

Kryptographie und Komplexität



Einheit 4.4

Semantische Sicherheit



1. Sicherheit partieller Informationen
2. Das Verfahren von Rabin
3. Sicherheit durch Randomisierung

Mehr als nur Sicherheit des Schlüssels

● Vollständiger Bruch des Verfahrens

- Angreifer ist in der Lage, den geheimen Schlüssel zu bestimmen
- Danach kann jede beliebige Nachricht vom Angreifer genauso effizient entschlüsselt werden wie vom vorgesehenen Empfänger

● Partieller Bruch des Verfahrens

- Angreifer kann mit einer gewissen Wahrscheinlichkeit Schlüsseltexte in akzeptabler Zeit dechiffrieren, ohne den Schlüssel kennen zu müssen

● Extraktion von Teilinformationen

- Angreifer kann aus Schlüsseltext spezifische Teilinformationen über den Klartext extrahieren
- Äquivalent zum Ziel, Chiffrierung eines Klartextes von einer beliebigen Strings zu unterscheiden (mit Wahrscheinlichkeit über 50%)
- Ein System ist **semantisch sicher**, wenn dies nicht möglich ist

- **Steckt Klartextinformation im Schlüsseltext?**

- z.B.: *Ist Klartext x eine gerade Zahl oder größer als $n/2$?*
- Information über einzelne Klartextbits kann sehr wertvoll sein und z.B. Antworten auf wichtige Ja/Nein Entscheidungen liefern

- **Nicht jede Information kann geheim bleiben**

- z.B. ändert RSA Verschlüsselung den Wert des Jacobi Symbols nicht
Ist $y = x^e \bmod n$ so ist $(\frac{y}{n}) = (\frac{x^e}{n}) = (\frac{x}{n})^e = (\frac{x}{n})$, da e ungerade ist

- **Extraktion von Teilinformation ist bei RSA genauso schwer wie partieller Bruch**

- Das Problem der vollständigen Entschlüsselung von RSA ist reduzierbar auf die Berechnung von **parity** bzw **half** (Beweis folgt)

- **RSA Sicherheit gegen partiellen Bruch ungewiß**

- Es gibt keinen Beweis, daß RSA genauso schwer wie Faktorisierung ist

Genauso schwer wie partieller Bruch

• Breche Schlüsseltexte mit ‘Orakel’ half

- Sei $\text{half}(y) = \begin{cases} 1, & n > x > n/2 \\ 0, & x < n/2 \end{cases}$ und $\text{parity}(y) = \begin{cases} 1, & x \text{ ungerade} \\ 0, & x \text{ gerade} \end{cases}$ für $y = e_K(x)$
- **half** und **parity** sind **gleich schwer** zu bestimmen
 - half**(y) liefert das erste ‘Bit’ des Klartextes, **parity**(y) das letzte
- Wegen $(2x)^e \bmod n = 2^e \cdot x^e \bmod n$ liefert **half**($2^e \cdot y$) das zweite ‘Bit’, **half**($2^{2e} \cdot y$) das dritte, bis nach $\|n\|$ Iterationen der Klartext feststeht

• Verfahren berechnet x durch Binärsuche

- Setze $[l..r] := [0..n]$
- Für $i := 0.. \|n\|$ setze $[l..r] := [(l+r)/2..r]$, falls **half**($2^{i \cdot e} \cdot y$) = 1
 $[l..r] := [l..(l+r)/2]$, falls **half**($2^{i \cdot e} \cdot y$) = 0
- Ergebnis nach $\|n\|$ Schritten ist $x = \lfloor r \rfloor$

‘Sichere’ Modifikation des RSA Verfahrens

● Schlüsselerzeugung

- Wähle zwei großer Primzahlen p und q , wobei $p, q \equiv 3 \pmod{4}$
(Zusatzbedingung nicht nötig, erleichtert Dechiffrierung)
- Setze $n = p * q$ und lege n offen, halte p und q geheim

● Verschlüsselungsverfahren

- Jeder Textblock wird als Binärdarstellung einer Zahl x interpretiert
- Verschlüsselung wird Quadrieren modulo n : $e_K(x) = x^2 \pmod{n}$
- Entschlüsselung wird Quadratwurzel modulo n : $d_K(y) = \sqrt{y} \pmod{n}$
(Nichttriviale Operation, wenn p, q nicht bekannt)

