Blockchiffren

Def: Eine Blockchiffre mit Blocklänge n ist ein symmetrisches Verschlüsselungssystem mit $M = C = A^n$.

Prop: Die Algorithmen $\mathcal{E}$ und $\mathcal{D}$ einer Blockchiffre definieren bijektive Funktionen.

Bew: Wegen $\mathcal{D}(k, \mathcal{E}(k,m)) = m$ wird keinen zwei Nachrichten der gleiche Chiffretext zugeordnet. Wegen $C = M$ wird jeder Nachricht daher genau ein Chiffretext zugeordnet. $\square$

Bezeichnet $S(A^n)$ die Menge aller bijektiven Abbildungen (Permutationen) von A^n in sich, so gilt also (etwas ungenau) $\mathcal{E}(k,\cdot), \mathcal{D}(k,\cdot) \in S(A^n)$.
Weiter erhalten wir Abbildungen $K \to S(A^n)$, $k \mapsto \mathcal{E}(k,\cdot)$ bzw. $k \mapsto \mathcal{D}(k,\cdot)$.

Blockchiffren

Beispiele:
- Die affin-linearen Chiffren (historisch).
- Substitutionchiffren (definiert durch koordinatenweise Anwendung eines Elements aus $S(A)$ auf A^n, historisch).
- DES (Data Encryption Standard, 1977, veraltet).
- AES (Advanced Encryption Standard, 2001, aktuell).

Allgemeine Elemente aus $S(A^n)$ können nur durch explizite Angabe aller Urbild-Bild Paare beschrieben werden (also $\#A^n$ viele, folglich auch $\#S(A^n) = (\#A^n)!$).

Ist $\mathcal{E}(k,\cdot)$ ein „allgemeines" Element aus $S(A^n)$, so entspricht k einer solchen Urbild-Bild Beschreibung. Wegen der großen Anzahl $\#A^n$ (Anzahl der möglichen Klartexte) ist dies nicht praktikabel.

Blockchiffren

Daher beschränkt man sich auf eine kleine Teilmenge von $S(A^n)$, welche als Ver- und Entschlüsselungsfunktionen vorkommen.
- Teilmenge soll trotzdem möglichst zufällig aus $S(A^n)$ gewählt aussehen.
- Teilmenge muß durch Schlüssel k (effizient) definierbar sein.

Anderer Ansatz für die Definition zufälliger Funktionen aus $S(A^n)$:
Sind r Verschlüsselungen zu berechnen, kann man auch einfach r zufällige Bilder wählen und diese dann den (zu einem späteren Zeitpunkt) gegebenen Klartexten zuordnen.

Beispiel:
- Vernam One-Time Pad.

One-Time Pad

Vernam's One-Time Pad (1917) ist wie folgt definiert:
- $M = C = K = \{0,1\}^n$.
- $c = \mathcal{E}(k,m) := m \oplus k$ (bitweises XOR = bitweise Addition in $\mathbb{Z}/2\mathbb{Z}$).
- $m = \mathcal{D}(k,c) := c \oplus k$.
- k wird zufällig gewählt und nur einmal (!) verwendet.

Entspricht der einmaligen Anwendung einer allgemeinen, zufälligen Permutation aus $S(\{0,1\}^n)$.

Known-Plaintext Angriff liefert $k = m \oplus c$. Unsicher unter mehrfacher Verwendung von k.

Aber: Unter einmaliger Anwendung vollkommen sicher!

One-Time Pad

Sicherheit durch Shannon bewiesen (1949).

Problem:
- Hoher Schlüsselverbrauch, daher ineffizient.
- Wie Schlüssel erzeugen?

Wurde angeblich für die Verbindung zwischen Washington und Moskau verwendet (und andere militärische/diplomatische Anwendungen).

Das One-time pad ist ein Spezialfall der Chiffre von Vignére, wenn nur einmal verwendet.

Perfekte Sicherheit

Theorie durch Shannon (1949). Sicherheit in Anwesenheit von Angreifern mit unbeschränkter Rechenleistung.

