Betriebsarten für Blockchiffren

Prof. Dr. Rüdiger Weis

TFH Berlin

Sommersemester 2008

Betriebsarten für Blockchiffren

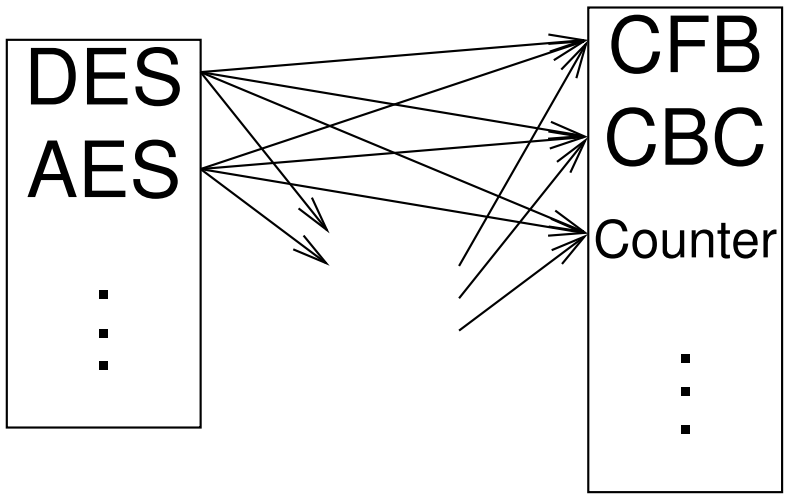
Was ist eine Betriebsart (engl. “Mode of Operation”)?

Blockchiffre wird genutzt, um etwas anderes zu realisieren:

- Z.B. eine Flusschiffre (synchrone oder selbstsynchr.),
- oder eine Blockchiffre (mit anderen Parametern),
- oder einen “Message Authentication Code” ($\rightarrow$ nächstes Kapitel),
- oder ...

Was ist eine Betriebsart?

- Die grundsätzliche Arbeitsweise einer Betriebsart ist unabhängig davon, welche Blockchiffre man verwendet.



Sicherheitsbeweise

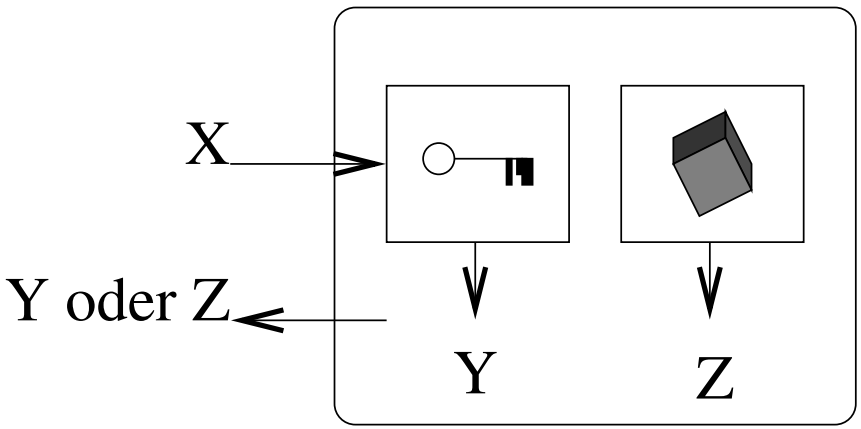
- Die Sicherheit einer Betriebsart kann man in vielen Fällen formal beweisen. Jeder Sicherheitsbeweis setzt voraus, dass die verwendete Blockchiffre selbst bestimmte Sicherheitskriterien erfüllt.
- Umgekehrt kann es Angriffe auf Betriebsarten geben, die nicht auf irgend einer Schwäche der verwendeten Blockchiffre beruhen.
  (Derartige Betriebsarten will man natürlich vermeiden . . .)

Flusschiffren aus Blockchiffren

Angriffsmodell I: “real or random”

Der Angreifer soll unterscheiden, ob ein gegebener Chiffretext einem von ihm gewählten Klartext entspricht (“real”) oder zufällig ist (“random”).
Dazu stellt er eine Testanfrage.

