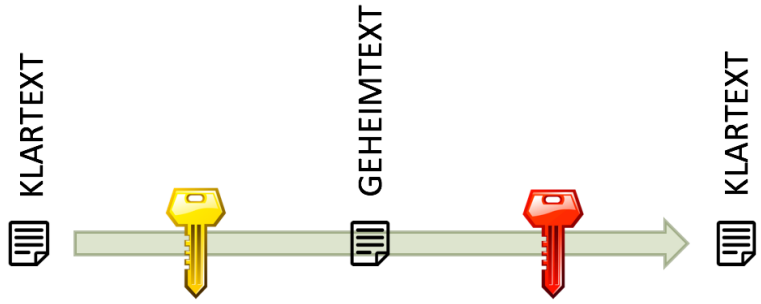
Modul 114: Asymmetrische Verschlüsselung

Die Problematik des Schlüsselaustauschs bei der symmetrischen Verschlüsselung

Bei den symmetrischen Verschlüsselungsverfahren wird beim Ver- und Entschlüsseln der gleiche Schlüssel verwendet. Diesen müssen Alice und Bob zuerst abmachen bzw. austauschen. Wenn nun Alice als erste Nachricht „Hi Bob, von nun an verwenden wir 3DES mit dem Schlüssel A1$-*Klx für die weiteren Nachrichten“ sendet, wird sich Mallory krummlachen. (Er kann ja alle Nachrichten abhören) Was gibt es für Alternativen? Alice könnte die gleiche Nachricht senden und anstelle eines Emails eine SMS verwenden oder persönlich übergeben. (Out of the band key exchange)

In einer Firma mit 20 Personen müssten also n*(n-1)/2 Schlüssel persönlich oder über einen anderen, zweiten Kanal ausgetauscht werden, das sind 190 „Tauschgeschäfte“. Bei 50 Personen sind dies bereits 1'225 Stück. Das ist in der Praxis kaum durchführbar. Ein zentraler Speicher um die Schlüssel abzulegen wäre auch keine gute Idee! (Diebstahl, Abhören bei der Abfrage der Schlüssel)

Die Lösung dieses Problems ist die asymmetrische Verschlüsselung. Es wird ein Schlüssel für die Verschlüsselung und ein anderer Schlüssel für die Entschlüsselung verwendet:



 Mit diesem Schlüssel wird die Nachricht verschlüsselt. Öffentlicher Schlüssel

 Mit diesem Schlüssel wird die Nachricht entschlüsselt. Privater Schlüssel

Welche Eigenschaften müssen diese **asymmetrischen** Schlüssel aufweisen?

- Der öffentliche Schlüssel muss die Nachricht so verschlüsseln, dass *nur* der private Schlüssel sie wieder entschlüsseln kann.
- Vom öffentlichen Schlüssel dürfen keine Rückschlüsse auf den privaten Schlüssel möglich sein
- Der öffentliche Schlüssel kann jedem (inkl. Mallory) zugänglich gemacht werden.
- Der private Schlüssel muss streng geheim gehalten werden!

Wie können wir uns diese beiden Schlüssel vorstellen?

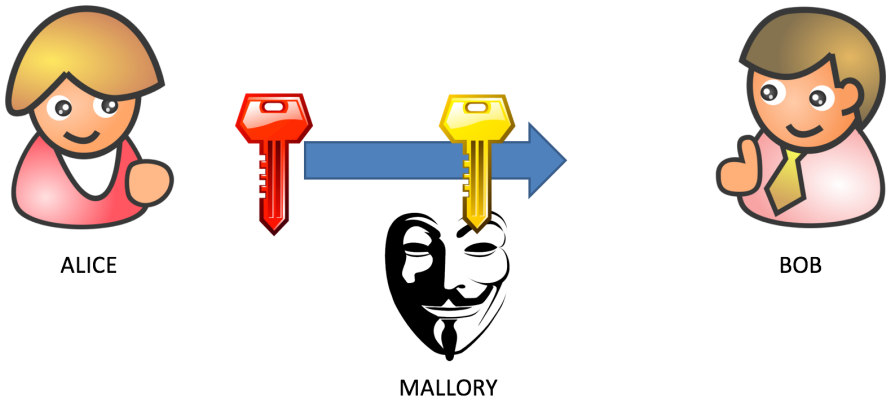
Phase 1: Ein „real-World“ Beispiel könnte ein kleiner Safe sein. Alice legt den öffentlichen Schlüssel auf den offenen Safe. Den privaten Schlüssel trägt sie immer auf sich.

