This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3036224, IEEE Transactions on Dependable and Secure Computing



# *RESCUE*: A Resilient and Secure Device-to-Device Communication Framework for Emergencies

Milan Stute, Florian Kohnhäuser, Lars Baumgärtner, Lars Almon, Matthias Hollick, Stefan Katzenbeisser, and Bernd Freisleben

**Abstract**—During disasters, existing telecommunication infrastructures are often congested or even destroyed. In these situations, mobile devices can form a backup communication network for civilians and emergency services using disruption-tolerant networking (DTN) principles. Unfortunately, such distributed and resource-constrained networks are particularly susceptible to a wide range of attacks such as terrorists trying to cause more harm. In this paper, we present *RESCUE*, a resilient and secure device-to-device communication framework for emergency scenarios that provides comprehensive protection against common attacks. *RESCUE* features a minimalistic DTN protocol that, by design, is secure against notable attacks such as routing manipulations, dropping, message manipulations, blackholing, or impersonation. To further protect against message flooding and Sybil attacks, we present a twofold mitigation technique. First, a mobile and distributed certificate infrastructure particularly tailored to the emergency use case hinders the adversarial use of multiple identities. Second, a message buffer management scheme significantly increases resilience against flooding attacks, even if they originate from multiple identities, without introducing additional overhead. Finally, we demonstrate the effectiveness of *RESCUE* via large-scale simulations in a synthetic as well as a realistic natural disaster scenario. Our simulation results show that *RESCUE* achieves very good message delivery rates, even under flooding and Sybil attacks.

## 1 Introduction

During floods, hurricanes, earthquakes, nuclear accidents, or terrorist attacks, fast disaster response can save human life, limit environmental damage, and reduce economic loss. Communication technologies are integral to disaster relief operations. However, panic reactions and physical damage often lead to inoperable local communication infrastructures, e.g., during the 2013 Typhoon Haiyan [1]. To provide an alternative to expensive satellite-based communication, many researchers have proposed to leverage the ad hoc and disruption-tolerant networking (DTN) capabilities of ubiquitous mobile devices, where all devices relay messages for others in the network [2], [3]. Nevertheless, due to its cooperative, distributed, and resource-constrained nature, DTN communication is vulnerable to many attacks [4]. Adversaries, e.g., terrorists, may exploit such vulnerabilities to subvert the communication system and disrupt disaster relief operations. Furthermore, people in panic may spam the network with messages and thereby unintentionally jeopardize availability. Existing DTN security solutions attempt to identify and then blacklist attackers [5]. However, such approaches can exhibit false positives, causing a valid user to be mistakenly identified as an adversary and be excluded from communication. Furthermore, Sybil attacks, where an adversary operates under multiple identities, have not been addressed in previous DTN research.

In this paper, we present *RESCUE*, a resilient and secure device-to-device communication framework for emergency scenarios which is the first work to provide *comprehensive* attack protection. *RESCUE*'s basic communication protocol relies on epidemic routing, authenticated and immutable messages, and an effective acknowledgment processing. This way, common attacks, such as message or routing manipulation, blackholing, or impersonation, are already prevented. Yet, as in today's Internet infrastructure [6], the key challenge is to defend against Denial of Service (DoS) attacks originating from *individuals* as well as *multiple identities* (Sybil attack) that flood the network.

For this purpose, *RESCUE* pursues a twofold mitigation technique. First, certificates are used to cryptographically bind users to network identifiers, which hinders the adversary from assuming multiple identities. Since traditional static certificate infrastructures may be unavailable in the disaster area, we propose a novel distributed approach that enables new users to obtain certificates from mobile authorities during crisis. Second, *RESCUE* applies a novel buffer management scheme called *source-elastic buckets (SEB)*


- *M. Stute, L. Almon, and M. Hollick are with Secure Mobile Networking Lab, Technische Universität Darmstadt, Germany. E-mails: {mstute,lalmon,mhollick}@seemoo.tu-darmstadt.de*
- *F. Kohnhäuser is with Security Engineering Group, Technische Universität Darmstadt, Germany. E-mails: kohnhaeuser@seceng.informatik.tu-darmstadt.de*
- *L. Baumgärtner is with Software Technology Group, Technische Universität Darmstadt, Germany. E-mail: baumgaertner@cs.tu-darmstadt.de*
- *S. Katzenbeisser is with Faculty of Computer Science and Mathematics, University of Passau, Germany. E-mail: Stefan.Katzenbeisser@uni-passau.de*
- *B. Freisleben is with Department of Mathematics & Computer Science, Philipps-Universität Marburg, Germany. E-mail: freisleb@informatik.uni-marburg.de*

that substantially increases message delivery rates, i. e., availability, in the presence of flooding attacks, both from individuals and multiple identities. At its core, SEB isolates authenticated messages and allocates buffer capacity fairly to all source nodes, effectively mitigating flooding attacks from individual nodes. We presented SEB's initial design in [7]. In this work, we extend SEB with priority sets to provide protection against Sybil attackers as well. As SEB relies on node-local decisions rather than a (complex) distributed protocol, it provides a *minimal surface to attacks* and causes *no network overhead* by design. In addition, instead of identifying and excluding misbehaving users from the network, our scheme provides a fair allocation of available resources to all users. Hence, *RESCUE* does not suffer from false positives, where a valid user is mistakenly excluded from the emergency communication. We make the following contributions:

- A mobile distributed certificate infrastructure tailored to disaster scenarios that hinders an adversary from assuming multiple identities to perform Sybil attacks (Section 5).
- A fair buffer management scheme that mitigates the effect of flooding attacks by individuals (Section 6).
- An extension to our buffer management scheme by priority sets that increases resilience against Sybil attacks, guarantees at least the performance of direct message delivery, and supports unregistered users without a certificate (Section 7).
- An evaluation of *RESCUE* using large-scale network simulations in both, synthetic and realistic disaster scenarios, demonstrating that *RESCUE* maintains very good delivery rates even under attack (Section 8).

Furthermore, Section 2 reviews related work, Section 3 depicts our system model, Section 4 describes our minimalistic communication protocol, and Section 9 concludes the paper and outlines areas for future work.

## 2 Related Work

Since our communication framework can withstand a variety of attacks (see Table 1), it supports and complements several existing secure opportunistic communication systems [8], [9], [10]. In particular, it complements a recently proposed framework for anonymous routing in DTNs [11]. We now review related work on network attacks and possible countermeasures.

**Flooding Attacks in DTNs.** Denial-of-Service (DoS) attacks on unauthenticated DTNs have been discussed in the literature, but contrary to previous findings [12], we show that authentication is essential for reliable operation (Section 8). In authenticated networks, [5] proposes to enforce rate limits hard-coded in certificates, using an active distributed protocol. Nodes exceeding their rate limit are blacklisted and excluded from the network. [13] proposes a similar scheme which also allows for bursty traffic. *In contrast to all previous works, we implicitly solve the problem of flooding attacks using a fair and elastic buffer management scheme which has the benefits of not requiring any pre-defined (and possibly arbitrary) limits and avoiding additional overhead in form of encounter records.*

**Sybil Attacks in Peer-to-Peer and Mobile Networks.** Previous works try to *identify* Sybil identities and then take appropriate actions to exclude them from the network. One approach exploits the users' social networks [14], [15]. However, this requires communication between peers, which is feasible for online peer-to-peer systems but not for DTN scenarios. In [16], Sybil identities are detected at direct neighbor nodes. This approach is suitable for proximity services, but not for DTNs where Sybil nodes might be multiple hops away. In [17], nodes bootstrap trust relationships randomly and then collaboratively filter bogus messages. *In contrast to existing work, we do not try to* identify *attackers but include their presence in our protocol design.*

**Secure Routing.** Encounter-based routing in DTNs is used to intelligently select forwarding nodes based on their contact history, which works well assuming repetitive mobility patterns. At the same time, it makes the network susceptible to blackhole attacks, where an attacker lies about past encounters to appear as a strong forwarder. Previous works have proposed to use signed encounter tickets that are exchanged upon contact [18], [19], [20], [21]. Unfortunately, exchanging and verifying these tickets introduces communication and computational overhead. *We use epidemic routing to thwart all routing attacks and mitigate the problems of increased message replicas using effective buffer management and prioritization.*

## 3 System Model

**Communication Model.** We support a wide range of communication models that are reasonable during emergencies, including *one-to-one* (contact with friends or family), *many-to-many* (within task forces or departments), or *one-to-many* (emergency notification broadcasts). Due to the inherent delay of DTN-based communication and our focus on emergency communication, we consider mainly small messages, such as text and distress messages (including additional information, such as GPS location of the sender), serving a similar purpose as the classic *112* or *911* emergency call. Compared to rich media (images, voice, video), information in text messages is more compact, thus, more suitable for DTN communication.

**Adversary Model.** We consider an adversary Adv who can mount network attacks and compromise network entities. Specifically, Adv can eavesdrop, manipulate, forge, or drop messages. Furthermore, Adv can assume a limited number of entities, either by compromising or stealing devices or by registering multiple times in our system. Unlike the classic Dolev–Yao adversary model, Adv controls only a part of the communication channel and a fraction of all network entities. Moreover, Adv cannot break cryptographic primitives or tamper with the *root authority* (see Section 5.1). In Table 1, we summarize well-known attacks [4] that Adv can mount, and list *RESCUE*'s countermeasures to prevent them.

