough for actual deployment, also on smartphones widely available *today*. Yet, our technique does not give up solid privacy guarantees, as we show next.

7.5.3. Security analysis

Our CDS construction satisfies the security goal of contact-hiding, as formalized in Section 7.4.

Theorem 13 *Our RSA-based CDS scheme from Section 7.5.1 is contact-hiding under the RSA assumption on safe moduli, in the random oracle model.*

Proof. Besides to the experiments $\text{Expt}^{\text{ch},b}$ from Figure 7.1 (including the modification proposed in Remark 5), we will refer to a set of auxiliary games (experiments) that will help us to prove that our CDS scheme is contact-hiding. For each of these games \mathbf{G} , let $W = \Pr[\mathbf{G}(\kappa, n) = 1]$ denote the probability that \mathbf{G} 's execution results in the output of 1. We will parametrize these games with a bit b and denote this with a superscript, e.g., \mathbf{G}^b .

Fix adversary \mathcal{A} and parameters $\kappa, n = n(\kappa)$. We assume that, for any protocol session π , session variable $\pi.\text{sid}$ holds the value computed in line 10 in Figure 7.2, after receiving first protocol message \mathcal{S} . Consider the following games:

Game \mathbf{G}_0^b . This game is identical to $\text{Expt}_{\text{CDS}, \mathcal{A}}^{\text{ch}, b}(\kappa, n)$.

Our goal is to show that $|W_0^0 - W_0^1|$ is bounded by a negligible function. This holds trivially if the adversary violates conditions (1) or (2) in Figure 7.1, as

this would directly imply $W_0^0 = W_0^1 = 0$. We hence assume in the following that adversary complies with the named conditions.

Game \mathbf{G}_1^b . Game \mathbf{G}_1^b is like Game \mathbf{G}_0^b , except that the simulation is aborted if, for any user $U \in \mathcal{U}$ and any two sessions run by U , a collision of session ids occurs, i.e., if there exist sessions $\pi \neq \pi'$ with $(\pi.\text{id}, \pi.\text{sid}) = (\pi'.\text{id}, \pi'.\text{sid})$.

Observe that session ids, as assigned in line 10 of the protocol, contain values θ that are freshly and independently picked for each session and carry about $\log_2 T > \kappa + \ell$ bits of entropy each. By the birthday paradox, the probability of collisions of session ids to occur is bounded by $q_s^2/T < q_s^2/2^{\kappa+\ell}$, where q_s denotes the total number of posed Discover queries.

Game \mathbf{G}_2^b . Recall that for the challenge session π^* we have that $\pi^*.\text{id}$ and $\pi^*.\text{partner}$ identify users in $\mathcal{U} = \{U_1, \dots, U_n\}$. Game \mathbf{G}_2^b is like Game \mathbf{G}_1^b , except that the simulator makes a priori guesses on the pseudonyms $\text{id}^*, \text{id}' \in \mathcal{U}$ that will be $\pi^*.\text{id}$ and $\pi^*.\text{partner}$, respectively. If one of these guesses later turns out to be incorrect, i.e., if adversary demands challenge session be run for other users, then the experiment outputs a random bit (i.e., the simulation aborts).

Game \mathbf{G}_3^b . Game \mathbf{G}_3^b is like Game \mathbf{G}_2^b , except that the simulator makes an a priori guess on user U^b such that $\{U^b\} = \text{CL}_b^* \setminus \text{CL}_{1-b}^*$, out of a set of size $|\mathcal{U}^b| \leq n$. Note that we assume the modification to experiment $\text{Expt}^{\text{ch},b}$ that is proposed in Remark 5. If the guess on U^b later turns out to be incorrect, then the experiment outputs a random bit (i.e., the simulation aborts).

Game \mathbf{G}_4^b . Let r^* be the value r computed in challenge session π^* for contact U^b (line 14). Game \mathbf{G}_4^b is like Game \mathbf{G}_3^b , except that all confirmation messages v (lines 15 and 16), that are computed in session π^* and all sessions π' with $\pi^*.\text{sid} = \pi'.\text{sid}$ in dependence on r^* , are consistently replaced by random values in the range $[0, T - 1]$.

Observe that named confirmation tags are computed from r^* by hashing this value, using hash function H . By the random oracle model, adversary can detect the difference between Games \mathbf{G}_3^b and \mathbf{G}_4^b only by querying (a string that contains) r^* to this oracle. However, the probability of this to happen can be bounded by $\text{Succ}_{\text{SRSA-GEN}}^{\text{srsa}}$ (cf. Definition 3), as discussed in Section 2.5.1.

In particular, by embedding an SRSA challenge (n, e, z) into parameters n, g, e of user U^b and into pseudonym id' , a solution to the challenge can be computed from any hash query on r^* . Moreover, the actions of all (honest) users continue to be simulatable, with the exception that for user id' a $\text{RequestCC}(U^b, \text{id}')$ query cannot be processed. This behavior, however, is compliant with rule (2) in $\text{Expt}^{\text{ch},b}$. For further details on the reduction we refer to Section 2.5.1,

Appendix A, and to the analysis by Gennaro, Krawczyk, and Rabin [82]. We conclude that, for a constant c ,

$$|\Pr[W_4^b] - \Pr[W_3^b]| \leq c \cdot \text{Succ}_{\text{SRSA-GEN}, \mathcal{A}'}^{\text{srsa}}(\kappa) \quad (\text{for an adversary } \mathcal{A}').$$

Game \mathbf{G}_5^b . Game \mathbf{G}_5^b is like Game \mathbf{G}_4^b , except that value θ for contact U^b , as computed by session π^* in line 6, is replaced by a random element: $\theta \leftarrow_R [0, T-1]$.

Observe that, in the protocol, θ is exclusively used to compute r^* in line 14 (and, correspondingly, in sessions π' with $\pi^*.\text{sid} = \pi'.\text{sid}$). As we decoupled this value from the remaining simulation in Game \mathbf{G}_4^b , the difference between W_4^b and W_5^b is bounded by the statistical difference of the two methods to generate θ . As discussed in Section 2.5.1 and [95], this difference is negligible.

Game \mathbf{G}_6^b . Game \mathbf{G}_6^b is like Game \mathbf{G}_5^b , except that, in session π^* , we replace index n , used for IHME encoding value θ for contact U^b , by a fixed (unused) index, e.g., $n = 0$ (cf. lines 7 and 9).

The change introduced in Game \mathbf{G}_6^b corresponds to the security experiment of IHME's index-hiding property (cf. Figure 4.1): As θ is chosen uniformly from $[0, T-1]$, which coincides with IHME's message space \mathcal{M} , we can readily construct an IHME adversary \mathcal{A}' from any distinguisher between Games \mathbf{G}_5^b and \mathbf{G}_6^b . In the reduction, the set of moduli of the contacts in CL_b^* is assigned to index set I_0 , while the set of moduli of the contacts in $\text{CL}_b^* \setminus \{U^b\}$ together with index $n = 0$ is assigned to I_1 . As messages M corresponding to the indices in $I_0 \cap I_1$ the θ -values for the contacts in $\text{CL}_b^* \setminus \{U^b\}$ are taken without modification. We conclude that

$$|\Pr[W_6^b] - \Pr[W_5^b]| \leq \text{Adv}_{\text{IHME}, \mathcal{A}'}^{\text{ihide}}(\kappa) \quad (\text{for an adversary } \mathcal{A}').$$

Consider, in Game \mathbf{G}_6^b , the existence of a session π' such that $(\pi'.\text{id}, \pi'.\text{partner}) = (\pi^*.\text{partner}, \pi^*.\text{id})$ and $\mathcal{D}^* \cap \pi'.\text{CL} \neq \emptyset$.

If such a session does not exist, then verification tags v assigned by session π^* for contact U^b are random and completely independent from U^b and the rest of the simulation (recall the changes introduced in Game \mathbf{G}_4^b). In particular, (a) the tag v that π^* sends in lines 18 and 19 contains no information about contact U^b , (b) IHME structure \mathcal{S}' that π^* sends in line 19 leaks no information about G^b (by an argument similar to the one in the hop to Game \mathbf{G}_6^b), and (c) a $\text{Reveal}(\pi^*)$ query unveils no information about U^b , as the test in line 22 corresponding to contact U^b will pass only with negligible probability $1/T < 2^{-(\kappa+\ell)}$. Recall that the protocol's first message, \mathcal{S} , sent in line 9, does not leak information about group G^b since the hop to Game \mathbf{G}_6^b .

If such a session π' does exist, then posing $\text{Reveal}(\pi')$ or $\text{Reveal}(\pi^*)$ queries is not allowed (cf. condition (1) in Figure 7.1). Although the verification tag for contact U^b that π^* sends in line 19 is not independent from U^b in the simulation (it is potentially also computed and expected by session π'), it is so from the point of view of the adversary, as the latter has no means to learn how this tag is processed within π' .

