Sichere Implementierung

Prof. Dr. rer. nat. Rüdiger Weis

TFH Berlin

Wintersemester 2007/2008

Von den Grundbausteinen zu sicheren Systemen

- Vorlesung bisher:
  - **Bausteine** für Kryptosysteme.

Von den Grundbausteinen zu sicheren Systemen

- Vorlesung bisher:
  - **Bausteine** für Kryptosysteme.
- Dieses Kapitel:
  - Naiver Einsatz der Bausteine kann gefährlich sein!
  - Beschreibung von Angriffszenarien, denen *gute* Kryptosysteme widerstehen können.

Von den Grundbausteinen zu sicheren Systemen

- Vorlesung bisher:
  - **Bausteine** für Kryptosysteme.
- Dieses Kapitel:
  - Naiver Einsatz der Bausteine kann gefährlich sein!
  - Beschreibung von Angriffszenarien, denen *gute* Kryptosysteme widerstehen können.
- Rest der Vorlesung:
  - Wie setzt man die Bausteine, die wir haben, sinnvoll ein?

Naiver Einstatz der PK-Kryptographie

- Öffentlicher Schlüssel (z.B. RSA: $\mathbf{n}, \mathbf{e}$)
- Klartext $\mathbf{X}$
- Öffentliche Verschlüsselungsfunktion $f(x)$
  (RSA: $f(x) = x^{\mathbf{e}} \bmod \mathbf{n}$)
- Chiffretext $\mathbf{Y} = f(\mathbf{X})$

Naiver Einstatz der PK-Kryptographie

- Öffentlicher Schlüssel (z.B. RSA: $\mathbf{n}, \mathbf{e}$)
- Klartext $\mathbf{X}$
- Öffentliche Verschlüsselungsfunktion $f(x)$
  (RSA: $f(x) = x^{\mathbf{e}} \bmod \mathbf{n}$)
- Chiffretext $\mathbf{Y} = f(\mathbf{X})$

Rate-Verifikationsangriff:

- Geben: Chiffretext $\mathbf{Y}$, gesucht Klartext $\mathbf{X}$.
- Für alle plausiblen $(*)$ Klartexte $\mathbf{X}^*$:
  - Wenn $f(X^*) = \mathbf{Y}$, dann gib $\mathbf{X}^*$ aus und STOP.

Der Angriff ist effizient, wenn **Entropie** des Klartextes klein ist.

Diskussion

- Der Rate-Verifikationsangriff ist ein sehr ernstes Problem – Nachrichten mit geringer Entropie kommen in der Praxis häufig vor.
- Ein Angreifer kann leicht eine begrenzte Anzahl (z.B., einige Milliarden) plausibler Klartexte zu generieren und zu überprüfen, ob einer davon der gesuchte Klartext ist.

Diskussion

- Der Rate-Verifikationsangriff ist ein sehr ernstes Problem – Nachrichten mit geringer Entropie kommen in der Praxis häufig vor.
- Ein Angreifer kann leicht eine begrenzte Anzahl (z.B., einige Milliarden) plausibler Klartexte zu generieren und zu überprüfen, ob einer davon der gesuchte Klartext ist.

  → **Gute Public-Key Verschlüsselung muss die Nachrichten deshalb randomisieren.**

Beispiel

- Sei Y eine verschlüsselte Nachricht an einen Aktienhändler. Absender ist ein hochrangiger Mitarbeiter von Merck. Handelt es sich um die Anweisung, die Schering-Aktien im Besitz der Firma Merck zu verkaufen?
- Man stelle sich ein Programm vor, das einige Millionen oder Milliarden Variationen der Anweisung "Schering verkaufen" erzeugt und verschlüsselt. Zu den generierten Klartexten würden z.B. gehören
  - "verkaufe die Schering-Aktien"
  - "verkaufe alle Schering-Aktien"
  - "verkaufe Schering-Aktien"
  - "verkaufe die Scherig-Aktien" (Tippfehler!)
  - "verkaufe alle Scherig-Aktien"
  - "verkaufe Scherig-Aktien"