Modell:
1. Betrachten Klartexte m als Zufallsvariablen mit Werten in M. („HEUTE“ kommt häufiger vor als „XZYQR“.)
2. Betrachten Schlüssel k als Zufallsvariablen mit Werten in K. (Die Schlüssel werden irgendwie zufällig gewählt.)
3. Annahme: m und k sind unabhängig.
4. Definieren Chiffretexte als Zufallsvariable $c = \mathcal{E}(k,m)$.

Als Ereignisraum können wir $M \times K$ nehmen.
Können $\Pr(c = c_0) > 0$ für alle $c_0 \in C$ annehmen (sonst C verkleinern), ebenso $\Pr(m = m_0) > 0$ für alle $m_0 \in M$.

Perfekte Sicherheit

Def: Ein symmetrisches Verschlüsselungssystem heißt perfekt sicher, wenn $\Pr(m = m_0 \,|\, c = c_0) = \Pr(m = m_0)$ für alle $m_0 \in M$, $c_0 \in C$.

Kenntnis von $c = c_0$ ergibt also keine weiteren Hinweise über den Wert von m.

Prop: Für ein perfekt sicheres Verschlüsselungssystem gilt $\#K \geq \#C$. Genauer gibt es für jedes $m_0 \in M$, $c_0 \in C$ ein $k_0 \in K$ mit $c_0 = \mathcal{E}(k_0, m_0)$.

Thm (Shannon): Es gelte $\#K = \#C$. Ein Verschlüsselungssystem ist genau dann perfekt sicher, wenn $\Pr(k = k_0) = 1/\#K$ für alle $k_0 \in K$ und wenn es für jedes $m_0 \in M$, $c_0 \in C$ genau ein $k_0 \in K$ gibt mit $c_0 = \mathcal{E}(k_0, m_0)$.

Folg: Das One-Time Pad ist bei gv. Schlüsselwahl perfekt sicher.

Perfekte Sicherheit

Bew Prop: Nach dem Satz von Bayes gilt $\Pr(c = c_0 \,|\, m = m_0) = \Pr(c = c_0)$ für jedes $c_0 \in C$. Wegen $\Pr(c = c_0) > 0$ ist c_0 der Chiffretext für ein m_0. Daher gibt es ein $k_0 \in K$ mit $c_0 = \mathcal{E}(k_0, m_0)$. $\square$

Bew Thm:
$\Rightarrow$. Die Existenz von k_0 folgt aus der Prop. Außerdem gilt $C = \{\mathcal{E}(k_0, m_0) \,|\, k_0 \in K\}$, woraus sich wegen $\#C = \#K$ die Eindeutigkeit von k_0 ergibt.
Sei $c_0 \in C$ und $k(m_0) \in K$ mit $\mathcal{E}(k(m_0), m_0) = c_0$. Dann gilt

$$\begin{aligned}\Pr(m = m_0 \,|\, c = c_0)\Pr(c = c_0) &= \Pr(c = c_0 \,|\, m = m_0)\Pr(m = m_0) \\ &= \Pr(k = k(m_0))\Pr(m = m_0).\end{aligned}$$

Aus $\Pr(m = m_0 \,|\, c = c_0) = \Pr(m = m_0)$ ergibt sich damit $\Pr(k = k(m_0)) = \Pr(c = c_0)$. Dies ist unabhängig von m_0, folglich sind alle Wahrscheinlichkeiten gleich und betragen $1/\#C = 1/\#K$.

Perfekte Sicherheit

Bew Thm:

$\Leftarrow$. Sei $k(m_0, c_0) \in K$ mit $\mathcal{E}(k(m_0, c_0), m_0) = c_0$. Dann gilt

$$\begin{aligned} \Pr(m = m_0 \,|\, c = c_0) &= \Pr(c = c_0 \,|\, m = m_0) \Pr(m = m_0) / \Pr(c = c_0) \\ &= \Pr(k = k(m_0, c_0)) \Pr(m = m_0) / \sum_{m_1 \in M} \Pr(m = m_1) \Pr(k = k(m_1, c_0)). \end{aligned}$$

Mit $\Pr(k = k(m_1, c_0)) = 1/\#K$ nach Voraussetzung ist
$\sum_{m_1 \in M} \Pr(m = m_1) \Pr(k = k(m_1, c_0)) = \sum_{m_1 \in M} \Pr(m = m_1)/\#K = 1/\#K$.
Einsetzen in obige Gleichung liefert $\Pr(m = m_0 \,|\, c = c_0) = \Pr(m = m_0)$. $\Box$

Bew Folg: Wegen $k = m \oplus c$ gibt es zu jedem m, c genau einen Schlüssel k mit $c = \mathcal{E}(k, m)$. $\Box$.