● System nur von theoretischem Interesse

- Ein partieller Bruch des Systems ist genauso schwer wie Faktorisierung
- Nach heutigem Stand der Technik semantisch sicher

- **Die Gleichung $x^2 \equiv y \pmod{p \cdot q}$ hat 4 Lösungen**

- y ist quadratischer Rest modulo p und modulo q
- Es folgt $y^{(p-1)/2} \equiv 1 \pmod{p}$ und $y^{(q-1)/2} \equiv 1 \pmod{q}$
- Damit $(\pm y^{(p+1)/4})^2 = y^{(p+1)/2} \equiv y \pmod{p}$ (analog für q)
- Wegen $p, q \equiv 3 \pmod{4}$ sind $\pm y^{(p+1)/4}$ und $\pm y^{(q+1)/4}$ wohldefiniert
- Nach dem chinesischen Restsatz hat jedes Paar von Kongruenzen
 $x \equiv \pm y^{(p+1)/4} \pmod{p}$ und $x \equiv \pm y^{(q+1)/4} \pmod{q}$
jeweils eine eindeutige Lösung modulo n
- Verschlüsselung ist nicht injektiv

- **Bestimmung des Klartextes**

$\mathcal{O}(\|n\|^3)$

- Berechne $y_p := q^{-1} \pmod{p}$ und $y_q := p^{-1} \pmod{q}$ a priori
- Berechne $x_i := \pm q \cdot y_p \cdot y^{(p+1)/4} \pm p \cdot y_q \cdot y^{(q+1)/4} \pmod{n}$
- Wähle nach Übertragung in Text den besten der vier Kandidaten aus
Bei großen Blöcken gibt es i.a. nur einen einzigen sinnvollen

Nichtdeterministische Dechiffrierung ist Basis für Sicherheit des Verfahrens

● Schlüsselerzeugung

- Wähle $p = 944123$ $q = 944147$
- Dann ist $n = 891390898081$
- Entschlüsselungskoeffizienten sind $y_p := p^{-1} \bmod q = 511400$
und $y_q := q^{-1} \bmod p = 432734$

● Verschlüsselung

- Alice verschlüsselt Klartext $x = 730581888230$
- Ergebnis ist $y = x^2 \bmod n = 408240297532$

● Entschlüsselung

- Bob berechnet $y^{(p+1)/4} \bmod p = 568209$, $y^{(q+1)/4} \bmod q = 576202$
- Bob berechnet $\pm q \cdot y_p \cdot y^{(p+1)/4} \pm p \cdot y_q \cdot y^{(q+1)/4} \bmod n$ und erhält
 $160809009851, 532577267684, 358813630397, 730581888230$
- Die letzte der vier Lösungskandidaten ist der Klartext

Berechnung des Klartextes x aus $y = x^2 \bmod n$ ist genauso schwer wie die Faktorisierung von $n = p \cdot q$

• Faktorisierung liefert den Klartext

- Wenn Eve n faktorisieren kann, dann hat sie alle Informationen, die im Rabin Verfahren zur Entschlüsselung erforderlich sind

• Quadratwurzelberechnung liefert Faktorisierung

- Wähle $a \in \{1 \dots n-1\}$ zufällig
- Ist $g = \gcd(a, n) \neq 1$ dann ist g echter Teiler von n
- Ansonsten berechne eine Quadratwurzel z von $x^2 \bmod n$
- Wenn $z \not\equiv \pm x \bmod n$ so ist $\gcd(z-x, n)$ echter Teiler von n

Für z gibt es vier Möglichkeiten

$$\gcd(z-x, n) = \begin{cases} n & z \equiv x \bmod p \wedge z \equiv x \bmod q \\ p & z \equiv x \bmod p \wedge z \equiv -x \bmod q \\ q & z \equiv -x \bmod p \wedge z \equiv x \bmod q \\ 1 & z \equiv -x \bmod p \wedge z \equiv -x \bmod q \end{cases}$$