In any case, we observe that the adversary cannot efficiently distinguish experiments \mathbf{G}_6^0 and \mathbf{G}_6^1 , i.e., we have $W_6^0 \approx W_6^1$. Putting everything together, we note that $\text{Adv}_{\text{CDS}, \mathcal{A}}^{\text{ch}}(\kappa, n) = |W_0^0 - W_0^1|$ is bounded by a negligible function, provided that the required assumptions hold. \square

7.6. A CDS construction based on NIKDS

We present a CDS that builds on ideas of the NIKDS-based mAHA scheme described in Section 5.5. Observe that, in contrast to the CDS in Section 7.5, the actual Discover protocol is deterministic.

7.6.1. Protocol specification

In our construction, we use a generic NIKDS scheme (cf. Definition 7) as central building block, e.g., the pairing-based construction from Section 2.3.1. The CDS is specified as follows:

InitUser

User U is initialized by running $\text{msk} \leftarrow \text{NSetup}(1^\kappa)$ and setting $U.\text{sk} \leftarrow \text{msk}$.

AddContact

The contact certificate issued by user U for identity $\text{id}_V \in \{0, 1\}^*$ of user V is computed as $\text{cc}_{U \rightarrow V} \leftarrow \text{NRegister}(\text{msk}, \text{id}_V)$, where $\text{msk} = U.\text{sk}$.

Discover

The specification of our Discover protocol is given in Figure 7.4. Besides the NIKDS, the algorithms make use of a hash function $H_1 : \{0, 1\}^* \rightarrow \{0, 1\}^\ell$, where $\ell = \ell(\kappa)$ is polynomially dependent on security parameter κ . As in Section 5.5.1, by $\text{Sort}(\mathcal{M})$ we denote the lexicographic ordering of \mathcal{M} . In regards to protocol's correctness, note that, for all contacts V common to U and U' , i.e., all users V such that $(V, \text{cc}_{V \rightarrow U}) \in \text{CL}_U$ and $(V, \text{cc}_{V \rightarrow U'}) \in \text{CL}_{U'}$, intermediate keys K' computed by U and U' in line 3 of the protocol are equal. Observe that the missing Diffie-Hellman key agreement (lines 1 and 3 in Figure 5.4) is not needed in protocols without key establishment.

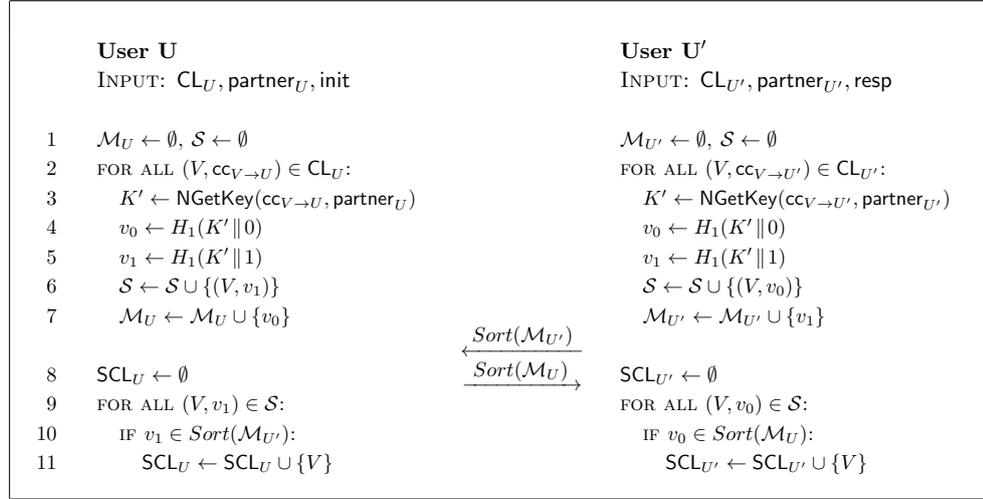


Figure 7.4.: NIKDS-based Discover protocol for private contact discovery

7.6.2. Efficiency analysis

Generally, the efficiency estimations from Sections 5.5.2 and 6.4 apply also to our CDS protocol. In particular, Discover has $O(n \log n)$ computational overhead, where $n = |\mathcal{G}|$ denotes the number of contacts per user. While the number of pairing evaluations (in the NIKDS) is only linear in the number of contacts, it is the $\text{Sort}(\mathcal{M})$ part that is responsible for the superlinear complexity. Observe that, in contrast to the mAHA protocol from Figure 5.4, our Discover protocol is a two-message protocol that is, moreover, deterministic (i.e., messages are dependent on the users and their certificates, but not on the session id).

7.6.3. Security analysis

Our CDS construction satisfies the security goal of contact-hiding, as formalized in Section 7.4.

Theorem 14 *Our NIKDS-based CDS from Section 7.6.1 is contact-hiding, given that NIKDS is IND-CIA secure, in the random oracle model.*

Proof. Besides to the experiments $\text{Expt}^{\text{ch},b}$ from Figure 7.1 (including the modification proposed in Remark 5), we will refer to a set of auxiliary games (experiments) that will help us to prove that our CDS scheme is contact-hiding. For each of these games \mathbf{G} , let $W = \Pr[\mathbf{G}(\kappa, n) = 1]$ denote the probability that \mathbf{G} 's execution results in the output of 1. We will parametrize these games with a bit b and denote this with a superscript, e.g., \mathbf{G}^b .

Fix adversary \mathcal{A} and parameters $\kappa, n = n(\kappa)$. Consider the following games:

Game \mathbf{G}_0^b . This game is identical to $\text{Expt}_{\text{CDS}, \mathcal{A}}^{\text{ch},b}(\kappa, n)$.

Our goal is to show that $|W_0^0 - W_0^1|$ is bounded by a negligible function. This holds trivially if the adversary violates conditions (1) or (2) in Figure 7.1, as this would directly imply $W_0^0 = W_0^1 = 0$. We hence assume in the following that adversary complies with the named conditions.

Game \mathbf{G}_1^b . Recall that for the challenge session π^* we have that $\pi^*.id$ and $\pi^*.partner$ identify users in $\mathcal{U} = \{U_1, \dots, U_n\}$. Game \mathbf{G}_1^b is like Game \mathbf{G}_0^b , except that the simulator makes a priori guesses on the pseudonyms $id^*, id' \in \mathcal{U}$ that will be $\pi^*.id$ and $\pi^*.partner$, respectively. If one of these guesses later turns out to be incorrect, i.e., if adversary demands challenge session be run for other pseudonyms, then the experiment outputs a random bit (i.e., the simulation aborts).

Game \mathbf{G}_2^b . Game \mathbf{G}_2^b is like Game \mathbf{G}_1^b , except that the simulator makes an a priori guess on user U^b such that $\{U^b\} = \mathbf{CL}_b^* \setminus \mathbf{CL}_{1-b}^*$, out of a set of size $|\mathcal{U}| = n$. Note that we assume the modification to experiment $\text{Expt}^{\text{ch},b}$ that is proposed in Remark 5. If the guess on U^b later turns out to be incorrect, then the experiment outputs a random bit (i.e., the simulation aborts).

Game \mathbf{G}_3^b . Game \mathbf{G}_3^b is like Game \mathbf{G}_2^b , except that, for all sessions π with $\{\pi.id, \pi.partner\} = \{id^*, id'\}$, confirmation messages v_0, v_1 for contact U^b , as computed in lines 4 and 5 of the protocol, are assigned via $v_d \leftarrow \bar{v}_d$, where $\bar{v}_d \in_R \{0, 1\}^\ell$, $d \in \{0, 1\}$, are fixed but random tokens. In particular, the v_d are assigned independently of NIKDS key K' .

As seen in Figure 7.4, the simulation of Game \mathbf{G}_2^b uses key $K' = \text{NSharedKey}(U^b.sk; id^*, id')$ exactly for the computation of the v_0, v_1 specified above, and nowhere else. The modification introduced in this game can be detected by adversary \mathcal{A} only by posing an H_1 query on (a string that contains) respective NIKDS's key K' . By embedding an OW-CIA challenge (cf. Definition 9) into credentials issued by U^b for pseudonyms id^*, id' , the probability of this to happen can be bounded by a negligible function (where q_{H_1} denotes the total number of posed H_1 queries):

$$|\Pr[W_3^b] - \Pr[W_2^b]| = q_{H_1} \text{Succ}_{\text{NIKDS}, \mathcal{A}'}^{\text{ow-cia}}(\kappa) \quad (\text{for an adversary } \mathcal{A}').$$

Observe that, in this step, we exploited condition (2) from experiment $\text{Expt}^{\text{ch},b}$: For common contact U^b , neither $\text{RequestCC}(U^b, id^*)$ nor $\text{RequestCC}(U^b, id')$ may be asked by the adversary.