## 4 Minimalistic Communication Protocol

This section describes *RESCUE*'s communication protocol, i.e., its routing protocol, message format, and acknowledgment processing. By employing a simple routing mechanism and a minimalistic frame format, *RESCUE* is immune to a large set of common attacks on DTN protocols (see Table 1).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3036224, IEEE Transactions on Dependable and Secure Computing



Table 1: Attack Resilience of *RESCUE*

| ATTACK | COUNTERMEASURE | SECTION |
|---|---|---|
| Routing manipulation<br>Message dropping<br>Blackholing | Epidemic routing | 4.1 |
| Message modification<br>Impersonation | Authentic immutable messages | 4.2 |
| ACK flooding | ACKs only for messages in buffer | 4.3 |
| Sybil attack | User registration | 5.3 |
| | Priority sets | 7.3 |
| Message flooding | Source-based elastic buckets | 6.2 |
| TTL spoofing | Source-based elastic buckets | 6.2 |

### 4.1 Epidemic Routing

Instead of relying on infrastructure, DTN-enabled devices exchange messages directly using Wi-Fi or Bluetooth. DTNs exploit user mobility to increase coverage. To this end, devices act as "data mules" that store their messages as well as messages from other users, carry them, and finally forward them to the destination upon contact. When a device is in communication range of another device, both devices replicate and transmit all messages not yet received by the other device. Therefore, every message is flooded to every node that comes into contact, spreading like an epidemic. We use *epidemic* routing [22] because there are no routing control messages, thus, mitigating all types of routing manipulation attacks. In addition, message dropping attacks have no effect, since messages are replicated to all available neighbors. Carried messages are stored in the node's *buffer* that we protect against flooding and Sybil attacks, as detailed in Sections 6 and 7, respectively. When two devices discover each other via Bluetooth or Wi-Fi beacon frames, they initiate a handshake. As part of the handshake, both devices first exchange metadata about carried messages and then start transferring messages that the other device is missing. However, due to limited buffer capacity and short contact times (e. g., two cars passing each other), not all messages might be exchanged. A message prioritization scheme (Section 6.3) determines which messages are exchanged first upon contact.

### 4.2 Authentic Immutable Messages

Each user possesses a unique *signature key pair* generated during initialization. The public signature key serves as a unique addressable *network identifier* similar to an IP address. The private signature key is used to sign outgoing messages. A message MSG contains the source network identifier $s$, the signature $\sigma_s$, and, if available, the identity certificate $\mathcal{C}$. The identity certificate is issued by a certficate authority and contains the network identifier as well as an identification token $T$ that we explain in Section 5. Identity certificates can be cached and only transmitted on demand to reduce overhead. In addition, a message MSG contains the destination network identifier $d$ (the public signature key of the destination), the creation time stamp $t$, the message *lifetime* $\Delta t$, and an optionally encrypted *payload* $\mathcal{P}$, resulting in the tuple:

$$\text{MSG} = (s, d, t, \Delta t, \mathcal{P}, \sigma_s, \mathcal{C}). \tag{1}$$

Devices verify messages at each hop by checking the message signature and, if available, the source's identity certificate. Devices discard messages if a check fails, so that corrupted messages do not propagate in the network.

We further define the message ID $m$ as a hash $h$ over all message fields: $m = h(s, d, t, \Delta t, \mathcal{P})$. The signature $\sigma_s$ is then calculated on $m$. We note that all header fields are *immutable*, that is, they are not changed in transit, which would be required for time-to-live (TTL) fields. Immutable fields allow the signature to protect the entire message and, thus, they prevent all message modification and impersonation attacks. Assuming that the clocks of all valid nodes are roughly synchronized,[1] the TTL of MSG can be locally computed by each node with:

$$\text{TTL} = t + \Delta t - t_{now}, \tag{2}$$

where $t_{now}$ is the current time. Nodes regularly remove *expired* messages (negative TTL) from their buffers.

### 4.3 Authentic Acknowledgments

*RESCUE* uses acknowledgments (ACKs) for *one-to-one* communication. Previous work [22] has shown that epidemic routing greatly benefits from ACKs since they free up buffer capacity for other messages. Upon receiving a message, the destination creates an ACK as a reply and forwards it with the same mechanism used for relaying regular messages. The ACK contains only the message ID $m$ and a signature from the destination $\sigma_d$:

$$\text{ACK} = (m, \sigma_d). \tag{3}$$

Upon receiving and verifying an ACK, intermediate nodes can safely remove the acknowledged message payload from their buffers. The ACK is stored until the corresponding message has expired. Attackers cannot forge ACKs, since they are cryptographically signed and, hence, cannot purge valid undelivered messages from the network. Since ACKs are small, they present a potential attack vector: by creating a large number of bogus ACKs, an attacker can exhaust the computational resources of the receiving nodes, because they have to verify each signature, leading to a DoS. To solve this problem, nodes in *RESCUE* only accept and process ACKs for messages they currently carry. This stops the spreading of bogus ACKs at the first valid node.

### 4.4 Storage Overhead

For each message, a node stores meta information $m$, $s$, $d$, $t$, and $\Delta t$ as well as $\mathcal{P}$ and $\sigma_s$ in its buffer. After receiving and verifying an ACK, the node deletes $\mathcal{P}$ and $\sigma_s$ and replaces them with $\sigma_d$. In comparison with an insecure scheme, the storage overhead is the signature $\sigma_s$ or $\sigma_d$ per message. Additionally, a node stores the identity certificate $\mathcal{C}$ for each other node $s$ that it carries messages for. If it no longer carries messages for $s$, a node can decide to delete $\mathcal{C}$ or keep it to speed up future transfers for messages from $s$.

1. Mobile nodes can synchronize their clocks via GPS or a cellular network. Even if synchronization opportunities are no longer available after a disaster, we can neglect small-scale clock drift that might occur since TTL is in the order of hours.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3036224, IEEE Transactions on Dependable and Secure Computing



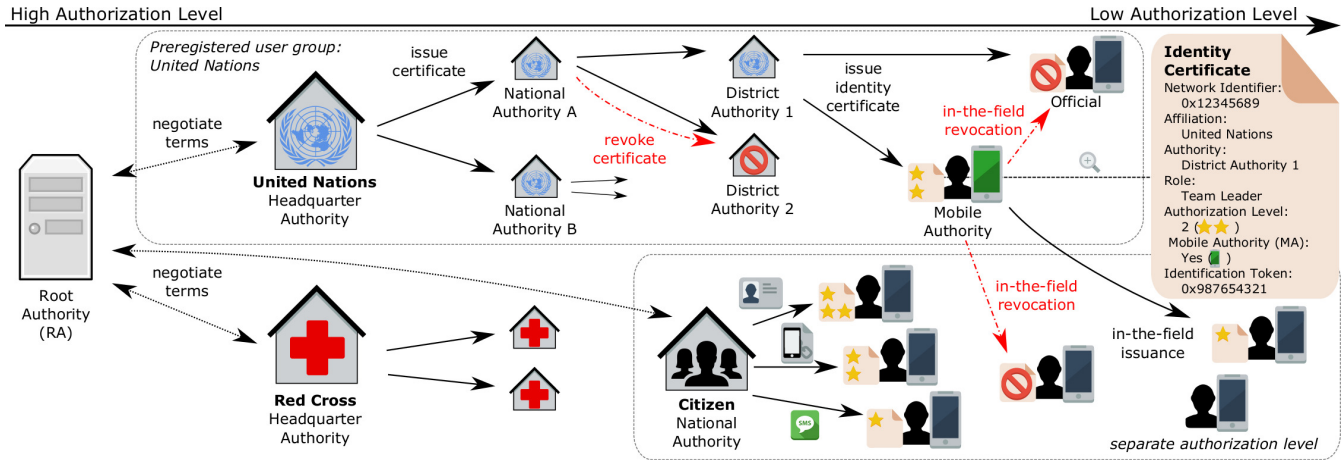



Figure 1: Illustration of our mobile distributed certificate infrastructure. The authorization level (number of stars) decreases from left to right. Registered users have a higher authorization level than unregistered users. Citizens are depicted in a box with a separate authorization level.

## 5 In-the-Field User Registration with Mobile Authorities

To establish trust relationships, we deploy so-called *identity certificates* that bind important properties in the emergency context (e. g., user role or affiliation) to the network identifier of users. In this section, we first describe our backbone certificate infrastructure. Next, we extend the backbone infrastructure by *mobile authorities*, enabling their operation during disasters in the field, where the backbone infrastructure is unavailable. Finally, we propose multiple user identity verification methods that hamper fake registrations with the certificate infrastructure. This way, an adversary is prevented from obtaining multiple identities, i.e., certified network identifiers, that could be used to perform distributed DoS attacks. Fig. 1 illustrates our certificate infrastructure.

