Post-Quantum Signaturverfahren Heute

Andreas Hülsing[1], Albrecht Petzoldt[1], Michael Schneider[1] und Sidi Mohamed El Yousfi Alaoui[2]

[1]TU Darmstadt
Hochschulstraße 10, 64289 Darmstadt, Germany
{huelsing | mischnei | apetzoldt}@cdc.informatik.tu-darmstadt.de

[2]Center for Advanced Security Research Darmstadt - CASED
Mornewegstraße 32, 64293 Darmstadt, Germany
elyousfi@cased.de

Zusammenfassung

In diesem Papier geben wir einen Überblick über aktuelle praktikable Post-Quantum Signaturverfahren. Wir stellen vier unterschiedliche Verfahren vor und erörtern deren Vor- und Nachteile. Es stellt sich heraus, dass die heute bereits praktikablen Post-Quantum Signaturverfahren aus dem Bereich der multivariaten Kryptographie und der hashbasierten Signaturverfahren kommen. Diese Verfahren bieten eine Performanz vergleichbar mit heutigen Signaturverfahren und eignen sich auch für die Umsetzung auf Ressourcen-beschränkten Geräten. Der Vorteil hashbasierter Signaturverfahren ist Ihre beweisbare Sicherheit, während sich Multivariate Signaturverfahren durch kleine Signaturen hervorheben. Post-Quantum Signaturverfahren aus dem Bereich der Gitter und der Kodierungstheorie haben noch kein praktikables Level erreicht, aber auch dort existieren gute Ansätze.

1 Einleitung

Digitale Signaturen sind eines der meist genutzten kryptographischen Primitive überhaupt. Digitale Signaturen werden verwendet, um die Authentizität von Softwareupdates oder - in Kommunikationsprotokollen wie SSL/TLS - die Authentizität des Kommunikationspartners sicherzustellen. In Form der qualifizierten elektronischen Signatur bilden sie heute ein rechtskräftiges elektronisches Äquivalent zur handschriftlichen Unterschrift. Dies zeigt, dass wir großes Vertrauen in die heute verwendeten Verfahren setzen. Die Fähigkeit eines Angreifers, Signaturen zu fälschen, hätte heutzutage fatale Folgen. Die heute hauptsächlich verwendeten Signaturverfahren sind RSA, DSA und EC-DSA (DSA auf elliptischen Kurven).

Die Sicherheit dieser Signaturverfahren beruht auf der Annahme, dass das Faktorisieren von Zahlen mit großen Primfaktoren bzw. das Berechnen diskreter Logarithmen in bestimmten Gruppen Probleme darstellen, die nicht effizient gelöst werden können. 1994 hat Shor [Shor94] jedoch gezeigt, dass beide Probleme effizient gelöst werden können, wenn einem Angreifer ein Quantencomputer ausreichender Größe zur Verfügung steht. Aus diesem Grund beschäftigt sich die Forschung heute mit der Suche nach Signaturverfahren, die auch Angriffen mit einem Quantencomputer stand halten. Solche Verfahren werden Post-Quantum Signaturverfahren genannt.

Bei der Entwicklung neuer Post-Quantum Signaturverfahren werden zwei Ansätze verfolgt. Die Sicherheit heutiger Signaturverfahren basiert auf der Sicherheit sogenannter Trapdoor-Einwegfunktionen. Solche Funktionen sind effizient zu berechnen, doch im Allgemeinen nicht effizient zu invertieren. Es existiert jedoch zu jeder solchen Funktion eine Information - die Trapdoor - die es ermöglicht, die Funktion effizient zu invertieren. Zusätzlich zur Trapdoor-Einwegfunktion benötigen all diese Signaturverfahren eine kryptographische Hashfunktion, mit deren Hilfe die zu signierende Nachricht auf eine Länge komprimiert werden kann, die das eigentliche Signaturverfahren verarbeiten kann. Der erste Ansatz zur Entwicklung von Post-Quantum Signaturverfahren besteht darin, alternative Trapdoor-Einwegfunktionen zu konstruieren, deren Sicherheit auf Problemen beruht, die nach heutigem Kenntnisstand auch mit Hilfe von Quantencomputern nicht effizient gelöst werden können. Diese Verfahren beschreiben wir in Abschnitt 2. Ein alternativer Ansatz besteht darin, die Sicherheitsanforderungen für ein Signaturverfahren zu minimieren. Anstatt die Existenz einer Trapdoor-Einwegfunktion und einer kryptographischen Hashfunktion vorauszusetzen, werden Signaturverfahren konstruiert, die ausschließlich eine kryptographische Hashfunktion verwenden. Diesen Ansatz beschreiben wir in Abschnitt 3. In Sektion 4 versuchen wir abschließend einen Vergleich anzustellen.

