Kryptografie und -analyse, Fragenkatalog

Dennis Rössel, Henry Haustein

Wir haben die Schlüsselverteilung in symmetrisch und asymmetrisch unterteilt. Was bedeutet das und was sind die Vorteile und Nachteile? Authentikations- und Konzelationssysteme?

Bei symmetrischen Verschlüsselungsverfahren wird der gleiche Schlüssel zum Ver- und Entschlüsseln verwendet. Dieser Schlüssel muss aber vorher über einen unsicheren Kanal ausgetauscht werden, was es Angreifern ermöglicht den Schlüssel abzufangen und selber die Kommunikation zu entschlüsseln. Allerdings ist die Performance symmetrischer Systeme sehr gut.

Bei asymmetrischen Verschlüsselungsverfahren gibt es 2 Schlüssel pro Teilnehmer: einen öffentlichen Schlüssel zum Verschlüsseln und einen privaten Schlüssel zum Entschlüsseln. Der öffentliche Schlüssel ist öffentlich, sodass jeder eine Nachricht für eine Person verschlüsseln kann, aber nur derjenige mit dem privaten Schlüssel diese wieder entschlüsseln kann. Es gibt also kein Problem beim Schlüsselaustausch, allerdings ist die Performance deutlich schlechter.

Bei symmetrischen Authentikationssystemen berechnet man für eine Nachricht eine MAC und der Empfänger berechnet aus der Nachricht auch die MAC und vergleicht mit der erhaltenen MAC.

Bei asymmetrischen Authentikationssystemen verschlüsselt der Sender die Nachricht mit seinem privaten Schlüssel, der Empfänger kann dann die verschlüsselte Nachricht mit dem öffentlichen Schlüssel entschlüsseln und weil nur der Besitzer des privaten Schlüssels diese Signatur berechnen konnte, ist klar, von wem die Nachricht kam.

Was besagt das Prinzip von Kerkhoff?

Die Sicherheit eines Verfahrens darf nicht von der Geheimhaltung des Verfahrens abhängen, sondern nur von der Geheimhaltung des Schlüssels.

Forscher sollen also das Verfahren überprüfen können um eventuelle Sicherheitslücken finden zu können.

Was ist informationstheoretisch perfekte Sicherheit? Geben Sie ein Beispielverfahren an! Wird das Verfahren praktisch angewendet?

Informationstheoretische Sicherheit: Selbst ein unbeschränkter Angreifer gewinnt aus seinen Beobachtungen keine zusätzlichen Informationen über den Klartext oder den Schlüssel.

- unbeschränkt: beliebiger Rechen- und Zeitaufwand
- zusätzliche Informationen: nicht besser als Raten

Ein System heißt informationstheoretisch sicher, wenn für alle Nachrichten und alle Schlüsseltexte gilt, dass die a posteriori Wahrscheinlichkeiten $\mathbb{P}(m \mid c)$ der möglichen Nachrichten nach Beobachtung eines gesendeten Geheimtextes gleich der a priori Wahrscheinlichkeiten $\mathbb{P}(m)$ dieser Nachrichten sind:

$$\forall m \in M \forall c \in C : \mathbb{P}(m \mid c) = \mathbb{P}(m)$$

Resultierende Anforderungen an die Schlüssel:

- $|K| \geq |C| \geq |M|$
- Bei einem System mit $|K| = |C| = |M|$ müssen die Schlüssel gleich wahrscheinlich sein.
- Die Wahl des Schlüssels muss zufällig erfolgen.

Ein Beispielverfahren ist das One-Time-Pad (Vernam-Chiffre): Klartext und Schlüssel sind gleich lang und es gilt $c_i = m_i \oplus k_i$, wobei jeder Schlüssel nur einmal verwendet wird und echt zufällig sein muss (nicht pseudo-zufällig, wie z.B. wenn Computer "Zufallszahlen" erzeugen). Die Zufälligkeit ist ein Problem, genau so wie der Schlüsselaustausch, weswegen das Verfahren praktisch nicht angewendet wird.

Welche weiteren Sicherheitsbegriffe gibt es?