Naiver Einsatz von RSA-Unterschriften

- Verifikationsschlüssel: $\mathbf{n}, \mathbf{e}$

Naiver Einsatz von RSA-Unterschriften

- Verifikationsschlüssel: $\mathbf{n}, \mathbf{e}$
- Finde zwei Nachrichten $\mathbf{X_1}$ und $\mathbf{X_2}$, und einen Faktor $\mathbf{b}$ mit

$$\mathbf{X_2} = \mathbf{X_1} * \mathbf{b}^{\mathbf{e}} \bmod \mathbf{n}.$$

Naiver Einsatz von RSA-Unterschriften

- Verifikationsschlüssel: $\mathbf{n}, \mathbf{e}$
- Finde zwei Nachrichten $\mathbf{X_1}$ und $\mathbf{X_2}$, und einen Faktor $\mathbf{b}$ mit

$$\mathbf{X_2} = \mathbf{X_1} * \mathbf{b}^{\mathbf{e}} \bmod \mathbf{n}.$$

- Es sei $\mathbf{X_1}$ ein harmloser Text, $\mathbf{X_2}$ enthalte dagegen eine für den Angreifer vorteilhafte Aussage.

Naiver Einsatz von RSA-Unterschriften

- Verifikationsschlüssel: $\mathbf{n}, \mathbf{e}$
- Finde zwei Nachrichten $\mathbf{X_1}$ und $\mathbf{X_2}$, und einen Faktor $\mathbf{b}$ mit

$$\mathbf{X_2} = \mathbf{X_1} * \mathbf{b}^{\mathbf{e}} \bmod \mathbf{n}.$$

- Es sei $\mathbf{X_1}$ ein harmloser Text, $\mathbf{X_2}$ enthalte dagegen eine für den Angreifer vorteilhafte Aussage.
- Veranlasse den Unterzeichner (= Inh. des geh. Schl.) dazu, $\mathbf{X_1}$ zu unterschreiben. Für die Unterschrift $\mathbf{S_1}$ unter $\mathbf{X_1}$ gilt:

$$\mathbf{X_1}^{\mathbf{e}} \bmod \mathbf{n} = \mathbf{S_1}.$$

Naiver Einsatz von RSA-Unterschriften

- Verifikationsschlüssel: $\mathbf{n}, \mathbf{e}$
- Finde zwei Nachrichten $\mathbf{X_1}$ und $\mathbf{X_2}$, und einen Faktor $\mathbf{b}$ mit

$$\mathbf{X_2} = \mathbf{X_1} * \mathbf{b}^{\mathbf{e}} \bmod \mathbf{n}.$$

- Es sei $\mathbf{X_1}$ ein harmloser Text, $\mathbf{X_2}$ enthalte dagegen eine für den Angreifer vorteilhafte Aussage.
- Veranlasse den Unterzeichner (= Inh. des geh. Schl.) dazu, $\mathbf{X_1}$ zu unterschreiben. Für die Unterschrift $\mathbf{S_1}$ unter $\mathbf{X_1}$ gilt:

$$\mathbf{X_1}^{\mathbf{e}} \bmod \mathbf{n} = \mathbf{S_1}.$$

- Nun ist es leicht, eine Unterschrift $\mathbf{S_1}$ unter $\mathbf{X_2}$ zu berechnen. (**Wie?**)

Diskussion

- Der Angriff mag nicht sonderlich plausibel erscheinen – immerhin muss man zwei *sinnvolle* Nachrichten $\mathbf{X_1}$ und $\mathbf{X_2}$ finden, die in einer sehr speziellen algebraischen Beziehung zueinander stehen.
- Wie schwierig das ist, hängt aber von Faktoren ab, die außerhalb der Kryptographie liegen, und die der Kryptograph nicht beeinflussen kann. (Welche Nachrichten sind *sinnvoll*?)

Diskussion

- Der Angriff mag nicht sonderlich plausibel erscheinen – immerhin muss man zwei *sinnvolle* Nachrichten $\mathbf{X_1}$ und $\mathbf{X_2}$ finden, die in einer sehr speziellen algebraischen Beziehung zueinander stehen.
- Wie schwierig das ist, hängt aber von Faktoren ab, die außerhalb der Kryptographie liegen, und die der Kryptograph nicht beeinflussen kann. (Welche Nachrichten sind *sinnvoll*?)

