Einführung zu RSA

Andreas Zweili, Ismail Cadaroski, Ivan Hörler, Michael Stratighiou

4. März 2017

Inhaltsverzeichnis

1 Einführung

Diese Arbeit gibt eine Einführung zu dem Verschlüsselungsalgorithmus RSA. Anhand von vereinfachten Rechnungen wird die Funktion des Algorithmus veranschaulicht und erklärt. In der Realität sind die verwendeten Zahlen jedoch um ein X-faches grösser. Die nachfolgende Zahl ist 1024 Bit gross. Der Leser kann sich also ungefähr vorstellen wie gross die Zahlen sind, wenn die heutige empfohlene Grösse bei 4096 Bit liegt.

RSA-1024 Primzahl

```
135066410865995223349603216278805969938881475605667027524485143851526510604859533833940287150571909441798207282164471551373680419703964191743046496589274256239341020864383202110372958725762358509643110564073501508187510676594629205563685529475213500852879416377328533906109750544334999811150056977236890927563
```

1.1 Geschichte

Im Jahre 1976 wurde von Whitfield Diffie und Martin Hellman eine Theorie zur Publickey-Kryptographie veröffentlicht [7], in welcher sie ein Konzept namens “Falltür” präsentieren. Dabei handelt es sich um mathematische Probleme, welche in eine Richtung sehr aufwändig und in die andere Richtung viel einfacher zu lösen sind.

Ronald L. Rivest, Adi Shamir und Leonard Adleman wollten nach der Veröffentlichung der Theorie von Herrn Diffie und Herrn Hellman beweisen, dass solche Falltüren nicht existieren. Dabei entdeckten sie jedoch genau solch eine Falltür. Daraus entwickelten sie dann den RSA Algorithmus welchen sie 1977 vorstellten [6]. RSA steht dabei für die Anfangsbuchstaben ihrer Familiennamen.

Im Jahre 2002 erhielten sie den Turing-Award für ihre Arbeit auf dem Gebiet der Kryptographie, welcher oft als Nobelpreis der Informatik bezeichnet wird.

1.2 Verwendung

RSA wird heute in einr Vielzahl von Programmen eingesetzt. Von besonderer Wichtigkeit sind hier folgende Systeme zu erwähnen.

Bankkarten nach dem EMV Standard

Dieser Standard definiert, wie der Chip auf den Karten zu funktionieren hat und wie die Authentifizierung gegenüber den Bankautomaten funktioniert.

HTTPS (TLS und X.509-Zertifikate)

HTTPS garantiert, dass die Zugriffe auf Websites, welche es unterstützen, vor Manipulationen sowie Spionage von Unbefugten geschützt sind. Dies ist insbesondere bei eBanking oder Websiten mit Logins essentiel wichtig. Ansonsten ist es ein Leichtes Konten zu übernehmen.

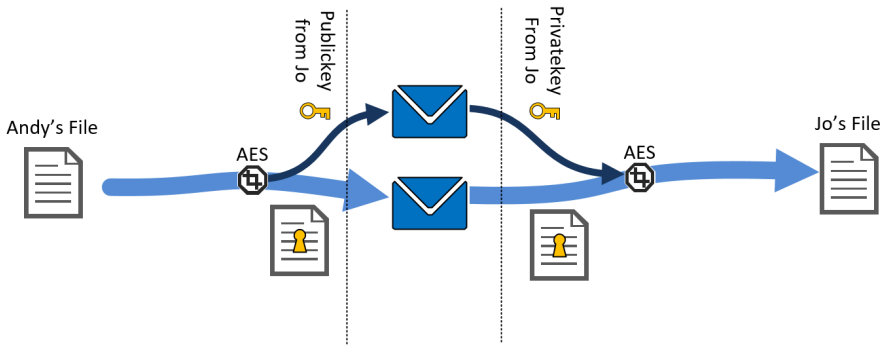
SSH (Secure Shell)

SSH ist ein Protokoll mit welchem man remote auf Unix Systeme Zugreifen kann. Am häufigsten wird es genutzt zur Administrierung von Servern oder zur Übertragung von Dateien.

OpenPGP

OpenPGP ist ein Verschlüsselungsverfahren, welches hauptsächlich bei der Verschlüsselung von Emails verwendet wird. Abseits davon wird es auch zur Signierung von Dateien eingesetzt.

Zusätzlich sollte noch erwähnt werden, dass RSA in den meisten Fällen nicht alleine eingesetzt wird, da die Performance von RSA im Vergleich zu symetrischen Verfahren sehr viel schlechter ist. Deshalb wird RSA oftmals nur zum Schlüsseltausch eingesetzt und eine symetrische Verschlüsselung zum Verschlüsseln der eigentlichen Daten.