1 Phase: real or random



X: Klartext (vom Angreifer gewählt)
Y: Chiffretext
Z: Zufälliger Bit-String; gleichlang wie Y.

Der Angreifer soll entscheiden, ob er Y oder Z erhalten hat.

Warum diese neue Definition?

Wir kennen bisher nur *binäre additive* Flusschiffren, basierend auf PZG.

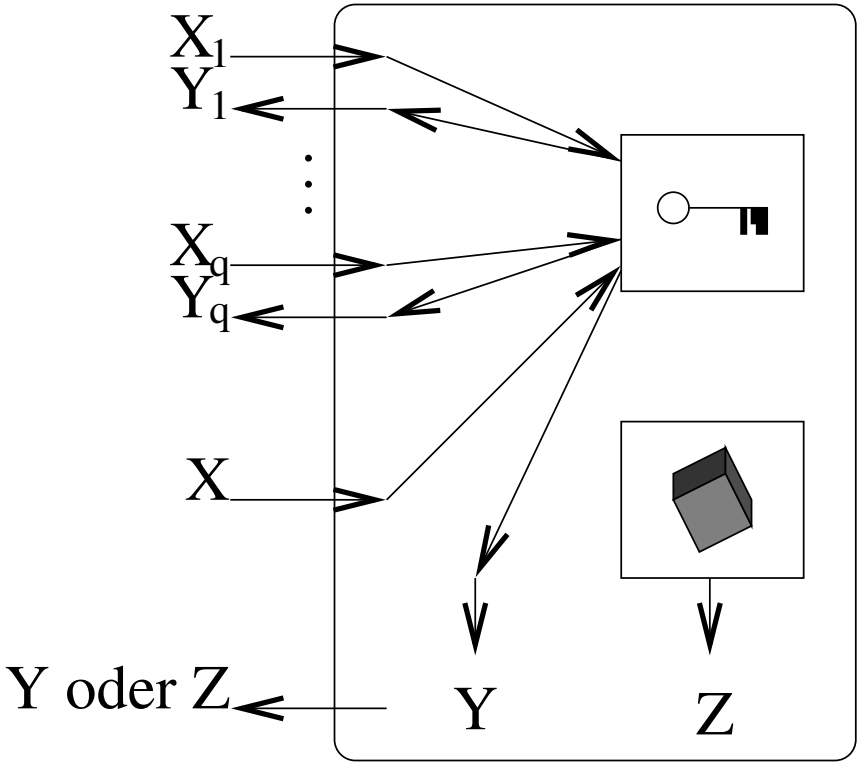
Man sieht leicht, dass deren Sicherheit äquivalent zur Sicherheit des verwendeten PZG ist.

Angriffsmodell I erlaubt es, auch die Sicherheit von anderen Flusschiffren zu untersuchen.

Angriffsmodel II: “real or random” mit zwei Phasen

1. Das Orakel wählt zufällig einen geheimen Schlüssel.
2. Fragephase: Der Angreifer wählt einen Klartext X_1, das Orakel antwortet mit dem (durch ehrliche Verschlüsselung gewonnenen) Chiffretext Y_1, der Angreifer wählt X_2, das Orakel antwortet mit Y_2, ...
3. Der Angreifer stellt wie bisher eine Testanfrage.
   D.h., er wählt den Klartext X, das Orakel antwortet entweder mit dem entsprechenden Chiffretext Y oder mit einem gleichlangen Zufallsstring Z.

Angriffsmodell II (2)