Phase 2: Bob kommt mit seiner Nachricht vorbei, legt diese in den Safe und verwendet zum abschliessen den öffentlichen Schlüssel. Mit diesem kann er den Safe abschliessen aber nicht mehr öffnen!

Phase 3: Alice findet den verschlossenen Safe vor. Mit dem privaten Schlüssel kann sie diesen öffnen und die Nachricht entnehmen. Sie lässt den leeren Safe wieder offen für die nächste Nachricht.

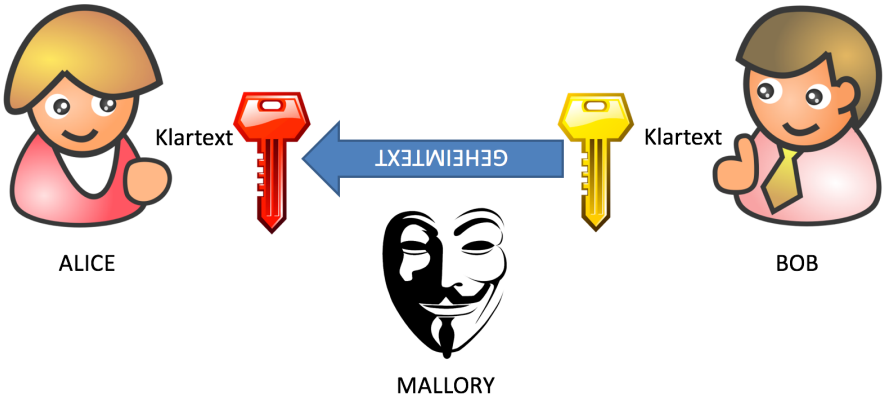
Schema der Kommunikation mittels asymmetrischer Verschlüsselung

Phase A: Schlüsseltausch



Alice generiert den öffentlichen und den privaten Schlüssel. Den öffentlichen Schlüssel sendet sie über den unsicheren Kanal an Bob. Mallory kann diesen auch lesen.

Phase B: Nachricht verschlüsseln und übertragen



Bob verschlüsselt die Nachricht mit dem öffentlichen Schlüssel und sendet diese an Alice. Mallory kann wiederum mithören. Alice entschlüsselt die Nachricht mit dem privaten Schlüssel.

Themen für Fortgeschrittene

Studieren Sie die Schlüsseltausch – Verfahren von „Diffie-Hellman“ und MQV!

RSA Verfahren (Ron Rivest, Adi Shamir, Leonard Adleman) S. 418

Der oben beschriebene Ablauf kann genau so mit dem RSA Verfahren durchgeführt werden. Die mathematischen Anforderungen sind nicht gross, sie beschränken sich auf die Modulo Operation.

Phase A: Schlüsselgenerierung

1) Alice muss die beiden Schlüssel (key pair) generieren. Dazu nimmt sie 2 Primzahlen *p,q*.	p = 5 q = 17
2) Alice errechnet die Modulzahl *n* aus den beiden Primzahlen. p * q	n = p * q n = 5 * 17 = 85

3) Alice errechnet die geheime Modulzahl φ(n) φ = Phi	φ(n) = (p-1) * (q-1) φ(n) = (5 - 1) * (17 – 1) = 64
4) Alice sucht eine Zahl *e*, welche teilerfremd zu φ(n) ist. (Es gibt keine Zahl ausser 1, welche e und φ(n) ohne Rest teilen, oder ggT(e, φ(n)) = 1) Tipp: Jede Primzahl erfüllt diese Bedingung.	e = 3
Der öffentliche Schlüssel besteht nun aus den beiden Zahlen n und e	Öffentlicher Schlüssel: n = 85 e = 3
5) Alice sucht nun den privaten Schlüssel *d*, dazu muss sie eine kleine Suche durchführen: d muss folgende Bedingung erfüllen: d * e = 1 (mod φ(n)) Das bedeutet das Produkt d*e muss bei der Division durch s einen Restwert von 1 ergeben. Alice sucht... Sie testet alle Zahlen d bis der Restwert der Division 1 ergibt.	Ausschnitt aus der Suche nach d