  → **Die Sicherheit guter Kryptosysteme für Digitale Unterschriften darf nicht davon abhängen, welche Nachrichten in einem gegebenen Kontext als sinnvoll akzeptiert werden!**

Beispiel für einen Denial-of-Service Angriff

- Nachrichten: Messwerte von einem Sensorknoten.

Beispiel für einen Denial-of-Service Angriff

- Nachrichten: Messwerte von einem Sensorknoten.
- Angriff: Wähle Unterschrift **S** zufällig, schicke "Messwert" $\mathbf{X} = \mathbf{S}^{\mathbf{e}} \bmod \mathbf{n}$ drahtlos an Empfänger.
- Empfänger: Gültig unterschriebene aber (wahrscheinlich) total unplausible Nachricht **X**.
- Mögliche Reaktion: "Sensorknoten defekt" – Empfänger ignoriert alle weiteren Nachrichten dieses Sensorknotens.

Kryptosystem für Digitale Unterschriften.

→ **Sichere Kryptosysteme für Digitale Unterschriften erlauben es dem Angreifer nicht einmal, Unterschriften unter zufällige Nachrichten zu fälschen.**

→ *Naiv eingesetztes RSA ist in diesem Sinne kein sicheres Kryptosystem für Digitale Unterschriften.*

Naiv eingesetzte Blockchiffren

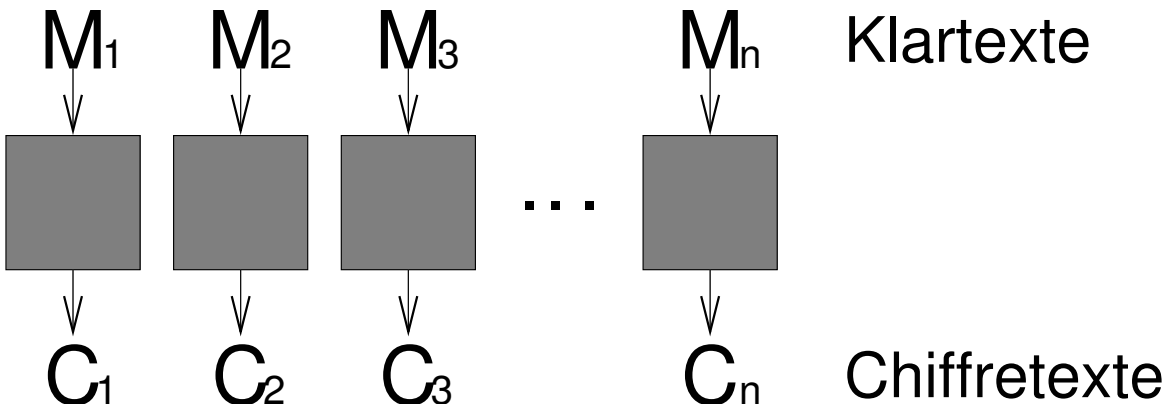
- "Von Haus aus" kann man eine Blockchiffre benutzen, um Nachrichten der gegebenen Blocklänge b (z.B. $b = 64$ bit, $b = 128$ bit, ...) zu verschlüsseln.

Naiv eingesetzte Blockchiffren

- "Von Haus aus" kann man eine Blockchiffre benutzen, um Nachrichten der gegebenen Blocklänge b (z.B. $b = 64$ bit, $b = 128$ bit, ...) zu verschlüsseln.
- Was tue ich mit längeren Nachrichten? Wir teilen M in b-bit Blöcke $M_i \in \{0,1\}^b$, ggf. mit "Padding" des letzten Blocks, und benutzen die Blockchiffre wie ein *Electronisches Codebuch* ("Electronic Codebook" – ECB):