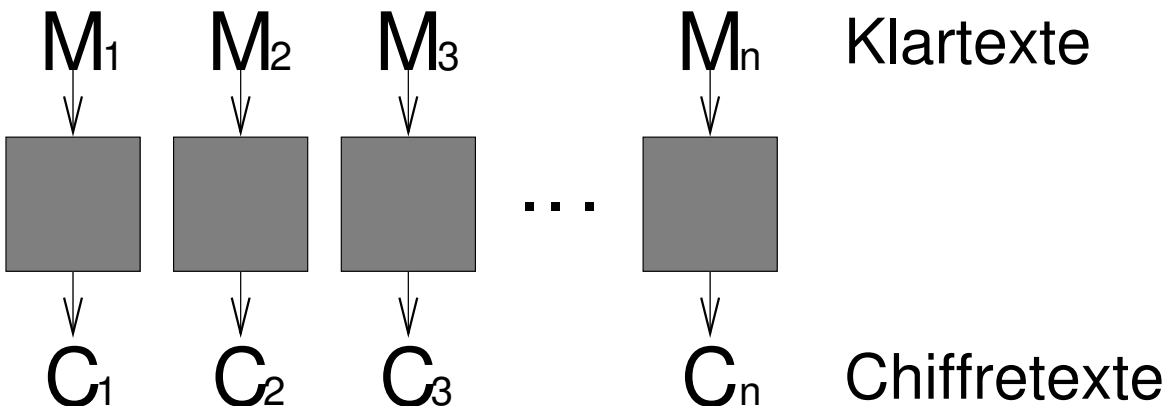
Bemerkung

Auch eine binäre additive Flusschiffre kann im Sinne des erweiterten Angriffsmodells sicher sein.

Nur muss sie zwischen den einzelnen Orakelfragen ihren internen Zustand beibehalten.

D.h., die ersten $|X_1|$ bit des Schlüsselstroms werden zur Berechnung von Y_1 herangezogen, die nächsten $|X_2|$ bit des Schlüsselstroms zur Berechnung von Y_2, ...

ECB (Electronic Codebook)



$$C_i := E_K(M_i).$$

Sicherheit des ECB Modus

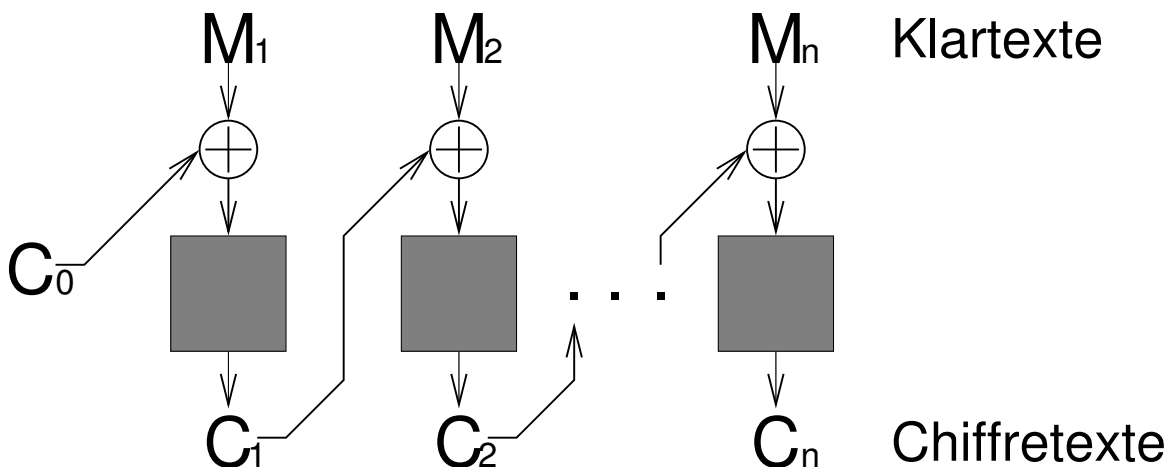
Der ECB Modus ist einer von vier im Zusammenhang mit dem DES „offiziell standardisierten“ “Modes of Operation”.

Der ECB Modus ist unsicher! (Warum?)

In vielen *schlechten* Krypto-Produkten wird der ECB-Modus trotzdem verwendet (da formal standardkonform).

Cipher Block Chaining (CBC)

Verknüpfen des Klartext-Blocks M_i mit schon bekannten Chiffretext-Block C_{i-1}; dann erst Anwenden von E:



CBC Modus

Verschlüsseln der Nachricht $(M_1, \ldots, M_n)$:

1. Wähle zufällig C_0.
2. Für $1 \leq i \leq n$: $C_i := E_K(M_i \oplus C_{i-1})$.
3. Gib den Chiffretext: $(C_0, \ldots, C_n)$ aus.

C_0 wird als "Initial Value" (IV) bezeichnet.
Wie entschlüsselt man?

