
Blockchiffren Designziele

Designziele:

- Konfusion, Verschleiern des Zusammenhangs zwischen Klar- und Chiffretext.
- Diffusion, Verteilen der Information im Klartext über den Chiffretext.
- Lawineneffekt, jedes Bit im Chiffretext soll von jedem Bit des Klartexts und des Schlüssels abhängen.
- Noch mehr: Jedes Bit des Chiffretext soll sich mit Wahrscheinlichkeit $1/2$ ändern, wenn irgendein Bit des Klartexts oder des Schlüssels geändert wird.

⇒ soll sich wie eine zufällige Funktion verhalten.

1

24. Oktober 2008

Blockchiffren Konstruktion

Der Eingabeblock wird durch mehrfache Anwendung einer Rundenfunktion bearbeitet. Für jede Runde wird ein Rundenschlüssel anhand eines Schlüsselschemas erzeugt.

Die Rundenfunktionen stellt im allgemeinen kein sicheres Verschlüsselungsverfahren dar, erst die mehrfache Anwendung ergibt die Sicherheit.

Bei den Rundenfunktionen werden im wesentlichen Feistel- und Substitutions-Permutationsnetzwerke unterschieden.

Die i -te Runde: $m_{i+1} = g(m_i, k_i)$.

2

24. Oktober 2008

Feistel Netzwerke

Blöcke in zwei gleiche Hälften aufteilen:

- $m_i = (L^i, R^i)$.

Verschlüsseln:

- $L^{i+1} \leftarrow R^i$,
- $R^{i+1} \leftarrow L^i \oplus f(L^{i+1}, k_i)$.

Entschlüsseln:

- $L^{i-1} \leftarrow R^i \oplus f(L^i, k_{i-1})$,
- $R^{i-1} \leftarrow L^i$.

f muß nicht injektiv sein.

3

24. Oktober 2008

Substitutions-Permutationsnetzwerke

Block und Schlüssel mit XOR verknüpfen:

- $b \leftarrow m_i \oplus k_i$.

Block in mehrere Teilblöcke aufteilen und auf die Teilblöcke eine Substitutionschiffre anwenden:

- $(b_j)_j \leftarrow b$,
- $b \leftarrow (\pi_S(b_j))_j$.

Die Bits der zusammengefaßten Teilblöcke permutieren:

- $m_{i+1} \leftarrow \pi_P(b)$.

(Die Permutation π_P wird in der letzten Runde aus Symmetriegründen nicht angewendet, dafür $m_{r+1} \leftarrow b \oplus k_{r+1}$.)

Entschlüsselung durch Invertierung dieser Runden.

4

24. Oktober 2008

Data Encryption Standard - DES

- Entwickelt um 1977
- Ist de-facto der internationale Standard für Bankensicherheit.
- Vermutlich der am meisten analysierte kryptographische Algorithmus.

- Kurze Geschichte:
 - Ausschreibung für ein Verfahren durch NBS (heute NIST).
 - Vorgänger Lucifer bei IBM entwickelt (Feistel, Coppersmith).
 - Untersuchung durch NSA und Veränderung der S-Boxen.
 - Geheimhaltung der Design-Kriterien bis in die 90er (Hintertür-Spekulationen).
 - 1990 Entdeckung der differentiellen Kryptanalyse, war NSA schon damals bekannt ...

5

24. Oktober 2008

DES

Blocklänge 64 Bit, Schlüssellänge 56 Bit.
Feistel Chiffre mit 16 Runden.

Die Feistel-Funktion $f: \{0, 1\}^{32} \times \{0, 1\}^{48} \rightarrow \{0, 1\}^{32}$ ist definiert wie folgt:

1. L^{i+1} wird auf 48 Bit erweitert und permutiert, so daß 16 Bits doppelt auftreten (Diffusion, Lawineneffekt).
2. XOR mit dem Rundenschlüssel k_i und Aufteilung in 8 Blöcke B_i von je 6 Bit.
3. Anwendung der S-Box $S_i: \{0, 1\}^6 \rightarrow \{0, 1\}^4$ auf B_i , $C_i = S_i(B_i)$.
4. Anwendung der Permutation P auf $C_1 \dots C_8$, Ergebnis ist $f(L^{i+1}, k_i)$.