2 Öffentlicher und privater Schlüssel

Als erster Schritt muss ein öffentlicher und privater Schlüssel (sozusagen ein Schlüsselpaar), konstruiert werden. Dazu wählen wir zwei zufällige Primzahlen, die wir in unserem Beispiel der Einfachheit halber klein halten und fangen mit der Konstruktion an.

2.1 Schlüsselkontruktion

In den folgenden Seiten berechnen wir :
N = RSA-Modul
p = Primzahl
q = Primzahl
e = Öffentlicher Verschlüsselungsexponent
d = Privater Verschlüsselungsexponent

Wobei $e + N$ den öffentlichen und $d + N$ den privaten Schlüssel bilden.

2.2 Konstruktion *N*

Es werden zwei verschiedene Primzahlen, die der Hersteller des Schlüssels selbst wählt, $p = 7$ und $q = 11$ verwendet und das Produkt aus diesen beiden Werten berechnet. Dieses Resultat N wird ein wichtiger Bestandteil, welchen wir beim Erstellen des privaten, sowie des öffentlichen Schlüssels wieder verwenden werden.

Theorie:

$$N = p \cdot q$$

Beispiel:

$$77 = 7 \cdot 11$$
$$N = 77$$

2.3 Konstruktion *m*

Danach rechnen wir Phi von N, um die Anzahl der teilerfremden Zahlen zu berechnen. Da p und q Primzahlen sind, wissen wir, dass Phi von $p = p - 1$ und Phi von $q = q - 1$ ist und erhalten als Phi v. $N = 60 = m$.

Theorie:

$$\varphi(N) = \varphi(p \cdot q)$$
$$\varphi(N) = \varphi(p) \cdot \varphi(q)$$
$$\varphi(N) = (p - 1) \cdot (q - 1)$$

Beispiel:

$$\varphi(N) = (7 - 1) \cdot (11 - 1)$$
$$\varphi(N) = 60$$
$$m = 60$$

2.4 Konstruktion *e*

Wir bestimmen eine zu $m = 60$ teilerfremde Primzahl, die grösser 1, aber kleiner m sein muss. Wir nehmen in unserem Beispiel $e = 7$.

2.5 Konstruktion *d*

Um die Nachricht zu entschlüsseln, werden wir d brauchen. Da $e \cdot d \bmod m = 1$ ist, muss d aus der Gleichung ausoperiert werden. Dies geschieht mit dem erweiterten, euklidischem Algorithmus und wird in der nachgehenden Tabelle Schritt für Schritt durchgerechnet.

Erweiterter Euklidischer Algorithmus:

A	B	Q	R	S	T	U	V	Berechnung:
m	*e*			1	0	0	1	Startwerte
60	7	8	4	0	1	1	-8	Q = A / B = 60 / 7 = 8 R = A % B = 60 % 7 = 4 S = U_{alt} = 0 T = V_{alt} = 1 U = $S_{alt} - (Q \cdot U_{alt}) = 1 - (8 \cdot 0) = 1$ V = $T_{alt} - (Q \cdot V_{alt}) = 0 - (8 \cdot 1) = -8$
7	4	1	3	1	-8	-1	9	A = B_{alt} B = R_{alt} Q = A / B = 7 / 4 = 1 R = A % B = 7 % 4 = 3 S = U_{alt} = 1 T = V_{alt} = –8 U = $S_{alt} - (Q \cdot U_{alt}) = 0 - (1 \cdot 1) = -1$ V = $T_{alt} - (Q \cdot V_{alt}) = 1 - (1 \cdot -8) = 9$
4	3	1	1	-1	9	2	-17	A = B_{alt} B = R_{alt} Q = A / B = 4 / 3 = 1 R = A % B = 4 % 3 = 1 S = U_{alt} = –1 T = V_{alt} = 9 U = $S_{alt} - (Q \cdot U_{alt}) = 1 - (1 \cdot -1) = 2$ V = $T_{alt} - (Q \cdot V_{alt}) = -8 - (1 \cdot 9) = -17$
	1			2	-17			Ergebnisse

Theorie:	Beispiel:
$S \cdot m + T \cdot e = ggT(60, 7)$	$2 \cdot 60 + (-17 \cdot 7) = 1$
	$120 - 119 = 1$
$T + m \cdot e \bmod m = ggT(60, 7)$	$-17 + 60 \cdot 7 \bmod 60 = 1$
$d \cdot e \bmod m = ggT(60, 7)$	$43 \cdot 7 \bmod 60 = 1$
$e^{ggT(60,7)} \bmod m = d$	$7^{-1} \bmod 60 = 43$

3 Verschlüsselung

Im Beispiel der Schlüsselkonstruktion werden die Variablen *e* und *N* als öffentlicher Schlüssel festgelegt. Dieser wird benötigt, um eine Nachricht für den dafür entsprechenden Empfänger zu verschlüsseln. Mit der daraus resultierenden Zahl, sowie dem privaten Schlüssel, welcher aus den Variablen *d* und *N* besteht, kann die Nachricht wieder entschlüsselt werden.