### 5.1 Static Authorities

The backbone certificate infrastructure consists of multiple hierarchically organized static Certificate Authorities (CAs). The root of these CAs constitutes a dedicated authority named *Root Authority (RA)* that serves as a trust anchor and its certificate is pre-deployed on all *RESCUE*-enabled devices. Before the actual crisis, the RA establishes relationships with organizations or governments that want to participate as *authorities* in the certificate infrastructure. All authorities initially undergo a rigorous audit, since their authenticity and trustworthiness are crucial to the overall security, and negotiate *user roles* as well as preconfigured *user groups* that the authority introduces to the network. For instance, in Fig. 1, the United Nations (UN) added the user roles *team leader* and *official*, and arranged a preconfigured user group *United Nations*, so users can specifically address all UN members when sending a message. Organizations manage their own certificate infrastructure and, therefore, maintain one or multiple, potentially hierarchically organized, CAs. On the lowest hierarchical CA level, CAs issue identity certificates to staff members. Furthermore, the overall infrastructure contains at least one authority that issues identity certificates to regular users, i.e., citizens.

Identity certificates bind the public signing keys of users, which function as their unique network identifiers (see Section 4), to user properties. Important properties in the emergency setting are the *affiliation* (e. g., UN, red cross), *user role* (e. g., citizen, physician) and *authorization level*, which indicate a user's permission level and trustworthiness. Figure 1 exemplifies the identity certificate of a UN team leader, and depicts the authorization level of entities by their position on the x-axis as well as stars in the certificate.

We further consider certificate revocation to defend against an adversary who obtains identity certificates by compromising user devices or infiltrating authorities. Other reasons for certificate revocation are, for instance, structural changes within organizations or users who quit organizations. *RESCUE* implements certificate revocation via certificate revocation lists (CRLs) that are broadcasted in the network. We distinguish between two different entities: authorities and users. An authority $\mathcal{A}$ can revoke an entity $\mathcal{E}$ if $\mathcal{A}$ has a higher authorization level than $\mathcal{E}$, and there is a certificate chain (i.e., a chain of trust) between $\mathcal{A}$ and $\mathcal{E}$. Upon the revocation of an authority, all certificates issued by the authority are regarded as invalid, depriving its power. In case a user identity certificate is revoked, the certificate is considered invalid and the respective user loses his or her role, authorization level, and any message transmission privileges (see Section 7).

### 5.2 Mobile Authorities

In a disaster area, infrastructure-based communication is mostly unavailable and, thus, users rarely have a connection to the static authorities in the backbone infrastructure. This is not an issue when established user properties are retrieved, since identity certificates can be verified and transmitted between users on demand (see Section 4). Nevertheless, operations that inevitably involve CAs, such as issuing new certificates or revoking existing certificates, cannot be performed when static authorities are unavailable.

Therefore, we propose that special privileged users employ their mobile devices to serve as *mobile authorities (MAs)* during a crisis. Since it is easier for an adversary to compromise MAs than static authorities, MAs have restricted

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3036224, IEEE Transactions on Dependable and Secure Computing



capabilities. In detail, they can only issue identity certificates for citizens, but not for specific user groups like red cross staff members. We argue that this is not a restriction, since professional emergency workers typically set up their systems prior to the disaster or outside the disaster area, where a connection to static authorities is available. Additionally, MAs are allowed to revoke identity certificates. An MA $\mathcal{M}$ can revoke a user $\mathcal{U}$, if $\mathcal{U}$ is a citizen, or if $\mathcal{U}$ has a lower authorization level than $\mathcal{M}$ and both belong to the same affiliation (Fig. 1).

Since malicious MAs can seriously harm the network, only privileged and trustworthy users with devices that satisfy certain security requirements are permitted to become MAs. In the initialization phase, each organization negotiates the maximum permitted number of MAs they introduce to the network, and then carefully selects those users qualifying to become MAs. MA users have a high social trust level and protect their devices using security mechanisms like a trusted execution environment, full disk encryption, and strong passwords. These mechanisms have shown to significantly increase the effort for physical attacks [23], giving MA users enough time to report and revoke stolen MA devices.

### 5.3 Secure Identity Verification Methods

In the following, we present methods that enable static and mobile authorities to identify registering entities based on hard-to-forge *identification tokens*. The identification token and method are part of the identity certificate and effectively hinder individuals from registering repeatedly and obtaining multiple identity certificates. Our goal is to increase the cost for fake registrations, such that bypassing our subsequently presented flooding and Sybil mitigations (see Sections 6 and 7) becomes uneconomical for an adversary. We assume that organizations and governments are already able to supply each staff member with exactly one identity certificate, e. g., by handing out preconfigured devices. Therefore, we focus on fake registrations of users with the authorization level "citizen". Citizens that employ stronger authentication methods during registration with CAs are considered more trustworthy, indicated by a higher authorization level in their identity certificate. Messages from users with high authorization levels are transmitted preferentially (see Section 6) to encourage citizens to ($i$) obtain identity certificates, and ($ii$) use strong identification methods during registration. Typically, as summarized in Table 2, the stronger the identity proof is, the more restrictive (i. e., less applicable) and time-consuming the verification process gets, resulting in limited practical usability. In the following, we first define identification tokens and then discuss the identification methods in detail.

**Identification Tokens.** *RESCUE* uses identification tokens to uniquely identify network nodes and implement its flooding protection mechanism (see Section 6). An identification token $T$ is calculated based on an identifier specific to one of the identification methods. For example, the identifier for the SIM-based identification method would be the phone number. Other methods use different identifiers, which we discuss later. To enforce a method-agnostic format for $T$, we apply a globally known hash function in the following way:

$$T = hash\left(\text{“Identification Method”} \mid\mid \text{“Identifier”}\right), \quad (4)$$

where $||$ is the concatenation operand. For example, an identification token resulting for a SIM-based registration would be calculated as $T_{\text{SIM}} = hash(\text{“SIM”} \mid\mid \text{“+491234567890”})$. This approach effectively and elegantly thwarts adversaries that attempt to receive multiple identities by registering with different MAs. Since the identification token for the same device and method is constant, all issued identity certificates would contain the same identification token and could, thus, be mapped to the same identity.

**Identification via Physical Presence.** This method constitutes a fallback solution only used by MAs during a crisis if all other verification methods are inapplicable. For this method, the identification token is generated from the devices's network identifier. While the identifier is not hard to forge (an adversary could easily generate several public keys to assume multiple identities), the identification method requires a user to physically approach an authority during registration, and thus spend physical effort. By spending this effort, we naturally limit the number of identity certificates that an adversary could receive using different network identifiers. MAs assert physical proximity of users by employing short range communication channels (e. g., QR codes, NFC, or Bluetooth) to transmit identity certificates. Furthermore, MAs manually confirm the issuance of each identity certificate to prevent an adversary from obtaining multiple certificates at once. The method has a weak identification strength, since an attacker can simply approach different MAs or, with some delay, the same MA repeatedly. Also, usability is poor, since users cannot register remotely.

**Identification via SIM.** The subscriber identity module (SIM) card is used as an identification token by requiring the user to enter a nonce that is sent via a call or SMS to the user's device during registration. Upon successful registration, the authority creates the identification token $T$ using the user's phone number as the identifier. The approach provides excellent usability and applicability, since it requires the user to take a minimum effort and SIM cards are available in many mobile devices. Nevertheless, an adversary can create fake users by using anonymous prepaid SIM cards. Also, since the approach requires a functioning cellular network, it is unsuitable for registering new users during a crisis.

**Identification via Remote Attestation.** Authorities can perform a remote attestation [24] with devices of registering users. This way, authorities obtain an attestation report that is signed with a device-unique secret attestation key and a certificate that testifies the validity of the attestation key. The public attestation key serves as the identifier for the identification token. To date, applicability is good, as recent Samsung [25], Windows [26], and Android [24] devices provide remote attestation capabilities. Additionally, remote attestation will become increasingly widespread with upcoming technologies [27], [28] and MAs could act as verifiers and thus identify new users during a crisis. Usually, a backbone in the form of stable connections, e. g.,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3036224, IEEE Transactions on Dependable and Secure Computing



Table 2: Overview of identity verification methods. Ratings scale from poor (⋆) to excellent (⋆⋆⋆⋆⋆).

| Identification Method | Strength | Usability | Applicability | Available in Crisis |
|---|---|---|---|---|
| Physical Presence | ⋆ | ⋆⋆ | ⋆⋆⋆⋆⋆ | ✓ |
| SIM | ⋆⋆⋆ | ⋆⋆⋆⋆⋆ | ⋆⋆⋆⋆ | ✗ |
| Remote Attestation | ⋆⋆⋆⋆ | ⋆⋆⋆⋆⋆ | ⋆⋆⋆ | ✓ |
| eID | ⋆⋆⋆⋆⋆ | ⋆⋆⋆ | ⋆⋆ | ✓ |

cable or satellite uplink, between the different systems that are responsible for the remote attestation is needed. For increased resilience and to keep functioning even under severely challenging network conditions, it is desirable to not rely on such a backbone but explore new decentralized and federated solutions [29], [30], [31]. In practice, though, remote attestation without backbone access has not yet been implemented.

**Identification via eIDs.** This method uses national electronic ID cards, which often provide identification capabilities, to identify a user. As an example, the eIDAS regulation defines electronic identification services in the entire European Union [32]. eIDAS specifies the restricted identification (RI) protocol, e. g., implemented in the German identity card since 2010. RI allows a service provider (SP) terminal to recognize an eID chip based on a chip-unique pseudonym. The chip-unique pseudonym acts as the identifier for the identification token. Using mobile devices as local terminals [33], authorities can act as SP terminals and securely identify eID cards of registering users. The approach provides a strong proof of identity as it is hard to forge eID cards or to obtain multiple valid eID cards including their PIN. Since MAs can in principle act as SP terminals, the approach is applicable during a crisis. As a downside, users must initially activate their eID cards and have them at hand.