Bemerkungen

In einem perfekt sicheren Verschlüsselungssystem erhält ein Angreifer aus dem Chiffretext keine Information über den Klartext oder den Schlüssel.

Für nicht perfekt sichere Systeme hat Shannon die folgenden Fragen untersucht (mehrfache Verschlüsselung mit dem gleichen Schlüssel):

1. Sei $c = \mathcal{E}(k, m)$. Ist $\mathcal{D}(k_1, c)$ ein „sinnvoller“ Klartext (z.B. deutsch) und $k_1 \neq k$, so heißt k_1 Nebenschlüssel. Wieviel Nebenschlüssel gibt es durchschnittlich, basierend auf der Redundanz der Klartexte?

Bemerkungen

Gibt es Nebenschlüssel, so kann ein Angreifer den Klartext nicht eindeutig reproduzieren.

(Für Chiffretext der Länge n: $s_n \geq \#K/(\#M)^{nr} - 1$. ($r$ Redundanz))

2. Wieviel Chiffretext ist durchschnittlich erforderlich, damit es keine Nebenschlüssel mehr geben kann?

Dann kann ein Angreifer den Schlüssel bestimmen.

($n_0 \approx \log_2(\#K)/(r \log_2(\#M))$ Elemente aus C.)

Entropie

Schlüsselkonzept zu Shannon's Untersuchung ist die Entropie einer Zufallsvariablen.

Sind die x_i für $1 \leq i \leq n$ die Werte der ZV X, so wird definiert
$H(X) = -\sum_{\Pr(X=x_i)>0} \Pr(X = x_i) \log_2(\Pr(X = x_i))$.

$-\log_2(\Pr(X = x_i))$ ist Länge in Bits der Information, daß x_i eintritt.
Also $H(X)$ erwartete (durchschnittliche) Information oder Unsicherheit.

$0 \leq H(X) \leq \log_2(n)$,

$H(X) = 0$ für $n = 1$, $H(X)$ max. für $\Pr(X = x_i) = 1/n$.

...

Beispiel: Würfeln mit fairem und „frisierten“ Würfel.

Exhaustive Search Aufwand $\approx 2^{H(k)}$ statt $\#K$.

Bemerkungen

Man kann also keine perfekte (informationstheoretische) Sicherheit erreichen, wenn man viel Klartext mit einem kleinen Schlüssel verschlüsseln will.

Man kann aber versuchen, komplexitätstheoretische Sicherheit zu erreichen.

Die Philosophie hierbei ist, daß $\mathcal{E}(k,\cdot)$ „sich wie eine zufällige Funktion verhält“.

Betriebsarten von Blockchiffren

Blocklänge ist fest und klein. Wie große Mengen an Daten verschlüsseln?

Blockchiffre geeignet verwenden:

- ECB Mode (Electronic Code Book)
- CBC Mode (Cipher Block Chaining)
- CFB Mode (Cipher Feedback)
- OFB Mode (Output Feedback)
- CTR Mode (Counter Mode)

Diese Betriebsarten (ohne CTR) wurden ursprünglich für DES entwickelt, können aber mit jedem Blockchiffre verwendet werden.

Sind standardisiert.

ECB – Electronic Code Book Mode

Einfachste Herangehensweise.

Klartext m in Blöcke der passenden Größe aufteilen $m = m_1 m_2 \cdots m_t$.
Letzten Block durch (zufällige) Bits ergänzen, falls nötig.

Verschlüsselung durch $c = c_1 c_2 \cdots c_t$ mit $c_i = \mathcal{E}(k, m_i)$.
Entschlüsseln durch $m_i = \mathcal{D}(k, c_i)$.

ECB – Electronic Code Book Mode

Eigenschaften:

- $m_i = m_j$ dann $c_i = c_j$, also Regelmäßigkeiten und Wiederholungen übertragen sich.
- Unabhängige c_i, Übertragungsfehler auf Block beschränkt.

