Vorlesung Sicherheit

Dennis Hofheinz

ITI, KIT

08.05.2017

Überblick

Überblick

Erinnerung

- Stromchiffren: „Simulation“ eines OTP
- Blockchiffren:
  - E-Funktion, Betriebsmodi $\rightsquigarrow \mathrm{Enc}$
  - Vor-/Nachteile ECB, CBC, CTR(, GCM)
  - DES: Struktur (Feistel: $\mathrm{F} \rightsquigarrow \mathrm{E}$)
  - DES-Problem: kurze Schlüssel
  - 2DES (unsicherer als man erwarten würde)

Überblick

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

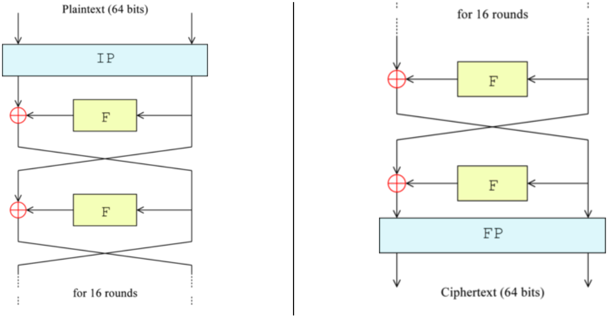
Advanced Encryption Standard (AES)

- 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[1]

[1] Strukturelle, aber impraktikable Angriffe existieren

Überblick

Erinnerung Feistelstruktur



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

Lineare Kryptoanalyse

- Grundidee: finde $\mathbb{F}_2$-lineare Abhängigkeiten zwischen den Bits von X und $Y := \mathrm{E}(K, X)$
  - Beispiel: $X_1 + X_7 + Y_3 + Y_8 + 1 = K_3 + K_{17} \bmod 2$
- Idealer Fall: K aus bekannten (X, Y)-Paaren herleitbar
- Bei Feistel-Verfahren (n Runden) indirekter Angriff möglich:
  1. Finde lineare Abhängigkeiten zwischen F-Ein- und -Ausgabe
  2. Erweitere Abhängigkeiten auf die ersten $n-1$ Feistel-Runden
  3. Vollständige Suche über letzten Rundenschlüssel $K^{(n)}$...
  4. ...überprüfe $K^{(n)}$-Kandidaten mittels linearer Abhängigkeit
  5. Wenn $K^{(n)}$ gefunden, suche nach $K^{(n-1)}$, danach $K^{(n-2)}$, usw.
- Bricht FEAL, bei DES besser als vollständige Suche
  (benötigt aber riesige Anzahl an Klartext-Chiffrat-Paaren)

Differentielle Kryptoanalyse

- Grundidee: betrachte Ausgabedifferenzen $\Delta_{\mathrm{out}} := Y \oplus Y'$ in Abhängigkeit von Eingabedifferenzen $\Delta_{\mathrm{in}} := X \oplus X'$
- Bei bestimmten Eingabedifferenzen (z.B. von S-Boxen) manche Ausgabedifferenzen wahrscheinlicher als andere
- Bei Feistel-Verfahren Angriff ähnlich wie bei linearer Analyse:
  1. Finde wahrscheinliche Paare $\Delta_{\mathrm{in}} \Rightarrow \Delta_{\mathrm{out}}$ zwischen Eingabe und Ausgabe von vorletzter Runde
  2. Vollständige Suche über letzten Rundenschlüssel $K^{(n)}$...
  3. ...überprüfe $K^{(n)}$-Kandidaten auf $\Delta_{\mathrm{in}} \Rightarrow \Delta_{\mathrm{out}}$-Konsistenz
- DES resistent gegen differentielle Analyse, FEAL nicht

Überblick

Überblick

Motivation

- **Diskussion:** Was soll eigentlich erreicht werden?
  Der Einfachheit halber Beschränkung auf passive Sicherheit/Angriffe

Semantische Sicherheit

- Semantische Sicherheit (Goldwasser und Micali 1982)

  Quelle: Wikipedia

- Idee: Chiffrat hilft nicht bei Berechnungen über Klartext
- Oder: alles, was *mit* C (effizient) über M berechnet werden kann, kann auch (effizient) ohne Chiffrat berechnet werden
- **Wichtig:** deckt so nur passive Angriffe ab

Semantische Sicherheit

Definition (Semantische Sicherheit, informell)

Ein symmetrisches Verschlüsselungsverfahren ist semantisch sicher, wenn es für jede M-Verteilung von Nachrichten gleicher Länge, jede Funktion f und jeden effizienten Algorithmus $\mathcal{A}$ einen effizienten Algorithmus $\mathcal{B}$ gibt, so dass

$$\Pr\left[\mathcal{A}^{\mathrm{Enc}(K,\cdot)}(\mathrm{Enc}(K,M)) = f(M)\right] - \Pr\left[\mathcal{B}(\varepsilon) = f(M)\right]$$

klein ist.

