Vorlesung Sicherheit

Dennis Hofheinz

ITI, KIT

17.04.2014

Logistisches

- Überschneidungsfreiheit Vorlesung: nachfragen
- Übungsblatt nicht vergessen
- **Frage:** Wie viele würden korrigiertes Übungsblatt nutzen?
- Skript: Projekt Ü-Ei ($\rightarrow$ `Florian.Boehl@kit.edu`)
- Demnächst: Kummerkasten

Erinnerung

- Sicherheit durch sichere Bausteine
- Erstes Thema: (symmetrische) Verschlüsselung $(\mathrm{Enc}, \mathrm{Dec})$
- Beispiele: Cäsar, Vigenère, One-Time-Pad
- OTP: $C = M \oplus K$ (unhandlich, veränderbar)
- Stromchiffren: „Simulation“ von OTP (veränderbar)

Überblick

Überblick

Struktur von Blockchiffren

- Zentraler Baustein: Funktion $\mathrm{E} : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$
- E bildet Schlüssel und Klartextblock auf Chiffratblock ab
- Für Entschlüsselung $\mathrm{D} : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$:

$$\forall K, M : \quad \mathrm{D}(K, \mathrm{E}(K, M)) \stackrel{!}{=} M$$

- Verschiedene Wege, E zu benutzen (Betriebsmodi)