Beweisbar sicher: Wenn ein Angreifer die Verschlüsselung knacken kann, so kann er auch das dahinterstehende mathematische Problem lösen.

Semantische Sicherheit: Ein System heißt semantisch sicher, wenn alles, was bei Kenntnis des zugehörigen Schlüsseltextes effizient über den Klartext berechnet werden kann, auch effizient ohne Kenntnis des Schlüsseltextes berechnet werden kann.

Ununterscheidbarkeit ("polynomielle Sicherheit"): Angreifer ist nicht in der Lage, zwei beliebige Nachrichten zu finden, so dass er die Verschlüsselung einer dieser Nachrichten korrekt der Nachricht zuordnen kann.
$\Rightarrow$ äquivalent zur semantischen Sicherheit

Schutzziele der Kryptographie

Die Schutzziele sind:

- Integrität: Die Nachricht wurde nicht verändert
- Zurechenbarkeit: Der Absender der Nachricht hat die Nachricht auch wirklich geschrieben (nur mit asymmetrischen Verfahren realisierbar)
- Vertraulichkeit: Unbefugte können die Nachrichten nicht lesen
- Verfügbarkeit (nicht mit Kryptographie realisierbar)

Warum benutzen wir nicht das informationstheoretisch perfekt sichere Verfahren in Bezug auf die Schutzziele?

Das One-Time-Pad ist ein symmetrisches Verfahren, damit kann Zurechenbarkeit nicht gewährleistet sein.

Was sind MM und PM? Beispiele? Sicherheit?

MM: Monoalphabetisch + Monografisch, Beispiel ist die Cäsar-Chiffre: Hier wird jeder Buchstabe um eine gewissen Anzahl an Positionen im Alphabet verschoben. Das Verfahren ist angreifbar mittels Häufigkeiten von einzelnen Buchstaben, Bi- und Trigrammen und Redundanz bei fehlenden Zeichen.

PM: Polyalphabetisch + Monografisch, Beispiel ist die Vigenère-Chiffre: Jeder Buchtstabe wird um so viele Zeichen im Alphabet verschoben, wie der Schlüssel an dieser Stelle vorgibt. In den meisten Fällen ist der Schlüssel kürzer als die Nachricht, dann wird der Schlüssel wiederholt. Mittels Kasiski-Test (Suche nach identischen Abschnitten im Schlüsseltext) bekommt man die Schlüssellänge. Damit teilt man dann den Schlüsseltext in Blöcke auf und z.B. das erste Zeichen eines jeden Blockes wurde mit dem selben Schlüssel verschlüsselt $\Rightarrow$ Häufigkeitsanalyse.

Wie funktioniert DES? Auf was baut es auf? Sicherheit?

Grundlage ist die Feistel-Chiffre mit 16 Runden. Bei der Feistel-Chiffre wird der zu verschlüsselnde Block in 2 Hälften aufgeteilt, die rechte Hälfte wird durch eine Rundenfunktion f geschickt und das Ergebnis mit der linken Hälfte $\oplus$, was zur neuen rechten Hälfte wird. Die ursprüngliche rechte Hälfte wird zur neuen linken Hälfte.

Die Blocklänge beim DES sind 64 Bit, die Schlüssellänge auch, wobei nur 56 Bit frei wählbar sind, der Rest sind Paritätsbits. Bevor es in die 16 Runden geht, findet eine Eingangspermutation statt und nach den 16 Runden werden noch mal linke und rechte Hälfte getauscht und durch die inverse Eingangspermutation geschickt.

Die rechte Hälfte in einer Runde ist 32 Bit lang, mittels Expansionsabbildung werden daraus 48 Bit, die mit dem Rundenschlüssel $\oplus$ werden. Danach wandern die 48 Bits in 8 Substitutionsboxen aus denen nur 32 Bit wieder herauskommen und diese werden noch mal mit einer Permutationsbox permutiert.

An den Substitutionsboxen hängt die Sicherheit, da diese nicht linear sind. Die lineare Kryptoanalyse versucht diese zu linearisieren und dann rückgängig zu machen. Ein weiteres Problem ist der kurze Schlüssel von nur 56 Bit, mit ausreichend Rechenleistung lässt sich jeder Schlüssel durchprobieren.

