Beweisbar sichere Verschlüsselung

ITS-Wahlpflichtvorlesung

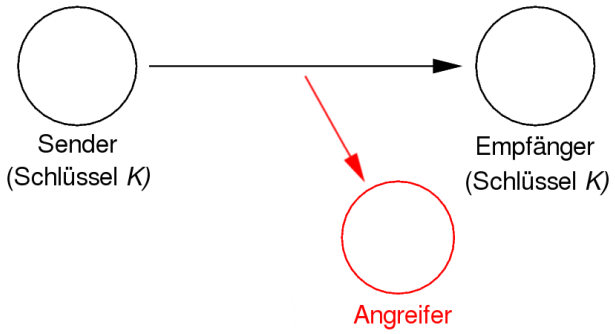
Dr. Bodo Möller
Ruhr-Universität Bochum
Horst-Görtz-Institut für IT-Sicherheit
Lehrstuhl für Kommunikationssicherheit
bmoeller@crypto.rub.de

$Id: bsv.ltx,v 1.108 2007/07/20 13:05:44 bm Exp $

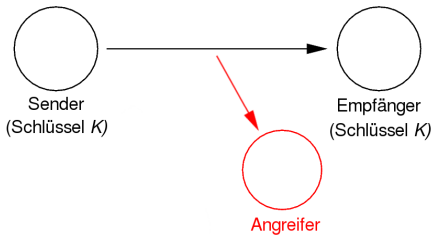
Einmal-Verschlüsselung

Für den Anfang ein bestimmter Spezialfall:

- *symmetrische* Verschlüsselung
- *Einmal*-Verschlüsselung
- *passiver* Angreifer



Einmal-Verschlüsselung (Forts.)



- *Symmetrisch:* Sender kennt gleichen Schlüssel K wie Empfänger
- *Einmal*-Verschlüsselung: Sender verschlüsselt nur eine Nachricht
- *Passiver Angreifer:* „liest mit“, sendet selbst nichts

Einmal-Verschlüsselung (Forts.)

Formalisierung als *Verschlüsselungsschema* $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ mit drei Algorithmen:

- *Schlüsselgenerierungsalgorithmus* $\mathcal{K}$ erzeugt Schlüssel:

  $$K \xleftarrow{\$} \mathcal{K}$$

- *Verschlüsselungsalgorithmus* $\mathcal{E}$ für Nachrichten $m \in \mathcal{M} \subseteq \{0,1\}^*$:

  $$\mathcal{E}_K(m) \in \{0,1\}^*$$

  m heißt *„Plaintext“.*
  $\mathcal{M}$ hat oft die Form $\bigcup_{l \in L} \{0,1\}^l$ mit $L \subseteq \mathbb{N}$.

- *Entschlüsselungsalgorithmus* $\mathcal{D}$ für Nachrichten $c \in \{0,1\}^*$:

  $$\mathcal{D}_K(c) \in \{0,1\}^* \cup \{\perp\}$$

  c heißt *„Ciphertext“.*
  Ergebnis $\perp$ zeigt Entschlüsselungsfehler an.

$\mathcal{K}$ und $\mathcal{E}$ sind *probabilistisch* ($\mathcal{E}_K$ ist also keine Funktion!),
$\mathcal{D}$ ist deterministisch.

Einmal-Verschlüsselung (Forts.)

Zwei Fragen für Verschlüsselungsschema $(\mathcal{K}, \mathcal{E}, \mathcal{D})$:

- Funktioniert es überhaupt richtig? *(Korrektheit)*
- Ist es „sicher“?

Korrektheit heißt einfach:
Für $K \xleftarrow{\$} \mathcal{K}$ und jedes $m \in \mathcal{M}$ gilt $\mathcal{D}_K\big(\mathcal{E}_K(m)\big) = m$.

Aber was heißt Sicherheit?

Einmal-Verschlüsselung (Forts.)

Sicherheit des Verschlüsselungsschemas: Formalisierung durch ein *Angriffsspiel*

- Als Einmal-Verschlüsselung betrachten
- Angreifer ist passiv

Erinnern an PRP-Angriffsspiel (pseudo-random permutation) für Blockchiffre:

Angreifer $\boldsymbol{A}$ hatte Zugriff auf E_K mit zufälligem Schlüssel K (Fall $b = 1$)
oder auf gleichverteilt zufällige Permutation (Fall $b = 0$),
sollte zwischen diesen Fällen unterscheiden.

„Vorteil“ maß seinen Erfolg dabei.