Überblick

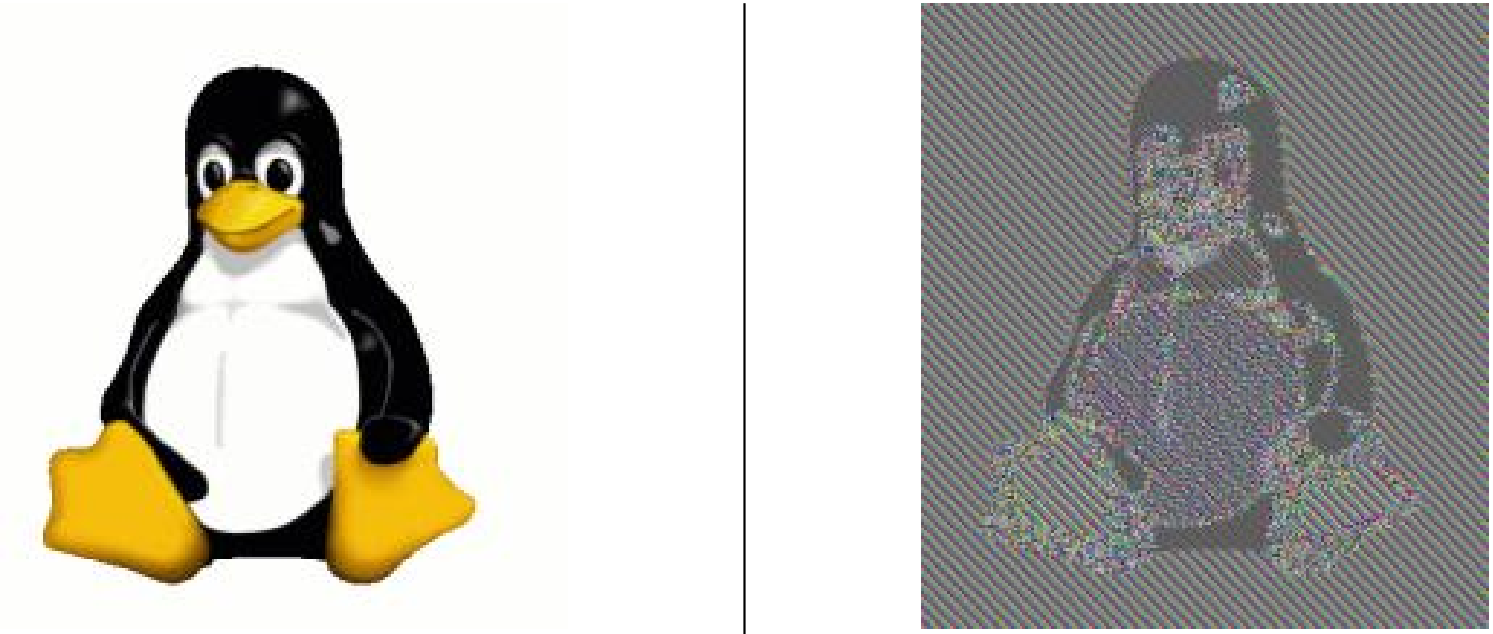
Electronic Codebook (ECB) Mode

- Erinnerung: $\mathrm{E}, \mathrm{D} : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$
- Einfachster Weg, zu verschlüsseln:
  - Teile M in ℓ-Bit-Blöcke $M_1, \ldots \in \{0,1\}^\ell$ auf
  - Setze $C := (C_1, \ldots)$ mit $C_i := \mathrm{E}(K, M_i) \in \{0,1\}^\ell$
  - Entschlüsselung funktioniert genauso, nur mit D
- **Frage:** Vorteile/Nachteile?

Eigenschaften des ECB

- Erinnerung: $C := (C_1, \dots)$ mit $C_i := \mathrm{E}(K, M_i)$
- Vorteile:
  - Einfach zu implementieren
  - Kein Zustands-Update, keine Synchronisation nötig
- Nachteile:
  - Gleiche Nachricht $\Rightarrow$ gleiches Chiffrat
  - Einfügen/Umsortieren von Chiffratblöcken möglich
- **Fun fact:** Bundestrojaner nutzt AES (gängiges E) im ECB-Modus (mit hartkodiertem Schlüssel)

Eigenschaften des ECB (Beispiel)



ECB-Verschlüsselung (links Nachricht, rechts Chiffrat) (Wikipedia)

- **Frage:** Wie können Nachteile behoben werden?

Cipher Block Chaining (CBC) Mode

- Erinnerung: $\mathrm{E}, \mathrm{D} : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$
- Problem des ECB: Chiffratblöcke „unabhängig“
- Idee des CBC: Chiffratblöcke verketten:
  - Teile M in ℓ-Bit-Blöcke $M_1, \ldots \in \{0,1\}^\ell$ auf
  - Setze $C_0 := IV$ (Initialisierungsvektor)
  - Setze $C_i := \mathrm{E}(K, M_i \oplus C_{i-1})$
  - Entschlüsselung: $M_i := \mathrm{D}(K, C_i) \oplus C_{i-1}$
- IV muss mit übertragen werden (oder konstant sein)
- **Frage:** Vorteile/Nachteile?

Eigenschaften des CBC

- Erinnerung: $C_i := \mathrm{E}(K, M_i \oplus C_{i-1})$
- CBC behebt Nachteile des ECB:
  - Gleiche Nachricht $\Rightarrow$ unterschiedliche Chiffrate
    (bei unterschiedlichen vorherigen Chiffraten)
  - Umsortierung von Chiffratblöcken führt zu fehlerhaften Blöcken
  - **Frage:** Welche Blöcke werden genau zerstört?
- Vorteile erkauft mit neuen Nachteilen:
  - *Ver*schlüsselung nicht parallelisierbar (C_{i-1} muss bekannt sein)
  - **Aber:** *Ent*schlüsselung parallelisierbar, (fast) wahlfreier Zugriff (**Frage:** wie?)
  - Chiffrate veränderbar (annähernd XOR-homomorph)

Einsatz des CBC

- Hauptproblem des CBC: „annähernde XOR-Homomorphie“
  - Ändern von C_i ändert entschlüsseltes M_{i+1}
  - Kann gewisse Informationen über M_{i+1} liefern
    (etwa: $M_{i+1} \oplus X$ noch „gültig“)
- Beispiele für konkrete Probleme, die hierdurch entstehen:
  - Angriffe auf TLS (wird noch besprochen)
  - Angriffe auf Linux-Festplattenverschlüsselung
    `http://www.jakoblell.com/blog/2013/12/22/`
    `practical-malleability-attack-against-cbc-encrypted-luks-partitions/`