- Technisch recht unhandlich (Quantoren!), aber handlichere äquivalente Begriffe (IND-CPA-Sicherheit) existieren
- Existenz von (mehrfach benutzbaren) semantisch sicheren Verfahren impliziert $P \neq NP$

Passive Sicherheit: IND-CPA

- IND-CPA: indistinguishability under chosen-plaintext attacks
- Schema IND-CPA-sicher $\Leftrightarrow$ kein effizienter Angreifer $\mathcal{A}$ kann Chiffrate von selbstgewählten Klartexten unterscheiden
  1. $\mathcal{A}$ erhält im Folgenden Zugriff auf $\mathrm{Enc}(K,\cdot)$-Orakel
  2. $\mathcal{A}$ wählt zwei Nachrichten $M^{(1)}, M^{(2)}$ gleicher Länge
  3. $\mathcal{A}$ erhält $C^* := \mathrm{Enc}(K, M^{(b)})$ für gleichverteiltes $b \in \{1,2\}$
  4. $\mathcal{A}$ gewinnt, wenn er b richtig rät
- $\forall\mathcal{A} : (\Pr[\mathcal{A} \text{ gewinnt}] - 1/2)$ klein $:\Leftrightarrow$ Schema IND-CPA-sicher
- Idee: Chiffrate ununterscheidbar (z.B. von Zufallschiffraten)

Theorem (Semantische Sicherheit $\Leftrightarrow$ IND-CPA, ohne Beweis)

Ein symmetrisches Verschlüsselungsverfahren ist genau dann semantisch sicher, wenn es IND-CPA-sicher ist.

Überblick

Beispiel: ECB Mode

- Erinnerung ECB: $C_i := \mathrm{E}(K, M_i)$

Theorem

Keine Blockchiffre ist im ECB Mode IND-CPA-sicher.

Beweis.

Betrachte folgendes $\mathcal{A}$:

1. $\mathcal{A}$ wählt $M^{(1)} \neq M^{(2)}$ beliebig
2. $\mathcal{A}$ erhält $C^* := \mathrm{Enc}(K, M^{(b)})$
3. $\mathcal{A}$ erfragt $C^{(1)} := \mathrm{Enc}(K, M^{(1)})$
4. $\mathcal{A}$ gibt 1 aus gdw. $C^* = C^{(1)}$

$\Pr[\mathcal{A} \text{ gewinnt}] = 1 \Rightarrow$ Schema IND-CPA-unsicher □

- Nutzt aus, dass „gleiche Nachricht $\Rightarrow$ gleiches Chiffrat“ gilt

Beispiel: CBC Mode

- Erinnerung CBC: $C_i := \mathrm{E}(K, M_i \oplus C_{i-1})$, wobei $C_0 := IV$
- ECB-Angriff funktioniert nicht bei CBC, wenn IV zufällig und für jedes Chiffrat neu gewählt
  - Gleiche Nachricht $\not\Leftrightarrow$ gleiches Chiffrat
- Annahme im Folgenden: IV für jede Verschlüsselung neu gleichverteilt gewählt und dem Chiffrat beigefügt
  - **Allerdings:** macht aktive Angriffe noch leichter (IV veränderbar)

Beispiel: CBC Mode

Theorem (IND-CPA-Sicherheit des CBC Mode, informell)

Ist $\mathrm{E}(K, \cdot) : \{0,1\}^\ell \to \{0,1\}^\ell$ *für zufälliges* K *ununterscheidbar von einer Zufallsfunktion* $\mathrm{R} : \{0,1\}^\ell \to \{0,1\}^\ell$*, dann ist die oben beschriebene Blockchiffre im CBC Mode IND-CPA-sicher.*

- **Bemerkung:** Analoge Aussage gilt auch für CTR und GCM

Beweisidee (Reduktion).