Consider, in Game \mathbf{G}_3^b , the existence of a session π' such that $(\pi'.id, \pi'.partner) = (\pi^*.partner, \pi^*.id)$ and $\mathcal{D}^* \cap \pi'.\text{CL} \neq \emptyset$.

If such a session does not exist, then verification tags $v_d^* = \bar{v}_d$ assigned by session π^* for contact U^b are random and completely independent from U^b and the rest of the simulation. In particular, (a) message $Sort(\mathcal{M})$ that π^* sends contains no information about contact U^b , and (b) a $Reveal(\pi^*)$ query unveils no information about U^b , as the test in line 10 corresponding to contact U^b will pass only with negligible probability $|Sort(\mathcal{M})|/2^\ell \leq m/2^\ell$.

If such a session π' does exist, then posing $Reveal(\pi')$ or $Reveal(\pi^*)$ queries is not allowed (cf. condition (1) in Figure 7.1). Although the verification tag for contact U^b that π^* sends in line 7 is not independent from U^b in the simulation (it is potentially also computed and expected by session π'), it is so from the point of view of the adversary, as the latter has no means to learn how this tag is processed within π' .

In any case, we observe that the adversary cannot efficiently distinguish experiments \mathbf{G}_3^0 and \mathbf{G}_3^1 , i.e., we have $W_3^0 \approx W_3^1$. Putting everything together, we note that $\text{Adv}_{\text{CDS}, \mathcal{A}}^{\text{ch}}(\kappa, n) = |W_0^0 - W_0^1|$ is bounded by a negligible function, provided that the required assumption on NIKDS holds. \square

The results of this thesis contribute to the understanding and deployment of privacy-preserving authentication in manifold ways. We summarize the major achievements in a short overview:

AHA with untrusted group authorities (Chapter 3)

In the setting of affiliation-hiding authentication (AHA), we analyze the means to repel the threat of dishonest or corrupt group authorities (GAs). We highlight that, while prior attempts to deal with GA corruption [104, 151] resulted in cumbersome settings where GAs are split in two or more (sub)authorities, our strengthened security model not only allows staying in the more practical one-GA setting, thus facilitating more robust and reliable constructions, but also defends against a wider range of attacks; for instance, it also averts GA attacks on session key security. We construct an RSA-based AHA protocol that is secure in the new model. In particular, our linkable scheme is the first *untraceable* one, i.e., even GAs cannot reveal identities of its users.

AHA in the multi-affiliation setting (Chapters 4 and 5)

We explore the multi-affiliation AHA (mAHA) setting where users are affiliated to multiple independent groups at the same time, and aim at learning from protocol sessions the set of groups they have in common. After motivating this scenario and proposing different ways to achieve the desired functionality, we select the most promising method (that bases on a new ‘IHME’ primitive) and construct two mAHA solutions (one RSA-based, one pairing-based). The efficiency of our protocols is $O(n)$ public key operations, where n is the number of affiliations per user. We expect this bound to be optimal.

Our multi-affiliation variant of AHA makes deployment of such privacy-aware authentication methods much more attractive and likely in practice, as this setting naturally arises in collaborative applications such as online social networks. At the same time our protocols offer the first satisfactory solution to

an open problem posed by Jarecki *et al.* in 2008 [95, p. 356], who ask for an efficient construction of mAHA.

Implementability of AHA (Chapter 6)

With the aim of obtaining meaningful performance evaluations of our mAHA solutions, we demonstrate multifaceted ways to further optimize our protocols in general, and our IHME primitive in particular (in respect to runtime and consumption of memory and bandwidth). We implement the protocols, and our measurements clearly document that mAHA protocols are now practically deployable on a wide range of architectures; for instance, assuming that participants are member of up to 50 groups, a full privacy-aware group discovery takes only 150 ms on an average PC.

For all non-trivial routines of the optimized protocols, we provide elaborate algorithmic descriptions that allow direct implementation. We also make proposals on parameter choices for multiple security settings.

Private discovery of common social contacts (Chapter 7)

We provide a treatment of the topic of *private contact discovery*. This setting assumes that participants individually manage lists of their respective friends ('contacts'). If two users jointly execute a contact-discovering protocol, on input their contact lists, the protocol identifies the set of contacts they have in common. This matching is performed in a privacy-preserving way, i.e., without disclosing non-matching contacts to the respective peer. After showing that all previously published approaches to this challenge suffer from severe privacy shortcomings, we construct two provably-secure solutions. The efficiency of our protocols is $O(n)$ public key operations, where n is the number of contacts per user.

During protocol design, we overcome several challenges. Amongst others, in order to prevent adversaries from arbitrarily expanding their contact lists to maximize the amount of information learned about the peer, we introduce the concept of *contact certification*. We also show, through experimental evaluation, that our solutions are practical enough to be deployed in real-world applications, including those running on mobile devices.

We stress that our contributions and treatments follow the rules of modern cryptography, i.e., we precisely specify execution and attack models, and offer elaborate security reductions to well-accepted hardness assumptions like RSA or the Bilinear Diffie-Hellman Problem to support the security of our protocols.

All presented models and protocols have been published in the proceedings of peer-reviewed international conferences (cf. Appendix B), with the exception of the protocol from Section 7.6, which was specifically designed for this thesis.

8.1. Directions for future research

The research field of affiliation-hiding authentication (AHA) is quite young. Although this thesis closes some open problems in this setting, a variety of topics is left for future research.

Multi-party mAHA

As we report in Section 1.3, some AHA constructions proposed in the literature generalize the standard two-party setting of authentication to a multi-party setting [93, 94, 155, 162]. Such a scenario is certainly interesting in practice, for instance in the context of online social networks.

However, the listed protocols lack support for users that are affiliated to multiple groups at the same time (mAHA). Corresponding sessions would privately detect the set of groups that *all* participants have in common. We propose to investigate whether the techniques we develop in Chapters 4 and 5 can be combined with the approaches from [93, 94, 155, 162] to obtain a first multi-party multi-affiliation AHA protocol.

mAHA in standard model

All schemes described in this thesis have security reductions that assume random oracles. Hence, an alternative *standard model* construction for mAHA (and likewise for a contact discovery scheme CDS) would be appealing. We observe that simply ‘adjusting’ our protocols towards standard model is not straight-forward. In particular, it is unclear how our main building blocks — Okamoto’s RSA-based key agreement and the NIKDS construction by Sakai, Ohgishi, and Kasahara — could be replaced by equivalent standard model tools.

Fairness in AHA/mAHA

Another interesting research direction is to consider *fairness* [4, 27] in AHA. Observe that, in all protocols discussed in Section 1.2, one of the authenticating parties will learn about matching groups before the other does. Moreover, this party can modify its last messages to trick its peer to assume that authentication failed.

The property of fairness ensures a balanced gain of knowledge of protocol participants even against insider adversaries. In other settings, fairness is often achieved via trusted third parties (that mostly stay offline) [4], or through protocols with a large number of rounds [27].

Privacy-aware revocation

In linkable AHA protocols [9, 47, 94, 95, 97], revocation of users is usually handled via revocation lists that are maintained and published by the GAs. A

consequence is that users lose their privacy upon revocation: once their pseudonym appears on the revocation list of their respective group, all their past AHA sessions can immediately be linked to this group.

Ways to handle this issue could be the introduction of a global revocation list (i.e., revocation would be ‘outsourced’ from the GAs to a single third party), or the limitation of all credentials’ validity to a short time frame. In this case, users would have to regularly contact their GAs to obtain new credentials, and revocation would be achieved by simply not renewing credentials of revoked users.

However, we wonder whether more practical ways to handle revocation in linkable AHA protocols do exist, that also preserve privacy of revoked users.

Private path discovery

In Chapter 7 we treat the topic of private discovery of common social contacts and give constructions that allow two users to detect the list of their common friends in a privacy-preserving way. Now, given two users, Alice and David, can they efficiently discover in a privacy-aware two-party protocol whether there exists a ‘chain of friendship’ between them? For example, can they efficiently discover whether there are also Bob and Charlie such that Alice is friend of Bob, Bob is friend of Charlie, and Charlie is friend of David?

Future research could answer the question whether the private discovery of such i -th grade contacts is possible without relying on trusted third parties.