In unserem Beispiel lautet der private Schlüssel also: 43 + 77
und der öffentliche Schlüssel: 7 + 77

3.1 Der eigentliche Akt der Verschlüsselung

Wollen wir nun eine Nachricht mit dem öffentlichen Schlüssel verschlüsseln, so dass sie nur noch für den Empfänger mit dem entsprechenden privaten Schlüssel zu entschlüsseln ist, gehen wir folgendermassen vor:

Wir kennen die beiden Zahlen des öffentlichen Schlüssels: 7 + 77

Unsere zu verschlüsselnde Nachricht x: 47 (muss kleiner sein als *N*) Wie bereits in einem früheren Kapitel erwähnt sind solche öffentlichen Schlüssel Primzahlen mit mehreren hundert Stellen, somit ist diese Regel im Normalfall irrelevant. Da wir aber in unserem Beispiel keine so grossen Primzahlen verwenden, müssen wir diesen Punkt beachten, um sicherzustellen dass wir auch ein korrektes Ergebnis erhalten.

Die Nachricht wird nun mit folgender Formel verschlüsselt:

Theorie:	Beispiel:
$y = x^e \bmod N$	$y = 47^7 \bmod 77$
	$y = 75$

75 (*y*) ist unsere verschlüsselte Nachricht, welche an den Empfänger übermittelt wird.

4 Entschlüsselung

Um die Nachricht zu entschlüsseln, muss zuerst d errechnet werden, dies geschieht mit hilfe des erweiterten, euklidischen Algorithmus. Diese Berechnung wurde bereits im Kapitel 2.5 erledigt. Unsere gesuchte Zahl lautet demnach 47 (d)

Da nun alle benötigten Variablen bekannt sind, kann die Nachricht mit folgender Formel entschlüsselt werden.

Theorie:	Beispiel:
$x = y^d \bmod N$	$x = 75^{43} \bmod 77$
	$x = 47$

5 Schwachstellen

Obwohl schon einige verkündet haben, die RSA Verschlüsselung geknackt zu haben, ist es bisher noch niemandem gelungen einer Überprüfung stand zu halten. Es gibt aber durchaus realistische Ideen wie die Verschlüsselung gebrochen werden kann, nachgehend stellen wir die wichtigsten vor.

5.1 Brute-force

Die Methode, alle möglichen Primzahlen von $\varphi = (p-1) \cdot (q-1)$ auszuprobieren, gilt als nicht einfacher, als N direkt zu faktorisieren.

5.2 Fakturierung durch die Kenntnis von *N*

Weil die Faktoren von N $\varphi(N)$ ermitteln lassen, kann d ermittelt werden. Die Erfinder von RSA berechneten, anhand eines Algorithmus von Richard Schroeppel und der Annahme, dass ein Annäherungsschritt 1ms benötigt, die Zerlegung von:

Zeichen	**Operationen**	**Zeit**
50	$1.4 \cdot 10^10$	3.9 Stunden
75	$9.0 \cdot 10^12$	104 Tage
100	$2.3 \cdot 10^15$	74 Jahre
200	$1.2 \cdot 10^23$	$3.8 \cdot 10^9$ Jahre

Wobei zu beachten ist, dass:

1. Diese Berechnungen der Entschlüsselungs-Zeiten überholt sind. (stand 1978)
2. 1996 schreibt Prof. Johannes Buchmann von der Universität Saarbrücken, dass ein parallelisiertes Netz von 250 Rechnern auf dem Campusareal für eine 130 stellige Zahl mehrere Wochen benötigt und sich mit mit drei zusätzlichen Dezimalstellen verdoppelt.
3. 2003 veröffentlichte Adi Shamir und Eran Tromer einen technischen Report, wie ein RSA Schlüssel von 1024 bit in unter einem Jahr gebrochen werden kann. [1]
4. Anfang 2017 mutmasste das Forschungs-journal nature.com über den Status der Entwicklung von Quantencomputern und dass deren Schritt aus dem Labor für dieses Jahr Realität werden könnte. Da diesen Rechnergenerationen eine überproportionale Beschleunigung nachgewiesen wurde, kann dies der RSA- Verschlüsselung schaden. [3]