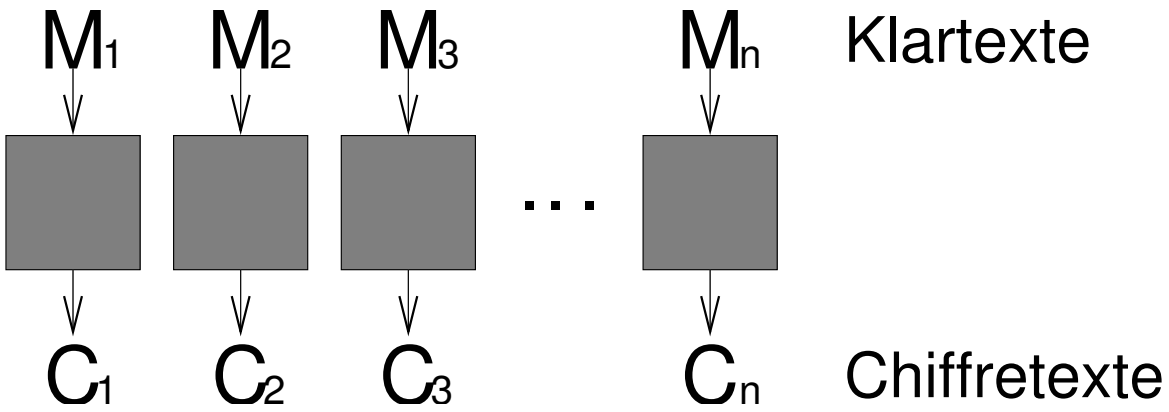
$$C_i := E_K(M_i).$$

Blockchiffren im ECB Modus



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

Blockchiffren im ECB Modus



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

Der ECB Modus ist unsicher! (Warum?)

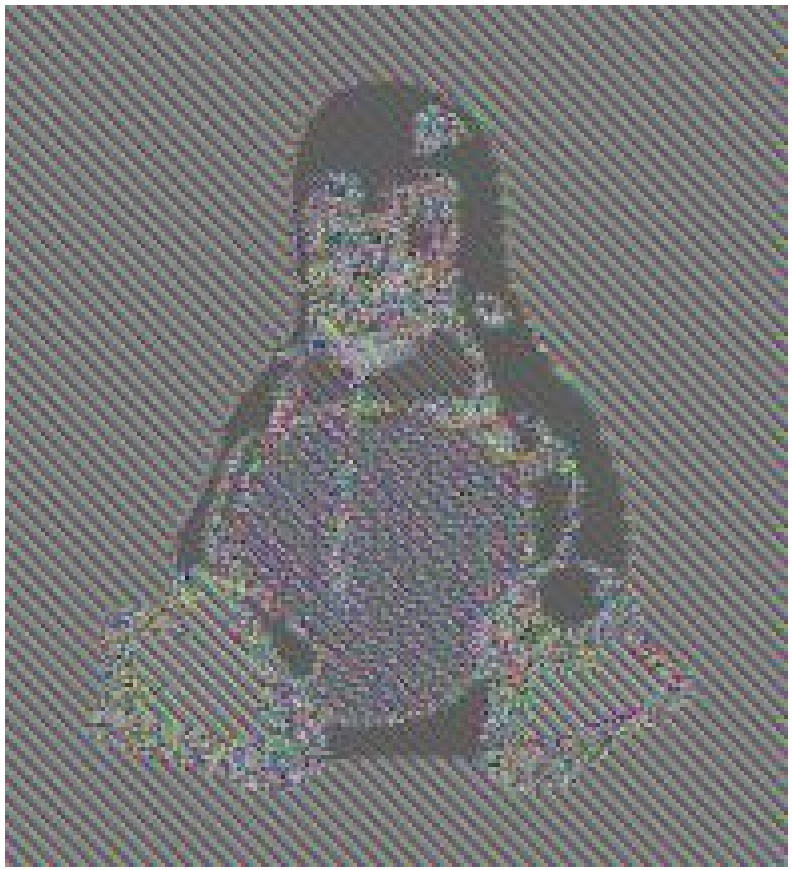
ECB in der Praxis

- Der ECB Modus ist einer von vier im Zusammenhang mit dem DES „offiziell standardisierten“ “Modes of Operation” (“Betriebsarten”).
- In vielen *(schlechten!)* DES- oder AES-basierten Krypto-Produkten wird der ECB-Modus verwendet (da formal standardkonform).