Eigenschaften des CBC Modus

Der CBC Modus verhält sich wie eine *selbstsynchr.* Flusschiffre. (Warum?)

Sicherheit gegen Angriffsmodell II verlangt hier nicht, den internen Zustand beizubehalten. (Warum?)

Sicherheitsproblem in der Praxis:

Eine Anwendung, die eine Blockchiffre im CBC Modus einsetzt und bei der der Angreifer die Wiederverwendung eines schon zuvor benutzten IV erreichen kann, sollte als unsicher betrachtet werden. (Warum?)

Output FeedBack (OFB)

Erzeugt pseudozufällig einen Schlüsselstrom

$$V_0 = \mathrm{IV}, \quad V_i = E_K(V_{i-1})$$

arbeitet wie eine binäre additive Flußchiffre:

$$C_0 := V_0, \quad C_i = V_i \oplus M_i, \quad M_i = V_i \oplus C_i (1 \leq i \leq n).$$

(Insbesondere verhält sich der OFB Modus also wie eine synchrone Flusschiffre.)

Cipher FeedBack (CFB)

Analog zum OFB Modus, nur wird der Chiffretext als Eingabe für E genutzt.
Erzeugt aus dem Schlüssel K und dem (nicht notw. geh.) IV einen pseudozufälligen Bit-Strom:

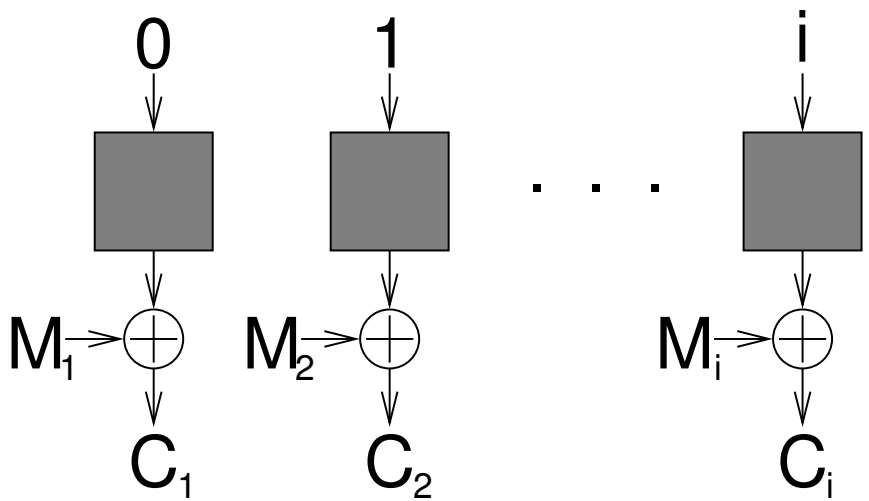
$$C_0 = \mathrm{IV}, \quad C_i = E_K(C_{i-1}) \oplus M_i \quad M_i = E_K(C_{i-1}) \oplus C_i.$$

Verhält sich der CFB Modus wie eine synchrone oder eine selbstsynchronisierende Flusschiffre?

Der Counter-Modus

Analog zum OFB Modus (und anderen binären additiven Flusschiffren) wird ein Klartext-unabhängiger Schlüsselstrom erzeugt und mit den Klartexten XOR-Verknüpft:

$$C_i := M_i \oplus E_K(i)$$



Der Counter-Modus (2)

Anders ausgedrückt: Der PZPG wird als PZG benutzt. Unter dem Schlüssel K wird der folgende Schlüsselstrom erzeugt: $E_K(0), E_K(1), E_K(2), \ldots$

Die als Input auftretenden nichtnegativen Zahlen werden wie üblich als Binärzahlen dargestellt.

Eine Variante des Counter-Modus

Sei E_K als Fkt. (bzw. Perm.) $E_K : \{0,1\}^b \to \{0,1\}^b$ definiert.

Die Nachricht $M = (M_1, \ldots, M_n) \in (\{0,1\}^b)^n$ wird wie folgt verschlüsselt:

1. Wähle $C_0 \in \{0, \ldots, 2^b - 1\}$ zufällig.
2. Für $0 \leq i \leq n-1$: $C_i := M_i \oplus E_K(C_0 + i \bmod 2^b)$
3. Chiffretext $(C_0, \ldots, C_n)$.