Da x zufällig ist, wird ein Teiler mit Wahrscheinlichkeit $1/2$ gefunden

Liefert **“RP” Algorithmus zur Faktorisierung von n**

● Unterscheidbarkeit von Schlüsseltexten

- Gegeben Verschlüsselungsfunktion f , zwei Klartexte x_1, x_2 und einen Schlüsseltext y mit der Eigenschaft $y=f(x_i)$ für ein i .
Bestimme i mit einer Fehlerwahrscheinlichkeit weniger als 50%
- Allgemeinste Form der Extraktion von Information aus Schlüsseltexten
- Kann ein Angreifer Schlüsseltexte nicht in polynomieller Zeit unterscheiden, so kann er keinerlei Information über den Klartext aus einem Schlüsseltext extrahieren
- Semantische Sicherheit $\hat{=}$ Nichtunterscheidbarkeit von Schlüsseltexten

● Nichtunterscheidbarkeit braucht Randomisierung

- Wenn die Verschlüsselungsfunktion f deterministisch ist, kann ein Angreifer $f(x_1)$ und $f(x_2)$ in polynomieller Zeit ausrechnen
- Ein deterministisches Verfahren kann nicht semantisch sicher sein
- Moderne Versionen von RSA ergänzen Zufallskomponenten

● Randomisierung der Verschlüsselungsfunktion f

- Verschlüsselung: $(y_1, y_2) := e_K(x) = (f(r), G(r) \oplus x)$
- Entschlüsselung: $d_K(y_1, y_2) := G(f^{-1}(y_1)) \oplus y_2$
 - r beliebige Zufallszahl
 - G ‘zufallstreue’ Expansionsfunktion (dargestellt als riesige Tabelle)
 - f Einwegfunktion (nicht in polynomieller Zeit umkehrbar)

● Intuitive Begründung der Sicherheit

- Um Information über x zu erhalten, braucht man Information zu $G(r)$
- $G(r)$ kann nur berechnet werden, wenn r vollständig bekannt ist
 - Teile von r reichen nicht aus, um Teile von $G(r)$ zu bestimmen
- r kann nur bestimmt werden, wenn $f^{-1}(y_1)$ vollständig berechnet wird
 - $f^{-1}(y_1)$ kann nicht in polynomieller Zeit berechnet werden

Detaillierter Beweis siehe Stinson §5.9.2

● Einfache Variante

- Verschlüsselung: $(y_1, y_2) := e_K(x) = (r^e \bmod n, G(r) \oplus x)$
- Entschlüsselung: $d_K(y_1, y_2) := G(y_1^d \bmod n) \oplus y_2$

● PKCS# Standard

- Verschlüsselung: $y := e_K(x) = ((x \oplus G(r)) \circ (r \oplus H(x \oplus G(r))))^e \bmod n$
 $r \in \{0, 1\}^k$ beliebige Zufallszahl
 $G: \{0, 1\}^k \rightarrow \{0, 1\}^l$ Expansionsfunktion
 $H: \{0, 1\}^l \rightarrow \{0, 1\}^m$ Kompressionsfunktion
Klartextlänge $\|x\| = l$, RSA Blocklänge $\|n\| = l+k+1$
Klartext wird zu $y_1 = x \oplus G(r)$ randomisiert
Zufallszahl wird zu $y_2 = r \oplus H(y_1)$ maskiert
- Entschlüsselung: Berechne $y_1 \circ y_2 := y^d \bmod n$
Bestimme $r = H(y_1) \oplus y_2$ und berechne $x = y_1 \oplus G(r)$

- **Ältestes/bedeutendstes Public-Key Verfahren**

- Ver-/Entschlüsselung ist Potenzieren mit e bzw. d modulo n
- Rahmenbedingung $d = e^{-1} \bmod \varphi(n)$
- n zusammengesetzt aus zwei großen Primzahlen p und q
- n, e liegt offen, d, p und q bleiben geheim

- **Schlüsselerzeugung benötigt Primzahltests**

- Schnellste Verfahren sind probabilistisch (Miller-Rabin)

- **Sicherheit basiert auf Faktorisierungsproblem**

- Schlüsselangriff äquivalent zu Lösung des Faktorisierungsproblems
- Faktorisierung ist subexponentiell $\mathcal{O}(e^{1.92 \cdot \|n\|^{1/3} \cdot (\log(\|n\|))^{2/3}})$
- Sicherheit nur noch für Schlüsselgröße über 1024 Bit
- Semantische Sicherheit nur mithilfe von Randomisierung