Bibliography

- [1] Martín Abadi. Private authentication. In Roger Dingledine and Paul F. Syver-son, editors, *Privacy Enhancing Technologies*, volume 2482 of *Lecture Notes in Computer Science*, pages 27–40. Springer, 2002.
- [2] Diego F. Aranha, Koray Karabina, Patrick Longa, Catherine H. Gebotys, and Julio López. Faster explicit formulas for computing pairings over ordinary curves. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 48–68, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Germany.
- [3] Diego F. Aranha, Julio López, and Darrel Hankerson. High-speed parallel software implementation of the η T pairing. In Josef Pieprzyk, editor, *Topics in Cryptology – CT-RSA 2010*, volume 5985 of *Lecture Notes in Computer Science*, pages 89–105, San Francisco, CA, USA, March 1–5, 2010. Springer, Berlin, Germany.
- [4] N. Asokan, Victor Shoup, and Michael Waidner. Optimistic fair exchange of digital signatures (extended abstract). In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT’98*, volume 1403 of *Lecture Notes in Computer Science*, pages 591–606, Espoo, Finland, May 31 – June 4, 1998. Springer, Berlin, Germany.
- [5] Giuseppe Ateniese, Emiliano De Cristofaro, and Gene Tsudik. (If) size matters: Size-hiding private set intersection. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011: 14th International Workshop on Theory and Practice in Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 156–173, Taormina, Italy, March 6–9, 2011. Springer, Berlin, Germany.

-
- [6] Giuseppe Ateniese, Jonathan Kirsch, and Marina Blanton. Secret handshakes with dynamic and fuzzy matching. In *ISOC Network and Distributed System Security Symposium – NDSS 2007*, San Diego, California, USA, February 28 – March 2, 2007. The Internet Society.
- [7] Yossi Azar, Andrei Z. Broder, Anna R. Karlin, and Eli Upfal. Balanced allocations. *SIAM J. Comput.*, 29(1):180–200, 1999.
- [8] Kooshiar Azimian, Javad Mohajeri, and Mahmoud Salmasizadeh. Weak composite Diffie-Hellman. *I. J. Network Security*, 7(3):383–387, 2008.
- [9] Dirk Balfanz, Glenn Durfee, Narendar Shankar, Diana K. Smetters, Jessica Staddon, and Hao-Chi Wong. Secret handshakes from pairing-based key agreements. In *IEEE Symposium on Security and Privacy*, pages 180–196. IEEE Computer Society, 2003.
- [10] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 566–582, Gold Coast, Australia, December 9–13, 2001. Springer, Berlin, Germany.
- [11] Mihir Bellare, Tadayoshi Kohno, and Victor Shoup. Stateful public-key cryptosystems: How to encrypt with one 160-bit exponentiation. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06: 13th Conference on Computer and Communications Security*, pages 380–389, Alexandria, Virginia, USA, October 30 – November 3, 2006. ACM Press.
- [12] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629, Warsaw, Poland, May 4–8, 2003. Springer, Berlin, Germany.
- [13] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Se-manko. The power of RSA inversion oracles and the security of Chaum’s RSA-based blind signature scheme. In Paul F. Syverson, editor, *FC 2001: 5th International Conference on Financial Cryptography*, volume 2339 of *Lecture Notes in Computer Science*, pages 319–338, Grand Cayman, British West Indies, February 19–22, 2001. Springer, Berlin, Germany.
- [14] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO’93*, volume

- 773 of *Lecture Notes in Computer Science*, pages 232–249, Santa Barbara, CA, USA, August 22–26, 1994. Springer, Berlin, Germany.
- [15] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT’96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416, Saragossa, Spain, May 12–16, 1996. Springer, Berlin, Germany.
- [16] Vicente Benjumea, Seung Geol Choi, Javier Lopez, and Moti Yung. Fair traceable multi-group signatures. In Gene Tsudik, editor, *FC 2008: 12th International Conference on Financial Cryptography and Data Security*, volume 5143 of *Lecture Notes in Computer Science*, pages 231–246, Cozumel, Mexico, January 28–31, 2008. Springer, Berlin, Germany.
- [17] Patrik Bichsel, Jan Camenisch, Gregory Neven, Nigel P. Smart, and Bogdan Warinschi. Get shorty via group signatures without encryption. In Juan A. Garay and Roberto De Prisco, editors, *SCN 10: 7th International Conference on Security in Communication Networks*, volume 6280 of *Lecture Notes in Computer Science*, pages 381–398, Amalfi, Italy, September 13–15, 2010. Springer, Berlin, Germany.
- [18] Eli Biham, Dan Boneh, and Omer Reingold. Breaking generalized Diffie-Hellmann modulo a composite is no easier than factoring. *Inf. Process. Lett.*, 70(2):83–87, 1999.
- [19] James Birkett and Douglas Stebila. Predicate-based key exchange. In Ron Steinfeld and Philip Hawkes, editors, *ACISP 10: 15th Australasian Conference on Information Security and Privacy*, volume 6168 of *Lecture Notes in Computer Science*, pages 282–299, Sydney, NSW, Australia, July 5–7, 2010. Springer, Berlin, Germany.
- [20] Ake Björck and Victor Pereyra. Solution of Vandermonde systems of equations. *Mathematics of Computation*, 24(112):893–903, 1970.
- [21] Ian Blake, Gadiel Seroussi, and Nigel Smart. *Elliptic curves in cryptography*. Cambridge University Press, New York, NY, USA, 1999.
- [22] Ian Blake, Gadiel Seroussi, and Nigel Smart. *Advances in Elliptic Curve Cryptography*. Cambridge University Press, New York, NY, USA, 2005.
- [23] Simon Blake-Wilson, Don Johnson, and Alfred Menezes. Key agreement protocols and their security analysis. In Michael Darnell, editor, *6th IMA International Conference on Cryptography and Coding*, volume 1355 of *Lecture*

- Notes in Computer Science*, pages 30–45, Cirencester, UK, December 17–19, 1997. Springer, Berlin, Germany.
- [24] Dan Boneh. The decision Diffie-Hellman problem. In *Third Algorithmic Number Theory Symposium (ANTS)*, volume 1423 of *Lecture Notes in Computer Science*. Springer, Berlin, Germany, 1998. Invited paper.
- [25] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Berlin, Germany.
- [26] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick McDaniel, editors, *ACM CCS 04: 11th Conference on Computer and Communications Security*, pages 168–177, Washington D.C., USA, October 25–29, 2004. ACM Press.
- [27] Fabrice Boudot, Berry Schoenmakers, and Jacques Traoré. A fair and efficient solution to the socialist millionaires’ problem. *Discrete Applied Mathematics*, 111(1-2):23–36, 2001.
- [28] Colin Boyd and Anish Mathuria. *Protocols for Authentication and Key Establishment*. Springer, 2003.
- [29] Robert W. Bradshaw, Jason E. Holt, and Kent E. Seamons. Concealing complex policies with hidden credentials. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick McDaniel, editors, *ACM CCS 04: 11th Conference on Computer and Communications Security*, pages 146–157, Washington D.C., USA, October 25–29, 2004. ACM Press.
- [30] Emmanuel Bresson, Olivier Chevassut, David Pointcheval, and Jean-Jacques Quisquater. Provably authenticated group Diffie-Hellman key exchange. In *ACM CCS 01: 8th Conference on Computer and Communications Security*, pages 255–264, Philadelphia, PA, USA, November 5–8, 2001. ACM Press.
- [31] Andrei Z. Broder and Michael Mitzenmacher. Using multiple hash functions to improve IP lookups. In *INFOCOM*, pages 1454–1463, 2001.
- [32] Christina Brzuska, Heike Busch, Özgür Dagdelen, Marc Fischlin, Martin Franz, Stefan Katzenbeisser, Mark Manulis, Cristina Onete, Andreas Peter, Bertram Poettering, and Dominique Schröder. Redactable signatures for tree-structured data: Definitions and constructions. In Jianying Zhou and Moti

- Yung, editors, *ACNS 10: 8th International Conference on Applied Cryptography and Network Security*, volume 6123 of *Lecture Notes in Computer Science*, pages 87–104, Beijing, China, June 22–25, 2010. Springer, Berlin, Germany.
- [33] Mike Burmester and Yvo Desmedt. A secure and efficient conference key distribution system (extended abstract). In Alfredo De Santis, editor, *Advances in Cryptology – EUROCRYPT’94*, volume 950 of *Lecture Notes in Computer Science*, pages 275–286, Perugia, Italy, May 9–12, 1994. Springer, Berlin, Germany.
- [34] Jan Camenisch, Nathalie Casati, Thomas Groß, and Victor Shoup. Credential authenticated identification and key exchange. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 255–276, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Berlin, Germany.
- [35] Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 08: 15th Conference on Computer and Communications Security*, pages 345–356, Alexandria, Virginia, USA, October 27–31, 2008. ACM Press.
- [36] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In Stanislaw Jarecki and Gene Tsudik, editors, *PKC 2009: 12th International Conference on Theory and Practice of Public Key Cryptography*, volume 5443 of *Lecture Notes in Computer Science*, pages 481–500, Irvine, CA, USA, March 18–20, 2009. Springer, Berlin, Germany.
- [37] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. Solving revocation with efficient update of anonymous credentials. In Juan A. Garay and Roberto De Prisco, editors, *SCN 10: 7th International Conference on Security in Communication Networks*, volume 6280 of *Lecture Notes in Computer Science*, pages 454–471, Amalfi, Italy, September 13–15, 2010. Springer, Berlin, Germany.
- [38] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118, Innsbruck, Austria, May 6–10, 2001. Springer, Berlin, Germany.
- [39] Jan Camenisch and Anna Lysyanskaya. An identity escrow scheme with appointed verifiers. In Joe Kilian, editor, *Advances in Cryptology –*

- CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 388–407, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Berlin, Germany.
- [40] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76, Santa Barbara, CA, USA, August 18–22, 2002. Springer, Berlin, Germany.
- [41] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 56–72, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Berlin, Germany.
- [42] Jan Camenisch and Markus Michels. Proving in zero-knowledge that a number is the product of two safe primes. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT’99*, volume 1592 of *Lecture Notes in Computer Science*, pages 107–122, Prague, Czech Republic, May 2–6, 1999. Springer, Berlin, Germany.
- [43] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups (extended abstract). In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO’97*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424, Santa Barbara, CA, USA, August 17–21, 1997. Springer, Berlin, Germany.
- [44] Jan Camenisch and Gregory M. Zaverucha. Private intersection of certified sets. In Roger Dingledine and Philippe Golle, editors, *FC 2009: 13th International Conference on Financial Cryptography and Data Security*, volume 5628 of *Lecture Notes in Computer Science*, pages 108–127, Accra Beach, Barbados, February 23–26, 2009. Springer, Berlin, Germany.
- [45] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 453–474, Innsbruck, Austria, May 6–10, 2001. Springer, Berlin, Germany.
- [46] Ran Canetti and Hugo Krawczyk. Universally composable notions of key exchange and secure channels. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*

- ence, pages 337–351, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer, Berlin, Germany.
- [47] Claude Castelluccia, Stanislaw Jarecki, and Gene Tsudik. Secret handshakes from CA-oblivious encryption. In Pil Joong Lee, editor, *Advances in Cryptology – ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 293–307, Jeju Island, Korea, December 5–9, 2004. Springer, Berlin, Germany.
- [48] Peter Chapman, David Evans, Yan Huang, and Sang Koo. Common Contacts — privacy-preserving shard contact computation.
- [49] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology – CRYPTO’82*, pages 199–203, Santa Barbara, CA, USA, 1983. Plenum Press, New York, USA.
- [50] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, 1985.
- [51] David Chaum and Jan-Hendrik Evertse. A secure and privacy-protecting protocol for transmitting personal information between organizations. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO’86*, volume 263 of *Lecture Notes in Computer Science*, pages 118–167, Santa Barbara, CA, USA, August 1987. Springer, Berlin, Germany.
- [52] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In Shafi Goldwasser, editor, *Advances in Cryptology – CRYPTO’88*, volume 403 of *Lecture Notes in Computer Science*, pages 319–327, Santa Barbara, CA, USA, August 21–25, 1990. Springer, Berlin, Germany.
- [53] David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *Advances in Cryptology – EUROCRYPT’91*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265, Brighton, UK, April 8–11, 1991. Springer, Berlin, Germany.
- [54] Shin-Yan Chiou, Shih-Ying Chang, and Hung-Min Sun. Common friends discovery with privacy and authenticity. In *IAS*, pages 337–340. IEEE Computer Society, 2009.
- [55] Seung Geol Choi, Kunsoo Park, and Moti Yung. Short traceable signatures based on bilinear pairings. In Hiroshi Yoshiura, Kouichi Sakurai, Kai Rannenberg, Yuko Murayama, and Shin-ichi Kawamura, editors, *IWSEC 06: 1st International Workshop on Security, Advances in Information and Computer*

- Security*, volume 4266 of *Lecture Notes in Computer Science*, pages 88–103, Kyoto, Japan, October 23–24, 2006. Springer, Berlin, Germany.
- [56] Henri Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag New York, 1993.
- [57] Craig Costello and Douglas Stebila. Fixed argument pairings. In Michel Abdalla and Paulo S. L. M. Barreto, editors, *Progress in Cryptology - LATIN-CRYPT 2010: 1st International Conference on Cryptology and Information Security in Latin America*, volume 6212 of *Lecture Notes in Computer Science*, pages 92–108, Puebla, Mexico, August 8–11, 2010. Springer, Berlin, Germany.
- [58] Emiliano De Cristofaro, Stanislaw Jarecki, Jihye Kim, and Gene Tsudik. Privacy-preserving policy-based information transfer. In Ian Goldberg and Mikhail J. Atallah, editors, *Privacy Enhancing Technologies*, volume 5672 of *Lecture Notes in Computer Science*, pages 164–184. Springer, 2009.
- [59] Dana Dachman-Soled, Tal Malkin, Mariana Raykova, and Moti Yung. Efficient robust private set intersection. In Michel Abdalla, David Pointcheval, Pierre-Alain Fouque, and Damien Vergnaud, editors, *ACNS 09: 7th International Conference on Applied Cryptography and Network Security*, volume 5536 of *Lecture Notes in Computer Science*, pages 125–142, Paris-Rocquencourt, France, June 2–5, 2009. Springer, Berlin, Germany.
- [60] Ivan Damgård. Payment systems and credential mechanisms with provable security against abuse by individuals. In Shafi Goldwasser, editor, *Advances in Cryptology – CRYPTO’88*, volume 403 of *Lecture Notes in Computer Science*, pages 328–335, Santa Barbara, CA, USA, August 21–25, 1990. Springer, Berlin, Germany.
- [61] Emiliano De Cristofaro, Jihye Kim, and Gene Tsudik. Linear-complexity private set intersection protocols secure in malicious model. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 213–231, Singapore, December 5–9, 2010. Springer, Berlin, Germany.
- [62] Emiliano De Cristofaro, Mark Manulis, and Bertram Poettering. Private discovery of common social contacts. In Javier Lopez and Gene Tsudik, editors, *ACNS 11: 9th International Conference on Applied Cryptography and Network Security*, volume 6715 of *Lecture Notes in Computer Science*, pages 147–165, Nerja, Spain, June 7–10, 2011. Springer, Berlin, Germany.
- [63] Emiliano De Cristofaro and Gene Tsudik. Practical private set intersection protocols with linear complexity. In Radu Sion, editor, *FC 2010: 14th In-*

- ternational Conference on Financial Cryptography and Data Security*, volume 6052 of *Lecture Notes in Computer Science*, pages 143–159, Tenerife, Canary Islands, Spain, January 25–28, 2010. Springer, Berlin, Germany.
- [64] Mongkol Dejnakarindra and David Banjerdpongchai. An algorithm for computing the analytical inverse of the Vandermonde matrix. In *3rd Asian Control Conference (ASCC)*, 2000.
- [65] Yvo Desmedt. Securing traceability of ciphertexts - towards a secure software key escrow system (extended abstract). In Louis C. Guillou and Jean-Jacques Quisquater, editors, *Advances in Cryptology – EUROCRYPT’95*, volume 921 of *Lecture Notes in Computer Science*, pages 147–157, Saint-Malo, France, May 21–25, 1995. Springer, Berlin, Germany.
- [66] Giovanni Di Crescenzo. Private selective payment protocols. In Yair Frankel, editor, *FC 2000: 4th International Conference on Financial Cryptography*, volume 1962 of *Lecture Notes in Computer Science*, pages 72–89, Anguilla, British West Indies, February 20–24, 2000. Springer, Berlin, Germany.
- [67] Giovanni Di Crescenzo, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. Conditional oblivious transfer and timed-release encryption. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT’99*, volume 1592 of *Lecture Notes in Computer Science*, pages 74–89, Prague, Czech Republic, May 2–6, 1999. Springer, Berlin, Germany.
- [68] Christopher P. Diehl, Galileo Namata, and Lise Getoor. Relationship identification for social network discovery. In *AAAI*, pages 546–552. AAAI Press, 2007.
- [69] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.
- [70] Whitfield Diffie, Paul C. van Oorschot, and Michael J. Wiener. Authentication and authenticated key exchanges. *Des. Codes Cryptography*, 2(2):107–125, 1992.
- [71] Régis Dupont and Andreas Enge. Provably secure non-interactive key distribution based on pairings. *Discrete Applied Mathematics*, 154(2):270–276, 2006.
- [72] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology – CRYPTO’84*, volume 196 of *Lecture Notes in Computer Science*,