Beispiel: Bei Bildern bleiben häufig Konturen erkennbar!

Probleme:

- Chiffretext zu Klartext am Anfang/Ende von Nachrichten extrahierbar.
- Block replay: Mischen/Einfügen von bekanntem Chiffretext möglich.

Anwendung: Besser nicht (u.U. Verschlüsselung von Schlüsseln).

CBC – Cipher Block Chaining Mode

Klartext m in Blöcke der passenden Größe aufteilen $m = m_1 m_2 \cdots m_t$.
Letzten Block durch (zufällige) Bits ergänzen, falls nötig.

Verschlüsselung durch $c = (c_0)c_1c_2 \cdots c_t$ mit $c_0 = IV$ und
$c_i = \mathcal{E}(k, m_i \oplus c_{i-1})$ für $i \geq 0$.

Entschlüsseln durch
$m_i = \mathcal{D}(k, c_i) \oplus c_{i-1}$ für $i \geq 0$.

IV ist zufällig gewählt, oder wird aus m erzeugt (so daß es nur für m vorkommt, z.B. die Verschlüsselung einer eindeutigen Nachrichtennummer).
Braucht nicht geheim gehalten zu werden.

CBC – Cipher Block Chaining Mode

Eigenschaften:

- Kontextabhängig: c_i hängt von c_j mit $j < i$ ab.
- Regelmäßigkeiten und Wiederholungen werden (durch unterschiedliche IV) verwischt.
- Fehler in c_i betrifft nur m_i und lokal m_{i+1}.
- Block replay nicht möglich.

Anwendung: Ist der Standardmodus. Verschlüsseln langer Nachrichten.

Padding

Klartext m in Blöcke der passenden Größe aufteilen $m = m_1 m_2 \cdots m_t$.
Padding = Letzten Block durch (zufällige) Bits ergänzen.

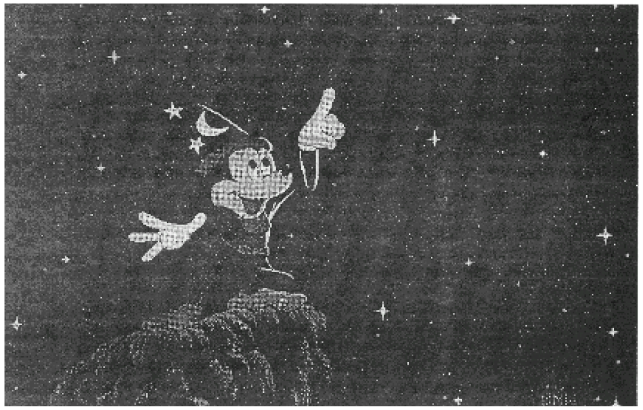
Vom Standpunkt der Kryptographie ist egal, wie man ergänzt (da jeder Klartext sicher verschlüsselt werden soll).

Ansätze:

- Eine Eins und soviele Nullen anhängen, wie nötig.
- Zufällige Bytes und Anzahl zu entfernender Bytes hinten anhängen.

Warum nicht nur Nullen anhängen?

Bild unverschlüsselt



(ausgeliehen von N. Smart, F. Vercauteren)

Bild verschlüsselt im ECB Mode

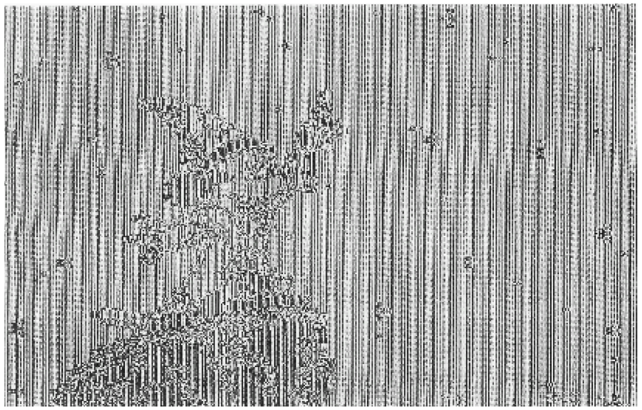


Bild verschlüsselt im CBC Mode