Offiziell standardisierte Betriebsarten

Im Zusammenhang mit dem DES:

ECB, CBC, OFB und CFB.

Im Zusammenhang mit dem AES:

ECB, CBC, OFB, CFB und Counter-Modus.

Das NIST hat die Standardisierung weiterer Modes of Operation angekündigt.

Abstrakte und konkrete Sicherheit des Counter-Modus

Man kann den PZPG E auch als PZFG F auffassen:
$F_K(X) = E_K(X)$.

Theorem

Wenn der PZPG E sicher gegen Chosen Plaintext Angriffe ist, dann ist der PZFG F sicher.

Beweis-Idee: Geburtstagsparadoxon! □

Abstrakte Sicherheit des Counter-Modus

- *Wenn der PZPG E sicher gegen Chosen Plaintext Angriffe ist, dann ist der Counter-Modus eine sichere Flusschiffre.*

Ein Angriff

Theorem ([ACM]: Angriff auf den Counter-Modus)

Seien $n > 4$ *und* $E : \{0,1\}^n \to \{0,1\}^n$ *eine Zufallsperm.*

Es gibt einen Chosen Plaintext Angriff mit $2^{n/2}$ *Orakelfragen, der* E *mit einem Vorteil über* $1/4$ *von einer Zufallsfunktion unterscheidet.*

Folgerung: Grenzen des Counter-Modus

[Grenze der Sicherheit des Counter-Modus]

Die Länge eines mit den Counter-Modus erzeugten Schlüsselstroms sollte

$$\underline{\textit{immer weit unter } n * 2^{n/2} \textit{ bit}}$$

liegen.

Bemerkung: Länge des Schlüsselstr.: $n * 2^{n/2}$.
Vorteil: $> \frac{1}{4}$.

Beispiel: DES ($\rightarrow$ Tafel)

Problematik

- Abstrakte Sicherheitsresultate weisen nur nach, dass ein System sicher ist, wenn die Sicherheitsparameter (hier: Schlüsselgröße und Blocklänge) *groß genug* sind!
  (Das zu wissen ist natürlich schon viel wert!)
- Aber: Wie groß ist „groß genug“?
- Schlimmer noch: Bei gegebenen Blockchiffren (z.B. DES) sind Schlüsselgröße und Blocklänge vorgegeben.

Die Konkrete Sicherheit des Counter-Modus

Theorem ([KSCM])

Eine Zufallspermutation $E : \{0,1\}^n \to \{0,1\}^n$ kann von einem Angreifer, der q Chosen Plaintext Orakelfragen stellt, höchstens mit dem Vorteil $q^2/2^{n+1}$ von einer Zufallsfunktion unterschieden werden.

Beweis: ($\to$ Übung)

Folgerung

Theorem

Sei E eine Zufallspermutation über $\{0,1\}^n$. Wir betrachten die Flusschiffre E-Counter und einen Angreifer entsprechend **Angriffsmodell I**. *Wenn die Testanfrage X maximal l bit lang ist, gilt:*
Kein Angreifer kann einen Vorteil über

$$\frac{\lceil l/n \rceil^2}{2^{n+1}}$$

erzielen.

Beweis: Folgerung aus Satz [KSCM]. □

Bemerkungen

- Aus Satz [ACM] ergibt sich, dass die genannte Grenze scharf ist (d.h., obere und untere Schranke liegen sehr dicht beisammen).
- Für **Angriffsmodell II**, d.h., bezogen auf Angreifer, bei denen die Orakelfragen $X_1, \ldots, X_q$ und die Testanfrage X zusammen nicht mehr als l bit lang sind, kann man ein analoges Resultat nachweisen. (Das Orakel behält zwischen den Orakelfragen seinen internen Zustand bei.)

Blockchiffren aus Blockchiffren

- Größere Blocks (und ggf. längere Schlüssel).
  - Der AES-Kandidat DEAL
- Gleiche Blockgröße, längere Schlüssel.
  - Doppel- und Dreifachverschlüsselung (kennen wir schon).
  - Die DESX-Konstruktion.