- pages 10–18, Santa Barbara, CA, USA, August 19–23, 1985. Springer, Berlin, Germany.
- [73] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988.
- [74] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO’86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer, Berlin, Germany.
- [75] Marc Fischlin. Anonymous signatures made easy. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007: 10th International Conference on Theory and Practice of Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 31–42, Beijing, China, April 16–20, 2007. Springer, Berlin, Germany.
- [76] Free Software Foundation. The GNU MP Bignum Library. <http://gmplib.org/>.
- [77] M. J. Freedman and A. Nicolosi. Efficient private techniques for verifying social proximity. In *IPTPS*, 2007.
- [78] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 1–19, Interlaken, Switzerland, May 2–6, 2004. Springer, Berlin, Germany.
- [79] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Appl. Math.*, 156:3113–3121, September 2008.
- [80] He Ge and Stephen R. Tate. Traceable signature: Better efficiency and beyond. In Marina L. Gavrilova, Osvaldo Gervasi, Vipin Kumar, Chih Jeng Kenneth Tan, David Taniar, Antonio Laganà, Youngsong Mun, and Hyunseung Choo, editors, *ICCSA (3)*, volume 3982 of *Lecture Notes in Computer Science*, pages 327–337. Springer, 2006.
- [81] Rosario Gennaro, Hugo Krawczyk, and Tal Rabin. Okamoto-Tanaka revisited: Fully authenticated Diffie-Hellman with minimal overhead. In Jianying Zhou and Moti Yung, editors, *ACNS 10: 8th International Conference on Applied Cryptography and Network Security*, volume 6123 of *Lecture Notes in*

- Computer Science*, pages 309–328, Beijing, China, June 22–25, 2010. Springer, Berlin, Germany.
- [82] Rosario Gennaro, Hugo Krawczyk, and Tal Rabin. Okamoto-Tanaka revisited: Fully authenticated Diffie-Hellman with minimal overhead. Cryptology ePrint Archive, Report 2010/068, 2010. <http://eprint.iacr.org/2010/068.pdf>.
- [83] Oded Goldreich and Vered Rosen. On the security of modular exponentiation with application to the construction of pseudorandom generators. *Journal of Cryptology*, 16(2):71–93, March 2003.
- [84] Gene H. Golub and Charles F. van Loan. *Matrix computations (3. ed.)*. Johns Hopkins University Press, 1996.
- [85] Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In C. G. Günther, editor, *Advances in Cryptology – EURO-CRYPT’88*, volume 330 of *Lecture Notes in Computer Science*, pages 123–128, Davos, Switzerland, May 25–27, 1988. Springer, Berlin, Germany.
- [86] Darrel Hankerson, Alfred Menezes, and Scott Vanstone. *Guide to Elliptic Curve Cryptography*. Springer, 2004.
- [87] Carmit Hazay and Yehuda Lindell. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. In Ran Canetti, editor, *TCC 2008: 5th Theory of Cryptography Conference*, volume 4948 of *Lecture Notes in Computer Science*, pages 155–175, San Francisco, CA, USA, March 19–21, 2008. Springer, Berlin, Germany.
- [88] Carmit Hazay and Kobbi Nissim. Efficient set operations in the presence of malicious adversaries. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 312–331, Paris, France, May 26–28, 2010. Springer, Berlin, Germany.
- [89] Jaap-Henk Hoepman. Private handshakes. In Frank Stajano, Catherine Meadows, Srdjan Capkun, and Tyler Moore, editors, *ESAS*, volume 4572 of *Lecture Notes in Computer Science*, pages 31–42. Springer, 2007.
- [90] H. Huang and Z. Cao. A novel and efficient unlinkable secret handshakes scheme. *Communications Letters, IEEE*, 13(5):363–365, 2009.
- [91] Yan Huang, Peter Chapman, and David Evans. Privacy-preserving applications on smartphones. In *6th USENIX Workshop on Hot Topics in Security*, 2011.

-
- [92] Bernardo A. Huberman, Matthew K. Franklin, and Tad Hogg. Enhancing privacy and trust in electronic communities. In *ACM Conference on Electronic Commerce*, pages 78–86, 1999.
- [93] Stanislaw Jarecki, Jihye Kim, and Gene Tsudik. Authentication for paranoids: Multi-party secret handshakes. In Jianying Zhou, Moti Yung, and Feng Bao, editors, *ACNS 06: 4th International Conference on Applied Cryptography and Network Security*, volume 3989 of *Lecture Notes in Computer Science*, pages 325–339, Singapore, June 6–9, 2006. Springer, Berlin, Germany.
- [94] Stanislaw Jarecki, Jihye Kim, and Gene Tsudik. Group secret handshakes or affiliation-hiding authenticated group key agreement. In Masayuki Abe, editor, *Topics in Cryptology – CT-RSA 2007*, volume 4377 of *Lecture Notes in Computer Science*, pages 287–308, San Francisco, CA, USA, February 5–9, 2007. Springer, Berlin, Germany.
- [95] Stanislaw Jarecki, Jihye Kim, and Gene Tsudik. Beyond secret handshakes: Affiliation-hiding authenticated key exchange. In Tal Malkin, editor, *Topics in Cryptology – CT-RSA 2008*, volume 4964 of *Lecture Notes in Computer Science*, pages 352–369, San Francisco, CA, USA, April 7–11, 2008. Springer, Berlin, Germany.
- [96] Stanislaw Jarecki and Xiaomin Liu. Unlinkable secret handshakes and key-private group key management schemes. In Jonathan Katz and Moti Yung, editors, *ACNS 07: 5th International Conference on Applied Cryptography and Network Security*, volume 4521 of *Lecture Notes in Computer Science*, pages 270–287, Zhuhai, China, June 5–8, 2007. Springer, Berlin, Germany.
- [97] Stanislaw Jarecki and Xiaomin Liu. Affiliation-hiding envelope and authentication schemes with efficient support for multiple credentials. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfssdóttir, and Igor Walukiewicz, editors, *ICALP 2008: 35th International Colloquium on Automata, Languages and Programming, Part II*, volume 5126 of *Lecture Notes in Computer Science*, pages 715–726, Reykjavik, Iceland, July 7–11, 2008. Springer, Berlin, Germany.
- [98] Stanislaw Jarecki and Xiaomin Liu. Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 577–594. Springer, Berlin, Germany, March 15–17, 2009.

-
- [99] Stanislaw Jarecki and Xiaomin Liu. Private mutual authentication and conditional oblivious transfer. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 90–107, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Berlin, Germany.
- [100] Stanislaw Jarecki and Xiaomin Liu. Fast secure computation of set intersection. In Juan A. Garay and Roberto De Prisco, editors, *SCN 10: 7th International Conference on Security in Communication Networks*, volume 6280 of *Lecture Notes in Computer Science*, pages 418–435, Amalfi, Italy, September 13–15, 2010. Springer, Berlin, Germany.
- [101] Don Johnson, Alfred Menezes, and Scott A. Vanstone. The Elliptic Curve Digital Signature Algorithm (ECDSA). *Int. J. Inf. Sec.*, 1(1):36–63, 2001.
- [102] Jakob Jonsson and Burt Kaliski. Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1 (RFC 3447), 2003.
- [103] Yutaka Kawai, Shotaro Tanno, Takahiro Kondo, Kazuki Yoneyama, Kazuo Ohta, and Noboru Kunihiro. Extension of secret handshake protocols with multiple groups in monotone condition. *IEICE Transactions*, 93-A(6):1122–1131, 2010.
- [104] Yutaka Kawai, Kazuki Yoneyama, and Kazuo Ohta. Secret handshake: Strong anonymity definition and construction. In Feng Bao, Hui Li, and Guilin Wang, editors, *ISPEC*, volume 5451 of *Lecture Notes in Computer Science*, pages 219–229. Springer, 2009.
- [105] Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Traceable signatures. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 571–589, Interlaken, Switzerland, May 2–6, 2004. Springer, Berlin, Germany.
- [106] Joe Kilian and Erez Petrank. Identity escrow. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO’98*, volume 1462 of *Lecture Notes in Computer Science*, pages 169–185, Santa Barbara, CA, USA, August 23–27, 1998. Springer, Berlin, Germany.
- [107] Lea Kissner and Dawn Xiaodong Song. Privacy-preserving set operations. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 241–257, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Berlin, Germany.

