Erinnerung Blockchiffre

Definition schlüsselabhängige Permutation

Seien F, F^{-1} pt Algorithmen. F heißt *schlüsselabhängige Permutation* auf ℓ Bits falls

1. F berechnet eine Funktion $\{0,1\}^n \times \{0,1\}^\ell \rightarrow \{0,1\}^\ell$, so dass für alle $k \in \{0,1\}^n$ die Funktion $F_k(\cdot)$ eine Bijektion ist.
2. $F_k^{-1}(\cdot)$ berechnet die Umkehrfunktion von $F_k(\cdot)$.

Definition Starke Pseudozufallspermutation (Blockchiffre)

Sei F eine schlüsselabhängige Permutation auf ℓ Bits. Wir bezeichnen F als *starke Pseudozufallspermutation (Blockchiffre)*, falls für alle ppt D gilt

$$\left|\mathrm{Ws}[D^{F_k(\cdot),F_k^{-1}(\cdot)}(1^n)=1] - \mathrm{Ws}[D^{f(\cdot),f^{-1}(\cdot)}(1^n)=1]\right| \leq \mathrm{negl}(n),$$

mit $k \in_R \{0,1\}^n$ und $f \in_R Perm_\ell$.

Angriffe auf Blockchiffren

Warnung:
Blockchiffren selbst sind **kein sicheres** Verschlüsselungsschema.

Angriffe: in aufsteigender Stärke

1. Ciphertext-only: $\mathcal{A}$ erhält $F_k(x_i)$ für unbekannte x_i.
2. Known plaintext: $\mathcal{A}$ erhält Paare $(x_i, F_k(x_i))$
3. Chosen plaintext: $\mathcal{A}$ wählt x_i und erhält $F_k(x_i)$.
   (entspricht PRP-Definition)
4. Chosen ciphertext: $\mathcal{A}$ wählt x_i, y_i und erhält $F_k(x_i), F_k^{-1}(y_i)$.
   (entspricht starker PRP-Definition)

Sicherheit: Ununterscheidbarkeit von echter PRP.

Konfusion und Diffusion

Ziel: Kleine Eingabedifferenzen erzeugen pseudozufällige Ausgaben.

Paradigma Konfusion und Diffusion

Rundeniterierte Vorgehensweise zur Konstruktion einer Blockchiffre

1. **Konfusion:** Permutiere kleine Bitblöcke schlüsselabhängig.
2. **Diffusion:** Permutiere alle Bits.

Bsp: F soll Blocklänge 128 Bits besitzen.

- Konfusion: Definiere schlüsselabhängige Permutation $f_1, \ldots, f_{16}$ auf 8 Bits. Sei $x = x_1 \ldots x_{16} \in (\{0,1\}^8)^{16}$. Definiere
$$F_k(x) = f_1(x_1) \ldots f_{16}(x_{16}).$$
- Diffusion: Permutiere die Bits von $F_k(x)$.
- Iteriere die obigen beiden Schritte hinreichend oft, damit kleine Eingabedifferenzen sich auf alle Ausgabebits auswirken.
- Beschreibungslänge von f_i: $8 \cdot 2^8$ Bits, $F: 16 \cdot 8 \cdot 2^8 = 2^{15}$ Bits.
- Länge einer echten Zufallspermutation: $128 \cdot 2^{128} = 2^{135}$ Bits.

Substitutions-Permutations Netzwerk (SPN)

Szenario: Verwende einen Masterschlüssel k.

- Berechne aus dem Masterschlüssel k Rundenschlüssel $k_1, \ldots k_r$ mittels eines sogenannten Keyschedule-Algorithmus.
- Die Permutationsfunktionen $f_1, \ldots, f_m$ werden fest und schlüsselunabhängig gewählt (sogenannte S-Boxen).

Beschreibung Substitutions-Permutations Netzwerk (SPN)

EINGABE: $f_1, \ldots, f_m, k \in \{0,1\}^n, x, \ell, r$

1. Berechne $k_1, \ldots, k_r \in \{0,1\}^\ell$ aus k. Setze $y \leftarrow x$.
2. For $i \leftarrow 1$ to r
   1. **Schlüsseladdition:** $y := y \oplus k_i$.
   2. **Substitution per S-Boxen:** $y := f_1(y_1) \ldots, f_m(y_m)$ mit $y = y_1 \ldots y_m$.
   3. **Permutation:** $y :=$ Permutation der Bits von y.

AUSGABE: $F_k(x) := y$

Beobachtung: F ist invertierbar, da jeder Schritt invertierbar ist.

Lawineneffekt

Ziel: Veränderung in Eingabebit wirkt sich auf alle Ausgabebits aus.

Beobachtung Notwendige Eigenschaften für Lawineneffekt

1. **S-Box:** Ändern eines Eingabebits verändert ≥ 2 Ausgabebits.
2. **Permutation:** Ausgabebits einer S-Box werden zu Eingabebits verschiedener S-Boxen.