Weitere Betriebsmodi

- Counter (CTR) Mode (ähnelt Stromchiffre)

$$C_i := \mathrm{E}(K, IV + i) \oplus M_i$$

  - Ähnliche Eigenschaften wie CBC (aber besser parallelisierbar)
  - **Allerdings:** wie CBC auch homomorph veränderbar
- **Deshalb:** Galois Counter Mode (GCM)
  - Authentifizierter CTR Mode (mit „Prüfsumme“)
  - Schützt gegen Manipulation der Chiffrate

Zusammenfassung Betriebsmodi

- Blockchiffre benutzt blockweise Funktion E in Betriebsmodus
- ECB: „rohe“ Funktion E, **nicht benutzen!**
- CBC,CTR: besser, schützt aber nur gegen Lauschangriffe
- GCM: Betriebsmodus der Wahl, noch nicht überall unterstützt
- **Später:** Formalisierung der Sicherheit von CBC/CTR
  - Klärt auch die Wahl von *IV*
- Mehr in Vorlesung „Symmetrische Verschlüsselungsverfahren“

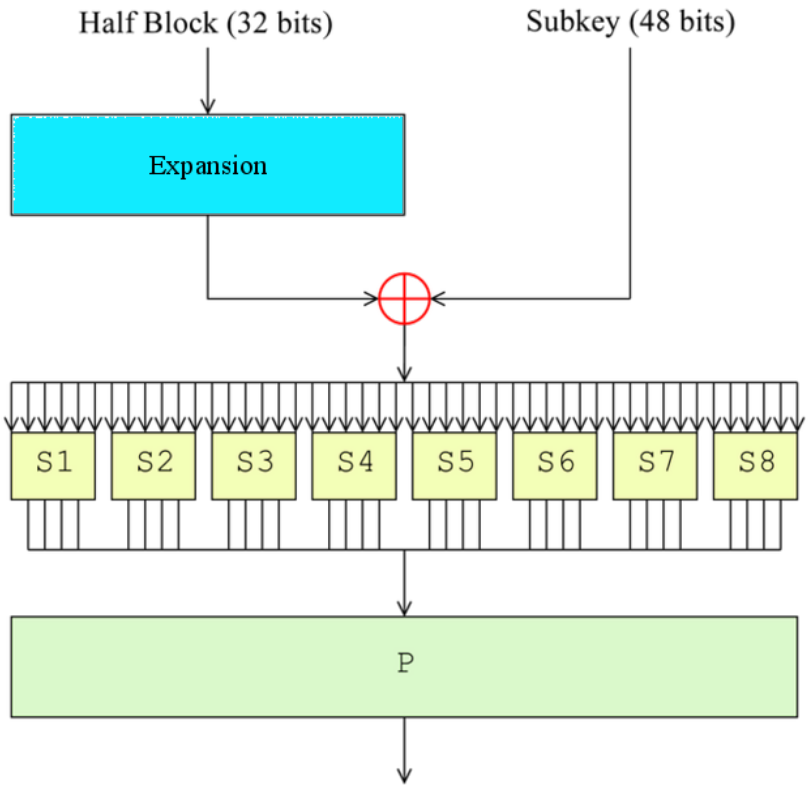
Überblick

Data Encryption Standard (DES)

- Erinnerung: $\mathrm{E} : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$ zentral
- DES: Beispiel für E mit $k = 56$ und $\ell = 64$
- Mittlerweile veraltet (zu kurzer Schlüssel)
- Aber: historisch und technisch interessant
  - Verwendete Feistel-Netzwerke interessante Struktur
  - Rundenfunktion F ohne Falltür $\to$ Falltürfunktion E
  - Strukturell ungebrochen[1]