- [108] Werner Koch. GNU Privacy Guard — The gcrypt Library. <http://www.gnupg.org/>.
- [109] Aleksandra Korolova, Rajeev Motwani, Shubha U. Nabar, and Ying Xu. Link privacy in social networks. In *ICDE*, pages 1355–1357. IEEE, 2008.
- [110] Aleksandra Korolova, Rajeev Motwani, Shubha U. Nabar, and Ying Xu. Link privacy in social networks. In James G. Shanahan, Sihem Amer-Yahia, Ioana Manolescu, Yi Zhang, David A. Evans, Aleksander Kolcz, Key-Sun Choi, and Abdur Chowdhury, editors, *CIKM*, pages 289–298. ACM, 2008.
- [111] Hugo Krawczyk. SIGMA: The “SIGn-and-MAC” approach to authenticated Diffie-Hellman and its use in the IKE protocols. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 400–425, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Berlin, Germany.
- [112] Hugo Krawczyk. HMQV: A high-performance secure diffie-hellman protocol. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 546–566, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Berlin, Germany.
- [113] Brian A. LaMacchia, Kristin Lauter, and Anton Mityagin. Stronger security of authenticated key exchange. In Willy Susilo, Joseph K. Liu, and Yi Mu, editors, *ProvSec 2007: 1st International Conference on Provable Security*, volume 4784 of *Lecture Notes in Computer Science*, pages 1–16, Wollongong, Australia, November 1–2, 2007. Springer, Berlin, Germany.
- [114] Laurie Law, Alfred Menezes, Minghua Qu, Jerome A. Solinas, and Scott A. Vanstone. An efficient protocol for authenticated key agreement. *Des. Codes Cryptography*, 28(2):119–134, 2003.
- [115] Daniel J. Lehmann. On primality tests. *SIAM J. Comput.*, 11(2):374–375, 1982.
- [116] Jin Li, Man Ho Au, Willy Susilo, Dongqing Xie, and Kui Ren. Attribute-based signature and its applications. In Dengguo Feng, David A. Basin, and Peng Liu, editors, *ASIACCS 10: 5th Conference on Computer and Communications Security*, pages 60–69, Beijing, China, April 13–16, 2010. ACM Press.
- [117] Chae Hoon Lim and Pil Joong Lee. A key recovery attack on discrete log-based schemes using a prime order subgroup. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO’97*, volume 1294 of *Lecture Notes in Computer Science*, pages 249–263, Santa Barbara, CA, USA, August 17–21, 1997. Springer, Berlin, Germany.

- [118] LinkedIn. Press center — about us. <http://press.linkedin.com/about>, 2011.
- [119] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In Howard M. Heys and Carlisle M. Adams, editors, *SAC 1999: 6th Annual International Workshop on Selected Areas in Cryptography*, volume 1758 of *Lecture Notes in Computer Science*, pages 184–199, Kingston, Ontario, Canada, August 9–10, 2000. Springer, Berlin, Germany.
- [120] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-based signatures. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 376–392, San Francisco, CA, USA, February 14–18, 2011. Springer, Berlin, Germany.
- [121] Mark Manulis, Benny Pinkas, and Bertram Poettering. Privacy-preserving group discovery with linear complexity. In Jianying Zhou and Moti Yung, editors, *ACNS 10: 8th International Conference on Applied Cryptography and Network Security*, volume 6123 of *Lecture Notes in Computer Science*, pages 420–437, Beijing, China, June 22–25, 2010. Springer, Berlin, Germany.
- [122] Mark Manulis and Bertram Poettering. Affiliation-hiding authentication with minimal bandwidth consumption. In Claudio Agostino Ardagna and Jianying Zhou, editors, *WISTP*, volume 6633 of *Lecture Notes in Computer Science*, pages 85–99. Springer, 2011.
- [123] Mark Manulis and Bertram Poettering. Practical affiliation-hiding authentication from improved polynomial interpolation. In *ASIACCS*, pages 286–295, 2011.
- [124] Mark Manulis, Bertram Poettering, and Gene Tsudik. Affiliation-hiding key exchange with untrusted group authorities. In Jianying Zhou and Moti Yung, editors, *ACNS 10: 8th International Conference on Applied Cryptography and Network Security*, volume 6123 of *Lecture Notes in Computer Science*, pages 402–419, Beijing, China, June 22–25, 2010. Springer, Berlin, Germany.
- [125] Mark Manulis, Bertram Poettering, and Gene Tsudik. Taming big brother ambitions: More privacy for secret handshakes. In Mikhail J. Atallah and Nicholas J. Hopper, editors, *Privacy Enhancing Technologies*, volume 6205 of *Lecture Notes in Computer Science*, pages 149–165. Springer, 2010.
- [126] Kevin S. McCurley. A key distribution system equivalent to factoring. *Journal of Cryptology*, 1(2):95–105, 1988.

-
- [127] Alfred Menezes, Paul van Oorschot, and Scott Vanstone. *Handbook of Applied Cryptography*. CRC Press, 2001.
- [128] Peter L. Montgomery. Speeding the Pollard and Elliptic Curve Methods of factorization. *Mathematics of Computation*, 48(177):243–264, 1987.
- [129] Michael Naehrig, Ruben Niederhagen, and Peter Schwabe. New software speed records for cryptographic pairings. In Michel Abdalla and Paulo S. L. M. Barreto, editors, *Progress in Cryptology - LATINCRYPT 2010: 1st International Conference on Cryptology and Information Security in Latin America*, volume 6212 of *Lecture Notes in Computer Science*, pages 109–123, Puebla, Mexico, August 8–11, 2010. Springer, Berlin, Germany.
- [130] Moni Naor. Deniable ring authentication. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 481–498, Santa Barbara, CA, USA, August 18–22, 2002. Springer, Berlin, Germany.
- [131] Samad Nasserian and Gene Tsudik. Revisiting oblivious signature-based envelopes. In Giovanni Di Crescenzo and Avi Rubin, editors, *FC 2006: 10th International Conference on Financial Cryptography and Data Security*, volume 4107 of *Lecture Notes in Computer Science*, pages 221–235, Anguilla, British West Indies, February 27 – March 2, 2006. Springer, Berlin, Germany.
- [132] Eiji Okamoto. Key distribution systems based on identification information. In Carl Pomerance, editor, *Advances in Cryptology – CRYPTO’87*, volume 293 of *Lecture Notes in Computer Science*, pages 194–202, Santa Barbara, CA, USA, August 16–20, 1988. Springer, Berlin, Germany.
- [133] Eiji Okamoto and Kazue Tanaka. Key distribution system based on identification information. *IEEE Journal on Selected Areas in Communications*, 7(4):481 – 485, May 1989.
- [134] Rasmus Pagh and Flemming Friche Rodler. Cuckoo hashing. *J. Algorithms*, 51(2):122–144, 2004.
- [135] Kenneth G. Paterson and Sriramkrishnan Srinivasan. On the relations between non-interactive key distribution, identity-based encryption and trapdoor discrete log groups. *Des. Codes Cryptography*, 52(2):219–241, 2009.
- [136] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO’91*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140, Santa Barbara, CA, USA, August 11–15, 1992. Springer, Berlin, Germany.

-
- [137] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
- [138] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. *J. Graph Algorithms Appl.*, 10(2):191–218, 2006.
- [139] Martin Raab and Angelika Steger. “Balls into Bins” — a simple and tight analysis. In Michael Luby, José D. P. Rolim, and Maria J. Serna, editors, *RANDOM*, volume 1518 of *Lecture Notes in Computer Science*, pages 159–170. Springer, 1998.
- [140] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565, Gold Coast, Australia, December 9–13, 2001. Springer, Berlin, Germany.
- [141] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairings. *Symposium on Cryptography and Information Security (SCIS)*, 2000.
- [142] Michelle Schatzman. *Numerical Analysis: A Mathematical Introduction*. Clarendon Press, Oxford, 2002.
- [143] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO’89*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252, Santa Barbara, CA, USA, August 20–24, 1990. Springer, Berlin, Germany.
- [144] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.
- [145] Michael Scott. Computing the Tate pairing. In Alfred Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 293–304, San Francisco, CA, USA, February 14–18, 2005. Springer, Berlin, Germany.
- [146] Michael Scott. Implementing cryptographic pairings. In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *Pairing*, volume 4575 of *Lecture Notes in Computer Science*, pages 177–196. Springer, 2007.
- [147] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology – CRYPTO’84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53, Santa Barbara, CA, USA, August 19–23, 1985. Springer, Berlin, Germany.