Beobachtung: Lawineneffekt

- Betrachten ein SPN mit 4 Bit S-Boxen und Blocklänge 128 Bit.
- 1-Bit Eingabedifferenz erzeugt mindestens eine 2-Bit Differenz.
- Eine 2-Bit Differenz resultiert in zwei 1-Bit Differenzen an verschiedenen S-Boxen in der nächsten Runde.
- Diese sorgen für mindestens 4-Bit Differenz, usw.
- D.h. jede Runde verdoppelt potentiell die beeinträchtigten Bits.
- Nach 7 Runden sind alle $2^7 = 128$ Bits von der Veränderung eines Eingabebits beeinträchtigt.

Angriff auf eine Runde eines SPN

Algorithmus Angriff auf eine Runde eines SPN

EINGABE: $x, y = F_k(x)$

1. $y :=$ Invertiere auf y die Permutation und die S-Boxen.
2. Berechne $k := x \oplus y$.

AUSGABE: k

Anmerkungen:

- Die Invertierung in Schritt 1 ist möglich, da sowohl die Permutation als auch die S-Boxen öffentlich sind.
- Nach Invertierung erhält man den Wert $x \oplus k$.

Distinguisher für 2 Runden

Bsp: Distinguisher für 2 Runden

- Wir betrachten Blocklänge 80 Bit und 4 Bit S-Boxen.
- Wähle x_i, die sich nur im ersten 4-Bit Block unterscheiden.
- Nach 1. Runde: Ausgaben unterscheiden sich in $\leq$ 4 Blöcken.
- Nach 2. Runde: Ausgaben unterscheiden sich in $\leq$ 16 Blöcken.
- D.h. nicht alle der 20 Ausgabeblöcke werden verändert.
- Können SPN leicht von Pseudozufallspermutation entscheiden.

Feistelnetzwerk

Szenario:

- Leite aus k Rundenschlüssel $k_1, \ldots, k_r$ ab.
- Teile Nachrichtenblock in linke Seite L_i und rechte Seite R_i.
- Sei n die Blocklänge. Definiere nicht notwendigerweise invertierbare Rundenfunktionen $f_i : \{0,1\}^{\frac{n}{2}} \to \{0,1\}^{\frac{n}{2}}$.
- Die Funktionen f_i hängen von den Rundenschlüsseln k_i ab.

Algorithmus Feistelnetzwerk

EINGABE: k, x, n, r

1. Leite $k_1, \ldots, k_r$ aus k ab.
2. Setze $(L_0 || R_0) := x$ mit $L_i, R_i \in \{0,1\}^{\frac{n}{2}}$.
3. For $i = 1$ to r
   1. Setze $L_i := R_{i-1}$ und $R_i := L_{i-1} \oplus f_i(R_{i-1})$.

AUSGABE: $F_k(x) := (L_r || R_r)$

Invertierung einer Feisteliteration: $R_{i-1} := L_i$ und $L_{i-1} := R_i \oplus f_i(L_i)$.

DES - Data Encryption Standard

Beschreibung von DES:

- Entwickelt 1973 von IBM, standardisiert 1976.
- DES besitzt Schlüssellänge 56 Bit und Blocklänge 64 Bit.
- Besteht aus Feistelnetzwerk mit 16 Runden.
- Aus Bits von k werden 48-Bit Schlüssel $k_1, \ldots, k_{16}$ ausgewählt.
- Rundenfunktionen f_i sind SPNs mit nicht invertierbaren S-Boxen.

Algorithmus Rundenfunktion f_i

EINGABE: $k_i, R_{i-1} \in \{0,1\}^{32}$

1. $y := $ Erweitere R_{i-1} auf 48 Bit durch Verdopplung von 16 Bits.
2. $y := y \oplus k_i$
3. $y :=$ Splitte y in 6-Bit Blöcke $y_1 \ldots y_8$ auf. Wende auf jedes y_i eine S-Box $S_i : \{0,1\}^6 \to \{0,1\}^4$ an. Permutiere das Ergebnis.

AUSGABE: $f_i(R_{i-1}) := y$

Die DES S-Boxen

DES S-Boxen:

- Alle 8 S-Boxen realisieren verschiedene Abb. $\{0,1\}^6 \to \{0,1\}^4$.
- Jede S-Box ist eine 4:1-Abbildung.
- D.h. jede S-Box sendet genau 4 Eingaben auf eine Ausgabe.
- Wechsel eines Eingabebits ändert mindestens zwei Ausgabebits.

Lawineneffekt bei DES:

- Wähle (L_0, R_0) und (L_0', R_0) mit 1-Bit Differenz in L_0, L_0'.
- (L_1, R_1) und (L_1', R_1') besitzen 1-Bit Differenz in R_1, R_1'.
- Durch f_2 erhält man mindestens eine 2-Bit Differenz in R_2, R_2'.
- D.h. (L_2, R_2) und (L_2', R_2') besitzen mind. eine 3-Bit Differenz.
- f_3 angewendet auf R_2, R_2' liefert mind. eine 4-Bit Differenz, usw.
- Nach 8 Runden erreicht man volle Diffusion auf alle Ausgabebits.