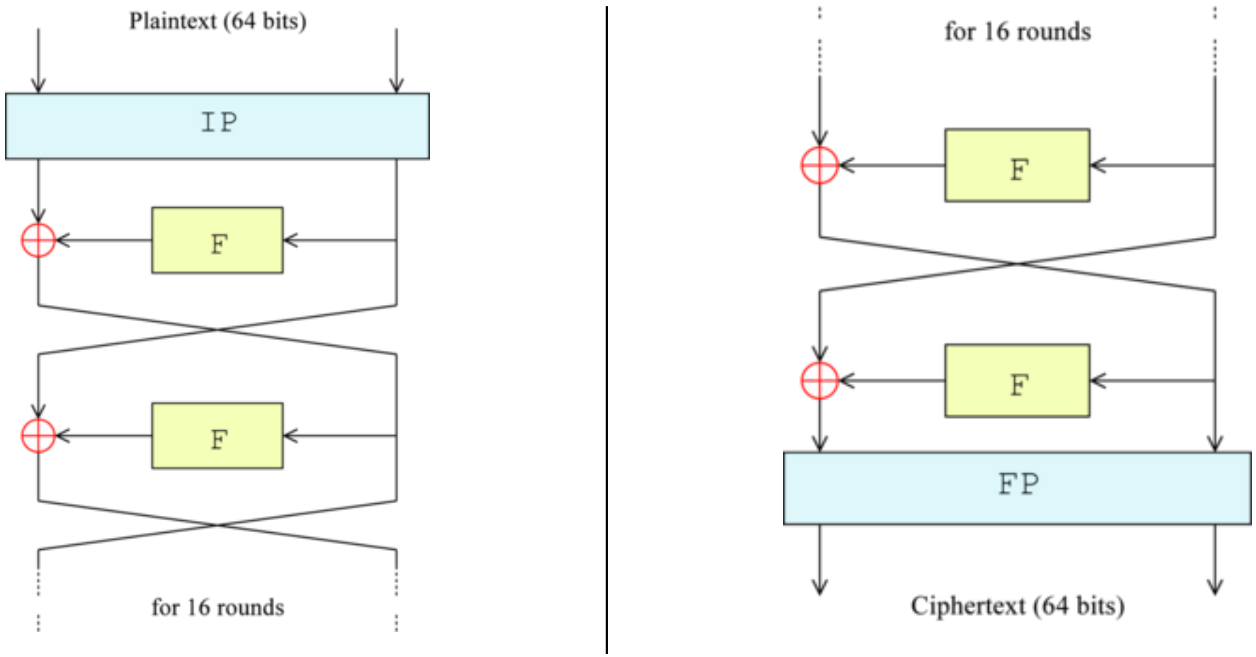
[1] lineare Kryptoanalyse besser als vollständige Suche, aber nicht praktikabel

DES-Rundenfunktion F



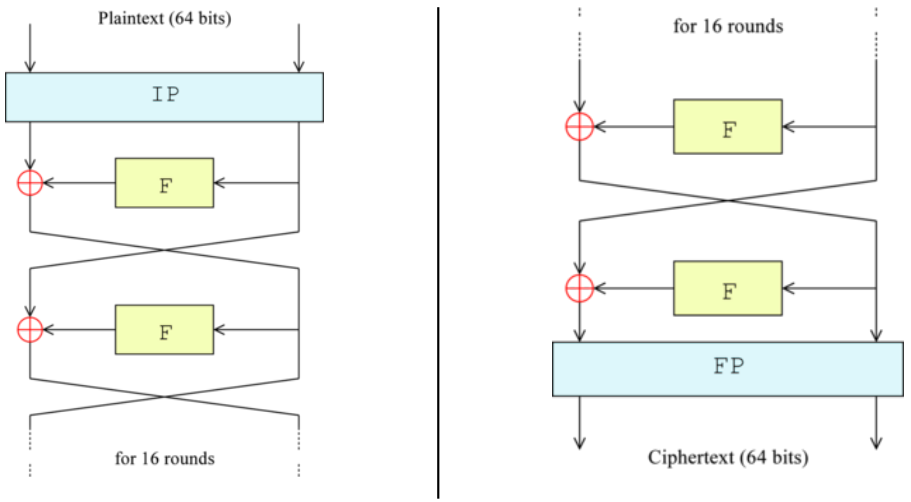
DES-Rundenfunktion $\mathrm{F} : \{0,1\}^{48} \times \{0,1\}^{32} \to \{0,1\}^{32}$ (Wikipedia)