d	e	d*e	ϕ(n)	d*e DIV ϕ(n)	d*e MOD ϕ(n)
39	3	117	64	1	53
40	3	120	64	1	56
41	3	123	64	1	59
42	3	126	64	1	62
43	3	129	64	2	1
44	3	132	64	2	4
45	3	135	64	2	7
46	3	138	64	2	10
47	3	141	64	2	13

129 / 64 = 2 Rest 1

Der geheime Schlüssel besteht nun aus den beiden Zahlen n und d	Privater Schlüssel: n = 85 d = 43

Einfacherer Berechnungsweg von d mit weniger Schritten
Im obigen Ansatz benötigen wir für die Berechnung des privaten Schlüssels 43 Versuche, das ist nicht optimal. Das kommt daher, dass wir viele Berechnungen durchführen, die unmöglich zum Resultat führen können. Wie können wir dies optimieren?
Wir kehren die Formel ein wenig um:

d * e = 1 (mod φ(n)) entspricht d = 1 (mod φ(n)) / e Division durch e

was ist 1 (mod φ(n)) eigentlich? Es entspricht der Zahlenreihe:

φ(n) + 1	65
φ(n) + φ(n) + 1	129
φ(n) + φ(n) + φ(n) + 1	193
φ(n) + φ(n) + φ(n) + φ(n) + 1	257

Zähler	ϕ(n)	(ϕ(n)*Zähler)+1	e	Divsion
1	64	65	3	21.6666667
2	64	129	3	43
3	64	193	3	64.3333333
4	64	257	3	85.6666667

Wenn nun die Zahl rechts dividiert durch e ganzzahlig ist, haben wir den privaten Schlüssel d gefunden! So haben wir den privaten Schlüssel in 2 Schritten (statt 43) gefunden!

Phase B: Nachrichtenaustausch

Bob hat den öffentlichen Schlüssel erhalten und möchte nun eine Nachricht versenden. Als Nachricht nehmen wir eine Zahl stellvertretend für Zeichen oder Bitfolgen.

Klartext *m*	m = 2
Geheimtext *c* = m^e (mod n)	c = 2^3 (mod 85) = 8 (mod 85) = 8

Alice erhält die Nachricht von Bob und entschlüsselt diese mit dem privaten Schlüssel

Geheimtext *c*	c = 8
Klartext m = c^d (mod n)	c = 8^{43} (mod 85) = 2

Mit diesen Werkzeugen können wir bereits einiges üben! ➔***1***

Sicherheit von RSA, welche Chance hat Mallory?

Mallory hört die Leitung ab und bekommt folgende Angaben:

- Öffentlicher Schlüssel bestehend aus n und e
- Nachricht c

Die daraus mögliche Angriffsform heisst „public key only“ Attacke. Mallory kann also selber auch eigene Nachrichten verschlüsseln. (Chosen plaintext) Jetzt muss er „nur noch“ den privaten Schlüssel herausfinden, die verschlüsselte Nachricht damit entschlüsseln und mit der Original – Klartextnachricht vergleichen.

Nehmen wir eine Schlüssellänge von 256 Bit für eine Hochrechnung. Bei einer vollständigen Suche wird im Durchschnitt nach der Hälfte der Versuche die Lösung gefunden, das bedeutet nach 2^{255} Versuchen. Das entspricht einer Zahl welche aus 77 Ziffern besteht.

10^{76}	Anzahl Versuche bis zum Knacken eines 256 Bit Schlüssels
10^{57}	Anzahl Atome im Universum
10^{18}	Alter des Universums in Sekunden

Wie findet er den privaten Schlüssel schneller? Dazu muss lediglich das Primzahlenpaar gefunden werden, welches die Modulzahl n erzeugt n = p * q
Mit diesen beiden Primzahlen kann er mit dem gleichen Verfahren wie es Alice angewandt hat den privaten Schlüssel suchen.