6

24. Oktober 2008

DES

Nach dem letzten Feistelschritt werden L^{16} und R^{16} vertauscht.

Dem gesamten Feistel Netzwerk wird eine Permutation $IP \in S(\{0, 1\}^{64})$ vorgeschaltet und IP^{-1} nachgeschaltet. Dies hat keine kryptographische Bedeutung.

Zusammenfassend also:

1. IP auf Block anwenden: $(L^0, R^0) \leftarrow IP(m)$.
2. 16 Runden Feistel-Netzwerk.
3. $R^{17} \leftarrow L^{16}$ und $L^{17} \leftarrow R^{16}$.
4. IP^{-1} auf Block anwenden: $c \leftarrow IP^{-1}(L^{17}, R^{17})$.

7

24. Oktober 2008

DES

Die k_i sind 48 Bit Schlüssel und setzen sich aus einer Auswahl permutierter Bits aus k zusammen:

1. k wird in zwei 28 Bit Blöcke aufgeteilt.
2. Diese werden in Abhängigkeit der Rundenanzahl rotiert.
3. Aus den 56 Bit werden 48 gemäß einer festen Regel ausgewählt.

Schwache Schlüssel k mit $\mathcal{E}(k, \cdot) = \mathcal{D}(k, \cdot)$:

- Linke und rechte Hälfte von k nur Einsen oder Nullen.
- Keine weiteren schwachen Schlüssel bekannt.

Semi-schwache Schlüsselpaare (k, l) mit $\mathcal{E}(k, \cdot) = \mathcal{E}(l, \cdot)$:

- 6 sind bekannt.

8

24. Oktober 2008

DES

Sicherheit von DES:

- Probleme im wesentlichen nur wegen zu kleiner Schlüssellänge.
- Stark gegen differentielle Kryptanalyse.
- Differentielle und lineare Kryptanalyse brauchen $> 2^{40}$ Klartexte, nicht praktikabel.
- Brute-Force (2^{56} Schritte) praktikabel.

Hardware:

- Deep Crack (1998): 250.000 Dollar, ein Schlüssel in 50h.
- Man kann davon ausgehen, daß es heute (bei den entsprechenden Organisationen) Hardware gibt, die das viel schneller schafft.

9

24. Oktober 2008

DES

Sicherheit von DES:

- Exportversionen von DES haben sogar nur 40 Bit Schlüssel!
- Triple-DES (EDE) mit ca. 112 Bit effektiver Schlüssellänge.

Moore's Law: Rechenleistung verdoppelt sich alle 18 Monate.

Also alle 15 Jahre 10 Bit Reduktion der Sicherheit in Bezug auf die benötigte Zeit.

Heutzutage werden mindestens 128 Bit Schlüssellänge und Blocklänge angestrebt.

10

24. Oktober 2008

Advanced Encryption Standard - AES

Reaktion auf den auslaufenden DES (langsamen Tripel-DES).

1997 Ausschreibung vom NIST für Nachfolger von DES.

- Blockchiffre mit 128 Bit Blocklänge, 128/192/256 Bit Schlüssellänge.
- Offene Dokumentation, Referenzimplementierungen.
- Bedingung: nicht patentiert, frei verwendbar.
- Offener, internationaler Prozess.
- 1999: Fünf Kandidaten: MARS, RC6, Rijndael, Serpent, Twofish.
- 2000/2001: Rijndael wird AES.

Alle fünf Kandidaten wurden als sicher eingestuft.

Wird von US Behörden benutzt (für classified SECRET ≥ 128 , für TOP SECRET ≥ 192). Weite Verbreitung (zu erwarten).

11

24. Oktober 2008

Endliche Körper

Sind sehr wichtig in der Kryptographie und Codierungstheorie (Übertragungsfehlerkorrektur). Nützlich bei der Beschreibung von Rijndael.

Ein Körper ist eine Menge, in der man wie in \mathbb{R} oder \mathbb{C} rechnen kann, also mit $+, -, \cdot, /$.

Endliche Körper haben nur endlich viele Elemente.

p Primzahl. Modulo p rechnen liefert $\mathbb{Z}/(p)$. Ist endlicher Körper \mathbb{F}_p .