Grundsätzliches Problem

Gegeben sei eine Blockchiffre E, deren (effektive) Schlüssellänge zu kurz ist (z.B. E=DES).

Gesucht ist eine Betriebsart für E, die nachweislich eine erhebliche Steigerung der effektiven Schlüssellänge erreicht. (Double-DES, Triple-DES, ...)

Die Betrachtung der abstrakten Sicherheit hilft hier überhaupt nicht weiter. (Warum?)

Beobachtungen und Vermutungen

- Doppelte Verschlüsselung trägt nicht zu einer (erheblichen) Steigerung der effektiven Schlüssellänge bei ($\rightarrow$ MITM Angriffe).
- Dreifache Verschlüsselung mit nur zwei Schlüsseln (wie bei Two-Key Triple-DES) ist verwundbar gegen Angriffe, die mit extrem vielen dem Angreifer bekannten Klartext-Chiffretext Paaren arbeiten.

Beobachtungen und Vermutungen (2)

- Man vermutet, dass dreifache Verschlüsselung tatsächlich eine signifikante Steigerung der effektiven Schlüssellänge erlaubt.
- Man vermutet auch, dass dreifache Verschlüsselung mit nur zwei Schlüsseln (wie bei Two-Key Triple-DES) eine signifikante Steigerung der Schlüssellänge erlaubt, *wenn* die Anzahl der Texte, die unter einem Schlüssel ver- oder entschlüsselt werden, beschränkt ist.

Diese Vermutungen zu beweisen oder zu widerlegen ist ein ungelöstes Problem!!!

Die DESX-Konstruktion

Sei E eine n-bit Blockchiffre mit k-bit Schlüsseln. Die Blockchiffre EX ist eine b-bit Blockchiffre mit $(k + 2b)$-bit Schlüsseln:

$$EX_{K,L_1,L_2}(M) = E_K(M \oplus L_1) \oplus L_2.$$

Die Blockchiffre Frugal-EX (oder "EX'") ist eine b-bit Blockchiffre mit $(k + b)$-bit Schlüsseln:

$$\text{Frugal-}EX_{K,L}(M) = E_K(M \oplus L) \oplus L.$$

DESX (Geschichte)

Die DESX-Konstruktion wurde um 1985 von Rivest in der Absicht vorgeschlagen, die effektive Schlüssellänge des DES zu steigern und trotzdem annähernd die Geschwindigkeit von Single DES zu erreichen.

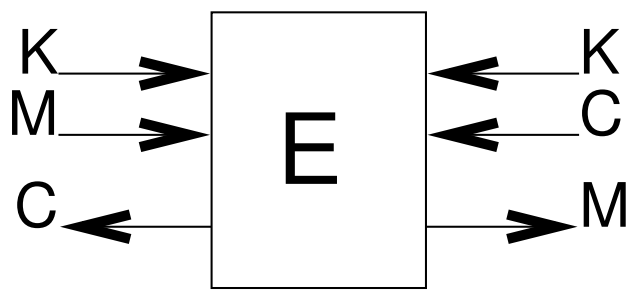
1996 gelang es Kilian und Rogaway im Rahmen eines formalen Modells, einen Sicherheitsbeweis für die DESX-Konstruktion zu führen – während ähnliche Versuche für Dreifach-Verschlüsselung bisher fehlschlugen.

Das Shannon-Modell für Blockchiffren

Der Angreifer betrachtet eine (bekannte) Blockchiffre mit k-bit Schlüsseln als Familie von 2^k Zufallspermutationen, auf die über ein Orakel zugegriffen wird.

Der Angreifer ruft ein Orakel E mit den Parametern (K, M) auf, um den Chiffretext $C = E_K(M)$ zu erfahren.