-
- [148] Zahava Shmueli. Composite Diffie-Hellman public-key generating systems are hard to break. Technical Report No. 356, Computer Science Department, Technion-Israel Institute of Technology, 1985.
- [149] Alessandro Sorniotti and Refik Molva. A provably secure secret handshake with dynamic controlled matching. In Dimitris Gritzalis and Javier Lopez, editors, *SEC*, volume 297 of *IFIP*, pages 330–341. Springer, 2009.
- [150] Alessandro Sorniotti and Refik Molva. Secret handshakes with revocation support. In Donghoon Lee and Seokhie Hong, editors, *ICISC 09: 12th International Conference on Information Security and Cryptology*, volume 5984 of *Lecture Notes in Computer Science*, pages 274–299, Seoul, Korea, December 2–4, 2009. Springer, Berlin, Germany.
- [151] Alessandro Sorniotti and Refik Molva. Federated secret handshakes with support for revocation. In Miguel Soriano, Sihan Qing, and Javier López, editors, *ICICS 10: 12th International Conference on Information and Communication Security*, volume 6476 of *Lecture Notes in Computer Science*, pages 218–234, Barcelona, Spain, December 15–17, 2010. Springer, Berlin, Germany.
- [152] Renwang Su. On the security of a novel and efficient unlinkable secret handshakes scheme. *Communications Letters, IEEE*, 13(9):712–713, 2009.
- [153] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.
- [154] The Facebook, Inc. Facebook’s statistics. <http://www.facebook.com/press/info.php?statistics>, 2011.
- [155] Gene Tsudik and Shouhuai Xu. A flexible framework for secret handshakes. In George Danezis and Philippe Golle, editors, *Privacy Enhancing Technologies*, volume 4258 of *Lecture Notes in Computer Science*, pages 295–315. Springer, 2006.
- [156] Damien Vergnaud. RSA-based secret handshakes. In Øyvind Ytrehus, editor, *WCC*, volume 3969 of *Lecture Notes in Computer Science*, pages 252–274. Springer, 2005.
- [157] Marco von Arb, Matthias Bader, Michael Kuhn, and Roger Wattenhofer. Veneta: Serverless friend-of-friend detection in mobile social networking. In *WiMob*, pages 184–189. IEEE, 2008.
- [158] D. Wallner, E. Harder, and R. Agee. Key management for multicast: Issues and architectures. RFC 2627 (Informational), 1999.

-
- [159] Harald J. Wertz. On the numerical inversion of a recurrent problem: the Vandermonde matrix. In *IEEE Transactions on Automatic Control*, 1965.
- [160] Chung Kei Wong, Mohamed G. Gouda, and Simon S. Lam. Secure group communications using key graphs. In *SIGCOMM*, pages 68–79, 1998.
- [161] Shouhuai Xu and Moti Yung. k-Anonymous secret handshakes with reusable credentials. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick McDaniel, editors, *ACM CCS 04: 11th Conference on Computer and Communications Security*, pages 158–167, Washington D.C., USA, October 25–29, 2004. ACM Press.
- [162] Shouhuai Xu and Moti Yung. K-anonymous multi-party secret handshakes. In Sven Dietrich and Rachna Dhamija, editors, *FC 2007: 11th International Conference on Financial Cryptography and Data Security*, volume 4886 of *Lecture Notes in Computer Science*, pages 72–87, Scarborough, Trinidad and Tobago, February 12–16, 2007. Springer, Berlin, Germany.
- [163] Naoyuki Yamashita and Keisuke Tanaka. Secret handshake with multiple groups. In Jae-Kwang Lee, Okyeon Yi, and Moti Yung, editors, *WISA 06: 7th International Workshop on Information Security Applications*, volume 4298 of *Lecture Notes in Computer Science*, pages 339–348, Jeju Island, Korea, August 28–30, 2006. Springer, Berlin, Germany.
- [164] Guomin Yang, Duncan S. Wong, Xiaotie Deng, and Huaxiong Wang. Anonymous signature schemes. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *PKC 2006: 9th International Conference on Theory and Practice of Public Key Cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 347–363, New York, NY, USA, April 24–26, 2006. Springer, Berlin, Germany.
- [165] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167. IEEE Computer Society, 1986.
- [166] Philip S. Yu, Jiawei Han, and Christos Faloutsos. *Link Mining: Models, Algorithms, and Applications*. Springer, 2010.
- [167] Elena Zheleva, Lise Getoor, Jennifer Golbeck, and Ugur Kuter. Using friendship ties and family circles for link prediction. In C. Lee Giles, Marc Smith, John Yen, and Haizheng Zhang, editors, *SNAKDD*, volume 5498 of *Lecture Notes in Computer Science*, pages 97–113. Springer, 2008.
- [168] Lan Zhou, Willy Susilo, and Yi Mu. Three-round secret handshakes based on ElGamal and DSA. In Kefei Chen, Robert H. Deng, Xuejia Lai, and Jianying

Zhou, editors, *ISPEC*, volume 3903 of *Lecture Notes in Computer Science*, pages 332–342. Springer, 2006.

On the security of RSA-based AHA

We show how the security reduction given by Gennaro, Krawczyk, and Rabin in [81, 82] for Okamoto's key establishment protocol (cf. Section 2.5.1) translates to the setting where element $g \in \mathbb{Z}_n$ is not a generator of $QR(n)$ but such that $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$. In the following, we refer to the individual paragraphs of [82, Section 3.2]. We list only the changes that need to be applied to the proof.

Identities and keys. [...] For Bob, *SIM* sets $H(id_B) = B = R$, where R is the input to *SIM* (note that R is random in \mathbb{Z}_N^\times).

Choosing a QR_N generator. *SIM* [...] chooses random $\bar{r} \leftarrow_R \mathbb{Z}_N^\times$, sets $r = \bar{r}^e$, and $g = (rB)^e$. Note that [...] g and B are random in \mathbb{Z}_N^\times and independent. Note moreover, that $\mathbb{Z}_N^\times = \langle -1 \rangle_N \times \langle g \rangle_N$ with probability $1/2$.

Session Interactions (non-test sessions). [...] Whenever Bob is activated in a session, *SIM* will set the value $\beta = (-1)^k g^b / \bar{r}$, where $k \leftarrow_R \{0, 1\}$, as the outgoing message from Bob [...]

Response to party corruption and session key queries (non-test sessions). In the third paragraph: Note that $\beta = (-1)^k g^b / \bar{r}$ for a known value $k \in \{0, 1\}$. Replace β by $(-1)^k \beta$, and keep all remaining computations.

Simulating the test session. [...] *SIM* sets this message to the value $\alpha = (-1)^{k'} (rB)^f S_A$, where $k' \leftarrow_R \{0, 1\}$ and r, B are as described at the beginning of the simulation. [...] In the second paragraph: Note that $\alpha = (-1)^{k'} (rB)^f S_A$ for a known value $k' \in \{0, 1\}$. Replace α by $(-1)^{k'} \alpha$, and keep all remaining computations.

Computing the forgery R^d . [...] since *SIM* chose $B = R$ then $(R^{2f})^d = \beta^{2f} / \bar{K}$. Using Lemma 1 and the fact that $2f$ is relatively prime to e we derive R^d from $(R^{2f})^d$.



Publication record

Author's publication record is provided below. The upper list enumerates treatments related to the topic of privacy-oriented authentication. All corresponding papers have been published on recognized international conferences and their contents are reflected in this thesis.

- Private Discovery of Common Social Contacts [62]
E. de Cristofaro, M. Manulis, B. Poettering
Applied Cryptography and Network Security (ACNS), 2011
- Affiliation-Hiding Authentication with Minimal Bandwidth Consumption [122]
M. Manulis, B. Poettering
Workshop on Information Security Theory and Practice (WISTP), 2011
- Practical Affiliation-Hiding Authentication from Improved Polynomial Interpolation [123]
M. Manulis, B. Poettering
ACM Symposium on Information, Computer and Communications Security (ASIACCS), Hong Kong, 2011
- Taming Big Brother Ambitions: More Privacy for Secret Handshakes [125]
M. Manulis, B. Poettering, G. Tsudik
Privacy Enhancing Technologies (PETS), Berlin, 2010
- Privacy-Preserving Group Discovery with Linear Complexity [121]
M. Manulis, B. Pinkas, B. Poettering
Applied Cryptography and Network Security (ACNS), Beijing, 2010
- Affiliation-Hiding Key Exchange with Untrusted Group Authorities [124]
M. Manulis, B. Poettering, G. Tsudik
Applied Cryptography and Network Security (ACNS), Beijing, 2010

The results from [62] will be submitted to the International Journal of Information Security (IJIS), following a corresponding invitation by the journal's editors.

In addition to the listed publications, the author contributed to the following projects. Observe that the works on redactable signatures and pseudorandom signatures are also related to the cryptographic treatment of privacy-preserving techniques.

- Redactable Signatures for Tree-Structured Data: Definitions and Constructions [32]
C. Brzuska, H. Busch, Ö. Dagdelen, M. Fischlin, M. Franz, S. Katzenbeisser, M. Manulis, C. Onete, A. Peter, B. Poettering, D. Schröder
Applied Cryptography and Network Security (ACNS), Beijing, 2010
- Pseudorandom Signatures (unpublished)
N. Fleischhacker, F. Günther, F. Kiefer, M. Manulis, B. Poettering
- Plaintext Awareness in Identity-Based Key Encapsulation (unpublished)
M. Manulis, B. Poettering, D. Stebila