- Elemente dargestellt durch $\{0, 1, \dots, p-1\}$.
- Beispiel $p = 5$: $-1 = 4$ weil $4 + 1 = 5 = 0$ in \mathbb{F}_p .
- Invertieren mit euklidischem Algorithmus zur ggT-Berechnung:
 $1 = ra + sp$ impliziert $r = 1/a$. Beispiel $1 = -3 \cdot 3 + 2 \cdot 5$, also $-3 = 1/3$.

12

24. Oktober 2008

Polynome

Polynome über (endlichen) Körpern K .

- Sind Ausdrücke der Form $a_0 + a_1x + a_2x^2 + \dots + a_r x^r$ mit Koeffizienten $a_i \in K$ und x einer „Variablen“.
- Gleichheit zweier Polynome genau dann, wenn die Koeffizienten vor x^i gleich sind für alle i .
- $+$, $-$, \cdot wie gewohnt und sinnvoll.
- Grad $\deg(f) := r$ wenn $f = a_0 + a_1x + a_2x^2 + \dots + a_r x^r$ und $a_r \neq 0$.
- $\deg(f) \leq 0$: f heißt konstant.

Beispiel:

- $(x+1)(x-1) = x^2 + x - x - 1 = x^2 - 1$.
- $2x+1 \neq 2x+2$, $x+2 \neq 2x+2$, $x \neq 0$.
- $\deg(x) = 1$, $\deg(x^2) = 2$, $\deg(1) = 0$, $\deg(0) = -\infty$.

13

24. Oktober 2008

Polynome

Menge der Polynome über K ist damit ein Ring (Körper ohne $/$).
Wird mit $K[x]$ bezeichnet.

Division mit Rest: Für $f, g \in K[x]$ schreibe $f = hg + r$ mit $h, r \in K[x]$ und $\deg(r) < \deg(g)$.

- Wie in der Schule möglich, ist analog zur Division mit Rest in \mathbb{Z} .
- h und r sind eindeutig bestimmt.
- f durch g teilbar $\Leftrightarrow r = 0$.
- Können damit in $K[x]$ modulo g rechnen, Ergebnisse sind die r .
- $f = x^2 + 1$, $g = x - 1$: $x^2 + 1 = x^2 - 1 + 2 = (x+1)(x-1) + 2$, also $h = x+1$ und $r = 2$.

14

24. Oktober 2008

Polynome

Primpolynom $f \in K[x]$: Kann nicht als Produkt nicht konstanter Polynome geschrieben werden (analog zur Primzahl).

- $x^2 - 1$ nicht prim: $x^2 - 1 = (x-1)(x+1)$.
- $x^2 + x + 1 \in \mathbb{F}_2[x]$ prim.

Thm: Jedes Polynom kann in ein Produkt von Primpolynomen zerlegt werden. Die Zerlegung ist bis auf Multiplikation mit Konstanten eindeutig.

- Nicht eindeutig: $x^2 - 1 = (x-1)(x+1) = (2x-2)(3x+3)$ in $\mathbb{F}_5[x]$.

Bew: Üblicherweise in der Algebra.

15

24. Oktober 2008

Endliche Körper

$K = \mathbb{F}_p$, $f \in K[x]$ Primpolynom. In $K[x]$ modulo f rechnen, ergibt $K[x]/(f)$. Ist endlicher Körper \mathbb{F}_{p^n} mit p^n Elementen.

- Primpolynome beliebigen Grads gibt es in Tabellen.
- Invertieren wieder mit euklidischem Algorithmus, jetzt für Polynome mit Hilfe der Polynomdivision. Beispiel: $K = \mathbb{F}_2$, $f = x^2 + x + 1$, $x \cdot (x+1) + f = 1$. Also $x = 1/(x+1)$ in \mathbb{F}_4 .
- In $K[x]/(f)$ gilt $f(x) = 0$.

Man schreibt häufig $\mathbb{F}_{p^n} = \mathbb{F}_p[\zeta]$ mit $f(\zeta) = 0$.

Elemente in \mathbb{F}_{p^n} können also durch Polynome vom Grad $\leq n-1$ in ζ über \mathbb{F}_p dargestellt werden. Es gilt $\#\mathbb{F}_{p^n} = p^n$.

16

24. Oktober 2008

Endliche Körper

\mathbb{F}_{p^n} auch ein n -dimensionaler \mathbb{F}_p -Vektorraum.