2 Neue Probleme

Ein großer Teil der Forschung im Bereich Post-Quantum Signaturverfahren befasst sich damit, Signaturverfahren zu konstruieren, deren Sicherheit sich auf Probleme reduzieren lässt, von denen man annimmt, dass sie auch mit Hilfe von Quantencomputern nicht effizient gelöst werden können. Diese Probleme kommen aus dem Bereich der Gitter, der Kodierungstheorie und der multivariaten Gleichungssysteme. Die Eigenschaften der resultierenden Signaturverfahren hängen vom jeweils verwendeten Problemfeld ab. Im Folgenden skizzieren wir die Eigenschaften für jedes Problemfeld im Allgemeinen und für jeweils ein Signaturverfahren pro Problemfeld im speziellen.

2.1 Gitterbasierte Signaturverfahren

Die gitterbasierte Kryptographie gründet auf der Arbeit von Ajtai [Ajta96]. Die Sicherheit von gitterbasierten Signaturverfahren basiert auf der Schwierigkeit bestimmter Gitterprobleme, für deren Beschreibung wir den Leser auf [BeBD09] verweisen. Eine Eigenschaft, die gitterbasierte Signaturverfahren einzigartig macht, ist die Existenz sogenannter *Worst-case to Average-case Reductions*. Mit Hilfe dieser Reduktionen wird gezeigt, dass eine zufällige Instanz eines Problems in einer hohen Gitter-Dimension m so schwer zu lösen ist wie die härteste Instanz des Problems in einer bestimmten, kleineren Dimension n. Was dies bedeutet, sei an einem Beispiel erläutert. Angenommen es gäbe solch eine Reduktion für RSA und sei $m = 2n$. Wenn man wüsste, dass es mindestens einen 1024 Bit RSA-Modul gibt, für den das Faktorisieren mindestens T Jahre dauert, so wüsste man ebenfalls, dass es für jeden zufällig gewählten 2048 Bit RSA-Modul mindestens T Jahre dauert eine Faktorisierung zu berechnen. Für praktikable Parameter lässt sich diese Reduktion jedoch im Allgemeinen nicht anwenden. Für das Verhältnis zwischen m und n gilt $m \sim n \log n$ ungefähr. Um die Reduktion zu nutzen, müssten nun die Parameter des Verfahrens (insbesondere die Dimension m) so groß gewählt werden, dass die unterliegenden Probleme auch in Dimension n schwer sind. Damit würden die meisten Verfahren ineffizient. Da für gitterbasierte Signaturverfahren nur einfache arithmetische Operationen benötigt werden, lassen sich diese im Allgemeinen sehr effizient implementieren. Ein Problem der gitterbasierten Signaturverfahren ist die Größe der Schlüssel

und Signaturen. Ein weiteres Problem sind fehlende Implementierungen, die einen praktischen Vergleich erlauben würden.

Nachdem es für das erste praktische gitterbasierte Signaturverfahren NSS [HoPS01] und dessen Nachfolger NTRU [HHP+03] immer wieder erfolgreiche Angriffe gibt [GeSz02], [Nguy06], ist das aktuell praktikabelste gitterbasierte Signaturverfahren Lyubashevskys Signaturverfahren [Lyub09]. Es ist beweisbar sicher und bietet Signatur- und Schlüssellängen die annähernd vergleichbar mit heutigen Verfahren sind. Die Laufzeiten für Schlüsselerzeugung, Signatur und Verifizierung liegen in $\tilde{\mathcal{O}}(n)$. Leider existiert bisher noch keine Implementierung dieses Verfahrens. In Tabelle 1 sind theoretisch vorhergesagte Parameter für eine Sicherheit von 82 Bit angegeben. Diese gelten als sicher bis zum Jahre 2020.

