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 $f_i : \{0,1\}^{\frac{n}{2}} \rightarrow \{0,1\}^{\frac{n}{2}}$.
- 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 $L_{i-1} := R_i \oplus f_i(R_{i-1})$.

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 den Bits 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.

Angriff auf eine Runde DES

Algorithmus Angriff auf eine Runde DES

EINGABE: (x, y) mit $x = (L_0, R_0)$ und $y = (L_1, R_1)$

1. Wir kennen ein Paar $(R_0, L_0 \oplus R_1)$ mit $f_1(R_0) = L_0 \oplus R_1$.
2. Berechne die Ausgaben für jede der acht S-Boxen.
3. Für jede Ausgabe gibt es 4 mögliche 6-Bit Eingaben, aus denen sich 4 Möglichkeiten für die 6 betreffenden Bits von k_1 ergeben.
4. Teste alle 4^8 Möglichkeiten für k_1 und verifiziere mittels (x, y).

AUSGABE: k_1

Laufzeit:

- Rekonstruieren k_1 mit Komplexität $4^8 = 2^{16}$ und einem Paar (x, y).

Angriff auf zwei Runden DES

Algorithmus Angriff auf zwei Runden DES

EINGABE: (x, y) mit $x = (L_0, R_0)$ und $y = (L_2, R_2)$

1. Setze $L_1 = R_0$ und $R_1 = L_0 \oplus f_1(R_0)$.
2. Verwende $(L_0, R_0), (L_1, R_1)$, um k_1 wie zuvor zu bestimmen.
3. Verwende $(L_1, R_1), (L_2, R_2)$, um k_2 wie zuvor zu bestimmen.

AUSGABE: k_1, k_2

Laufzeit:

- Wir benötigen Laufzeit 2^{17} mit einem Paar (x, y).
- Kann reduziert werden, wenn wir das DES-Keyscheduling berücksichtigen, bei dem Bits von k_1 einige Bits von k_2 festlegen.

Angriff auf drei Runden DES

Eigenschaften des DES-Keyschedule:

- Sei $k = k^{(1)}k^{(2)} \in \{0,1\}^{56}$ der Masterschlüssel mit $k^{(i)} \in \{0,1\}^{28}$.
- Für jeden Rundenschlüssel $k_i \in \{0,1\}^{48}$ gilt: Die ersten 24 Bits von k_i werden aus $k^{(1)}$ gewählt, die zweiten 24 Bits aus $k^{(2)}$.

Idee zum Angriff auf drei Runden DES:

- Sei ein Paar $(x, y) = ((L_0, R_0), (L_3, R_3))$ gegeben.
- Die Eingaben von f_1 bzw. f_3 sind R_0 bzw. R_3.
- Die Ausgaben von f_1 bzw. f_3 sind $(L_0 \oplus L_2)$ bzw. $(L_2 \oplus R_3)$.
- Da L_2 unbekannt ist, sind beide Ausgaben unbekannt.
- Allerdings ist das XOR der Ausgaben von f_1, f_3 bekannt: $L_0 \oplus R_3$.
- Wir raten $k^{(1)}$ und $k^{(2)}$ separat und Testen dieses XOR.

Angriff auf drei Runden DES

Algorithmus Angriff auf drei Runden DES

EINGABE: $(x_1, y_1), (x_2, y_2)$

1. Rate $k^{(1)}$. Berechne die ersten 24 Bits von k_1 und k_3.
2. Berechne die Eingabe der S-Boxen $S_1, \ldots, S_4$ von f_1.
3. Berechne die Eingabe der S-Boxen $S_1, \ldots, S_4$ von f_3.
4. D.h. wir kennen dieselben 16 Ausgabebits von f_1 und f_3.
5. Vergleiche das XOR dieser Bits mit $L_0 \oplus R_3$ für beide Paare (x_1, y_1), (x_2, y_2). Bei Ungleichheit verwerfe $k^{(1)}$.
6. Verfahren analog beim Raten des Teilschlüssels $k^{(2)}$.

AUSGABE: Masterschlüssel $k = k^{(1)}k^{(2)}$

Laufzeit:

- Annnahme: Inkorrekte Teilschlüssel $k^{(1)}$ stimmen auf den 16 Bits in Schritt 5 mit Ws $\frac{1}{2^{16}}$ überein. Daher genügen zwei Paare (x_i, y_i).
- Raten für $k^{(i)}$ jeweils 28 Bits, d.h. die Gesamtkomplexität ist 2^{29}.

Die (Un-)Sicherheit von DES

Sicherheit von DES:

- Bester praktischer Angriff ist noch immer die Brute-Force Suche.
- Die folgende Tabelle gibt eine Übersicht über DES Kryptanalysen.

Jahr	Projekt	Zeit
1997	DESCHALL, Internet	96 Tage
1998	distributed.net, Internet	41 Tage
1998	Deep Crack, 250.000 Dollar Maschine	2 Tage
2008	COPACOBANA, 10.000 Euro FPGAs	1 Tag

- Das Design von DES ist gut, nur die Schlüssellänge ist zu kurz.
- Die Blocklänge von 64 Bits von DES gilt als zu kurz (s. Folie 78).