

Einführung zu RSA

Andreas Zweili, Ismail Cadaroski, Ivan Hörler, Michael Stratighiou

31. Januar 2017

Inhaltsverzeichnis

1 Einführung	3
1.1 Geschichte	3
1.2 Verwendung	4
2 Öffentlicher und Privater Schlüssel	5
2.1 Schlüsselkonstruktion	5
2.2 Konstruktion N	5
2.3 Konstruktion m	6
2.4 Konstruktion e	6
2.5 Konstruktion d	6
3 Verschlüsselung	8
3.1 Der eigentliche Akt der Verschlüsselung	8
4 Entschlüsselung	9
5 Schwachstellen	9
5.1 Brute-force	9
5.2 Fakturierung durch die Kenntnis von N	9
5.3 Berechnung von $\varphi(N)$ ohne Fakturierung von N	10
5.4 zu kleine Multiplikator-Primzahlen	11
5.5 Riehmman Hypotese	11
5.6 Social Engineering 1	11
5.7 Social Engineering 2	11

1 Einführung

Diese Arbeit wird eine Einführung zu dem Verschlüsselungsalgorithmus RSA geben. 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

```
13506641086599522334960321627880596993888147560566
70275244851438515265106048595338339402871505719094
41798207282164471551373680419703964191743046496589
27425623934102086438320211037295872576235850964311
05640735015081875106765946292055636855294752135008
52879416377328533906109750544334999811150056977236
890927563
```

1.1 Geschichte

Im Jahre 1976 wurde von Whitfield Diffie und Martin Hellman eine Theorie zu 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 das 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 Nobel Preis für Informatik bezeichnet wird.

1.2 Verwendung

RSA wird heute in eine Vielzahl an 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 das die Zugriffe auf Website welche es unterstützen, vor Manipulationen sowie Spionage von Unbefugten geschützt sind. Dies ist insbesondere bei eBanking oder Websites mit Logins essentiell wichtig. Ansonsten ist es ein Leichtes Konten zu übernehmen.

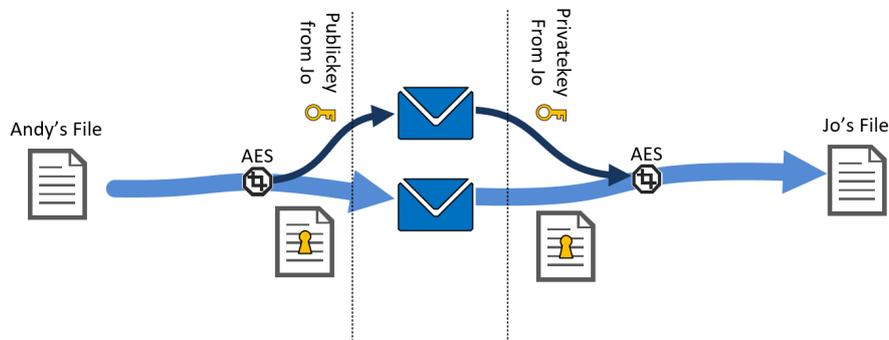
SSH (Secure Shell)

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 das RSA in den meisten Fällen nicht alleine eingesetzt wird da die Performance von RSA im Vergleich zu symmetrischen Verfahren sehr viel schlechter ist. Deshalb wird RSA oftmals nur zum Schlüsseltausch eingesetzt und eine symmetrische 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 2 zwei zufällige Primzahlen die wir in unserem Beispiel zur Einfachheit halber klein halten und fangen mit der Konstruktion an.

2.1 Schlüsselkonstruktion

In den folgenden Seiten berechnen wir :

N = RSA-Modul

p = Primzahl

q = Primzahl

e = Öffentlicher Verschlüsselungsexponent

d = Privater Verschlüsselungsexponent

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

2.2 Konstruktion N

Es werden zwei verschiedenen Primzahlen, die der Hersteller des Schlüssels selbst wählt, $p = 7$ und $q = 11$ verwendet (in Realen Fällen werden kompliziertere Zahlen

gewählt, jedoch halten wir sie hier der Einfachheit halber klein) und das Produkt aus diesen beiden Werten berechnet. Dieses Resultat N wird ein wichtiger Anteil, den wir im Privat und Public Schlüssel brauchen 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 das Phi von $p = p - 1$ und Phi von $q = q - 1$ ist und erhalten als Phi von $N = 60 = m$.

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 $e = 7$ da sie nicht durch 60 teilbar ist und beide den ggT 1 besitzen.

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 erweiter-

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

Theorie:

$$S \cdot m + T \cdot e = \text{ggT}(60, 7)$$

$$T + m \cdot e \text{ mod } m = \text{ggT}(60, 7)$$

$$d \cdot e \text{ mod } m = \text{ggT}(60, 7)$$

$$e^{\text{ggT}(60,7)} \text{ mod } m = d$$

Beispiel:

$$2 \cdot 60 + (-17 \cdot 7) = 1$$

$$120 - 119 = 1$$

$$-17 + 60 \cdot 7 \text{ mod } 60 = 1$$

$$43 \cdot 7 \text{ mod } 60 = 1$$

$$7^{-1} \text{ mod } 60 = 43$$

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 das 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 öffentliche 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 das wir auch ein korrektes Ergebnis erhalten.)

Die Nachricht wird nun mit folgender Formel verschlüsselt:

Theorie:

$$y = x^e \bmod N$$

Beispiel:

$$y = 47^7 \bmod 77$$

$$y = 75$$

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 mithilfe des erweiterten euklidischen Algorithmus. Diese Berechnung wurde bereits im Kapitel ?? erledigt. Unsere gesuchte Zahl lautet demnach 47 (d)

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

Theorie:

$$x = y^d \bmod N$$

Beispiel:

$$x = 75^{43} \bmod 77$$

$$x = 47$$

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 der Code zerbrochen werden kann, nachgehend stellen wir die Wichtigsten Methoden vor.

5.1 Brute-force

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

5.2 Fakturierung durch die Kenntnis von N

Weil die Faktoren von N den $\varphi(N)$ ermitteln lassen kann auch d ermittelt werden. Die Erfinder RSA selbst, berechneten anhand eines Algorithmus von Richard Schroepel und der Annahme das 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

1. Diese Berechnungen der Entschlüsselungs-Zeiten sind überholt. (stand 1978)
2. 1996 schreibt der Prof. Johannes Buchmann von der Universität Saarbrücken das 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 Quanten Computern 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 $\varphi(N)$ lautet:

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

5.3 zu kleine Multiplikator-Primzahlen

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 zerlegt zu werden.

5.4 Riehmann Hypotese

Die Riehmann Hypothese beschreibt ein bisher ungelöstes Mathematisches Problem. Sollte sich die Theorie der Riehmann Hypothese bewarheiten könnten daraus Primzahlen abgeleitet werden auf dessen 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 mit entwenden der Schlüssel vom Zielgerät geschehen.

5.6 Social Engineering 2

Eine weitere Methode ist das Täuschen durch nachfolgendem Beispiel: Durch das Abfangen einer Nachricht kann ein Angreifer damit noch nichts anfangen da sie 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. Dieser 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, 2017. 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%BCsslung.
- [10] Wikipedia. RSA (cryptosystem) — Wikipedia, Die freie Enzyklopädie, 2016. [https://en.wikipedia.org/wiki/RSA_\(cryptosystem\)](https://en.wikipedia.org/wiki/RSA_(cryptosystem)).
- [11] Wikipedia. RSA-Kryptosystem — Wikipedia, Die freie Enzyklopädie, 2016. <https://de.wikipedia.org/wiki/RSA-Kryptosystem>.