Wodurch zeichnet sich die kryptografische Güte der Rundenfunktion f aus?

Eine gute S-Box erfüllt folgende Eigenschaften:

- Vollständigkeit: jedes Outputbit hängt von jedem Inputbit ab
- Avalanche: Änderung eines Input-Bits ändert $\approx 50\%$ der Outputbits, striktes Avalanche-Kriterium: Änderung von $\geq 50\%$ der Outputbits
- Nichtlinearität: Jedes Outputbits hängt nicht linear von den Inputbits ab

Man kann diese Eigenschaften anhand der Abhängigkeitsmatrix überprüfen.

Bei einer Feistel-Chiffre ist die Vollständigkeit erst nach 3 Runden erfüllt.

Wie funktioniert Diffie-Hellman? Und was ist das?

Diffie-Hellman ist ein asymmetrisches Schlüsselaustauschverfahren. Öffentlich bekannt sind eine Primzahl p und ein Generator g der Gruppe $\mathbb{Z}_p^*$. Dabei wählen die beiden Partner x_A und x_B im Geheimen und

berechnen

$$y_A = g^{x_A} \mod p$$
$$y_B = g^{x_B} \mod p$$

und schicken sich y_A und y_B. Dann kann von beiden Partnern der gemeinsame Schlüssel k berechnet werden:

$$k = y_B^{x_A} \mod p$$
$$k = y_A^{x_B} \mod p$$

Die Sicherheit beruht auf dem Diffie-Hellman-Problem, was auf dem Problem des diskreten Logarithmus beruht. Es ist (bisher) sicher gegen passive Angriffen, aber nicht gegen aktive Angriffe (z.B. Man-in-the-Middle).

DH-Problem: gegeben p, g, y_A, y_B, finde $g^{x_A x_B} \mod p$

DL-Problem: bestimme $\log_g(y_A) \mod p$

Wie funktioniert RSA? Schlüsselgenerierung (Bedingungen, $\Phi(n)$, ...), Ver- und Entschlüsselung?

Schlüsselgenerierung: zwei große Primzahlen p und q, berechne $n = p \cdot q$. Wähle öffentlichen Schlüssel mit $1 < k_e < \Phi(n)$ und $\text{ggT}(k_e, \Phi(n)) = 1$, berechne privaten Schlüssel mit $k_d = k_e^{-1} \mod \Phi(n)$, wobei

$$\Phi(n) = (p-1)(q-1)$$

Verschlüsselung: $c = m^{k_e} \mod n$
Entschlüsselung: $m = c^{k_d} \mod n$

Ist das gerade eben skizzierte RSA-System sicher?

Langer Schlüssel (Stand der Technik 2048 Bit), Primzahlen dürfen nicht zu nah beieinander sein und sollten etwa gleiche Länge haben

Nutzung verschiedener n's für verschiedene Nutzer (sonst Common Modulus Attack)

Verhinderung passiver Angriffe durch Nutzung einer Zufallszahl ⇒ sonst Durchprobieren aller Klartexte, bis man Schlüsseltext findet

Verhinderung aktiver Angriffe durch hinzufügen von Redundanz ⇒ RSA ist ein Homomorphismus bezüglich der Multiplikation: m_1, s_1 und m_2, $s_2 \rightarrow m_1 m_2$, $s_1 s_2$ ist gültig

Wie funktioniert das ElGamal-Kryptosystem? Wie lautet das DH-Problem? Warum ist es sicher?

Schlüsselgenerierung: Jeder Teilnehmer

- wählt Primzahl p und Generator $g \in \mathbb{Z}_p^*$
- wählt zufällige Zahl k_d mit $0 \leq k_d \leq p-2$
- berechnet $k_e = g^{k_d} \mod p$

Verschlüsselung: Wahl einer Zufallszahl r mit $0 \leq r \leq p-2$

$$c_1 = g^r \mod p$$
$$c_2 = m \cdot k_e^r \mod p$$

Entschlüsselung: $m = \frac{c_2}{c_1^{k_d}} \mod p$

Sicherheit beruht auf dem DL-Problem: $k_d = \log_g(k_e) \mod p$

Was sind Betriebsarten? Was kann man damit erreichen?