## 6 Local Buffer Management

Within the *buffer*, a node stores unacknowledged and unexpired messages. If there are many such messages, a node might not have the resources to store them all: at some point, it needs to decide which messages to keep and which to drop. Given a set of messages and a buffer with a node-defined *capacity* $C$, the *buffer management* has to decide which messages to store in the buffer without exceeding its capacity to enforce:

$$\sum_{\text{MSG}} |\text{MSG}| \leq C. \tag{5}$$

Besides this hard constraint, buffer management can have multiple optimization goals, e. g., throughput maximization, delay minimization, or delivery reliability. In this work, we are concerned with security, in particular resilience against DoS attacks. To this end, we first motivate the need for security mechanisms in the buffer management, define security requirements, and then present a novel secure buffer management strategy, *Source-based Elastic Buckets (SEB)*, which achieves protection against flooding attacks.

### 6.1 Security Requirements and Design

Poor buffer management schemes can expose a network to flooding attacks. For example, malicious nodes can exploit trivial schemes such as first-in first-out (FIFO) queues to replace valid messages with bogus ones [34]. The goals of our buffer management scheme are: ($i$) a single attacker can only occupy a "fair share" (we formally discuss this in sec:seb:basic) of available buffer space; ($ii$) maximization of buffer utilization to increase message redundancy and, thus, the delivery rate; and ($iii$) a MA compromise does not compromise the network (Sybil attacks are discussed in Section 7). To reach these goals, we apply a *locality* principle [35] to allow nodes to decide locally and independently which messages to store. Hence, nodes do not need to trust and verify third-party information, which keeps the attack surface small. Furthermore, bandwidth efficiency is increased, since control messages need not be exchanged.

### 6.2 Source-based Elastic Buckets (SEB)

We now present our novel buffer management strategy *Source-based Elastic Buckets (SEB)* that, by design, prevents valid messages from being purged from the network during flooding attacks. The basic idea is that all messages from a source $s$ are placed in an isolated bucket $B_s$. All buckets are stored as a map $\mathcal{B}$ that uses identification tokens as its key.

$$B_s = \mathcal{B}(T_s). \tag{6}$$

Since *RESCUE* uses identification tokens as authenticated identifiers and ensures message authenticity through digital signatures, an adversary cannot forge messages in a way that they occupy buckets of valid users. SEB is fair in the sense that each bucket $B_s$ has a *guaranteed capacity* of $C_n = C/n$, with $n$ being the number of currently allocated buckets (number of source nodes that a respective node currently carries messages from). The *occupancy* of a single source bucket $O(b)$ is a non-negative number and is subject to

$$\sum_{b \in \mathcal{B}} O(b) \leq \sum_{b \in \mathcal{B}} C_n = C. \tag{7}$$

We further define the surplus $S(b)$ as the (possibly negative) difference between the occupancy and the guaranteed capacity:

$$S(b \in \mathcal{B}) = C_n - O(b). \tag{8}$$

If $s$ does not exhaust its guaranteed capacity ($S(b) > 0$), because it has not sent "enough" messages, $S(b)$ is provided to other buckets requiring it, which means that their surplus can become negative. However, when $s$ sends a message at a later point, overdrawn buckets ($S(b) < 0$) are emptied first. These *elastic quotas* allow full exploitation of the local buffer capacity, while maintaining strict message separation of different sources. Algorithm 1 shows SEB's message insertion procedure. We define the argument of the minimum function as $\arg\min_{x \in S} f(x) = \{x \in S : f(x) = \min_{y \in S} f(y)\}$. Algorithm 1 first inserts a new message in the corresponding source bucket $B_s$ (l. 3). Until the total occupancy meets $C$ to satisfy Eq. (5) (l. 4), the algorithm drops messages (l. 7) from the bucket with the smallest surplus (l. 5) in order of the messages' ranks (l. 6). We expain the message

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3036224, IEEE Transactions on Dependable and Secure Computing



**Algorithm 1** Message Insertion using SEB

**Input:** MSG {*The message to be inserted*}
**Input:** $\mathcal{B}$ {*The set of all buckets*}
**Input:** $C$ {*Total capacity*}
1: $T_s :=$ source's identification token from MSG
2: $B_s := \mathcal{B}(T_s)$ {*Select source bucket*}
3: $B_s := B_s \cup$ MSG {*Add new MSG to the source bucket*}
4: **while** $\sum_{b \in \mathcal{B}} O(b) > C$ **do**
5: $B' := \arg\min_{b \in \mathcal{B}} S(b)$ {*Bucket with smallest surplus*}
6: MSG$' := \arg\min_{m \in B'} MR(m)$ {*message with the lowest message rank*}
7: $B' := B' \setminus$ MSG$'$ {*Drop MSG$'$ from bucket*}
8: **end while**

rank function in Section 6.3. Ties are broken at random if there are two or more elements with the smallest message rank. To assert that the buffer converges to a stable state, tie breaking needs to be consistent, i. e., the same tie needs to be broken consistently at a single node. We implement this by comparing the salted hashes [36] of node's identification tokens, while the salt is drawn at random once by each node. A tie is then broken by the smaller hash value.

SEB's robustness relies on the fact that messages are source-authenticated, and on the high costs of registering multiple identities in *RESCUE*. Without the latter costs, an attacker could assume multiple identities, flood the network with messages and, thus, hijack a disproportional amount of buffer capacity. In addition, SEB mitigates TTL spoofing attacks where an attacker sets excessively high values for $\Delta t$ to maximize the lifetime of its messages: by separating messages of different sources, an attacker would only be able to replace its own messages.

### 6.3 Multi-Factor Message Rank

Within each bucket, SEB uses *Message Rank (MR)* for prioritization. MR prioritizes: ($i$) acknowledgments, ($ii$) messages with the largest TTL, and ($iii$) messages with the smallest payload size. Carrying messages with a large TTL increases the probability that they will be delivered before expiration (we confirm this in Section 8), while small messages take less time for transmission, and help to prevent buffer fragmentation. Upon device contact, messages exchanged first have a higher chance of actually being transmitted to the next hop and eventually reaching their destination. A sending node transfers messages in its buffer to a receiving node $R$ in the following order: ($i$) messages destined for $R$, ($ii$) own messages, ($iii$) messages from other registered users, ($iv$) all other messages. Messages in each category are sorted by MR. MR only relies on fields in the message header. Since they are immutable, MR results in the same order independent of the order in which messages were received, which means that the buffers of two nodes will converge to a stable state if the contact duration is long enough. This is a problem that has been ignored by the research community and is reflected by the fact that the most popular network simulator for DTN research, ONE [37], only implements non-converging random and FIFO-based dropping strategies.

Table 3: All *priority sets* used in *RESCUE*.

| LEVEL | CONTAINED NODES | PURPOSE | SECTION |
|---|---|---|---|
| 0 | Local (own) | DD lower bound | 7.2 |
| 1 | Social net./by MA | Sybil protection | 7.3 |
| 2 | Other registered | Flooding protection | 6.2 |
| 3 | Unregistered | Best effort | 7.4 |

## 7 LOCAL PRIORITY SETS

Until now, we have assumed that MAs behave correctly and cannot be compromised by an adversary. We recognize that this is a strong assumption, since MA devices may be stolen or infected with malware. However, if we lift our assumption on secure MAs, our buffer management presented in Section 6 is vulnerable to Sybil attacks. This is because an adversary that gets hold of an MA can generate as many certificates as it wants. Since SEB allocates buffer resources fairly among all nodes, a Sybil attacker would receive an unfair amount of buffer space. In this section, we secure *RESCUE* even against such Sybil attackers by leveraging the concept of *secure message copies* using *priority sets* where nodes prioritize messages originating from a certain set of other nodes in the network.

In the following, we first introduce the concept of *secure copies*, explain the workings of *priority sets*, and discuss Sybil-secure fill strategies. Finally, we explain how priority sets also help to securely support unregistered users.

### 7.1 Secure Copies

In direct delivery (DD) forwarding, nodes do not carry messages for others, but only deliver their own messages when they actually encounter the destination. Obviously, this diminishes the advantage of having "data mules," but DD has a desirable security property: *it is inherently immune to flooding attacks even from Sybil attackers* since each node simply does not carry messages for other nodes. In other words, DD ensures that there is always one copy of every message in the network, namely in the buffer of the source node. We call this a *secure copy*, i. e., a message copy that an attacker cannot remove or replace. In the case of DD, the number of secure copies per message is exactly one. Though this is a very simple strategy, other buffer management schemes fail to achieve this guarantee. For example, when using FIFO, the node's own messages might be replaced by more recently received messages of other nodes. In the following, we increase the number of secure copies to improve delivery reliability.