DES-Feistelstruktur



DES-Feistelstruktur (links Anfang, rechts Ende) (Wikipedia)

DES-Feistelstruktur



- Eingangs- und Ausgangspermutation historisch bedingt
- **Entscheidend:** Entschlüsselung muss F nicht invertieren!
- Entschlüsselung hat Feistel-Struktur wie Verschlüsselung
  (nur mit F-Teilschlüsseln in umgekehrter Reihenfolge)

Überblick

2DES

- DES-Schlüssel zu kurz (56 Bits)
- Naive Verbesserung: 2DES
  - $K := (K_1, K_2) \in (\{0,1\}^{56})^2$
  - $\mathrm{E}_{\mathrm{2DES}}(K, M) := \mathrm{E}_{\mathrm{DES}}(K_2, \mathrm{E}_{\mathrm{DES}}(K_1, M))$
  - Erst mit K_1, dann mit K_2 DES-verschlüsseln
- Problem: 2DES nicht wesentlich sicherer als DES

Meet-in-the-Middle-Angriff auf 2DES

- Erinnerung: $\mathrm{E}_{\mathrm{2DES}}(K, M) := \mathrm{E}_{\mathrm{DES}}(K_2, \mathrm{E}_{\mathrm{DES}}(K_1, M))$
- Gegeben: M, $C = \mathrm{E}_{\mathrm{2DES}}(K, M)$, gesucht: $K = (K_1, K_2)$
  1. Berechne Liste aller $C_{K_1'} := \mathrm{E}_{\mathrm{DES}}(K_1', M)$ (d.h. für alle K_1')
  2. Sortiere Liste lexikographisch (damit binäre Suche möglich)
  3. Berechne nacheinander $C_{K_2'} := \mathrm{D}_{\mathrm{DES}}(K_2', C)$
  4. Wenn $C_{K_2'} = C_{K_1'}$, gib (K_1', K_2') aus
- Bei mehreren Kandidaten (K_1', K_2') erneute Suche
- Zeitaufwand $\mathbf{O}(56 \cdot 2^{56})$, Platzbedarf $64 \cdot 2^{56} + \varepsilon$ Bits

3DES

- DES zu unsicher, 2DES nicht so sicher wie erhofft
- Triple-DES (3DES)
  - $K := (K_1, K_2, K_3) \in (\{0,1\}^{56})^3$
  - $\mathrm{E}_{\mathrm{3DES}}(K, M) := \mathrm{E}_{\mathrm{DES}}(K_3, \mathrm{D}_{\mathrm{DES}}(K_2, \mathrm{E}_{\mathrm{DES}}(K_1, M)))$
  - Mit K_1 ver-, mit K_2 ent-, dann mit K_3 verschlüsseln
- Meet-in-the-Middle anwendbar, Aufwand $\mathbf{O}(2^{112})$
- Bessere (aber unpraktikable) Angriffe existieren
- Hauptgrund für Verwendung: benutzt DES als black box

Überblick

Advanced Encryption Standard (AES)

- Erinnerung: $\mathrm{E} : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$ zentral
- AES: Beispiel für E mit $k \in \{128, 192, 256\}$ und $\ell = 128$
- Entwickelt von Daemen und Rijmen, standardisiert 2000
- Keine Feistel-Struktur
- Nach heutigem Kenntnisstand sicher[2]

[2] Strukturelle, aber impraktikable Angriffe existieren