Diese vier Beispiele zeigen auf, wie unvorhersehbar die Standhaftigkeit eines Schlüssels in Bezug auf Zeit ist. Gemäss Adi Shamir lautet die Formel zur Zerlegung von $\varphi(N)$:

$$\varphi = 2 \cdot \text{kgV}\left(\frac{p-1}{2}, \frac{q-1}{2}\right)$$

5.3 Zu kleine Multiplikatorprimzahlen

Da die Sicherheit von RSA darauf beruht, dass die Fakturierung von Primzahlen Zeit benötigt, ist sie auch nur so stark wie die Grösse der Primzahl q die multipliziert mit p den Modulus ergibt. Ist q oder p kleiner als 100 Stellen, wird daraus nicht ein Schlüssel > 10^{200} entstehen und damit die Verschlüsselung zwar schneller geschehen aber sie ist auch gefährdeter durch Brute-force Attacken oder Fakturierung geknackt zu werden.

5.4 Die Riehmann Hypothese

Die Riehmann Hypothese beschreibt ein bisher ungelöstes mathematisches Problem. Sollte sich die Theorie der Riehmann Hypothese bewarheiten könnten daraus Primzahlen abgeleitet werden auf deren Basis die Zerlegung von N einfacher und schneller ausgeführt werden kann.

5.5 Social Engineering 1

Die direkteste Methode an einen Teil oder den ganzen Schlüssel zu gelangen ist das Hacken oder Stehlen. Einerseits kann dies mittels Trojaner oder dann direkt durch entwenden der Schlüssel vom Zielgerät geschehen.

5.6 Social Engineering 2

Eine weitere Methode ist das Täuschen durch nachfolgendes Beispiel: Durch das Abfangen einer Nachricht kann ein Angreifer damit noch nichts anfangen, da die Nachricht mit dem Schlüssel des Empfängers Verschlüsselt ist. Möchte er diese nun entschlüsseln, muss er an den Schlüssel des Empfängers kommen. Dazu kann er die Datei wiederum mit einem ihm bekannten Schlüssel verschlüsseln und sie dem Empfänger erneut und gegebenenfalls unter Verschleierung seiner Identität zustellen. Der Empfänger wird nun die Datei mit seinem Schlüssel entschlüsseln und nichts damit anfangen können da sie immer noch mit dem Schlüssel des Angreifers verschlüsselt ist. Bringt nun der Angreifer durch Geschick den Empfänger dazu ihm diese entschlüsselte, vermeintlich defekte Datei zuzusenden, kann er sie mit seinem Schlüssel entschlüsseln und den Inhalt lesen.

6 Referenzen

Literatur

[1] Eran Tromer Adi Shamir. Factoring large numbers with the twirl device, 2003. http://www.wisdom.weizmann.ac.il/~tromer/papers/twirl.pdf.

[2] Prof. Johannes Buchmann. Faktorisierung grosser zahlen, 1996. http://www.spektrum.de/magazin/faktorisierung-grosser-zahlen/823255.

[3] Davide Castelvecchi. Quantum computers ready to leap out of the lab in 2017, 20017. http://www.nature.com/news/quantum-computers-ready-to-leap-out-of-the-lab-in-2017-1.21239?xing_share=news.

[4] L. Van Houtven. Crypto 101, 2016. https://www.crypto101.io.

[5] Manuel Pöter. Kryptographie - public-key verfahren am beispiel von rsa, 2001/2002. http://www.manuel-poeter.at/tutorials/FBA-V1.0.PDF.

[6] R.L. Rivest und A. Shamir und L. Adleman. A method for obtaining digital signatures and public-key cryptosystems, 1978. https://people.csail.mit.edu/rivest/Rsapaper.pdf.

[7] Whitfield Diffie und Martin Hellman. New directions in cryptographie, 1976. https://www-ee.stanford.edu/%7Ehellman/publications/24.pdf.

[8] Wikipedia. Diffie-Hellman-Schlüsselaustausch — Wikipedia, Die freie Enzyklopädie, 2016. https://de.wikipedia.org/wiki/Diffie-Hellman-Schl%C3%BCsselaustausch.

[9] Wikipedia. Hybride Verschlüsselung — Wikipedia, Die freie Enzyklopädie, 2016. https://de.wikipedia.org/wiki/Hybride_Verschl%C3%BCsselung.

[10] Wikipedia. RSA (cryptosystem) — Wikipedia, Die freie Enzyklopädie, 2016. https://en.wikipedia.org/wiki/RSA_(cryptosystem).

[11] Wikipedia. RSA-Kryptosystem — Wikipedia, Die freie Enzyklopädie, 2016. https://de.wikipedia.org/wiki/RSA-Kryptosystem.

this Document is Typset with

LaTeX