### 7.2 Priority Sets Overview

We apply the idea of *secure copies* to SEB to mitigate Sybil attacks. In particular, we propose to prioritize certain buckets in SEB (e. g., the node's own bucket) such that they are emptied last when the buffer capacity is exceeded. We model the general assignment of node identities (and their buckets) to priorities via *priority sets* (PS). How we implement the assignment of node identities to PS, i. e., deciding which nodes' messages should be prioritized, is crucial for the system's security and is the core question that we address

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3036224, IEEE Transactions on Dependable and Secure Computing



in this section. Making individual local decisions makes it hard for an attacker to appear in all PS, thus preventing that their messages fill the buffers of all other nodes. By using PS, each node essentially gains a number of *secure relays* that prioritize messages for it, effectively increasing the number of *secure copies*. Next, we show how the PS concept integrates with SEB (Section 6), and discuss how PS can be used to protect against Sybil attacks in Section 7.3.

To integrate PS in SEB, we need to ensure that buckets of nodes in a PS are emptied last. We generalize this approach by allowing an arbitrary number of PS *levels* $l$. For the remainder of this work, we denote $\mathcal{S}_l$ as the priority set at level $l$. The set with the *lowest level* has the *highest priority*. Formally, $\mathcal{S}_l$ are pairwise disjoint subsets of $\mathcal{B}$:

$$\bigcup_{l=0,\dots} \mathcal{S}_l = \mathcal{B} \quad \text{and} \quad \mathcal{S}_l \cap \mathcal{S}_k = \emptyset, \quad l \neq k$$

To use priority sets with SEB, we adapt Algorithm 1 to start removing messages from the buckets with the lowest priority. In particular, we change Line 5 to first select the non-empty PS with the lowest priority, i. e., largest level $l$. Then, we select the bucket with the smallest surplus as follows:

5a: $l' := \max_l \{l : \mathcal{S}_l \neq \emptyset\}$
5b: $B' := \arg\min_{b \in \mathcal{S}_{l'}} S(b)$

To achieve the performance of DD (at least one secure copy per message), $\mathcal{S}_0$ only includes the local node, such that messages of the local node are removed last. We explain more PS levels in the following sections and we summarize all PS levels used in *RESCUE* in Table 3.

### 7.3 A Sybil-secure Priority Set

How to select the nodes that are to be put in $\mathcal{S}_1$ is key to achieving protection against Sybil attacks. The selection strategy needs to ensure that the nodes residing within each set are (preferably) different for each node and it is hard for the Sybil attacker to become a member of many of those sets. In the following, we discuss two PS fill strategies that are secure against Sybil attacks and can be practically used in emergency scenarios. The members of this set are selected either ($i$) by exploiting social relationships between the nodes, e. g., by leveraging phone numbers in the address book of the users' smart phones, or ($ii$) by letting the MA assign the set during registration. We now present both approaches in detail.

**Pre-registration: Social Networks.** Social networks have been considered as a solution for effectively detecting Sybil identities [14]. While the particular method [14] is unpractical in DTNs (it needs to perform online verifications), we borrow the idea of using social networks as a defense against Sybil attacks. We exploit the user's social network to prioritize messages from direct neighbors in the user's social graph, e. g., those nodes that are in the local node's address book. Since the attacker has no control over uncompromised devices, they cannot forcibly add themselves to others' address books, thus, they will not be able to appear in the PS of legitimate nodes. This option is only available for users that were able to register with an authority before the disaster, e. g., using their SIM card (Table 2), and were able to resolve the phone number to the *RESCUE* public key via a central server similar to secure messaging applications (e. g., Signal). For all users that registered during a disaster with an MA, a different method is required.

**Post-registration: MA-assigned.** We assume that most users will only register post-disaster and, thus, cannot use social contacts to fill their PS. However, we can leverage the trustworthiness of MAs by letting MAs suggest identities for the priority set during registration. We propose that the suggested identities are those that recently registered with the MA.[2] Since an attacker is not able to manipulate the PS of nodes that registered with the MA *before* being compromised, only nodes that registered *after* an MA compromise may be affected by PS manipulations.[3] The latter may experience a decreased service quality since their identities will not appear in the PS of other nodes. This effectively reduces the number of secure copies for their messages to one, which is the same for other unregistered users. Other than that, they are not affected negatively by registering with a compromised MA.

For both strategies, we need to determine the size of $\mathcal{S}_1$ (number of contacts in the users' address books and size of MA-suggested list, respectively) in order to be effective against Sybil attackers. In Section 8, we empirically show that a small PS size is sufficient to withstand Sybil attacks.

### 7.4 Supporting Unregistered Users

Apart from Sybil attacks, PS enable us to solve another remaining problem: secure support for unregistered nodes. To this end, we simply introduce another PS with a priority level higher than the one used in Section 7.3. This essentially assigns all remaining buffer capacity to unregistered nodes. These nodes will consequently receive the lowest quality-of-service level since their messages will be dropped first. However, we show later that in case the network is not fully congested (i. e., not during a flooding attack), unregistered users receive a quality-of-service level similar to that of registered users. Even when the network is under a flooding attack, performance never drops below the DD lower bound. For unregistered nodes, the identification token $T_s$ required for Algorithm 1 is derived from the network identifier.

## 8 Experimental Evaluation

In this section, we evaluate the behavior of our security mechanisms in large networks using the Opportunistic Network Environment (ONE) simulator [37] which is a well-accepted tool in the DTN research community. We first describe our evaluation scenario and present the performance results of *RESCUE* under flooding and Sybil attacks. Finally, we repeat the experiments under an accurate realistic mobility model for large-scale natural disasters. *RESCUE* is resilient against several attacks, since it leverages concepts from related work that have been shown to be secure. Therefore, we refrain from studying those attacks via experiments

2. An MA will consequently not assign any identities to the $\mathcal{S}_1$ set of the first user that registers. To also assign identities to the first users, an MA could use a set of known identities that registered pre-disaster.

3. Using this method, an adversary may be present in the PS of legitimate nodes by registering at an MA during disasters. However, the adversary may only register few identities this way, since it is hard to register multiple times at an MA (see Section 5.3).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3036224, IEEE Transactions on Dependable and Secure Computing



and focus on evaluating *RESCUE*'s novel mechanisms to protect against flooding and Sybil attacks.

### 8.1 Scenarios

We consider two scenarios as detailed in Table 4: a *synthetic* scenario to isolate the effect of two distinct attacks on the network (Sections 8.2 and 8.3) and a *Typhoon Haiyan* scenario to assess performance under more realistic conditions (Section 8.4). In both scenarios, we consider three different node classes: *pre-registered*, *post-registered*, and *unregistered*. During the course of the simulation, all *post-registered* nodes start unregistered, and become registered until the end of the simulation linearly over time. All nodes listed as *unregistered* in Table 4 remain unregistered. For simplicity, we only evaluate two *authorization levels* (Section 5): registered and unregistered. In the *Typhoon Haiyan* scenario, we have additional *roles* such as injured and healthy citizens, urban search-and-rescue teams (USRT), and UN officials, which all have distinct mobility patterns (refer to [38] for details). We use epidemic routing and compare four different buffer management strategies: (FIFO) uses a first-in first-out queue as prioritization and drop strategy, (MR) uses Message Rank instead of a FIFO queue, (SEB) employs our Source-based Elastic Buckets and uses all priority set levels except for $\mathcal{S}_1$, and (PS) makes use of *all* priority set levels. For the evaluation of the Sybil attack scenario, we additionally include the direct delivery (DD) buffer management strategy as a benchmark. We choose a small buffer capacity to exaggerate the effect of the different buffer management strategies. The message size is fixed to avoid fragmentation effects in the buffers. The simulation parameters in the *Typhoon Haiyan* scenario were chosen in accordance to Stute et al. [38] for comparison reasons. We use the ONE simulator v1.6.0 [37] for our experiments and, unless stated otherwise, show the average over ten runs with different seeds.

### 8.2 Flooding Attack

We first evaluate the resilience of different buffer management strategies against a small number (5 %) of attackers injecting bogus messages at a high rate in the *synthetic* scenario. The attackers maximize message lifetime of their messages: they address their messages to nonexistent destinations, so that acknowledgments are never returned; and they set the message lifetime to a value that is larger than the simulation time to keep their messages persistent unless they are dropped by the buffer management. Valid users choose the destination randomly among all other valid users. Figure 3 shows the overall delivery rate and delay for different attacker injection rates $R_\epsilon$ as a function of the valid users' injection rate $R_n$. Note that $R_\epsilon$ and $R_n$ are aggregate rates, e. g., $R_\epsilon = 1R_n$ means that all attackers inject as many messages as all valid users combined. To better understand the results, Fig. 3 differentiates between registered and unregistered users, and includes the network-wide copies per message during the attack. We make multiple observations.

**Importance of ACKs.** In a benign setting, ACKs help to keep buffers clean (SEB and MR). Once a message is delivered (Fig. 2a), the number of copies in the network reduces about as quickly as they increased (Fig. 3a) yielding perfect (i. e., 100 %) delivery rates.

**FIFO vs. MR.** Using FIFO as a buffer management strategy does not yield satisfactory results even in a benign scenario because buffer states do not converge: FIFO always accepts an incoming message even if it has previously been dropped. Performance further decreases as the attackers' injection rate is increased (Fig. 2c). On the other hand, prioritizing messages by deadline is apparently a very effective metric to achieve 100 % message delivery in less than 1 hour (MR in Fig. 2a) but, at the same time, is tremendously susceptible to flooding attacks, as it uses the TTL for prioritization that the attacker manipulates by setting an arbitrarily large message lifetime, thus, reducing the message delivery rate to about 5 % (Fig. 2c). This occurs since the attacker is able to quickly remove virtually all valid message copies in the network (Fig. 3b).

