IT-Sicherheit Seminar

Post-Quantum Kryptographie

Eingereicht von: Alexander Stein
E-Mail: winf100313@fh-wedel.de

Referent: Prof. Dr. Gerd Beuster
E-Mail: gb@fh-wedel.de

Fachhochschule Wedel
Feldstraße 143
22880 Wedel

Inhaltsverzeichnis

1. Einleitung

Kryptographie beschäftigt sich hauptsächlich, aber nicht ausschließlich, mit der Sicherung von Kommunikation und ihrer Übertragung im Internet. Kryptographie ist aber auch in vielen anderen Bereichen von großer Bedeutung. So muss im Bereich der Autoindustrie beispielsweise gewährleistet sein, dass jeweils nur ein Schlüssel ein bestimmtes Auto öffnen kann, während im Bereich der Medizin das Speichern und Verwahren von sensiblen persönlichen Daten ermöglicht werden muss. Zudem ist das Verwahren von vertraulichen Informationen ebenfalls bei Festplattenverschlüsselung wichtig, um ein weiteres Beispiel zu nennen.

Krypto Systeme versuchen die Vertraulichkeit, die Integrität sowie die Authentizität von zu übermittelnden Daten und Informationen zu gewährleisten, damit Datentransfer sicher und ohne Einschränkungen ablaufen kann. Bei einem Angriff werden diese zuvor erwähnten Aspekte beeinträchtigt und gefährdet.

Post-Quantum Kryptographie betrachtet Kryptographie unter der Annahme, dass dem Angreifer ein leistungsstarker Quantencomputer zur Verfügung steht. Diese Entwicklung würde kryptografische Systeme, die heute Anwendung finden und heutzutage relativ sicher sind, beeinflussen. Post-Quantum Kryptosysteme verfolgen das Ziel sicher zu bleiben, auch wenn diese neue Technologie des Quantencomputers verfügbar wird. In diesem relativ neuen Forschungsbereich wurden schon erste Erfolge verzeichnet. So wurden bereits mathematische Operationen gefunden, die sogar mit Quantenalgorithmen nicht deutlich schneller geknackt werden können. Die zentrale Herausforderung von Post-Quantum Kryptographie ist es, genauso benutzbar und flexibel wie herkömmliche Kryptographie zu sein, ohne Sicherheitseinbußen nach sich zu ziehen.

Ein typischer Angriff versucht Informationen, die zwischen zwei Partnern übertragen werden, mitzuschneiden und beeinträchtigt damit die Vertraulichkeit der Informationen. Oder es werden eigene Informationen eingeschleust, welches die Integrität verletzt. Außerdem kann auch die generelle Verfügbarkeit von Informationen eingeschränkt werden, um Informationen vorzuenthalten. Um dies zu verhindern verwenden Kommunikationspartner Verschlüsselungssysteme.

Die zwei praktikabelsten Verschlüsselungssysteme normaler Kryptographie sind RSA und ECC. Diese werden in einem großen Teil von verschlüsselter Kommunikation eingesetzt. Bei

diesen beiden Verschlüsselungssystemen handelt es sich um asymmetrische Verschlüsselungsverfahren. Wenn sich Quantencomputer aber so entwickeln wie derzeit angenommen wird, sind sowohl RSA als auch ECC nicht mehr sicher. Der Markt befindet sich genauso wie die Forschung in einem Wettrennen gegen die Zeit, um Post-Quantum Algorithmen zu implementieren, bevor leistungsstarke Quantencomputer zur Verfügung stehen und Kommunikation beeinträchtigt werden kann.

In dieser Seminararbeit präsentiere ich die Ergebnisse von Daniel J. Bernstein und Tanja Lange aus ihrem Paper "Post-quantum cryptographie - dealing with the fallout of physics success". In einer Einleitung führe ich in das Thema der Kryptographie ein und stelle zwei Quantenalgorithmen vor, die bisherige Verfahren beeinträchtigen. Weiter werde ich zwei Kryptosysteme vorstellen, welche eine Alternative in Bezug auf Post-Quantum Kryptographie zu derzeitigen Systemen bieten. Abschließend wird ein Ausblick gegeben und die Ergebnisse werden zusammengefasst.

2. Quantenalgorithmen

Im Folgenden werden zwei Quanten-Algorithmen vorgestellt, die die meisten der heute verwendeten kryptographischen Systeme zumindest beeinträchtigen. Der Shor Algorithmus hat dabei Einfluss auf die asymmetrischen Verschlüsselungssysteme und der Grover Algorithmus beeinträchtigt die symmetrischen Verschlüsselungssysteme.

2.1 Shor Algorithmus

Das Verfahren RSA verwendet einen public-key, der aus dem Produkt N zweier geheimer Primzahlen p und q ermittelt wird. Die Sicherheit dieses Verfahrens basiert allein darauf, dass es in angemessener Zeit nicht möglich ist, eine große Zahl N in ihre Faktoren p und q zu zerlegen.

Peter Shor ermittelte 1994 ein Quanten-Fakturierungsverfahren, dass für eine beliebige positive Zahl N die Primzahlfaktoren ermitteln kann. (vgl. Shor 1995) Im Jahr 2001 wurde mittels dieses Algorithmus auf einem Quantencomputer mit 3 Quibits die Zahl 15 in ihre Faktoren 3 und 5 zerlegt. (vgl. Bernstein/Lange 2017:4) Dies hat gravierende Folgen. Sobald der Shor Algorithmus mit einer hohen Anzahl Quibits verwendet wird, ist das meist genutzte asymmetrische Verschlüsselungsverfahren RSA nicht mehr sicher. Dies gilt ebenso für alle

anderen Verfahren, die auf dem Problem der Fakturierung beruhen. Shor entwickelte zusätzlich einen Algorithmus, welcher auf ähnliche Weise in der Lage ist, mit der Addition von elliptischen Kurven umzugehen. (vgl. ebd.) Dadurch wird auch das nächst beliebteste Verfahren ECC wird mehr sicher sein.

Shors Algorithmus basiert auf der grundlegenden Idee, dass Fakturierung auf die Bestimmung der Ordnung zurückgeführt werden kann. Der Algorithmus berechnet mittels Quantenberechnungen alle potentiellen Teiler von N gleichzeitig und bildet sie in einem Quantenstatus ab. Anhand der Quanten-Fouriertransformation wird dann ein Ergebnis ausgegeben, welches mit hoher Wahrscheinlichkeit ein echter Faktor von N ist. (vgl. Shor 1995)

An diesem Algorithmus wurde seitdem weiter geforscht. Inzwischen gibt es Variationen und die erforderte Leistung ist bekannt. Eine Variation des Algorithmus von Beauregard benötigt 2n+3 Quibits bei einer Laufzeit von $O(n^3 \log n)$. (vgl. Bernstein/Lange 2017:4) Dabei ist n die Anzahl von Bits, die zur Speicherung von N nötig ist. Die Laufzeit kann sogar auf bis zu $O(n^2+o(1))$ reduziert werden, bei Erhöhung der verfügbaren Quibits, denn es können mehrere Operationen parallel durchgeführt werden. (vgl. ebd.)

Will man den Shor Algorithmus auf heutige Schlüsselstandards anwenden, benötigt man allerdings Milliarden von Operationen und Millionen von Quibits. (vlg. ebd.) Es ist sehr wahrscheinlich, dass Quantencomputer nicht in auf eine solche Größe skalierbar sind.

2.2 Grover Algorithmus

Viele weitere kryptographische Verfahren sind durch einen Algorithmus von Grover betroffen, den er 1996 veröffentlichte. Dieser Algorithmus legte die Grundlage für die meisten Applikationen von Quantencomputern.

Ursprünglich war der Algorithmus dafür gedacht, in ungeordnete Datenbanken der Größe N mit $\sqrt{N}$ Quantenoperation zu suchen. (vgl. Grover 1996) Geordnete Datenbanken können allerdings mit O(log n) durchsucht werden und sind daher immer noch vorteilhafter.

Der Grover-Algorithmus ermöglicht, genau betrachtet, keine direkte Suche in unsortierten Datenbanken, sondern die Umkehrung einer endlichen Funktion y = f(x), denn zu einem gegebenen Wert y entspricht ein Wert x = f^{-1} (y) der Suche nach einem Wert x im

Definitionsbereich von f. Wichtig ist dabei die Komplexität der der Funktion, da eine Laufzeit von √N eine Menge Quibit Operationen benötigt. (vgl. Bernstein/Lange 2017:4)

Zur Veranschaulichung wird nun ein Beispiel aus der Kryptographie angeführt:

Der Nutzer hat einen 128-Bit Klartext "7" und "8" mit dem 128-Bit AES Schlüssel k verschlüsselt und versendet seinen verschlüsselten Text c = ((AES(k, 7), AES(k, 8)). Wir definieren f(x) = (AES (x, 7), AES (x, 8)) - c. Diese Funktion f kann leicht berechnet werden (ca. 20.000 Bit Operationen) und Grovers Algorithmus findet die Umkehrfunktion in etwa 2^{64} Quantenoperationen. (vgl. ebd.) Die berechnete Umkehrfunktion ist mit hoher Wahrscheinlichkeit k. Auch wenn es theoretisch möglich ist, ist es doch sehr unwahrscheinlich, dass ein Schlüssel gewählt wurde, für den es weitere Schlüssel gibt, für die (AES (x, 7), AES (x, 8)) = (AES (k, 7), AES (k, 8)) gilt. (vgl. ebd.) Es kann also angenommen werden, dass die Umkehrfunktion korrekt ermittelt wurde.

Der Grover Algorithmus ermöglicht demzufolge eine Entschlüsselung mit einer Laufzeit von O(√N). Dies ist eine deutliche Beschleunigung zu O(N) bei herkömmlichen Methoden. Die Laufzeit befindet sich allerdings noch in derselben Komplexitätsklasse wie Pre-Quantum Verfahren. Im Vergleich zu Shors Algorithmus wird Grovers Algorithmus eher als ungefährlich für derzeitige kryptographische Systeme eingeschätzt.

Sobald eine hohe Anzahl von Quibits zur Verfügung steht, kann Grovers Algorithmus wirksam gegen symmetrische Verschlüsselungssysteme mit einem 2^{128} Sicherheitsstandard werden, wie beispielsweise 128-Bit AES Verschlüsslungen. Jedoch sollten solche Systeme problemlos auf 256-Bit umsteigen können. Symmetrische Verfahren können demnach mit leichten Anpassungen bei einem Angriff, der den Grover Algorithmus nutzt, sicher bleiben. (vgl. ebd.)

Die Beschleunigung der Laufzeit von N zu √N unter Verwendung des Grover Algorithmus ist nicht so extrem, wie die des Shor Algorithmus.

3. Post-Quantum Kryptographie

Name	function	pre-quantum security level	post-quantum security level
post-quantum security level			
AES-128	block ctpher	128	64 (Grover)
AES-256	block ctpher	256	128 (Grover)
Salsa20	Stream cipher	256	128 (Grover)
GMAC	MAC	128	128 (no impact)
Poly1305	MAC	128	128 (no impact)
SHA-256	hash function	256	128 (Grover)
SHA-3	hash function	256	128 (Grover)
Public-key cryptography			
RSA-3072	encryption	128	broken (Shor)
RSA-3072	signature	128	broken (Shor)
DH-3072	key exchange	128	broken (Shor)
DAS-3072	signature	128	broken (Shor)
256-bit ECDH	key exchange	128	broken (Shor)
256-bit ECDSA	signature	128	broken (Shor)

Tabelle 1: Übersicht kryptographischer Verfahren (Bernstein/Lange 2017: 3)

Tabelle 1 fasst den Effekt von Shors und Grovers Algorithmus auf die üblichen kryptographischen Verfahren zusammen. Es mag der Eindruck entstehen, dass mit dem Aufkommen von Quantencomputern der Abstieg von asymmetrischer Verschlüsselung einhergeht und von dort an nur noch symmetrische Verschlüsselungsverfahren mit längeren Schlüsseln zurückbleiben. Allerdings sind RSA und ECC nicht die einzigen Public-Key Systeme.

Im nächsten Abschnitt werden verschiedene Verfahren betrachtet, die den bisher präsentierten Angriffen standhalten können. Bisher ist es niemandem gelungen, den Shor-Algorithmus auf diese Verfahren anzuwenden.

Bei der Auswahl der Schlüsselgröße ist der Grover Algorithmus zu berücksichtigen, ebenso weitere Quantenverfahren, wie der Quanten Walk, (vgl. Bernstein/Lange 2017: 5) ein weiterer Quantenalgorithmus. Ist die Schlüsselgröße zu hoch angesetzt, werden wesentlich höhere Laufzeitkosten verursacht. Der Einfluss des Grover Algorithmus muss noch genauer betrachtet werden, um die kleinstmögliche Schlüsselgröße zu ermitteln, die sowohl Sicherheit als auch schnelle Verarbeitung ermöglicht.

Im Folgenden werden zwei der alternative Krypto Systeme vorgestellt, die von Bernstein und Lange ausgewählt wurden. Die in ihrem Paper ausgewählten Krypto Systeme sind vor allem Verfahren, die seit Jahrzenten bekannt sind und noch nicht mit herkömmlichen Methoden geknackt werden konnten und diese zählen somit auch dazu. Bisher ist es auch noch niemandem gelungen, den Shor-Algorithmus auf diese Verfahren anzuwenden.

3.1 Code-based Encryption

Sehr zuverlässige Geräte verwenden einen "error-correcting code". Dabei werden beispielsweise 64 Bits logischen Speichers auf 72 Bits physikalischem Speicher abgebildet. Eine 72 x 64 "Generator Matrix" G ordnet jedem der 72 physikalischen Bits eine Summe Modulo 2, von logischen Bits zu. Der Code ist so konzipiert, dass jeder einzelne Fehler an einem Bit korrigiert werden kann und auch doppelte Fehler erkannt werden können.

Error-correcting Codes können skaliert werden, sodass mehrere Fehler in größeren Blöcken gleichzeitig erkannt werden können. Diese Technologie wird in einer Vielzahl von Applikationen verwendet, unter anderem in Festplatten, Satellitenkommunikation und fehlersicherer Quantenberechnung. Dieser error-correcting Code wird auch als Gobba Code bezeichnet.

1978, als public-key Verfahren noch neu waren, hatte R. J. McEliece die Idee, eine solche Generator Matrix als öffentlichen Schlüssel zu verwenden. Dabei wird ein Code verschlüsselt, indem eine bestimmte Anzahl von Fehlern hinzugefügt wird. McElieces generierte dann einen privaten Code, der die einzelnen Fehler beheben kann, was allein mit der Generator Matrix nicht möglich ist. (vgl. McEliece 1978)

Für dieses Verfahren wird erst eine Goppa Code Matrix G erstellt, die in der Lage ist, eine bestimmte Anzahl von Fehlern t, zu korrigieren. Diese Matrix G wird mit einer invertierbaren "Scrambler"-Matrix S und einer Permutationsmatrix P multipliziert, um G' zu erhalten. Also ist G' = S*G*P. (vgl. ebd.) Die Matrix G' dient als öffentlicher Schlüssel. Zusätzlich zum öffentlichen Schlüssel wird beim Verschlüsseln ein zufälliger Fehlervektor addiert. Dieser Fehlervektor darf maximal t Fehler beinhalten. Der private Schlüssel besteht aus den Matrizen S, G und P. Mittels der inversen Matrizen S^{-1} und P^{-1} kann die Nachricht entschlüsselt werden und mittels des Goppa Codes der Fehler behoben werden.

Eine einfache aber langsame Angriffsstrategie gegen McEllieces System ist "information set decoding" (ISD). Ein Informations-Set ist dabei eine Ansammlung von Codewort Positionen, die den Rest des Codeworts bestimmen. ISD rät ein Information-Set, unter der Annahme, dass der verschlüsselte Text an diesen Stellen fehlerfrei ist und benutzt lineare Algebra, um das Codewort zu bestimmen. Wenn der Verschlüsselungstext die gleiche Anzahl an Fehlern aufweist, muss das Codewort korrekt sein. (vgl. Bernstein/Lange 2017: 4)

ISD ist sehr langsam, weil die Chance ein fehlerfreies Informations-Set zu erhalten bei großen Matrizen relativ gering ist. Die Anzahl von Versuchen ist im Durchschnitt $\left(c + o(1)\right)^w$, dabei ist w die Anzahl der platzierten Fehler, c ist eine Konstante größer 1, die von dem Verhältnis zwischen Reihen und Spalten abhängt, o(1) konvergiert gegen 0 für $w \to \infty$. (vgl. ebd.)

Für ISD wurden nach der Veröffentlichung von dutzenden Angriffs-Papers viele Verbesserungen gefunden, doch die benötigte Zeit liegt immer noch bei $\left(c + o(1)\right)^w$. McEllieces konzipierte sein Modell mit einem Sicherheitslevel von 2^{64}. Ein erfolgreicher Angriff auf dieses System benötigt auch noch heutzutage, etwa 40 Jahre später, über 2^{60} CPU Zyklen. (vgl. ebd.) Den Entschlüsselungscode selbst herauszufinden würde mit bekannten Algorithmen noch länger dauern. Die einzige Post-Quantum Veränderung liegt in der Konstante c, welche unter der Verwendung des Grover Algorithmus durch √n ersetzt werden kann.

Niederreiter verfeinerte das Verfahren, indem er den öffentlichen Schlüssel in eine systematische Form brachte. Dabei sind die ersten k physischen Bits genau die gleichen, wie die ersten k logischen Bits. So ist die erste k x k Submatrix der Generator Matrix eine Identitätsmatrix und muss nicht transportiert werden. Eine weitere Verbesserung, die Niederreiter an dem Modell vorgenommen hat, war, "Symptome" an Stelle eines gigantischen Codeworts zu verschicken. Dies verringert die Größe des Verschlüsselungstextes auf 200 Bytes und ist somit in der Lage, ein hohes Sicherheitslevel aufrecht zu erhalten. (vgl. ebd.)

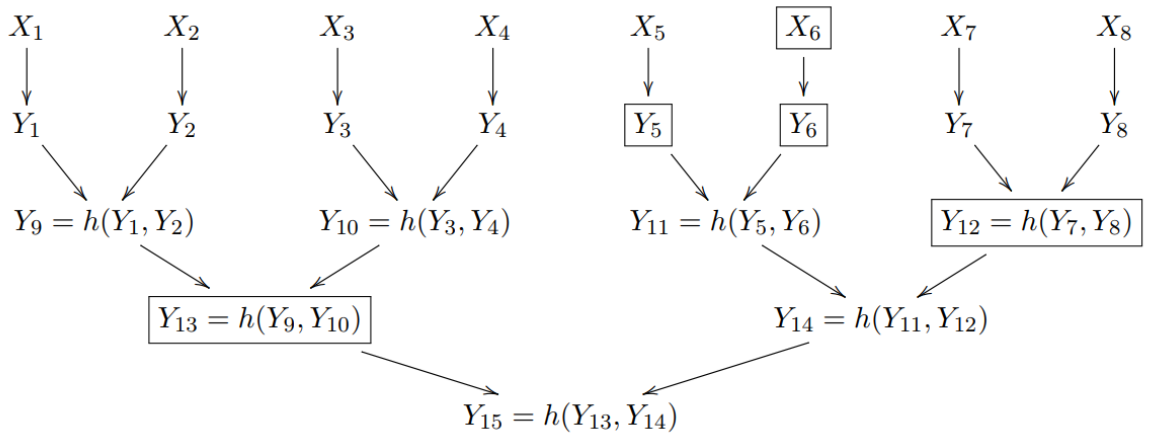
Das Hauptproblem in der Praxis stellt die Schlüsselgröße dar, selbst in der systematischen Form beträgt sie rund ein Megabyte, wenn ein hohes Sicherheitsniveau angestrebt wird. Es gibt Variationen, die sich diesem Problem angenommen haben und den Schlüssel komprimiert haben. Allerdings sind sie meist gerade dadurch angreifbarer geworden. Das einzig bewährte Verfahren ist das McElliece System mit der Erweiterung von Niederreiter.

3.2 Hash-based Signatures

Eine Hashfunktion bildet Daten von unterschiedlicher Größe auf einen Wert mit einer festen Größe ab. Die für die Kryptographie wichtige Eigenschaft dabei ist, dass sie eine Einwegfunktion ist. Das bedeutet, dass aus dem erzeugten Hashwert nur mit sehr großem Aufwand der Eingabewert ermittelt werden kann. 1975 wurde ein Verfahren entwickelt, dass unter dem Namen Lamport-Diffie-Einmal-Signaturverfahren (englisch: „Lamport signature“) bekannt geworden ist, welches eine kollisionsresistente Hashfunktion berechnet. (vgl. Bernstein/Lange 2017:10)

Signaturen werden wie folgt generiert. Der Nutzer sucht sich 2 zufällige Strings x_0 und x_1. Der öffentliche Schlüssel ermittelt sich aus den jeweiligen Hashwerten der Strings (h(x_0), h(x_1)). Dabei ist h die Hashfunktion, diese darf jedem bekannt sein. Sobald eine Unterschrift geprüft werden soll, gibt man x_0 bekannt. Der Prüfer berechnet h(x_0) und erkennt den Hashwert aus dem öffentlichen Schlüssel wieder. Da die Hashfunktion eindeutig ist, war der Schlüssel richtig. Für die zweite Signatur wird dann x_1 verwendet. (vgl. Lamport 1979) Für n Nachrichten müssen allerdings auch n Schlüssel generiert werden, um die Sicherheit des Schlüssels zu gewährleisten. Mehrfachnutzung eines Schlüssels ist aus Sicherheitsgründen nicht möglich.

Das offensichtliche Problem sind große öffentliche Schlüssel, die nur einmal verwendet werden können. (vgl. Bernstein/Lange 2017:11) Aus diesem Grund stellte R.C. Merkle 1979 vor 2^k öffentliche Schlüssel zu kombinieren um alle 2^k Signaturen zu überprüfen. Dafür werden 2^k Schlüsselpaare erstellt und die öffentlichen Schlüssel werden an die Blätter eines binären Baums gehängt (vgl. Merkle 1979:41), der k + 1 Ebenen besitzt. In Grafik 1 sehen wir ein Beispiel für einen solchen Baum mit 2^3= 8 Blättern.



Grafik 1: Binärer Baum mit gehashten Schlusselpaaren. [Bernstein/Lange 2017:12]

Diese Schlüssel werden, wie in der Grafik veranschaulicht, kombiniert. Es wird bei den Blättern gestartet und jeweils 2 werden in einem neuen Schlüssel zusammengefasst. So wird durch den ganzen Baum iteriert, bis am Ende die Wurzel $Y_{2^{k+1}-1}$ berechnet wird. (vgl. ebd.) Diese ist der öffentliche Schlüssel des Systems.

Der öffentliche Schlüssel ist jetzt ein einziger Hashwert. Dafür benötigen die Signaturen mehr Informationen, um sie zu verifizieren. Zusätzlich zu den geheimen Schlüsseln X_i werden noch Y_i und alle Kinder der Knoten auf dem Weg von X_i zur Wurzel $Y_{2^{k+1}-1}$ mitgegeben. Soll z.B. X_2 verwendet werden, werden die Werte Y_1, Y_2, Y_9, Y_{13} mitgegeben. So können alle Hashwerte berechnet werden und die Signatur bestätigt werden. (vgl. Bernstein/Lange 2017:11)

Der Nutzen dieses Verfahren besteht darin, dass ein, mit einem öffentlichen Schlüssel versehenes, Dokument versendet werden kann und zu einem späteren Zeitpunkt signiert werden kann. Somit wird der Inhalt des Dokuments im Vorhinein kommuniziert, es gilt rechtlich allerdings erst als bestätigt sobald die Signatur bekanntgegeben wird. Nur der Ersteller des Dokuments kann die passende Signatur wissen. (vgl. Merkle 1979:34)

Hashfunktionen werden in allen Signaturen Systemen verwendet. Standard Hashfunktionen sind nur von Grovers Algorithmus betroffen, dadurch sind sie ein guter Kandidat für Post-Quantum Signaturen. Die Sicherheit von Hashfunktionen ist schon lange erforscht und daher stabil und die Rechenoperationen sind schnell. (vgl. Bernstein/Lange 2017:11)

Es gibt viele Variationen und Verbesserungen, die bessere Einmalsignaturen verwenden, die Signaturengröße verringern oder mehrdimensionale Bäume verwenden. (vgl. ebd.) Ein Verfahren, dass auf XMSS basiert, wird bald von der Internet Research Task Force (IRTF) eingesetzt. Das U.S. National Institute for Standards and Technologie (NIST) hat angedeutet, dass ein neues Signatursystem eingeführt wird, das auf Hashfunktionen basiert. (vgl. ebd.)

Besonders wichtig ist es, nie denselben Schlüssel mehrfach zu benutzen. In komplexeren Systemen, die virtuelle Maschinen, also share signing keys, oder Ähnliches benutzen ist dies ein großes Problem. Für solche Systeme gibt es Lösungen, diese verursachen aber längere Schlüssel oder eine längere Generierungszeit der Schlüssel. (vgl. ebd.)

4. Post-Quantum Kryptographie in der Umsetzung

Einige Organisationen, die sich auf die Standardisierung von Kryptosystemen spezialisiert haben, haben erkannt, dass zu Kryptosystemen, die auch Angriffen von Quantencomputern standhalten können, gewechselt werden muss. Dies ist eine entscheidende Entwicklung, da viele Applikationen voraussetzen, dass alle Partner dieselben Kryptosysteme benutzen.

Die Internet Engineering Task Force (IETF) hatn mit der IRTF die Standardisierung zu hashbasierten Signaturverfahren fast abgeschlossen. NIST haben Kandidaten für Standardisierung angefordert, die Frist lief November 2017 ab und geht jetzt für drei bis fünf Jahre in die Evaluierungsphase. Andere Organisationen, die sich aktuell mit Standardisierungsprozessen von Kryptosystemen beschäftigen, sind ETSI mit dem "quantum-safe" Programm, ISO mit SC27 WG2 und OASIS mit dem KMIP Standard. Außerdem gibt es das EU-H2020 PQCRYPTO Projekt, welches sich mit Forschung, Design und Implementierung von neuen Systemen beschäftigt.

5. Fazit und Ausblick

Abschließend lässt sich sagen, dass Post-Quantum Kryptographie ein relevantes und aktuelles Thema ist und auch noch für einige Zeit bleiben wird. Die Forschung hat viele neue Möglichkeiten aufgezeigt, mit Public-Key Verfahren und Signaturen umzugehen. Manche dieser Ansätze existieren in der Theorie schon länger, würden aber deutlich höhere Kosten in der praktischen Anwendung verursachen und sind daher nicht effizient einsetzbar. Andere Ansätze scheinen leichter umsetzbar zu sein, sind aber weder lange auf dem Markt, noch

wurden sie ausreichend getestet. Zudem besteht die Gefahr, dass sie schon bald geknackt werden könnten. (vgl. Bernstein/Lange 2017: 13) Diese Tatsachen stehen einer Standardisierung auch dieser Kryptosysteme im Wege. Auf diesem Gebiet sind viele Bereiche noch ungeklärt. Daher muss weiterhin geforscht werden. Bevor Lösungen entwickelt werden können, die die Bedürfnisse des Marktes zufriedenstellend erfüllen und zugleich großflächig umsetz- und einsetzbar sind, muss noch viel getan werden.

6. Literaturverzeichnis

Bernstein, Daniel J. und Lange, Tanja (2017): Post-quantum Cryptography - Dealing with the Fallout of Physics Success. In: International Association for Cryptologic Research. 1-20. <https://cr.yp.to/papers/fallout-20170409.pdf>. [Letzter Zugriff am 01.10.2018]

Grover, Lov K. (1996): A fast quantum mechanical algorithm for database search. In: Proceedings. 212-219. <https://arxiv.org/pdf/quant-ph/9605043.pdf>. [Letzter Zugriff am 01.10.2018]

Lamport, Leslie (1979): Constructing Digital Signatures from a One Way Function. In: SRI International Technical Report. <http://lamport.azurewebsites.net/pubs/dig-sig.pdf>. [Letzter Zugriff am 15.10.2018]

McEliece, R. J. (1978): A public-key cryptosystem based on algebraic coding theory. In: The Deep Space Network Progress Report. 42(44). 114-116. <http://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF>. [Letzter Zugriff am 01.10.2018]

Merkle, Ralph Charles (1979): Security, Authentication, and Public Key Systems. Dissertation. In: Stanford Electronics Laboratories. California. <http://www.merkle.com/papers/Thesis1979.pdf>. [Letzter Zugriff am 15.10.2018]

Micciancio, Daniele und Regev, Oded (2009): Lattice-based Cryptography. In: Bernstein,

Shor, Peter W. (1995): Polynomal-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. In: Proceedings. <https://arxiv.org/pdf/quant-ph/9508027.pdf>. [Letzter Zugriff am 30.10.2018]