Ausgeschriebene Informatik-Unterrichtsmitschrift (inklusive Hausaufgaben) vom 28.02.07 V.2.

Valentina Tamer

RSA-Verschlüsselung

Legende

M (message) = Nachricht im Klartext
p, q = (sehr große) Primzahlen
N = öffentliches Schloss (N= p*q) [public key]
e = (halb)öffentlicher Schlüssel zum Codieren für den gilt:
1<e<N; e ≠ Teil von N: e ≠ p ; e ≠ q; e ≠ Teil von (p-1)(q-1)
d (decypher) = private key zum Decodieren
C (cypher) = verschlüsselte Nachricht

Formeln

Verschlüsseln:
C= M^e mod N
[a mod b= 1
(a-1) mod b = 0]

Entschlüsseln:
M= C^d mod N

Berechnung von N:
p*q= N

Berechnung von d:
e*d mod (p-1)(q-1)=1
⇔
(K*Z+1):e = d

Z= (p-1)(q-1)
K*Z= Vielfaches von Z (K= Anzahl)

Ungeschickte Wahl von p und q

1. Zu kleine Primzahlen
2. Eine sehr kleine und eine sehr große Primzahl
3. Zwei nah beieinander liegende Primzahlen

Gründe:

1. Wenn jemand den private key zum Entschlüsseln der Nachricht haben will, muss er N mit der Primfaktorzerlegung zu p und q machen (da man sie zur Berechnung des keys braucht). Wenn die Person beim Ausprobieren mit 2 anfängt und dann immer höher geht, findet er recht schnell die beiden Primfaktoren.
2. Dasselbe gilt hierbei, da man, wenn man die eine (Kleine) hat, die andere automatisch auch kennt (durch ganz kurzes Berechnen)
3. Bei der Primfaktorzerlegung kann man, anstatt bei der kleinsten möglichen Primzahl, auch bei der Größten anfangen, die die Wurzel von N ist. (Sobald man beim Versuchen alle Primzahlen zwischen 2 und dieser Wurzel ausprobiert hat und zu keinem Ergebnis kommt, muss der public key N falsch angegeben und selbst eine Primzahl sein)
   Wenn der „Spion“ jetzt also bei der Wurzel anfängt und dann rückwärts geht, kommt er bei zwei sich sehr ähnlichen Primzahlen auch sehr schnell auf das Ergebnis, da sie nicht groß von der Wurzel abweichen.

…und wie rechnet man mit Nachrichten eigentlich?
Da man mit Buchstaben, die nicht für Variablen stehen, keine mathematischen Rechnungen anstellen kann, gibt es mehrere Möglichkeiten, sie sozusagen in Zahlen umzuwandeln, z.B.:
A= 01
B= 02
C= 03
…
Dieses Verfahren werden wir in weiteren Beispielen verwenden.

Öfter benutzt wird aber der **ASCII**:
American
Standard
Code for
Information
Interchange,
bei dem das lateinische Alphabet bei 65 anfängt anstatt mit 01,
also
A= 65
B= 66
C= 67
…
(Hier gibt es für alle erdenklichen Zeichen eine Zahl, deswegen musste das System um einiges erweitert werden (auf doppelte Größe), als erst alle Zeichen des amerikanischen Sprachraums integriert waren (allein für Deutsch ü, ä, ö, ß; für Französisch, Griechisch usw. noch einiges mehr))

Berechnung von d
(K*Z+1):e= d

Beispiel:
e= 7
p= 3
q= 5

					(x mod 7 = 0?)
K		Z		x	x mod e = 0?
2	*	8	+1	=17	nein
3	*	8	+1	=25	nein
4	*	8	+1	=33	nein
5	*	8	+1	=41	nein
6	*	8	+1	=49	ja