**SEB Mitigates Flooding Attacks for All Registered Users.** SEB uses MR as a secondary metric within each bucket. Therefore, SEB can achieve the same delivery rate as standalone MR (plots in Fig. 2a overlap), but maintains a high delivery rate of more than 90 % even under the flooding attack (Fig. 2c). In fact, as we vary the flooding injection rate, performance only decreases for unregistered users (Fig. 2g), while all other groups (Figs. 2d to 2f) *remain unaffected*. Since SEB's delivery rate did not change from $R_\epsilon = 1R_n$ to $2R_n$, we abstain from studying further increased $R_\epsilon$ values.

Table 4: Simulation Settings

| Scenario | *Synthetic* | *2013 Typhoon Haiyan* |
|---|---|---|
| Mobility | Map RWP (*Helsinki*) | *Natural Disaster* [38] |
| Speed | 90 % ped., 10 % car | 100 % pedestrian |
| Duration | 12 h (+5 h cool down) | 168 h = 7 days |
| Dimensions | 4500 × 3400 m$^2$ | 5000 × 7000 m$^2$ |
| Total Nodes | 1000 | 500 (7 roles [38]) |
| Unregistered | — | injured citizens |
| Post-registered | 800 | healthy citizens |
| Pre-registered | 200 | all others |
| Message Rate $R_n$ | 0.1 s$^{-1}$ (10 s interval) | 0.1 s$^{-1}$ (10 s interval) |
| Message Size | 25 KB | 25 KB |
| Message Lifetime | 5 h | 12 h |
| Buffer Capacity | 5 MB | 20 MB |
| Buffer Mgmt. | FIFO, MR, SEB, PS, DD | |
| Routing | Epidemic | |
| Radio Link | Bluetooth (2 Mbit/s at 10 m range) | |

### 8.3 Sybil Attack

In our next experiment, we evaluate the impact of a Sybil attacker on *RESCUE*, again in the *synthetic* scenario. The Sybil attacker is a single node (physical position) that can generate an unlimited number of registered identities, e. g., by compromising an MA. To cause maximum harm, the attacker executes a flooding attack as in Section 8.2 but uses a new identity for each injected message, therefore undermining our fair buffer management. Below, we compare the results of DD, SEB, and PS. To isolate the effect of our Sybil-secure priority set, we include an additional strategy PSo which only uses $\mathcal{S}_0$ and $\mathcal{S}_1$ (i. e., this strategy does not relay messages for unregistered nodes and registered

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3036224, IEEE Transactions on Dependable and Secure Computing



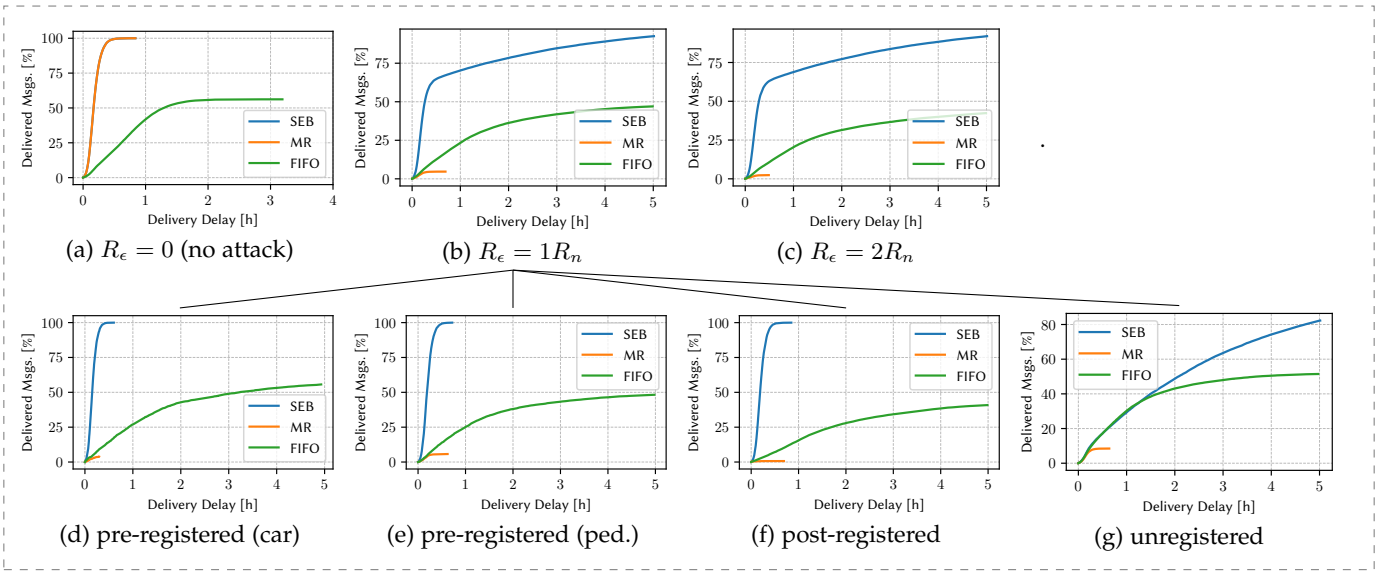



Figure 2: **Flooding attack.** Delivered messages over the delivery delay for different attacker injection rates $R_\epsilon$ (*top*) and user groups at $R_\epsilon = 1R_n$ (*bottom*)

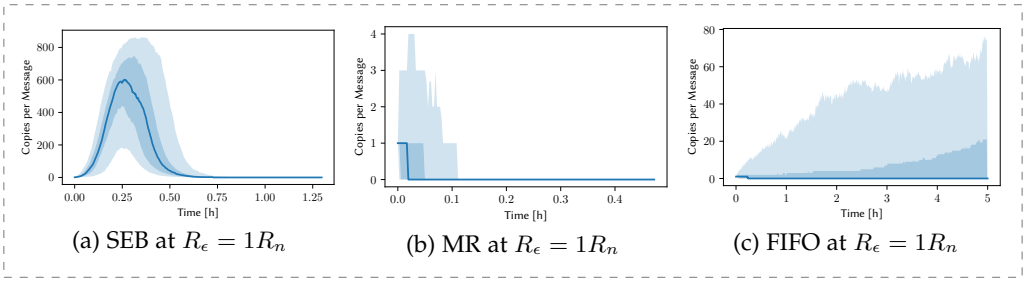



Figure 3: **Flooding attack.** Message copies over time of the *pre-registered (pedestrian)* node group at $R_\epsilon = 1R_n$ for simulation run 1. The center line shows the median, the shaded areas the 10th, 30th, 70th, and 90th percentiles.

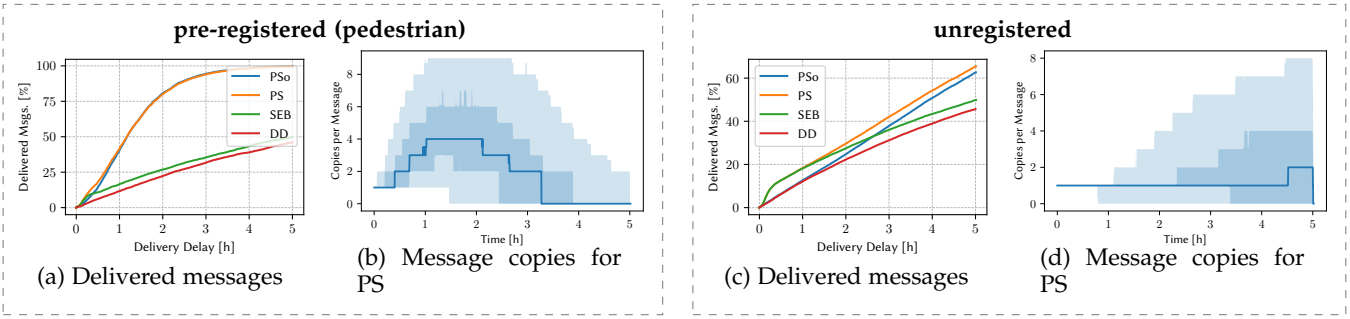



Figure 4: **Sybil attack.** Delivered messages over the delivery delay and message lifetime at $R_\epsilon = 1R_n$.

nodes that are not in the Sybil-secure priority set). We omit the results for FIFO and MR since they already performed poorly under a non-Sybil attack. We set the Sybil-secure priority set size $|\mathcal{S}_1|$ to 10. For the pre-registered nodes, the sets are generated from a fixed-degree random graph, while the sets of the post-registered nodes are determined by the $|\mathcal{S}_1|$ most recently post-registered nodes. For space reasons, Fig. 4 shows only the results for the pre-registered and unregistered group.