2.2 Multivariate Signaturverfahren

Vielversprechende Signaturverfahren wurden basierend auf der Schwierigkeit, nichtlineare multivariate Gleichungssysteme zu lösen, konstruiert. Allen Verfahren ist gemein, dass sie sehr schnelle Laufzeiten haben, da nur einfache arithmetische Operationen benötigt werden. Daher eignen sich diese Verfahren besonders für die Verwendung auf Ressourcenbeschränkten Geräten. Darüber hinaus produzieren die Verfahren kurze Signaturen. Allerdings existiert für keines der Verfahren ein Sicherheitsbeweis. Die gebotene Sicherheit besteht, wie auch bei den heute verwendeten Signaturverfahren, darin, dass es bis heute niemandem gelungen ist, die Verfahren zu brechen. Zudem benötigen die Verfahren große Schlüssel. Als Beispiel geben wir das 2005 vorgestellte Rainbow Signaturverfahren [DiSc05] . In Tabelle 1 finden sich Signatur- und Schlüsselgrößen sowie Laufzeiten einer nicht optimierten Implementierung für 80 Bit Sicherheit.

2.3 Codebasierte Signaturverfahren

Der Vollständigkeit halber seien hier auch codebasierte Signaturverfahren erwähnt. Codebasierte Signaturverfahren basieren auf der Schwierigkeit von Problemen der Kodierungstheorie. Auch hier sei der Leser für eine genauere Beschreibung auf [BeBD09] verwiesen. Das einzige annähernd effiziente codebasierte Signaturverfahren ist CFS [CoFS01]. Der Vorteil dieses Verfahrens sind die mit wenigen hundert Bits kurzen Signaturen und eine schnelle Verifikation. Allerdings wurde der ursprüngliche Sicherheitsbeweis für CFS widerlegt [FGO+10] und das Verfahren produziert große öffentliche und private Schlüssel von nahezu einem Megabyte. Weiterhin benötigt eine CFS Signatur viel Zeit, da nicht jeder Hashwert signiert werden kann. Aus diesem Grund wird der ursprünglichen Nachricht ein Zähler angehängt und dieser solange inkrementiert, bis ein Hashwert entsteht, der signiert werden kann. Bevor dieses Problem nicht gelöst ist, sind codebasierte Signaturen fern jedweder Praxistauglichkeit. Da uns keine Implementierung vorliegt und auch keine Veröffentlichung, die Performanzwerte für aktuell sichere Parameter angibt, wurde CFS nicht in Tabelle 1 aufgenommen.

3 Minimale Sicherheitsanforderungen

Einen anderen Weg als die im letzten Kapitel beschriebenen Ansätze verfolgt die Forschung im Bereich der hashbasierten Signaturverfahren. Alle bisher genannten Signaturverfahren benötigen zusätzlich eine kollisionsresistente Hashfunktion, um Nachrichten beliebiger Länge zu Bitstrings fester Länge zu komprimieren, die das eigentliche Signaturverfahren verarbeiten kann. Bereits Ende der 80er Jahre hat Ralph Merkle [Merk90] vorgeschlagen, ein Signatur-

verfahren einzig aus kollisionsresistenten Hashfunktionen oder Blockchiffren zu konstruieren. Dabei wird aus einer Blockchiffre eine kollisionsresistente Hashfunktion konstruiert. Die existierenden hashbasierten Signaturverfahren verwenden somit kein konkretes Problem, sondern eine beliebige kryptographische Hashfunktion oder Blockchiffre. Die Auswahl an solchen Funktionen ist groß und es existieren sowohl sehr effiziente als auch beweisbar sichere Hashfunktionen. Aktuelle hashbasierte Signaturverfahren haben kleine Schlüssel und Laufzeiten für Signatur und Verifikation entsprechen heutigen Signaturverfahren. Zudem existieren inzwischen Sicherheitsbeweise für alle aktuellen hashbasierten Signaturverfahren. Auf der anderen Seite sind die Signaturen etwas größer als bei herkömmlichen Signaturverfahren und die Schlüsselerzeugung benötigt mehr Zeit als bei heutigen Verfahren. Ein weiterer möglicher Nachteil besteht darin, dass die Anzahl der mit einem Schlüsselpaar erstellbaren Signaturen bei der Schlüsselerzeugung festgelegt werden muss und dass hashbasierte Signaturverfahren zustandsbehaftet sind. Das bedeutet, dass sich der geheime Schlüssel mit jeder Signatur ändert. Dieser Nachteil birgt jedoch wiederum den Vorteil, dass diese Verfahren Vorwärtssicherheit bieten. Vorwärtssicherheit besagt, dass ein Angreifer, der zu einem Zeitpunkt t Kenntnis über den geheimen Schlüssel erlangt, trotzdem keine Signaturen für einen beliebigen Zeitpunkt t' < t fälschen kann.

Das ausgereifteste hashbasierte Signaturverfahren ist XMSS [BuDH11]. Es ist beweisbar sicher wenn es mit einer zweites Urbild resistenten Hashfunktion und einer pseudozufälligen Funktionsfamilie instanziiert wird. Eine (Hash-)Funktion ist zweites Urbild resistent, wenn sie effizient zu berechnen ist, es einem Angreifer jedoch nicht möglich ist, zu einer gegebenen Eingabe eine weitere Eingabe zu finden, so dass die Funktionswerte kollidieren. Diese Anforderung ist strikt schwächer als Kollisionsresistenz und ist im Gegensatz zur Kollisionsresistenz nicht anfällig für Geburtstagsangriffe. Eine Funktionsfamilie ist pseudozufällig, wenn es einem Angreifer nicht möglich ist, eine zufällig gewählte Funktion der Familie von einer zufällig gewählten Funktion aus allen Funktionen mit derselben Ein- und Ausgabelänge zu unterscheiden. Beide Eigenschaften werden von aktuellen kryptographischen Hashfunktionen erfüllt. Ebenso kann aus einer Blockchiffre eine Funktion konstruiert werden, die diese Eigenschaften erfüllt. Dies ermöglicht es auch existierende Hardwarebeschleunigung für AES auszunutzen. Die Performanz von XMSS hängt von der Performanz der verwendeten Funktion ab. Bei Verwendung von AES oder SHA-2 ist die Performanz vergleichbar mit der heutiger Signaturverfahren. Laufzeiten sowie Signatur- und Schlüsselgrößen für unterschiedliche Sicherheitslevel finden sich in Tabelle 1.

Darüber hinaus hat XMSS minimale Sicherheitsanforderungen. 1990 hat Rompel gezeigt [Romp90], dass die notwendige und hinreichende Bedingung für Signaturverfahren eine Einwegfunktion ist. Eine Einwegfunktion ist eine Funktion, die effizient berechnet, aber nicht effizient invertiert werden kann. Aus der theoretischen Kryptographie ist bekannt, dass die Existenz von Einwegfunktionen die Existenz von zweites Urbild resistenten Hashfunktionen und pseudozufälligen Funktionsfamilien impliziert. Dies zeigt, dass es eine Instanziierung von XMSS gibt, solange es ein Signaturverfahren gibt.

4 Vergleich

Es zeigt sich, dass es sich bei den praxistauglichen Post-Quantum Signaturverfahren um hashbasierte und multivariate Signaturverfahren handelt. Es existieren jeweils Implementierungen auf unterschiedlichen Plattformen und die Verfahren übertreffen in manchem Parameter heutige Signaturverfahren. Hashbasierte Signaturverfahren kommen mit einem Sicher-

heitsbeweis und bieten Vorwärtssicherheit, während multivariate Signaturverfahren besonders schnelle und einfach zu implementierende Algorithmen bieten. Gitterbasierte Verfahren sind vielversprechend, insbesondere auf Grund der starken Sicherheitsgarantien die sie bieten. Es fehlt jedoch eine Implementierung, um die Performanz der Algorithmen zu evaluieren. Ebenso wäre es wünschenswert, wenn sich Verbesserungen bezüglich der Signatur- und Schlüsselgrößen erzielen ließen. Codebasierte Signaturverfahren scheiden auf Grund der extrem großen Schlüssel und langer Signaturzeiten vorerst aus.

Tabelle 1 zeigt Performanzwerte für XMSS, Rainbow und Lyubashevsky Signaturen. Ein Verfahren mit einer Bitsicherheit von 80 Bit wird nach der Heuristik von Lenstra [Lens06] bis 2018 als sicher eingestuft, 82 Bit genügen bis zum Jahre 2020, 100 Bit bis 2048 und 146 Bit bis 2117. Im Falle von XMSS können alle Schlüsselpaare für 2^{20} Signaturen verwendet werden. Im Falle von Lyubashevskys Signaturverfahren wurden Quantencomputer bei der Berechnung der Bitsicherheit berücksichtigt. Dies gilt nicht für die übrigen Verfahren. Die ersten beiden Zeilen zeigen den Trade-Off zwischen Signaturgröße und Laufzeit bei XMSS und Zeile drei zeigt Laufzeiten für XMSS bei Verwendung des neuen Intel AES-NI Befehlssatzes. Die Werte für XMSS stammen aus [BuDH11], die Werte für Lyubashevsky aus [RüSc10]. Die Werte für Rainbow wurden mit einer eigenen Implementierung ermittelt. Diese verwendet Optimierungen die in [PeBB10] beschrieben sind.

Tabelle 1: Performanz. * Messungen auf Intel(R) Core(TM) i5 CPU M540 @ 2.53GHz mit Intel AES-NI, ** Messungen auf Intel(R) Core(TM) i3 CPU M 350 @ 2.27 GHz

	Signieren (ms)	**Verifizieren (ms)**	**Signatur (bit)**	**öffentlicher Schlüssel (bit)**	**geheimer Schlüssel (bit)**	**Bitsicherheit**
XMSS-SHA-2*	15,17	1,02	16.664	13.568	280	146
XMSS-AES-NI*	1,72	0,11	19.608	7.296	152	82
XMSS-AES*	2,87	0,22	19.608	7.296	152	82
Rainbow**	0,69	0,21	344	84.944	156.368	80
Lyubashevsky	-	-	100.480	21.360	21.840	82
RSA 2048*	3,08	0,09	$\leq$ 2.048	$\leq$ 4.096	$\leq$ 4.096	87

5 Literatur

[Ajta96] Ajtai, Miklos: Generating hard instances of lattice problems (extended abstract). In: STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, ACM, New York, 1996.

[BeBD09] Bernstein, Daniel J.; Buchmann, Johannes; Dahmen, Erik (Eds.): Post-Quantum Cryptography. Springer, 2009.

[BuDH11] Buchmann, Johannes; Dahmen, Erik; Hülsing, Andreas: XMSS - A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions. In: Post-Quantum Cryptography, LNCS 7071, Springer, Berlin / Heidelberg, 2011.

[CoFS01] Courtois, Nicolas; Finiasz, Matthieu; Sendrier, Nicolas: How to Achieve a McEliece-Based Digital Signature Scheme. In: Advances in Cryptology — ASIACRYPT 2001, Springer, Berlin / Heidelberg, 2001.

[DiSc05] Ding, Jintai; Schmidt, Dieter: Rainbow, a New Multivariable Polynomial Signature Scheme. In: Applied Cryptography and Network Security, LNCS 3531, Springer, Berlin / Heidelberg, 2005.

[FGO+10] Faugère, Jean-Charles; Gauthier, Valérie; Otmani; Ayoub; Perret, Ludovic; Tillich, Jean-Pierre: A Distinguisher for High Rate McEliece Cryptosystems. In: Cryptology ePrint Archive: Report 2010/331, IACR, 2010.

[GeSz02] Gentry, Craig; Szydlo, Mike: Cryptanalysis of the Revised NTRU Signature Scheme. In: Advances in Cryptology — EUROCRYPT 2002, Springer, Berlin / Heidelberg, 2002.

[HHP+03] Hoffstein, Jeffrey; Howgrave-Graham, Nick; Pipher, Jill; Silverman, Joseph; Whyte, William: NTRUSign: digital signatures using the NTRU lattice. In: CT-RSA'03 Proceedings of the 2003 RSA conference The cryptographers' track, Springer, Berlin / Heidelberg 2003.

[HoPS01] Hoffstein, Jeffrey; Pipher, Jill; Silverman, Joseph: NSS: An NTRU Lattice-Based Signature Scheme. In: Advances in Cryptology — EUROCRYPT 2001, Springer, Berlin / Heidelberg 2001.

[Lens06] Lenstra, Arjen K.: Key lengths. In: The Handbook of Information Security, Wiley, 2006.

[Lyub09] Lyubashevsky, Vadim: Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In: Advances in Cryptology – ASIACRYPT 2009, Springer, Berlin / Heidelberg, 2009.

[Merk90] Merkle, Ralph: A Certified Digital Signature. In: Advances in Cryptology - CRYPTO' 89 Proceedings, Springer, Berlin / Heidelberg, 1990.

[Nguy06] Nguyen, Phong Q.: A Note on the Security of NTRUSign. In: Cryptology ePrint Archive, Report 2006/387, IACR, 2006.

[PeBB10] Petzoldt, Albrecht; Bulygin, Stanislav; Buchmann, Johannes: CyclicRainbow – A Multivariate Signature Scheme with a Partially Cyclic Public Key. In: Progress in Cryptology - INDOCRYPT 2010, Springer, Berlin / Heidelberg, 2010.

[Romp90] Rompel, John: One-way functions are necessary and sufficient for secure signatures. In: STOC '90: Proceedings of the twenty-second annual ACM symposium on Theory of computing, ACM, New York, 1990.

[RüSc10] Rückert, Markus; Schneider, Michael: Estimating the Security of Lattice-based Cryptosystems. In: Cryptology ePrint Archive, Report 2010/137, IACR, 2010.

[Shor94] Shor, Peter W.: Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In: Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1994), IEEE Computer Society Press, 1994.