(K*Z+1): e = 49:7 = 7

d= 7

Beispiel einer vollständigen Rechnung:
Ermitteln Sie N, d und C

Entschlüsseln Sie C zu M

Vorgehensweise: In Zweierblöcken

p= 13
q= 11
e= 3

W	I	L	D	S	C	H	W	E	I	N
23	09	12	04	19	03	08	23	05	09	14

V	O	R	A	U	S
22	15	18	01	21	19

N= p*q = 13*11 = 143

C= M^e mod N
C= 23^3 mod 143 = 057
[Bemerkung: C kann höchstens N-1 sein, deswegen muss die Zahl mit so vielen Nullen vorne ergänzt werden, bis sie so viele Stellen wie N hat, in diesem Fall drei.]

d:
Z= (p-1)(q-1) = 120

(K*Z+1):e

K		Z		x	(x mod 3 = 0?) x mod e = 0?
2	*	120	+1	=241	nein
3	*	120	+1	=361	nein
.					
.					
.					
18	*	120	+1	=2161	nein
19	*	120	+1	=2281	nein
20	*	120	+1	=2401	nein

-> Anscheinend wurde bei der Auswahl der Keys ein Fehler begangen:
Der Grund liegt wohl beim Z, das durch e teilbar ist, was bedeutet, dass es K-mal immer glatt durch e geteilt werden kann, und mit 1 addiert niemals dazu imstande sein kann.
Also haben wir raus gefunden: e ≠ Teiler von Z[=(p-1)(q-1)]

Wählen wir also eine vollständig funktionierende Möglichkeit:

Ermitteln Sie N, d und C

Entschlüsseln Sie C zu M

Vorgehensweise: In Zweierblöcken (damit die Zahlen nicht zu groß für unsere Rechner werden)

p= 7
q= 3
e= 5
(Die Zahlen sollten größer gewählt werden, übersteigen aber momentan unsere Möglichkeiten)
Nachricht= Horologie

N= p*q = 7*3 = 21

d:
Z= (p-1)(q-1)= 12

				x mod e = 0?
K		Z	x	x mod 5 = 0?
2	*	12 +1 =	25	ja

d= (K*Z+1):e = 5

M=

H	O	R	O	L	O	G	I	E
09	15	18	15	12	15	08	10	05

M= 09 15 18 15 12 15 08 10 05

C= M^e mod N

C= 9^5 mod 21 = 18	C= 15^5 mod 21 = 15	C= 8^5 mod 21 = 08
C= 15^5 mod 21 = 15	C= 12^5 mod 21 = 03	C= 10^5 mod 21 = 19
C= 18^5 mod 21 = 09	C= 15^5 mod 21 = 15	C= 5^5 mod 21 = 17

C= 18 15 09 15 03 15 08 19 17

M= C^d mod N

M= 18^5 mod 21 = 09	M= 15^5mod 21 = 15	M= 8^5mod 21 = 08
M= 15^5mod 21 = 15	M= 3^5mod 21 = 12	M= 19^5mod 21 = 10
M= 9^5mod 21 = 18	M= 15^5mod 21 = 15	M= 17^5mod 21 = 05

M= 09 15 18 15 12 15 08 10 05 = HOROLOGIE

Vigenère-Verschlüsselung

M= Wildschwein voraus
Code= Kuehlerhaube

Was zutun ist:
Es wird die Stellenzahl, die der n-te Buchstabe des Codeworts im Alphabet hat, genommen, um 1 verringert, und zur Stellenzahl, die der n-te Buchstabe des Klartexts im Alphabet hat, addiert. Das Summe ist die Stellenzahl, an der der verschlüsselte Buchstabe steht.

Klartext	**WILDSCHWEINVORAUS**
Code	**KUEHLERHAUBEKUEHL**
Verschlüsselung	**GTPKDGYDECOZYLEBD**

Vorteile bei Vigenère:
Die Entschlüsselung ist schwerer als bei Caesar, da man dieses System durch ein Codewort auf jeden einzelnen Buchstaben im Wort und nicht das ganze Alphabet anwendet.
Dadurch ist das Problem der erkennbaren Buchstabenhäufigkeit in der deutschen Sprache behoben.

Nachteile bei Vigenère:
Es ist immer noch ein symmetrisches Verfahren, der key könnte also abgefangen und benutzt werden.