Baue aus IND-CPA-Angreifer $\mathcal{A}$ einen E/R-Unterscheider $\mathcal{B}$. □

- **Achtung:** zeigt noch nicht, dass E ununterscheidbar von R

Beispiel: Stromchiffren

- Erinnerung: Stromchiffre nutzt $\mathrm{SC} : \{0,1\}^k \to \{0,1\} \times \{0,1\}^k$
- Annahme: K-Update über mehrere Verschlüsselungen hinweg

Theorem (IND-CPA-Sicherheit von Stromchiffren, informell)

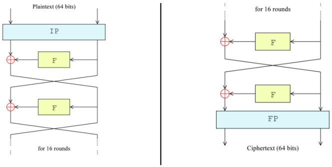
Ist $\mathrm{SC}(K)$ für zufälliges K ununterscheidbar von zufälligem $U_{\{0,1\}\times\{0,1\}^k} \in \{0,1\} \times \{0,1\}^k$, dann ist die entstehende zustandsbehaftete Stromchiffre IND-CPA-sicher.

Beweisidee (Reduktion).

Baue aus einem beliebigem gegebenem IND-CPA-Angreifer $\mathcal{A}$ einen $\mathrm{SC}(K)/U_{\{0,1\}\times\{0,1\}^k}$-Unterscheider $\mathcal{B}$ $\square$

Überblick

Weiteres Beispiel: Feistel-Schema



Theorem (Sicherheit des Feistel-Schemas, informell)

Ist $\mathrm{F}(K,\cdot)$ *(mit zufälligem* K*) ununterscheidbar von einer zufälligen Funktion, dann ist* $\mathrm{Enc}(K,\cdot)$ *ununterscheidbar von einer zufälligen invertierbaren Funktion.*

Überblick

Zusammenfassung symmetrische Verschlüsselung

- Stromchiffren schneller, aber weniger gute Kandidaten
- Struktur von Blockchiffren reichhaltiger
  - E-Funktion, Betriebsmodi
  - Feistel-Struktur (DES)
  - Angriffsstrategien: lineare/differentielle Kryptoanalyse
- Sicherheit grundsätzlich formalisierbar
  - Sicherheitsreduktion erlaubt es, sich auf zugrundeliegende Bausteine (z.B. E) zu konzentrieren
  - Aber: sichere Verschlüsselung impliziert $P \neq NP$
    (Ausnahme: begrenzte Nachrichtenlänge/unbegrenzter Schlüssel)

Aktuelle Forschung

- Algebraische Kryptoanalyse
  1. Drücke Rundenfunktion algebraisch ($\mathbb{F}_2$-GLS) aus
  2. Versuche, algebraische Abhängigkeiten zu erweitern
  3. Löse entstehendes GLS z.B. mit Gröbnerbasisalgorithmen
- Alternativen zu Feistel (Sponge Functions)
- Seitenkanalangriffe und Leakage Resilience
  - Beispiel: Zeitmessung(en) von Verschlüsselungen
  - Laufzeit hängt subtil von verwendetem Schlüssel ab
  - Manchmal statistische Analyse möglich
  - Aber: manchmal algorithmische Gegenmaßnahmen möglich

Überblick

Überblick

Was ist eine Hashfunktion?

- Kurzer „Fingerabdruck“ großer Daten:

$$\mathrm{H} : \{0,1\}^* \rightarrow \{0,1\}^k$$

- Anwendungen:
  - Vergleich großer Dateien: überprüfe, ob Download korrekt war
  - (vorgreifend) Signaturen: signiere $\mathrm{H}(M)$ anstellt von M
  - Generell wichtiger kryptographischer Baustein

    Beispiel (vorgreifend): aktiv sichere Verschlüsselung

Anforderungen an eine Hashfunktion

- Kurzer „Fingerabdruck“ großer Daten:

$$\mathrm{H} : \{0,1\}^* \to \{0,1\}^k$$

- Keine „Kollisionen“ ($X \neq X'$ mit $\mathrm{H}(X) = \mathrm{H}(X')$)
- Für unsere Zwecke (kryptographische Hashfunktion):

Definition (Kollisionsresistenz, informell)

Eine Hashfunktion H ist *kollisionsresistent*, wenn jeder effiziente Algorithmus nur mit kleiner Wahrscheinlichkeit eine Kollision findet.