Man will Nachrichten, die Länger als 1 Block sind, verschlüsseln.

Was ist Electronic Code Book und was ist daran das Problem?

Beim ECB verschlüsselt man jeden Block einzelnen und hängt die Blöcke aneinander. Allerdings führt das dazu, dass gleiche Klartextblöcke zu gleichen Schlüsseltextblöcken werden und der Angreifer damit eine gewisse Struktur erkennen kann. Zudem kann ein Angreifer auch zusätzliche Blöcke hinzufügen oder vorhandene Blöcke entfernen, ohne dass dies erkennbar wäre.

Wie funktioniert Cipher Block Chaining?

Bevor eine Nachricht verschlüsselt wird, wird diese mit dem vorherigen Schlüsseltextblock $\oplus$ (für den ersten Block wird die Nachricht mit einem Initialvektor IV $\oplus$).

Bei der Entschlüsselung wird der Schlüsseltext zuerst entschlüsselt und dann mit dem vorherigen Schlüsseltext $\oplus$.

Cipher Feedback Mode

Schieberegister wird mit IV gefüllt, verschlüsselt und mit der Nachricht $\oplus$. Der so entstandene Schlüsseltext wandert in das Schieberegister.

Kryptoanalyse

Bei der differentiellen Kryptoanalyse schickt man Klartextpaare mit bestimmten Differenzen durch den Verschlüsselungsalgorithmus und beobachtet die Output-Differenzen. Daraus versucht man dann wahrscheinliche Schlüssel abzuleiten.

Die lineare Kryptoanalyse versucht man die Verschlüsselung durch eine lineare Funktion zu approximieren.

AES

Klartextblöcke sind für AES nur für 128 Bit standardisiert, die Anzahl der Runden hängt von der Schlüssellänge ab (10 - 14 Runden).

Die Nachricht wird mit dem ersten Teilschlüssel $\oplus$, danach folgenden die r Runden. In jeder Runde finden folgende Operationen statt:

- SubBytes: Substitution
- ShiftRow: zyklische Verschiebung der Zeilen

- MixColumn: Substitution auf Spaltenbasis
- $\oplus$ Rundenschlüssel

In der letzten Runde fällt das MixColumn weg.

Bei der Entschlüsselung wendet man die Inversen Funktionen an, allerdings kann man hier die Reihenfolge gleich wie bei der Verschlüsselung lassen.

Elliptische Kurven

Eine elliptische Kurve ist eine implizit definierte Funktion $y^2 = x^3 + ax + b$. In der Kryptografie braucht man nicht-singuläre Kurven, d.h. $4a^3 + 27b^2 \neq 0$.

Auf elliptischen Kurven kann man Punkte addieren, geometrisch ist $P + Q$:

- Gerade durch P und Q legen, diese schneidet die Kurve in exakt einem Punkt R'
- Spiegelung des Punktes R' an der x-Achse liefert $R = P + Q$.

Eine Punktverdoppelung $2P$ ist geometrisch:

- Tangente an die Kurve im Punkt P schneidet die Kurve in exakt einem Punkt R'
- Spiegelung des Punktes R' an der x-Achse liefert $R = 2P$.

Schlüsselaustausch auf Basis elliptischer Kurven funktioniert so, dass man wie bei Diffie-Hellman sich x_A und x_B wählt Punktverdopplungen/-additionen durchführt. Zusätzlich berechnet man noch mod p.

$$\begin{aligned} Q_A &= x_A \cdot P \mod p \\ Q_B &= x_B \cdot P \mod p \end{aligned}$$

und schicken sich Q_A und Q_B. Dann kann von beiden Partnern der gemeinsame Schlüssel k berechnet werden:

$$\begin{aligned} k &= x_A \cdot Q_B \mod p \\ k &= x_B \cdot Q_A \mod p \end{aligned}$$