Thm: Für jedes n ist \mathbb{F}_{p^n} bis auf Isomorphie eindeutig bestimmt.

Thm: Es gibt $w \in \mathbb{F}_{p^n}$, so daß sich jedes $a \in \mathbb{F}_{p^n}$ mit $a \neq 0$ in der Form $a = w^s$ für ein s mit $0 \leq s \leq p^n - 2$ schreiben läßt.

Bew: Üblicherweise in der Algebra.

$(\mathbb{F}_2)^8$:

- Sind Bytes.
- Addition entspricht XOR.
- $-1 = 1$, daher Subtraktion gleich Addition.
- Identifikation mit \mathbb{F}_{2^8} unter einer Basis $1, \zeta, \dots, \zeta^7$.

17

24. Oktober 2008

AES - Rijndael

Basiert auf (verallgemeinertem) Substitutions-Permutationsnetzwerk.

- 128 Bit Blocklänge,
- 128/192/256 Bit Schlüssellänge,
- entsprechend 10/12/14 Runden.

Im folgenden $(\mathbb{F}_2)^8 \cong \mathbb{F}_2[\zeta] \cong \mathbb{F}_{2^8}$ als \mathbb{F}_2 -Vektorräume mit $\zeta^8 + \zeta^4 + \zeta^3 + \zeta + 1 = 0$ und $(c_7, \dots, c_0) \mapsto \sum_{i=0}^7 c_i \zeta^i$.

Der jeweils zu bearbeitende Block b (state block) und Rundenschlüssel k_i im Netzwerk ist eine Matrix $\in (\mathbb{F}_{2^8})^{4 \times 4}$. Hierbei zeilenweise arbeiten: Ein Block oder Rundenschlüssel (b_0, \dots, b_{15}) ergibt die Matrix $(b_{4(i-1)+j-1})_{1 \leq i, j \leq 4}$.

Details im FIPS-197 und unter <http://www.nist.gov/aes>.

18

24. Oktober 2008

AES - Rijndael

Die Operationen auf b in den einzelnen Runden sind:

AddRoundKey(i):

- $b \leftarrow b + k_i$.

SubstBytes:

- $b \leftarrow$ Koeffizientenweise Anwendung von $\pi_S \in S(\mathbb{F}_{2^8})$ auf b .

ShiftRows:

- Zeile v in b um $v - 1$ Positionen nach links rotieren, $1 \leq v \leq 4$.

MixColumns:

$$\bullet b \leftarrow M \cdot b \text{ mit } M = \begin{pmatrix} \zeta & \zeta+1 & 1 & 1 \\ 1 & \zeta & \zeta+1 & 1 \\ 1 & 1 & \zeta & \zeta+1 \\ \zeta+1 & 1 & 1 & \zeta \end{pmatrix}.$$

19

24. Oktober 2008

AES - Rijndael

Ausführung der Runden:

1. Eingabe $b \leftarrow m$.
2. AddRoundKey(i) für $i = 0$.
3. Für $i = 1, \dots, n - 1$: SubstBytes, ShiftRows, MixColumns, AddRoundKey(i).
4. Für $i = n$: SubstBytes, ShiftRows, AddRoundKey(i).
5. Ausgabe $c \leftarrow b$.

Für Entschlüsselung inverse Operationen benutzen.

Diffusion durch ShiftRows und MixColumns.

Konfusion durch SubstBytes und AddRoundKey.

20

24. Oktober 2008

AES - Rijndael

Blocklänge $b = 4, 6, 8$ des Schlüssel k in Worten (4 Bytes).

Schlüsselschema zur Bestimmung der k_i (KeyExpansion):

- Schreibe $k = (w_0, \dots, w_{b-1})$ mit $w_j \in (\mathbb{F}_{2^8})^4$.
- Für $j \geq b$: $w_j \leftarrow w_{j-b} + f(j, w_{j-1})$, wobei f unten definiert ist.
- $k_i \leftarrow (w_{4i}, \dots, w_{4i+3})$ für $0 \leq i \leq n$.