Kriegen wir hier etwas Ähnliches hin?

Einmal-Verschlüsselung (Forts.)

- Für Angreifer wieder zwei Szenarien zu unterscheiden
- z. B. „richtige“ Verschlüsselung und etwas Ähnliches
- Es geht um Einmal-Verschlüsselung, also ein einziger Verschlüsselungsvorgang
- Was wird verschlüsselt ...?
  - In der Praxis ist oft bekannt, welche Plaintexte vermutlich vorkommen
  - Um viele denkbare Szenarien abzudecken: Lassen wir den Angreifer wählen!

RoR für Einmal-Verschlüsselung

Ein *real-or-random*-Angriffsspiel für Verschlüsselungsschema $(\mathcal{K}, \mathcal{E}, \mathcal{D})$:

- Angreifer $\boldsymbol{A}$ wählt Plaintext m als Anfrage an *Verschlüsselungsorakel* E, erhält dazu Ciphertext c.
  $\boldsymbol{A}$ gibt schließlich ein Bit $\widetilde{b}$ aus.
- Im Fall $b = 1$ (*real*):
  $K \xleftarrow{\$} \mathcal{K}$
  Das Verschlüsselungsorakel gibt $\mathcal{E}_K(m)$ zurück.
- Im Fall $b = 0$ (*random*):
  $K \xleftarrow{\$} \mathcal{K}$
  Das Verschlüsselungsorakel gibt $\mathcal{E}_K(m_0)$ zurück für m_0 mit $|m_0| = |m|$, m_0 gleichverteilt zufällig (Bitstring gleicher Länge wie m).

Hier ist $\boldsymbol{A}$ aktiv tätig (wählt m), obwohl wir einen passiven Angreifer modellieren.
→ *Chosen plaintext attack* (*CPA*)

RoR für Einmal-Verschlüsselung (Forts.)

RoR-OTCPA-Angriffsspiel für symmetrische Verschlüsselung:

- real-or-random
- one-time
- chosen plaintext attack

(Achtung: Die Terminologie ist nicht immer einheitlich,
Angaben zum Angriffsspiel nicht immer komplett.)

"Real"-Fall mit Verschlüsselungsorakel E_1,
"Random"-Fall mit Verschlüsselungsorakel E_0

Wir können hier einen *RoR-OTCPA-Vorteil* (advantage) von $\boldsymbol{A}$ definieren,
ähnlich wie beim PRP-Angriffsspiel:

$$\mathrm{Adv}^{\text{RoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} = \Pr\left[\boldsymbol{A}^{E_1(\cdot)} \Rightarrow 1\right] - \Pr\left[\boldsymbol{A}^{E_0(\cdot)} \Rightarrow 1\right],$$

wobei $\boldsymbol{A}$ das Verschlüsselungsorakel nur einmal verwenden darf.

RoR für Einmal-Verschlüsselung (Forts.)

Genau wie zuvor (bei PRP):

Vorteil lässt sich auch über die Wahrscheinlichkeit ausdrücken,
dass $\boldsymbol{A}$ das versteckte Bit b errät.

$$\begin{aligned}\mathrm{Adv}^{\text{RoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} &= \Pr\left[\boldsymbol{A}^{E_1(\cdot)} \Rightarrow 1\right] - \Pr\left[\boldsymbol{A}^{E_0(\cdot)} \Rightarrow 1\right] \\ &= 2\Big(\Pr_{b\in_{\$}\{0,1\}}\left[\boldsymbol{A}^{E_b(\cdot)} \Rightarrow b\right] - \frac{1}{2}\Big)\end{aligned}$$

RoR für Einmal-Verschlüsselung (Forts.)

RoR-OTCPA-Sicherheit: kein denkbarer Angreifer kann
(mit praktikablem Ressourcenaufwand) einen Vorteil

$$\mathrm{Adv}^{\text{RoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} \;=\; 2\Big(\Pr{}_{b\in_\$\{0,1\}}\big[\boldsymbol{A}^{E_b(\cdot)} \Rightarrow b\big] - \frac{1}{2}\Big)$$

erreichen, der größer ist als verschwindend gering.

Aber was besagt Sicherheit in diesem Sinn?

- Laut Definition: Erfolgreich ist ein Angreifer,
  der „echtes“ Verschlüsselungsorakel E_1
  vom „zufälligen“ Verschlüsselungsorakel E_0 unterscheiden kann:
  E_1 verschlüsselt m mit $\mathcal{E}_K(\cdot)$, E_0 verschlüsselt gleich langen Zufallswert
- