

Kryptographie – Chancen und Risiken

Johannes Buchmann und Tsuyoshi Takagi

Fachbereich Informatik, Technische Universität Darmstadt

Einleitung

Seit Jahrhunderten benutzen Menschen Kryptographie, um Nachrichten und Dokumente geheim zu halten.

Heutzutage ist die Kryptographie eine Schlüsseltechnik zur Absicherung des Internets und anderer Computernetzwerke. Verschlüsselungsverfahren garantieren die Vertraulichkeit von Nachrichten und Dokumenten. Digitale Signaturen beweisen ihre Authentizität und sorgen für Verbindlichkeit. Die meisten wichtigen Anwendungen wie Internet Browser, Email-Programme und Betriebssysteme sind für den Einsatz von Kryptographie vorbereitet bzw. verwenden diese auch.

Weil Kryptographie für die Computersicherheit so wichtig ist, müssen kryptographische Verfahren nachhaltig sicher bleiben. Aber nachweisbar sichere kryptographische Verfahren sind bis heute unbekannt. Dies gefährdet die Langzeitsicherheit kryptographischer Verfahren.

Dieser Artikel widmet sich der nachhaltigen Sicherheit kryptographischer Verfahren. Unsere These: Es gibt heutzutage kryptographische Verfahren, die ihre Absicherungsaufgaben zuverlässig erfüllen. Aber diese Verfahren können unsicher werden, weil immer wieder neue Angriffe gefunden werden. Es ist darum nötig, die Sicherheit kryptographischer Verfahren genau zu studieren und ihre Schwachstellen zu finden. Außerdem ist es erforderlich, Alternativen zu den heute verwendeten kryptographischen Verfahren bis zur Einsatzbereitschaft vorzubereiten und die Sicherheitsinfrastrukturen so zu konstruieren, dass unsichere Verfahren leicht gegen sichere ausgetauscht werden können. Dies ist eine notwendige Voraussetzung für die nachhaltige Sicherheit von IT-Systemen.

In unserem Artikel erläutern wir zuerst an einem Beispiel, wo kryptographische Verfahren im Internet-Alltag zum Einsatz kommen. Hierbei gehen wir besonders auf das viel benutzte RSA-Verschlüsselungs- und Signaturverfahren ein und zeigen, was man mit diesen Verfahren alles machen kann. Die Sicherheit von RSA wird diskutiert und wir erläutern einige Angriffe auf das RSA-Verfahren. Weiterhin erklären wir, wie ein neuer Computertyp, der Quantencomputer, RSA unsicher machen wird und beschreiben neue Verschlüsselungsverfahren, die voraussichtlich nicht von Quantencomputern angegriffen werden können. Zuletzt erläutern wir noch einmal zusammenfassend, wie nachhaltige kryptographische Sicherheit erreicht werden kann.

Jeder benutzt Kryptographie

Fast jeder, der das Internet benutzt, verwendet Kryptographie. Dazu sei ein kleines Beispiel betrachtet.

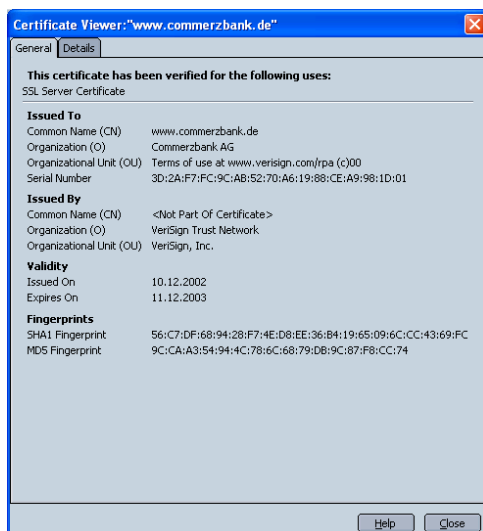
Wer mit dem Netscape Navigator oder dem Internet Explorer die Internetseite www.commerzbank.de öffnet, sieht unten rechts ein kleines geschlossenes Vorhängeschloss. Das bedeutet: die Verbindung zwischen dem Browser und der Commerzbank ist verschlüsselt. Genauere Informationen erhält, wer das kleine Schloss anklickt. Wer dagegen www.google.de

wählt, sieht ein geöffnetes Schloss (Netscape Navigator) oder nichts (Internet Explorer). Die Verbindung ist nicht verschlüsselt.

Wie funktioniert die Verschlüsselung bei der Verbindung zur Commerzbank? Woher bekommt mein Browser den nötigen Schlüssel? Welches Verfahren wird verwendet? Verwendet wird das Public-Key-Verschlüsselungsverfahren RSA und das Protokoll SSL (Secure Socket Layer). Wir erläutern kurz und vereinfacht, wie die Verschlüsselung bei SSL abläuft.

Zuerst erzeugt der Browser einen geheimen Schlüssel für das schnelle Verschlüsselungsverfahren AES (Advanced Encryption Standard), der die Verbindung absichern soll. Dieser Schlüssel wird an die Commerzbank geschickt. Alle Informationen, die von der Commerzbank kommen und alle Nachrichten, die der Kunde an die Commerzbank schickt, werden dann mit diesem Schlüssel verschlüsselt. Aber wieso bleibt der AES-Schlüssel geheim, wenn der Browser ihn an die Commerzbank schickt? Die Antwort, er wird selbst verschlüsselt und zwar mit dem RSA-Verfahren.

Das RSA-Verfahren ist ein Public-Key-Verfahren. Zum RSA-Verschlüsseln braucht der Browser einen öffentlichen RSA-Verschlüsselungsschlüssel. Diesen Schlüssel darf jeder kennen. Man kann ihn nur zum Verschlüsseln benutzen. Die Commerzbank schickt den Verschlüsselungsschlüssel in einem Zertifikat an den Browser. Das Zertifikat kann man sich ansehen, wenn man beim Netscape Navigator zuerst das kleine Schloss anklickt und dann den View-Button.



Beim Internet Explorer genügt es, das kleine Schloss anzuklicken. Zum Entschlüsseln verwendet die Commerzbank einen privaten Entschlüsselungsschlüssel. Diesen Schlüssel hält die Commerzbank natürlich geheim.

Warum werden zwei verschiedene Verschlüsselungsverfahren verwendet, AES und RSA? RSA ist relativ langsam. Eine ganze Internet-Seite mit RSA zu verschlüsseln würde viel zu lange dauern. AES ist dagegen schnell. Aber AES kann man nur verwenden, wenn man vorher einen geheimen Schlüssel ausgetauscht hat. Und dafür braucht man das Public-Key-Verfahren RSA.

SSL wird inzwischen sehr oft eingesetzt. Kauft ein Kunde bei www.amazon.de ein Buch und bezahlt es mit seiner Kreditkarte, so wird die Kreditkartennummer mit SSL vertraulich an Amazon geschickt. Buht ein Kunde bei www.centralticket.de ein Ticket der Centralstation, so wird der Buchungsvorgang mit SSL abgesichert. Es gibt hierfür noch viele andere Beispiele.

Digitale Signaturen

Woher weiß mein Browser, dass er tatsächlich den öffentlichen Schlüssel der Commerzbank hat? Das Zertifikat, das den öffentlichen Schlüssel enthält wurde von VeriSign Inc. digital signiert. Die digitale Signatur kann VeriSign mit einem geheimen Schlüssel erzeugen. Der Browser kann digitale Signaturen von VeriSign mit einem eingebauten öffentlichen Schlüssel von VeriSign verifizieren.

Jeder, der auf die Commerzbank-Seite geht, weiß also: Ich bin mit der Commerzbank verbunden und nicht mit einem Betrüger. Wenn er VeriSign und seinem Browser glaubt. Man sagt: die Seite der Commerzbank ist authentifiziert. Doch ist VeriSign glaubwürdig? Die Firma VeriSign lebt davon, dass stimmt, was sie bestätigt. Wenn bekannt werden würde, dass VeriSign falsche Informationen bestätigt, dann wäre das sehr schlecht für das Ansehen und das Geschäft von VeriSign. Ist der öffentliche Schlüssel von VeriSign korrekt? Das kann der Nutzer überprüfen. Im Netscape Navigator geht das so: Edit -> Preferences -> Privacy and Security -> Certificates -> Manage Certificates -> Authorities -> VeriSign Class 3 Public Primary Certificate -> View. Hier erhält der Benutzer den Fingerprint von VeriSign. Diesen kann er überprüfen indem er bei VeriSign anruft und nach dem Fingerprint fragt. Diesen vergleicht er dann mit dem Fingerprint des Browsers.

Digitale Identitäten

Die Commerzbank identifiziert sich bei jedem Webbrowser mit ihrem Zertifikat. Aber woher weiß die Commerzbank, wer mit ihr verbunden ist? Die Kunden haben meistens noch keine Zertifikate. Daher müssen sie sich mit einer PIN identifizieren, wie überall im Internet.

Teilnehmernr. (7 bzw. 10-stellig):

Bankleitzahl (8-stellig):

PIN (5-stellig):

Einige Beispiele hierfür die Internet-Buchhandlung www.amazon.de, das Internet-Auktionshaus www.ebay.de, der Buchungsservice von www.bahn.de. Bei der Anmeldung müssen die Nutzer ihren Namen, ihre Anschrift und ihre Email-Adresse angeben und eine PIN vereinbaren. Dies ist aufwändig.

Die PIN müssen sich die Nutzer irgendwo aufschreiben. Oder sie müssen immer dieselbe PIN verwenden. Beides ist nicht besonders sicher.

Dabei ist es viel einfacher, wenn jeder Internet-Nutzer ein eigenes Zertifikat, also eine digitale ID wie die Commerzbank, hat. Dann muss bei der Anmeldung nur das Zertifikat übertragen werden und Commerzbank, ebay, Amazon und auch alle anderen wissen, mit wem sie verbunden sind. Außerdem können die Internetnutzer vertrauliche Emails austauschen und digitale Signaturen ausstellen. Und Gebäudezutritt, Zeiterfassung und Computerzugang können gleich mit erledigt werden.

So funktioniert RSA-Verschlüsselung

RSA ist das älteste Public-Key-Verschlüsselungsverfahren. RSA wurde 1978 von Ron Rivest, Adi Shamir und Len Adleman erfunden. Für diese Erfindung bekamen Rivest, Shamir und Adleman im Jahre 2002 den Turing Award, den angesehensten Informatik-Preis der Welt.

Wie erzeugt die Commerzbank ihre RSA-Schlüssel? Sie wählt zwei ungefähr 157-stellige Primzahlen P und Q und berechnet ihr Produkt $N = PQ$. Die Primzahlen P und Q werden so gewählt, dass N eine 1024-Bit-Zahl ist. Zusätzlich erzeugt sie zwei natürliche Zahlen e, d , und zwar so, dass das Produkt ed bei der Division durch $(P-1)(Q-1)$ den Rest 1 läßt. Wie das genau gemacht wird, erklären wir hier nicht. Mehr Details findet man im Buch [Buc03]. Die Zahl d ist der geheime RSA-Schlüssel der Commerzbank. Das Zahlenpaar (e, N) ist der öffentliche RSA-Schlüssel der Commerzbank. N wird auch als RSA-Modul bezeichnet.

Wie verschlüsselt der Browser den AES-Schlüssel? Zuerst schreibt er ihn als Bit-String und ergänzt diesen Bit-String so, dass er die Länge 1024 hat und somit eine Zahl m aus der Menge $\{0, 1, \dots, N-1\}$ darstellt (Padding). Die Zahl m , der Klartext, wird verschlüsselt, indem der Chiffretext $c = m^e \bmod N$ berechnet wird. Hierbei bedeutet $\bmod N$, dass nur der Rest der bei

der Division durch N entsteht betrachtet wird. Der Chiffretext ist also die e -te Potenz mod N von m .

Wie entschlüsselt die Commerzbank c ? Sie muss die e -te Wurzel aus c mod N berechnen. Die Commerzbank kennt den geheimen Schlüssel d und berechnet $m = c^d \text{ mod } N$. Dann schreibt sie m als Binärzahl, entfernt die ergänzten Bits und erhält den AES-Schlüssel.

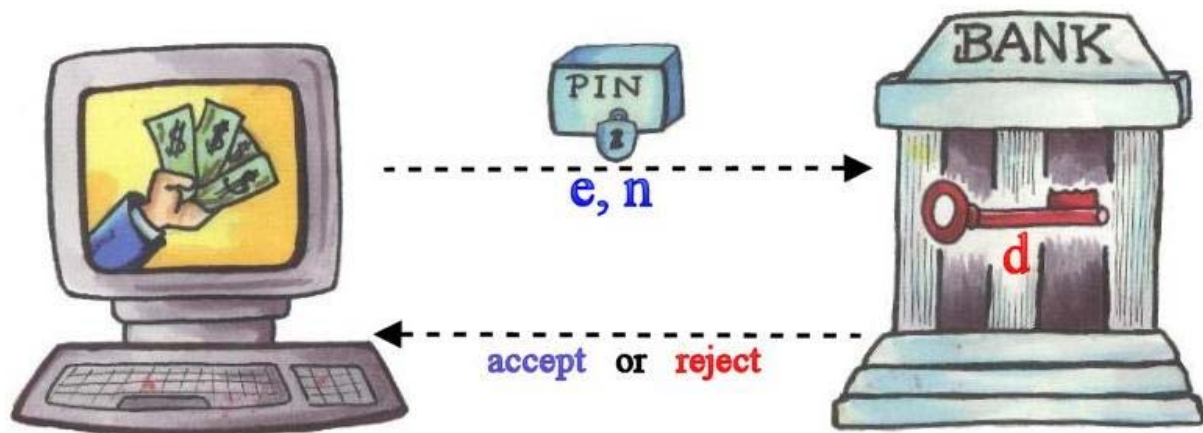
So funktionieren RSA-Signaturen

VeriSign hat auch einen öffentlichen (e, N) und einen geheimen RSA-Schlüssel d . VeriSign signiert das Zertifikat der Commerzbank so: Von dem Zertifikat wird ein digitaler Fingerprint erzeugt. Der wird wie bei der Verschlüsselung verlängert (Padding). Das ergibt eine Zahl x . Die Signatur ist $s = x^d \text{ mod } N$. Wer verifizieren will, dass die Signatur korrekt ist, besorgt sich das Zertifikat von VeriSign, bestimmt x und prüft, ob $x = s^e \text{ mod } N$ ist.

Wieso ist das eine Signatur? VeriSign kennt das geheime d und kann darum die Signatur s berechnen. Ohne Kenntnis von d kann keiner s berechnen. VeriSign berechnet die Signatur s nur, wenn VeriSign den Inhalt des Zertifikats überprüft hat und bestätigen will. Jeder kann kontrollieren, ob die Signatur korrekt ist.

Wieso ist RSA sicher?

Wer N faktorisieren, also aus N die Primfaktoren P und Q berechnen kann, der kann auch den geheimen Schlüssel d herausbekommen. Wenn N eine 1024-Bit-Zahl ist, kann das heutzutage keiner. Wie schwer Faktorisieren heutzutage ist, lässt sich unter www.crypto-world.com nachlesen.



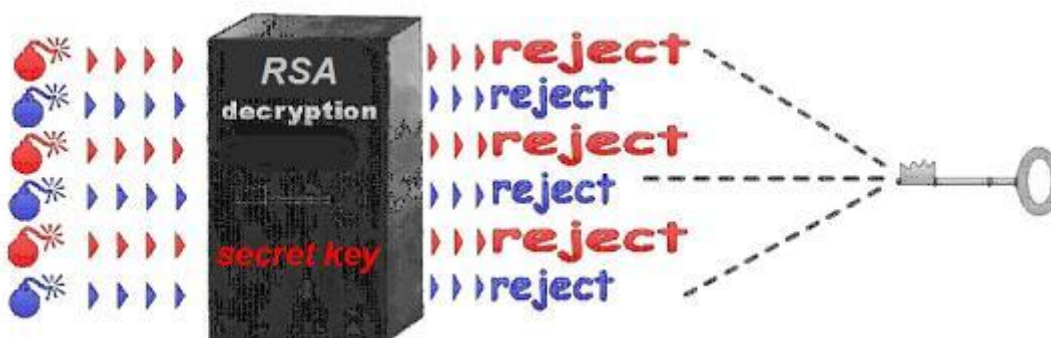
Wer RSA brechen will, also aus dem RSA-Chiffretexten ohne Kenntnis des RSA-Schlüssels etwas über den Klartext erfahren will, braucht nicht unbedingt den RSA-Modul zu faktorisieren. Wir geben ein Beispiel für einen anderen Angriff.

Wir wollen eine PIN mit RSA verschlüsseln. Die PIN ist eine vierstellige Zahl. Der entsprechende Chiffretext ist $c = \text{PIN}^e \text{ mod } N$. Ein Angreifer kann die PIN durch ausprobieren ermitteln. Er verschlüsselt einfach alle möglichen PINs, also 0001, 0002, usw. Sobald er eine PIN gefunden hat, für die der Chiffretext mit m übereinstimmt, weiß er: das ist die gesuchte PIN.

Der beschriebene Angriff wird abgewehrt, indem der Klartext, also die PIN, vor dem Verschlüsseln mit entsprechend gewählten Bits länger gemacht wird. Das Verfahren heißt RSA-OAEP (Optimal Asymmetric Encryption Protocol). OAEP ist in [PKCS] genau beschrieben. Berechnet wird also der Schlüsseltext $c = (\text{OAEP}(\text{PIN}))^e \text{ mod } N$. $\text{OAEP}(\text{PIN})$ ist

die verlängerte PIN. Wenn der Empfänger die PIN entschlüsselt, berechnet er zuerst $OAEP(PIN) = c^d \bmod N$. Dann extrahiert er die PIN, indem er die überflüssigen Bits entfernt. Ein Angreifer kann den Klartext nicht erraten, weil er deutlich länger als die PIN ist, und es viel zu viele mögliche Klartexte gibt. Man kann mathematisch beweisen, dass RSA-OAEP unter plausiblen Voraussetzungen sicher ist.

Ein anderer RSA-Angriff nutzt einen Beschleunigungstrick für die RSA-Entschlüsselung aus. Der Angriff wurde von Boneh und Brumley vorgeschlagen [BB03]. Der Beschleunigungstrick hat die Eigenschaft, dass Nachrichten m , die kleiner sind als der Primfaktor P des RSA-Moduls wesentlich schneller entschlüsselt werden als Nachrichten, die größer als P sind. Der Angreifer schickt verschlüsselte Nachrichten, die nicht das richtige Format haben an einen Webserver, der den Beschleunigungstrick einsetzt. Der Webserver entschlüsselt die Nachricht, erkennt, dass sie nicht das richtige Format haben und weist sie sofort zurück.



Nachrichten, die kleiner als der Primfaktor P sind, weist der Webserver schneller zurück als Nachrichten, die größer als P sind, weil er solche Nachrichten schneller entschlüsseln kann. Der Angreifer kann also feststellen, ob der geheime Primfaktor P größer ist als die von ihm geschickte Nachricht oder kleiner. Wenn der Angreifer seine Nachrichten geschickt wählt, kann er den Primfaktor P herausfinden. Welchen Schutz gibt es vor diesem Angriff? Wer vor der Entschlüsselung den Chiffretext in geeigneter Weise ändert und diese Änderung nach der Entschlüsselung rückgängig macht, kann sich leicht gegen den Angriff schützen.

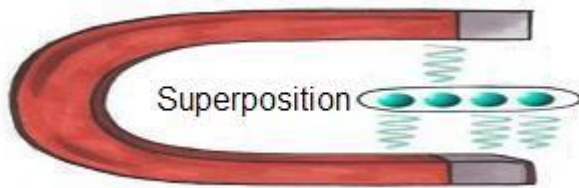
Quantencomputer machen Faktorisieren leicht

Anfang des 20. Jahrhunderts formulierten Physiker wie Max Planck, Niels Bohr, Albert Einstein, Werner Heisenberg, Erwin Schrödinger, Paul Dirac und Max Born die Quantentheorie. Sie revolutionierte die Physik. Im Jahre 1996 zeigte Peter Shor [Sho94], wie unter Ausnutzung der Gesetze der Quantentheorie ein völlig neuartiger Computer gebaut werden kann, für den es sehr leicht ist, RSA-Moduln zu faktorisieren, also das RSA-Verschlüsselungsverfahren zu brechen. Im Folgenden beschreiben wir in Umrissen den Algorithmus von Shor.

Der Algorithmus von Shor zerlegt den RSA-Modul N in seine Primfaktoren P und Q . Der Shor-Algorithmus findet mit dem Quantencomputer Zahlen x und r mit der Eigenschaft, dass N ein Teiler von $x^r - 1$ ist. Aus der Zahlentheorie ist bekannt, dass dann mit hoher Wahrscheinlichkeit eine der Zahlen $x^{r/2} - 1$, $x^{r/2} + 1$, $x^{r/4} - 1$, $x^{r/4} + 1$, $x^{r/8} - 1$, usw. nur noch durch eine der beiden Primfaktoren P oder Q teilbar ist aber nicht mehr durch das Produkt $N = PQ$. Nachdem er r gefunden hat, berechnet der Algorithmus dann sukzessive $\text{ggT}(x^{r/2} - 1, N)$, $\text{ggT}(x^{r/2} + 1, N)$, $\text{ggT}(x^{r/4} - 1, N)$, usw. bis ein Teiler von N gefunden ist. Die Zahlen x und r können auch mit einem klassischen Computer berechnet werden. Er kann zum Beispiel x vorwählen und danach alle Werte für r zwischen 1 und N ausprobieren. Das dauert aber viel

zu lange. Der Quantencomputer kann dagegen die möglichen Werte für r alle gleichzeitig verarbeiten und ist darum viel schneller.

Wie funktioniert ein Quantencomputer? Klassische Computer arbeiten mit zwei Zuständen: 0 oder 1. Sämtliche Informationen sind Kombinationen von Nullen und Einsen. Alle Berechnungen bestehen aus ganz einfachen Berechnungen mit Nullen und Einsen. Quantencomputer arbeiten mit Qubits (Quantenbits). Ein Qubit kann gleichzeitig 0 und 1 sein (Superposition). Wird es gemessen, dann nimmt es einen der Werte 0 oder 1 an. Die 0 ergibt sich mit bestimmter Wahrscheinlichkeit und die 1 auch. Weil Qubits eine Superposition von Zuständen sind, können Quantencomputer viele Rechnungen parallel durchführen.



Wie findet ein Quantencomputer die Zahlen x und r ? Die Zahl x wird festgelegt. Alle möglichen Kandidaten a für r kommen durch Superposition in ein Quantenregister. Der Quantencomputer belegt ein

weiteres Quantenregister mit den entsprechenden Werten $x^a - 1$. Das zweite Register wird gemessen. Damit liegt der Inhalt des zweiten Registers fest. Messung des ersten Registers kann dann nur noch Werte ergeben, die sich um Vielfache von r unterscheiden. Eine Quantenfouriertransformation und eine weitere Messung liefert r . Der Quantencomputer ist deshalb so schnell, weil er alle Werte a und $x^a - 1$ gleichzeitig in die Quanten-Register einbringen kann und weil er die Zahl r schnell extrahieren kann.

Quantensichere Kryptoverfahren

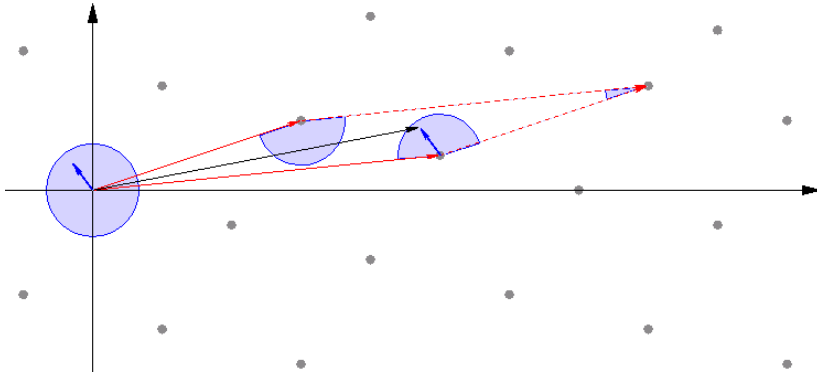
Wir wissen heute noch nicht, ob und wann es Quantencomputer gibt, die groß genug sind, um RSA und die anderen gängigen Public-Key-Verfahren zu brechen. Manche Wissenschaftler denken, dass Quantencomputer einen genauso dramatischen Fortschritt machen werden wie die heute bekannten Computer. Andere glauben, dass die Entwicklung so verlaufen wird wie bei der Kernfusion: Viel Forschungsgeld und wenig Erfolg. Aber eins steht fest. Wenn es solche Quantencomputer gibt, dann braucht man völlig neue Public-Key-Verfahren. Wenn man die erst sucht, wenn große Quantencomputer schon gebaut werden können, ist es zu spät.

Tatsächlich gibt es einige Public-Key-Verfahren, die nach heutiger Überzeugung sicher bleiben, selbst wenn es Quantencomputer gibt. Diese Verfahren sind aber viel zu langsam und auch noch nicht gut genug untersucht.

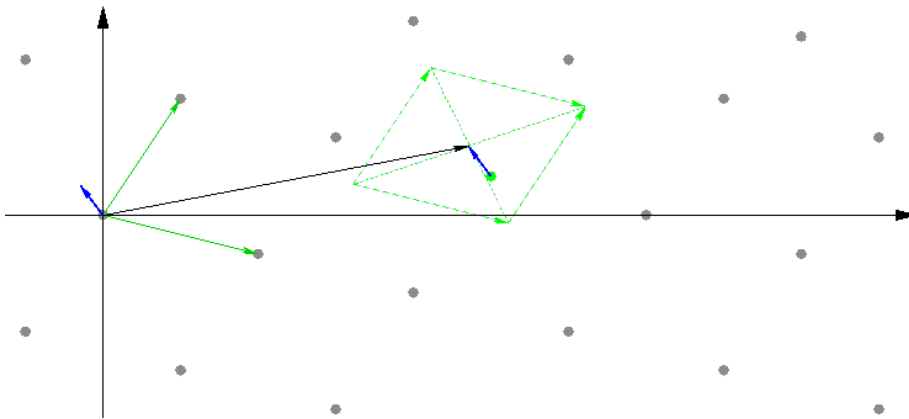
Wir erläutern ein solches Verfahren. Es wurde 2001 von Micciancio erfunden. Der öffentliche Schlüssel in diesem Verfahren ist die Basis (rot gezeichnet) eines Gitters. Man erhält durch Addition und Subtraktion der Basisvektoren alle Gitterpunkte (graue Punkte). In unserer Zeichnung ist das Gitter zweidimensional. Um Sicherheit zu garantieren, muss die Dimension viel höher sein.

Der Klartext, der verschlüsselt werden soll, ist ein kurzer Vektor. Er ist in unserer Zeichnung blau dargestellt. Der blaue Kreis ist der Bereich, aus dem der Vektor gewählt werden darf.

Der Chiffretext (schwarzer Pfeil) ist der Vektor, den man erhält, wenn man den Klartext durch Addition von Gittervektoren in das Parallelogramm verschiebt, das von den Basisvektoren aufgespannt wird.



Der geheime Schlüssel ist eine Basis desselben Gitters mit kürzeren Basisvektoren. Mit dieser Basis ist es leicht, den Gitterpunkt zu finden, der dem Chiffretext am nächsten liegt und daraus durch Differenzbildung den Klartext zu rekonstruieren.



C. Ludwig hat gefunden, dass für eine sichere Verwendung des Micciancio-Verfahrens mindestens die Dimension 800 gewählt werden muss. Der Schlüssel hat eine Größe von 1,6 MByte [Lud03]. Seine Berechnung dauert ca. 50 Stunden. Eine Verschlüsselung braucht 0,29 Sekunden. Eine Entschlüsselung 75 Minuten. Dies ist ein Anfang aber die Ergebnisse sind noch zu schlecht. Hier besteht noch viel Forschungsbedarf.

Nachhaltige kryptographische Sicherheit?

Kryptographie kann heute gut verwendet werden, um multifunktionale digitale Identitäten einzuführen.

Wer nachhaltige Sicherheit will, sollte die heute als sicher geltenden kryptographischen Verfahren einsetzen, ihre Sicherheit sorgfältig untersuchen, Alternativen vorbereiten und Infrastrukturen schaffen, die einen einfachen Austausch unsicherer Komponenten ermöglichen. In diesem Sinne versuchen die Arbeitsgruppen der Autoren einen Beitrag zur nachhaltigen Sicherheit von IT-Systemen zu leisten.

Literatur

- [BB03] D.Boneh and D.Brumley; "Remote Timing Attacks are Practical", 12th Usenix Security Symposium, USENIX, pp.1-14, 2003.
- [Buc03] J.Buchmann; "Einführung in die Kryptographie", 3. Auflage, Springer-Verlag 2003.

- [IT03] T. Izu, T. Takagi; “Exceptional Procedure Attack on Elliptic Curve Cryptosystems”, PKC 2003, LNCS 2274, Springer-Verlag, 2002.
- [Lud03] C.Ludwig; “The Security and Efficiency of Micciancio’s Cryptosystem”, Technical Report, No. TI-7/02, Fachbereich Informatik, TU Darmstadt, 2002.
- [PKCS] Public-Key Cryptography Standards, PKCS \# 1. <http://www.rsasecurity.com/rsalabs/pkcs/>
- [Sho94] P.Shor, “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”, In Proceedings, FOCS ’94, pp.124-134, IEE Press, 1994.
- [VSB⁺01] L.Vadersypen, M.Steffen, G.Breytea, etc; “Experimental Realization of Shor’s Quantum Factoring Algorithm using Nuclear Magnetic Resonance”, Nature, 414, pp.883-887, 2001.

Informationen zum Fachgebiet Theoretische Informatik an der TU Darmstadt

Zentraler Forschungsgegenstand der Arbeitsgruppe ist die Entwicklung sicherer, in der Praxis einsetzbarer Public-Key-Infrastrukturen (PKI). PKIs benötigen sichere kryptographische Verfahren. Wir überprüfen bekannte Krypto-Verfahren auf Sicherheit und Effizienz. Wir entwickeln neuartige Kryptoverfahren, deren Sicherheit auf alternativen mathematischen Problemen beruht und die auch noch sicher sind, wenn die bekannten Verfahren gebrochen werden. Diese Kryptoverfahren stellen wir in der Open Source Bibliothek FlexiProvider (www.flexiprovider.de) zur Verfügung. Eine PKI muss auf die vorhandene Infrastruktur aufsetzen und sich in bestehende Arbeitsabläufe integrieren. Wir entwickeln die PKI-Software FlexiTrust (www.flexisecure.de) mit einem hohen Grad an Flexibilität, die diesen Anforderungen gerecht wird.

Ansprechpartner:

Prof. Dr. Johannes Buchmann
<http://www.informatik.tudarmstadt.de/TI/>
e-mail: buchmann@cdc.informatik.tu-darmstadt.de
Tel: +49-6151-16-3416, Fax: +49-6151-16-6036
Alexanderstraße 10
D-64283 Darmstadt

Informationen zum Fachgebiet Kryptographische Protokolle an der TU Darmstadt

Das Fachgebiet Kryptographische Protokolle (KP) im Fachbereich Informatik befasst sich in Forschung und Lehre mit den Sicherheitsprotokollen in der Theorie und Anwendungen der Kryptographie. Forschungsschwerpunkte des Fachgebiets sind:

- Beweisbare Sicherheit
- Effiziente Kryptographie
- Sichere Implementierung
- Sicherheitsanwendungen

Ansprechpartner:

Dr. Tsuyoshi Takagi, Juniorprofessor
<http://www.informatik.tu-darmstadt.de/KP/>
e-mail: takagi@informatik.tu-darmstadt.de
Tel: +49-6151-16-6167, Fax: +49-6151-16-6036
Alexanderstr.10
D-64283 Darmstadt