Definition von f :

- SubstWord: π_S auf $(\mathbb{F}_{2^8})^4$ erweitern durch byteweise Anwendung.
- RotWord: $\text{RotWord}(B_0, B_1, B_2, B_3) = (B_1, B_2, B_3, B_0)$, $B_i \in \mathbb{F}_{2^8}$.
- Für $j = 0 \bmod b$ ist $f(j, w_{j-1}) = \text{SubstWord}(\text{RotWord}(w_{j-1}) + (\zeta^{j/b-1}, 0, 0, 0))$.
- Für $b > 6$ und $j = 4 \bmod b$ ist $f(j, w_{j-1}) = \text{SubstWord}(w_{j-1})$.
- In allen anderen Fällen ist $f(j, w_{j-1}) = w_{j-1}$.

21

24. Oktober 2008

AES - Rijndael

Definition von $\pi_S \in S(\mathbb{F}_{2^8})$:

- $\phi \in S(\mathbb{F}_{2^8})$ ist eine affin-lineare Abbildung auf \mathbb{F}_{2^8} als \mathbb{F}_2 -Vektorraum.
- Damit $\pi_S(x) := \phi(1/x)$ für $x \neq 0$ und $\pi_S(0) := \phi(0)$.

Definition von $\phi \in S(\mathbb{F}_{2^8})$:

- $\psi: \mathbb{F}_{2^8} \rightarrow \{g \in \mathbb{F}_2[x] \mid \deg(g) < 8\}$ mit $\psi(\sum_{i=0}^7 b_i \zeta^i) := \sum_{i=0}^7 b_i x^i$, ψ ist \mathbb{F}_2 -linear und bijektiv.
- $\phi(f) := \psi^{-1}((x^4 + x^3 + x^2 + x + 1)\psi(f) + x^6 + x^5 + x + 1 \bmod x^8 + 1)$.
- $\phi^{-1}(g) = \psi^{-1}((x^6 + x^3 + x)\psi(g) + x^2 + 1 \bmod x^8 + 1)$.
- Hierbei bezeichnet \bmod den Rest nach Division mit $x^8 + 1$.

Da ϕ affin-linear und bijektiv ist, kann man ϕ auch mit einer inv-baren Matrix aus $(\mathbb{F}_2)^{8 \times 8}$ definieren (siehe z.B. das FIPS-197 Dokument).

22

24. Oktober 2008

AES - Rijndael

Rijndael ist sehr schnell, besonders auf Chipkarten (im Vergleich doppelt so schnell wie andere Verfahren).

Sicherheit von Rijndael:

- Rundenzahl recht knapp gehalten.
- Wird immer noch intensiv untersucht.

Die Struktur von Rijndael ist sehr algebraisch. Daher gibt es eine einfache, geschlossene algebraische Formel für die Verschlüsselung.

- Manche sehen dies als möglichen Angriffspunkt an.
- Es könnte aber auch helfen, die Sicherheit zu beweisen.
- XL (extended linearisation): Verfahren zum Lösen von Gleichungssystemen mit vielen Unbekannten.
- Effizienz von XL?

23

24. Oktober 2008

Angriffe auf Blockchiffren

Brute-Force, exhaustive search (nach dem Schlüssel), Meet-in-the-middle, CPA.

- Alles oder trickreich ausprobieren \rightarrow effektive Schlüssellänge.

Differentielle Kryptoanalyse

- Man untersucht, wie Änderungen (Differenzen) am Klartext sich durch die Runden fortpflanzen, u. mit welcher Wahrscheinlichkeit.

Lineare Kryptoanalyse

- Man untersucht, mit welcher Wahrscheinlichkeit lineare Relationen zwischen Klartext- und Chiffretextbits gelten.

Diese Techniken sind statistisch und ziemlich detailliert.

Lassen sich (im wesentlichen) nicht auf DES und Rijndael anwenden.

24

24. Oktober 2008

Angriffe auf Blockchiffren

Timing Angriffe, Power Analysis:

- wenn Ausführungszeit oder Energieverbrauch vom Schlüssel abhängt.
- Zeit- bzw. Energiemessung liefert Information über den Schlüssel.
- Gegenmaßnahme: Algorithmus entsprechend modifizieren.

Differential Fault Analysis:

- Hardware-mäßiger Eingriff auf Bits und Programmausführung (z.B. in Chipkarte).
- Speziell Related-Key Angriff.
- Dann Untersuchung der geänderten/fehlerhaften Verschlüsselung.
- Starke Anforderung an die Hardware ...

Diese Angriffe sind auch für andere kryptographische Algorithmen relevant.