Wie findet er die beiden Primzahlen p und q, welche die Modulzahl bilden?
Dieser Vorgang nennt sich **Faktorisierung**. Für die Faktorisierung gibt es unterschiedliche Ansätze. Wer sich jetzt denkt „es gibt nicht so viele Primzahlen, die Kombination wird sich schnell finden lassen“ der täuscht sich.
Die Firma RSA Laboratories hat einen Wettbewerb ausgeschrieben um Schlüssel durch die Community knacken zu lassen.
http://www.emc.com/emc-plus/rsa-labs/historical/the-rsa-challenge-numbers.htm

Der längste Schlüssel welcher erfolgreich faktorisiert wurde hat die Länge von 768 Bit. Bereits die Schlüssellänge von 1024 Bit scheint im Moment sicher. Das sind lediglich 309 Ziffern!

Wie konnte denn ein 768 Bit Schlüssel geknackt werden? Dazu werden cleverere Verfahren als eine vollständige Suche verwendet. Interessierte können weiterforschen nach Themen wie:

- Low Exponent Attacke
- Quantencomputer
- Twinkle
- Chinesischer Restsatz
- Auch im CrypTool sind 6 Verfahren verbaut, welche die Faktorisierung durchführen

Themen für Fortgeschrittene zum Thema Verschlüsselung

- Pretty Good Privacy (PGP) zum Beispiel mit „Portable PGP“
- Handbuch CrypTool (als PDF hinterlegt), auch als Referenz geeignet
- Verschlüsselung mittels elliptischer Kurven
- Steganographie
- Public Key Infrastructure (PKI)

Aufgabe 1 (RSA Verfahren)

Als Einstieg werden Sie einfache Modulo Berechnungen durchführen:

89 (mod 10) =	45 (mod 8) + 19 (mod 8) =
54 (mod 5) =	45 (mod 8) * 19 (mod 8) =
22 (mod 16) =	(4 * 13) (mod 9) =

Welche Zahlenkombinationen sind teilerfremd?

46 53	
67 45	
45 93	

Nun generieren Sie selber ein RSA Schlüsselpaar mit den Anfangswerten:

p = 7 q = 11 e = 13

Berechnung öffentlicher Schlüssel:

Privater Schlüssel: (Nehmen Sie Excel zur Hilfe, Modulo Rechnung mittels =REST(x;y)

Übernehmen Sie die Rolle von Bob und verschlüsseln Sie die Nachricht m = 2

Übernehmen Sie die Rolle von Alice und entschlüsseln Sie den soeben berechneten Geheimtext! Was fällt Ihnen auf?

Testen Sie die Resultate mit dem Online RSA Simulator!
http://www.mathe-online.at/materialien/Franz.Embacher/files/RSA/

Aufgabe 2 (RSA Angriffe)

Nun holen wir uns Hilfe vom CrypTool! (Teamarbeit)
Sie verkörpern zuerst Alice, welche mit Hilfe des CrypTools ein Schlüsselpaar generiert:
→ Einzelverfahren → RSA Demo → RSA – Kryptosystem
Wählen Sie selber Primzahlen oder lassen Sie diese generieren. Auch den öffentlichen Schlüssel e können Sie selber wählen wenn Sie wollen. Verschlüsseln Sie nun einen Buchstaben.
Notieren Sie auf einem Blatt die Angaben, welche Sie gemacht haben. (oder Printscreen)
Jetzt senden Sie die Informationen übers Internet, dazu notieren Sie auf einem Papier die Angaben

- N (Modulzahl) Bsp: 6683454961
- e (öffentlicher Schlüssel) Bsp: 73
- c (verschlüsselte Nachricht) Bsp: 0289501616

Jetzt tauschen Sie die Zettel aus und übernehmen die Rolle von Mallory!
Versuchen Sie den übermittelten Buchstaben zu knacken, als Hilfestellung können Sie die Funktion „Faktorisieren einer Zahl“ innerhalb der RSA Demo nutzen.

Informieren Sie sich über Primzahlen und machen Sie Experimente auf der folgenden Webseite:
http://www.arndt-bruenner.de/mathe/scripts/primzahlen.htm
Lesen Sie auch den Abschnitt „Sicheres Homebanking mit 128 Bit?“