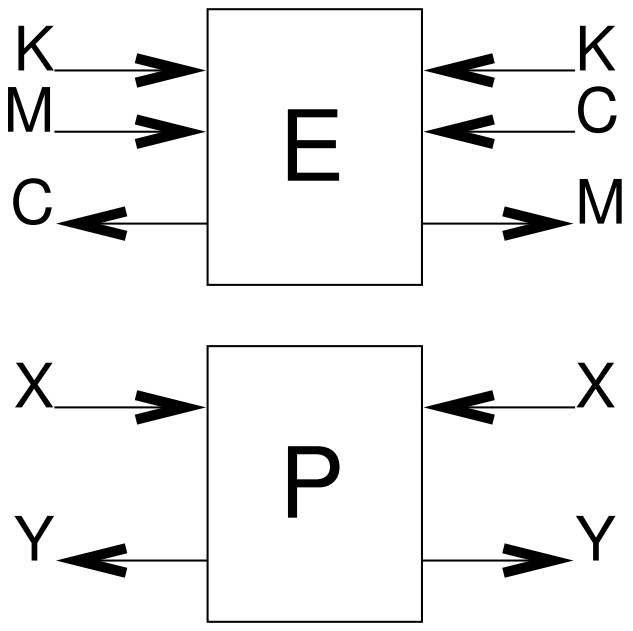
Er ruft ein Orakel E^{-1} mit den Parametern (K, C) auf, um den Klartext $M = E_K^{-1}(C)$ zu erfahren.



Anwendung des Shannon-Modells auf DESX

Der Angreifer hat Zugriff auf die E- und E^{-1}-Orakel, sowie auf Orakel, die eine Permutation $P : \{0,1\}^n \rightarrow \{0,1\}^n$ und deren Umkehrung P^{-1} realisieren.

Er soll entscheiden, ob P eine Zufallspermutation ist, oder intern entsprechend der DESX- bzw. Frugal-DESX-Konstruktion dargestellt wird.



Sicherheit von DESX

Theorem

Sei E eine n-bit Blockchiffre mit k-bit Schlüsseln.

Man betrachte einen Angreifer, der e Fragen an die $E^{\pm 1}$ Orakel stellt, und höchstens p Fragen an die $P^{\pm 1}$ Orakel.

Der Angreifer versucht, zu entscheiden, ob P eine Zufallspermutation ist, oder eine DESX Chiffre.

Der Vorteil dieses Angreifers beträgt maximal

$$\frac{2ep}{2^{k+n}}.$$

Konsequenzen für die Sicherheit der DESX-Konstruktion

- p: Anzahl der unter einem Schlüssel zu verarbeitenden Klar- und Chiffretexte
  ($*$ unter Kontrolle des *Sicherheitsarchitekten* einer Anwendung $*$)
- E: n-bit Blockchiffre mit k-bit Schlüsseln, die bestimmten „vernünftigen“ Sicherheitskriterien gerecht wird
- e: Rechenaufwand des Angreifers
- $2ep/2^{k+n}$: maximaler Vorteil des Angreifers

Beispiel auf der Basis von DES

Setze $n = 64$ und $k = 55$. (Warum *nicht* $k = 56$?)

Sei $p \leq 2^{32}$. Sei $e \leq 2^{80}$.

Der Vorteil eines Angreifers beträgt maximal

$$2 * 2^{32+80}/2^{55+64} = \frac{2^{113}}{2^{119}} = 2^{-6}.$$

Sicherheit von Frugal-DESX

Theorem

Sei E eine n-bit Blockchiffre mit k-bit Schlüsseln.

Man betrachte einen Angreifer, der e Fragen an die $E^{\pm 1}$ Orakel stellt, und höchstens p Fragen an die $P^{\pm 1}$ Orakel.

Der Angreifer versucht, zu entscheiden, ob P eine Zufallspermutation ist, oder eine Frugal-DESX Chiffre.

Der Vorteil dieses Angreifers beträgt maximal

$$\frac{2ep}{2^{k+n}}.$$

Schlussbemerkungen

Bemerkungen:

- Sowohl für die DESX-Konstruktion als auch für die Frugal-DESX Konstruktion gibt es Angriffe, die die in den entsprechenden Sätzen angegebenen Schranken für den Vorteil fast erreichen. Die angegeben Schranken sind also scharf!
- Überraschenderweise ist die Frugal-DESX Konstruktion ähnlich sicher wie die DESX-Konstruktion.

Beweise:

- Auf die Beweis für die Sätze über die Sicherheit von DESX und FrugalDESX verzichten wir hier.

Danksagungen

Danksagungen

- Nach einer Vorlesung von Stefan Lucks
- Erstellt mit Freier Software