Einführung zu RSA

Andreas Zweili, Ismail Cadaroski, Ivan Hörler, Michael Stratighiou

31. Dezember 2016

Inhaltsverzeichnis

1 Einführung

Diese Arbeit wird eine Einführung zu dem Verschlüsselungsalgorithmus RSA geben. Anhand von vereinfachten Rechnungen wird die Funktion des Algorthmus 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. In welcher sie ein Konzept Namens FF-alltürppräsentieren. 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. 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 Websiten mit Logins essentiel wichtig. Ansonsten ist es ein Leichtes Konten zu übernehmen.

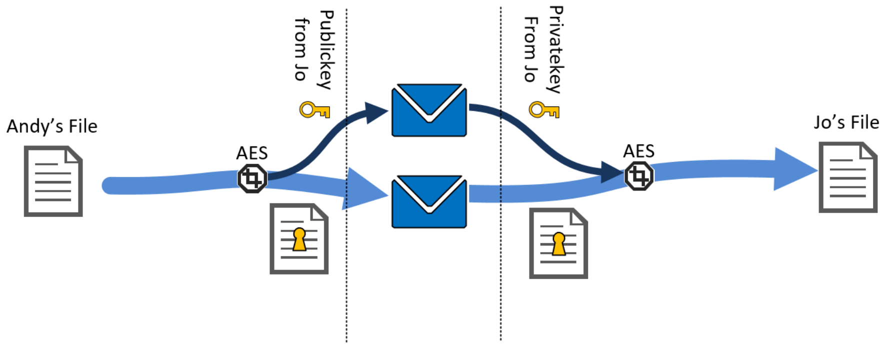
SSH (Secure Shell)

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 das 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 Verschlüsseln

TODO: Sind das wirklich alles Sections? Ich habe sie jetzt mal in Subsections geändert. Ist evtl. eher Fett gemeint? Ismail was hast du hier gemeint? Ist der Titel dieses Kapitels korrekt?

2.1 Schlüsselkontruktion

$N = \text{Privatschlüssel} \quad p = \text{primzahl} \quad q = \text{primzahl}$
$e = \text{Teilerfremder Wert} \quad d = \text{modular inverse}$
$N = 77 \quad p = 7 \quad q = 11 \quad e = 7 \quad d = 43$

Es werden zwei verschiedene Primzahlen 7 und 11 gewählt und das Produkt daraus gerechnet.

Gleichung erstellen nach :

2.2 Konstruktion *N*

$$N = p \cdot q$$

$$77 = 7 \cdot 11$$

$$\mathrm{N} = 77$$

2.3 Konstruktion *m*

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

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

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

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

$$\varphi(N) = 60$$

2.4 Konstruktion *(e)*

Wir bestimmen eine zu $N = 60$ teilerfremde Zahl $e < N$ -> 7

Platzhalter

2.5 Konstruktion *d*

Um *d* zu Konstruieren müssen wir den erweiterten euklidischen Algorithmus mit m anwenden

$$60 = 8 \cdot 7 + 4 \tag{2.5.1}$$
$$7 = 1 \cdot 4 + 3 \tag{2.5.2}$$
$$4 = 1 \cdot 3 + 1 \tag{2.5.3}$$
$$3 = 3 \cdot 1 + 0 \tag{2.5.4}$$

Danach verwenden wir Formelnr. 2.5.3 und formen die Gleichung nach 1 um

$$
\begin{aligned}
&1 = 4 - 1 \cdot 3\\
&1 = 4 - 1 \cdot (7 - 1 \cdot 4)\\
&1 = 4 - 1 \cdot 7 + 1 \cdot 4\\
&1 = (-1) \cdot 7 + 2 \cdot 4\\
&1 = (-1) \cdot 7 + 2 \cdot (60 - 8 \cdot 7)\\
&1 = 2 \cdot 60 - 17 \cdot 7 \text{ (mod)}\\
&1 = 43 \cdot 7 \text{ mod} 60 \text{ } (60 - 17 = 43)\\
&1 = d^e \text{ mod } n \text{ / (Umformen nach } d \text{)}\\
&d = 7^{-1} \text{ mod } 60\\
&d = 43
\end{aligned}
$$

3 Entschlüsselung

Hallo Dies ist ein Test

4 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.

4.1 Brut-force

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

4.2 Fakturierung durch die Kenntnis von *N*

Weil die Faktoren von N den φN ermitteln lassen kann auch d ermittelt werden. Die Erfinder RSA selbst, berechneten anhand eines Algorithmus von Richard Schroeppel 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]

Diese drei Beispiele zeigen auf wie unvorhersehbar die Standhaftigkeit eines Schlüssels in Bezug auf Zeit ist.

Die Formel zur Zerlegung von φN lautet:

$$\varphi = 2 \cdot \text{kgV}\left(\frac{p-1}{2}, \frac{q-1}{2}\right) \tag{4.2.1}$$

4.3 Berechnung von φN ohne Fakturierung von N

Natürlich lässt sich φN auch ohne Fakturierung von N ermitteln wenn d bekannt ist oder ermittelt werden kann. Da d jedoch ein multiplikator von φN ist, ist sein wert nicht leichter zu ermitteln als die Fakturierung von N ist.

4.4 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 > $10^2 00$ entstehen und damit die Verschlüsselung zwar schneller geschehen aber sie ist auch gefährdeter durch Brute-force Attacken oder Fakturierung zerlegt zu werden.

4.5 Gleiche φ N

4.6 Riehmann hypotese

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

4.7 Social Engineering

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.

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

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

[5] 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.

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

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

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

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

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