Wie schwach ist der ECB Modus?

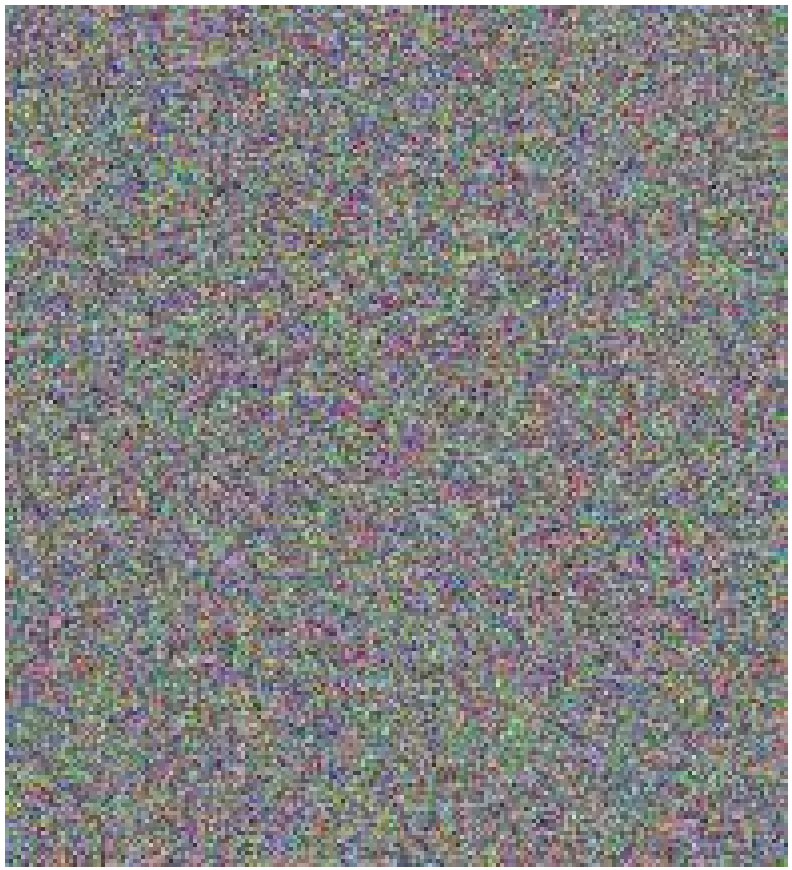
- Bei ECB-verschlüsselten Nachrichten kann der Angreifer ggf. sich wiederholende Textfragmente entdecken.
- Da sich bei “normalen” Nachrichten Textteile kaum wiederholen, dürfte dies eher eine theoretische Schwäche sein . . . oder?

Ein ECB-verschlüsselter Chiffretext



Der zugehörige Klartext (als Bitmap gespeichert):

Das gleiche Bild, auf sichere Weise verschlüsselt:



Keine Authentizität

- Beispiel: Verschlüsselted Datenpaket mit einer dem Angreifer bekannten IP-Adresse im (verschlüsselten) Header ...
- Das Datenpaket wird an einem Gateway entschlüsselt und innerhalb eines lokalen und sicheren Netzes an den vorgesehenen Empfänger weitergeleitet.
- Ein aktiver Angreifer kann das die Empfängeradresse manipulieren – und die entschlüsselte Nachricht an die eigene Adresse weiterleiten.

Vertraulichkeit und Authentizität

Man betrachte Anwendungen mit dem Schutzziel *Vertraulichkeit*, aber ohne das Schutzziel *Authentizität:*

→ **Verschlüsselte Daten nicht vor Manipulationen zu schützen, kann deren Vertraulichkeit gefährden!**

©opyleft

©opyleft

- Erstellt mit Freier Software
- Nach einer Vorlesung von Stefan Lucks.
- © Rüdiger Weis, Berlin 2008
- unter der GNU Free Documentation License.