**Sybil Attacker Replaces All Non-*secure Copies*.** In Fig. 4a, we can see that the delivery rate for SEB significantly decreases under a Sybil attack compared to Fig. 2e. With PS, the situation improves enormously. The *secure copies* in $\mathcal{S}_1$ work as intended, so that there is a small number ($\leq |\mathcal{S}_l|$) of message copies that propagate through the network. Conversely, this experiment also shows that all $|\mathcal{S}_l|$ with $l > 1$ are completely vulnerable to a Sybil attack: the attacker is able to effectively replace all non-*secure copies*, which is confirmed by the almost identical performance of PSo and PS (overlap in Fig. 4a). Therefore, the network already operates at its lower performance limit. Further increasing the attacker's injection rate (beyond $R_\epsilon = 1R_n$) or the number of attackers should, thus, not affect the network's performance. This behavior demonstrates the *resilience* aspect of *RESCUE*:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3036224, IEEE Transactions on Dependable and Secure Computing



even under a strong attack, *RESCUE* continues to operate considerably well but with reduced performance.

**Direct Delivery Defines the Lower Bound Performance.** In Fig. 4c, we show the performance for unregistered users. We see that the performance (even with PS) is significantly lower than for registered users. However, all of our buffer management strategies perform better than DD, asserting our claim of achieving a lower bound performance. SEB performs slightly better as it can take advantage of the epidemic flooding at the beginning of the simulation when the attacker's messages have not yet fully saturated the network. PS performs better because some of the unregistered nodes become registered while their messages traverse the network and, thus, are prioritized by other nodes which also receive the new identity certificate. Figure 4d shows the average message copies which increase towards the end of the message lifetime.

### 8.4 2013 Typhoon Haiyan Scenario

To evaluate how *RESCUE* would perform in a disaster scenario, we repeat our experiments under an accurate human mobility model for large-scale natural disasters [38]. This model features seven different node roles (including citizens and professional disaster response teams (DRTs)), various points-of-interest (base camps, etc.), and time-of-day dependent activities. Depending on the group, we consider that users are registered (✓) or unregistered (✗). We evaluate *RESCUE* in the *Typhoon Haiyan* scenario (city of Tacloban, Philippines) with a total of 500 nodes. We show the inter-group message delivery rates for PS in Figs. 5 and 6. For space reasons, we only show the results for no attacker (Fig. 5) and for the Sybil attack with a flooding rate of $R_\epsilon = 1R_n$ but with different values for $|\mathcal{S}_1|$ (Fig. 6). The Sybil attack is the same as in Section 8.3 with the difference that the attacker node is chosen randomly from the DRO group, since this group is the best connected one due to high contact rates.

**Impact of Mobility Model.** In comparison to our *synthetic* scenario, the most striking difference is that we do not achieve perfect delivery rates when no attackers are present (compare Fig. 2a and Fig. 5). These differences are caused by the underlying mobility, e. g., some nodes (injured citizens) do not move at all. In addition, the *synthetic* scenario features fast-moving cars and a higher node density. Since cars might not be usable due to blocked roads, we focus on pedestrians only. Still, PS achieves significantly better results than FIFO. This suggests that PS is applicable even to benign settings.

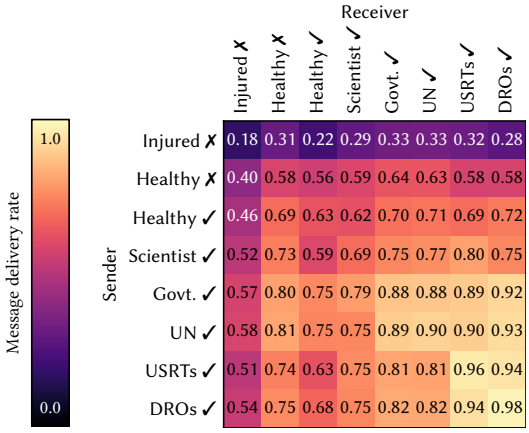



Figure 5: **Typhoon Haiyan scenario. No attack ($R_\epsilon = 0$).** Message delivery rates between roles using PS.

**Effectively No Service for Unregistered Citizens.** Unregistered users (✗) are almost completely denied service under attack, experiencing a delivery rate of 19 % at most (Fig. 6). Those citizens already receive reduced service in a benign scenario (Fig. 5). The direct-delivery performance that is achieved under attack does not suffice to maintain a reasonable delivery rate. This discrepancy emphasizes the importance of user registration especially under a Sybil attack.

**Impact of PS Size.** Under attack, delivery rates drop significantly for all groups (Fig. 6a). While well-connected registered groups can still achieve inter-group delivery rates above 90 %, unregistered groups are effectively cut off from communication. The situation for registered nodes improves when we increase $|\mathcal{S}_1|$ from 10 up to 50 (Figs. 6b and 6c): at $|\mathcal{S}_1| = 50$, the delivery rate is only 3–12 % lower than in a benign setting. However, as expected, increasing $|\mathcal{S}_1|$ has no effect on unregistered users.

**Delivery Asymmetries.** We detect asymmetries in the delivery rates between certain group pairs in Fig. 5 such as the registered DROs and the unregistered injured citizens (0.54 vs. 0.28). These asymmetries can be explained by our PS levels: registered users can utilize *secure relays* than unregistered ones. These asymmetries might influence the design of applications building on *RESCUE*.

### 8.5 Threats to Validity

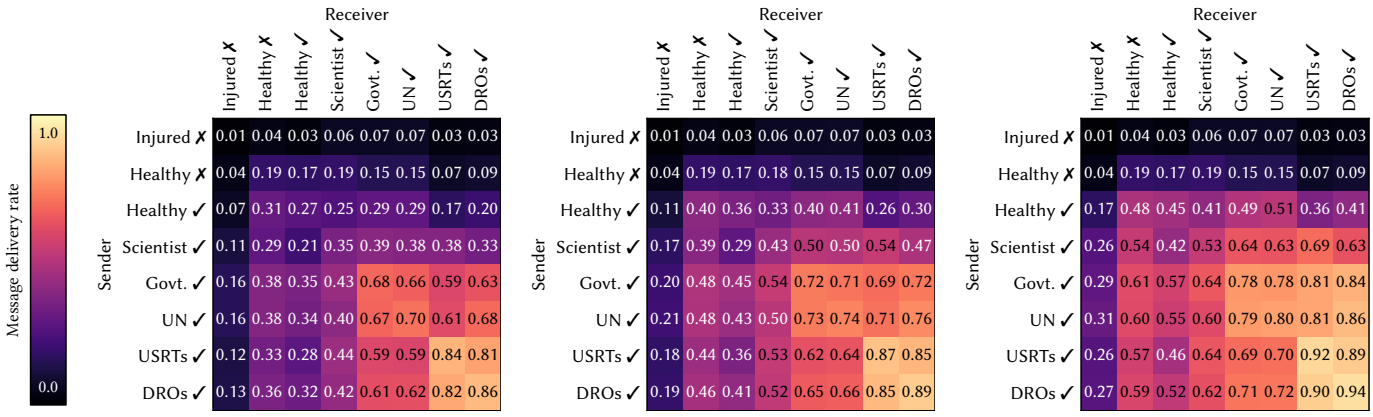
We designed *RESCUE* to be agnostic to the underlying mobility model. While this makes our approach applicable to different scenarios, we acknowledge that—as with any DTN—the nodes' mobility behavior governs its overall performance, i. e., reliability and delay. This means that *RESCUE*'s performance might vary in different scenarios. To assess the extend to which mobility influences performance, we considered two distinct scenarios that cover two extremes: ($i$) a well-connected network with several fast-moving nodes and ($ii$) a poorly connected network with slowly moving nodes. We expect that performance in other scenarios with similar features will fall between these two scenarios. Therefore, we can view the results presented in this section as an operating range when assessing whether *RESCUE* should be deployed in a particular (other) scenario.

## 9 Conclusion

In this work, we presented *RESCUE*, a communication framework for resilient and secure disruption-tolerant emergency communication on mobile devices. *RESCUE* is the first secure emergency communication solution that allows users to join the network during disasters when infrastructure networks are unavailable by deploying mobile authorities in the field. In addition, we are the first to present a buffer management approach to mitigate flooding

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3036224, IEEE Transactions on Dependable and Secure Computing






(a) $|\mathcal{S}_1| = 10$ (b) $|\mathcal{S}_1| = 20$ (c) $|\mathcal{S}_1| = 50$

Figure 6: **Typhoon Haiyan scenario. Sybil Attack ($R_\epsilon = 1R_n$).** Message delivery rates between roles using PS.

attacks even in the presence of Sybil attackers. In particular, our solution uses a minimalistic communication protocol and implements flooding mitigation via *source-based elastic buckets* that prevent attackers from purging valid messages. We also leveraged the concept of *secure copies* to implement *priority sets* to offer protection against Sybil attacks. We evaluated our solution in a synthetic scenario and have shown that, under flooding attacks, *RESCUE* maintains a delivery rate of 100 % for all registered users, while unregistered users experience a drop of up to 20 %. In the presence of a Sybil attacker, *RESCUE* maintains a delivery rate close to 100 % for all registered users, while unregistered users can still deliver more than 60 % of their messages. Finally, we confirmed that *RESCUE* performs well in a realistic natural disaster scenario and showed that the priority set size can be increased to improve the delivery success under a Sybil attack. In future work, we will integrate our buffer management strategies into an existing DTN implementation to evaluate performance and energy efficiency on mobile devices. In addition, we want to further investigate the relation between messages copies and delivery rates to appropriately dimension the priority sets. Finally, the proposed solution should be investigated in applications of the Internet-of-Things or in Fog computing scenarios. Here, we find similar heterogeneous device setups and often fluctuating network connectivity.

## Acknowledgments

This work has been co-funded by the LOEWE initiative (Hesse, Germany) within the emergenCITY center and by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

## References

[1] IRIN News, "Life-saving radio begins broadcasting in typhoon-hit Tacloban," Nov. 2013. [Online]. Available: http://www.irinnews.org/report/99132/life-saving-radio-begins-broadcasting-typhoon-hit-tacloban

[2] Z. Lu, G. Cao, and T. La Porta, "Networking smartphones for disaster recovery," in *IEEE PerCom*, 2016.

[3] A. Martín-Campillo, J. Crowcroft, E. Yoneki, and R. Martí, "Evaluating opportunistic networks in disaster scenarios," *Journal of Network and Computer Applications*, 2013.

[4] H. Yih-Chun and A. Perrig, "A survey of secure wireless ad hoc routing," *IEEE Security & Privacy Magazine*, vol. 2, no. 3, pp. 28–39, 2004.

[5] Q. Li, W. Gao, S. Zhu, and G. Cao, "To lie or to comply: Defending against flood attacks in disruption tolerant networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, 2013.

[6] J. Zheng, Q. Li, G. Gu, J. Cao, D. K. Yau, and J. Wu, "Realtime DDoS Defense Using COTS SDN Switches via Adaptive Correlation Analysis," *IEEE TIFS*, 2018.

[7] F. Kohnhäuser, M. Stute, L. Baumgärtner, L. Almon, S. Katzenbeisser, M. Hollick, and B. Freisleben, "SEDCOS: A secure device-to-device communication system for disaster scenarios," in *IEEE Conference on Local Computer Networks (LCN)*, 2017.

[8] T. Hossmann, P. Carta, D. Schatzmann, F. Legendre, P. Gunningberg, and C. Rohner, "Twitter in disaster mode: security architecture," in *ACM Special Workshop on Internet and Disasters*, 2011.

[9] S. G. Weber, Y. Kalev, S. Ries, and M. Mühlhäuser, "MundoMessage: Enabling trustworthy ubiquitous emergency communication," in *ACM IMCOM*, 2011.

[10] Briar Project, "Website," 2017. [Online]. Available: https://briarproject.org

[11] K. Sakai, M.-T. Sun, W.-S. Ku, and J. Wu, "A framework for anonymous routing in delay tolerant networks," in *International Conference on Network Protocols (ICNP)*. IEEE, 2017, pp. 1–10.

[12] J. Burgess, G. D. Bissias, M. D. Corner, and B. N. Levine, "Surviving attacks on disruption-tolerant networks without authentication," in *ACM MobiHoc*, 2007.

[13] T. N. D. Pham, C. K. Yeo, N. Yanai, and T. Fujiwara, "Detecting flooding attack and accommodating burst traffic in delay-tolerant networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 795–808, 2018.

[14] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "SybilGuard: defending against sybil attacks via social networks," in *ACM SIGCOMM*, vol. 36, 2006.

[15] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, "SybilLimit: A near-optimal social network defense against sybil attacks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, 2010.

[16] D. Quercia and S. Hailes, "Sybil attacks against mobile users: Friends and foes to the rescue," 2010.

[17] S. Trifunovic, M. Kurant, K. A. Hummel, and F. Legendre, "Preventing spam in opportunistic networks," 2014.

[18] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in DTNs," in *IEEE INFOCOM*, 2009.

[19] F. Li, J. Wu, and A. Srinivasan, "Thwarting blackhole attacks in disruption-tolerant networks using encounter tickets," in *IEEE INFOCOM*, 2009.

[20] ——, "Thwarting blackhole attacks in disruption-tolerant networks using encounter tickets," in *IEEE INFOCOM*, 2009.

[21] Q. Li and G. Cao, "Mitigating routing misbehavior in disruption tolerant networks," *IEEE TIFS*, vol. 7, no. 2, 2012.

[22] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Computer Networks*, vol. 51, no. 10, 2007.

[23] S. Skorobogatov, "The bumpy road towards iphone 5c nand mirroring," *arXiv preprint arXiv:1609.04327*, 2016.

[24] Android Developers, "Safetynet attestation api," 2018, https://developer.android.com/training/safetynet/attestation.

[25] Samsung, "White paper: An overview of Samsung KNOX," 2013.

[26] Alan Meeus, "Windows 10 Mobile security guide," 2017, https://technet.microsoft.com/en-us/itpro/windows/keep-secure/windows-10-mobile-security-guide.

[27] Joseph Yiu, "ARMv8-M architecture technical overview," 2015, https://community.arm.com/docs/DOC-10896.

[28] Trusted Computing Group, "TPM 2.0 Mobile Reference Architecture Specification," 2014, https://www.trustedcomputinggroup.org/tpm-2-0-mobile-reference-architecture-specification/.

[29] F. Kohnhäuser, N. Büscher, and S. Katzenbeisser, "A practical attestation protocol for autonomous embedded systems," in *IEEE EuroS&P*, 2019, pp. 263–278.

[30] T. Abera, R. Bahmani, F. Brasser, A. Ibrahim, A.-R. Sadeghi, and M. Schunter, "Diat: Data integrity attestation for resilient collaboration of autonomous systems," in *NDSS*, 2019.

[31] S. Wedaj, K. Paul, and V. J. Ribeiro, "Dads: Decentralized attestation for device swarms," *ACM TOPS*, vol. 22, no. 3, 2019.

[32] Regulation of the European parliament and of the council, "EUR-Lex - 32014R0910," 2014.

[33] A. Wiesmaier, M. Horsch, J. Braun, F. Kiefer, D. Hhnlein, F. Strenzke, and J. Buchmann, "An efficient mobile PACE implementation," in *ACM ASIACCS*, 2011.

[34] F. C. Lee, W. Goh, and C. K. Yeo, "A queuing mechanism to alleviate flooding attacks in probabilistic delay tolerant networks," in *IEEE AICT*, 2010.

[35] W. Galuba, P. Papadimitratos, M. Poturalski, K. Aberer, Z. Despotovic, and W. Kellerer, "Castor: Scalable secure routing for ad hoc networks," in *IEEE INFOCOM*, 2010.

[36] R. Morris and K. Thompson, "Password security: A case history," *Bell Laboratories*, 1978. [Online]. Available: https://web.archive.org/web/20130821093338/http://cm.bell-labs.com/cm/cs/who/dmr/passwd.ps

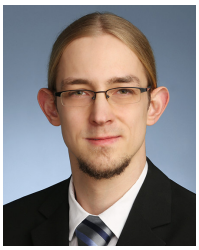
[37] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *ICST SIMUTools*, 2009.

[38] M. Stute, M. Maass, T. Schons, and M. Hollick, "Reverse engineering human mobility in large-scale natural disasters," Nov. 2017.


**Milan Stute** received his Ph.D. degree in computer science from TU Darmstadt in 2020 and is now a postdoctoral researcher at the Secure Mobile Networking Lab. His research focuses on denial-of-service attacks and prevention mechanisms for distributed wireless networks. He is recipient of several awards including outstanding Bachelor and Master thesis awards as well as ACM MobiCom Best Community Paper and Demo Awards.

**Florian Kohnhäuser** received his Ph.D. degree in computer science from TU Darmstadt in 2019. From 2015 to 2019 he worked as a research associate and doctoral candidate at the Security Engineering Group. His research interests include system and network security for mobile and embedded devices. More particularly, his research focuses on the design, development, and analysis of cryptographic protocols to verify the integrity of embedded systems (remote attestation).

**Lars Baumgärtner** received his Ph.D. degree in computer science from the University of Marburg, Germany. Since 2019, he has been a research staff member in the Software Technology Group at the TU Darmstadt. His research interests include computer/network security, resilient communication using embedded systems and mobile devices, and delay-tolerant networking for emergency scenarios in particular.



**Lars Almon** received his master degrees in computer science and IT security from TU Darmstadt in 2015. He started working as a research associate and doctoral candiate at the Secure Mobile Networking Lab in 2015. His research focuses on secure wireless communication protocols for mobile and resource constraint devices. Especially the design and management of wireless testbeds.

**Matthias Hollick** is currently heading the Secure Mobile Networking Lab at TU Darmstadt. After receiving the Ph.D. degree from TU Darmstadt in 2004, he has been researching and teaching at TU Darmstadt, Universidad Carlos III de Madrid, and the University of Illinois at Urbana–Champaign. His research focus is on resilient, secure, privacy-preserving, and QoS-aware communication for mobile and wireless systems and networks.

**Stefan Katzenbeisser** received his Ph.D. degree from the Vienna University of Technology, Austria. Since 2019, he has been a professor at University of Passau, previously at TU Darmstadt. His current research interests include embedded security, data privacy and cryptographic protocol design. He has authored over 200 scientific publications and served on the program committees of several workshops and conferences devoted to information security.

**Bernd Freisleben** received his Ph.D. degree and Habilitation from TU Darmstadt, in 1985 and 1993, respectively. He is a full professor in the Department of Mathematics and Computer Science, University of Marburg, Germany, and a part-time professor in the Department of Electrical Engineering and Information Technology, TU Darmstadt. His current research interests include mobile computing, wireless networks, computation- and data-intensive applications, and IT security.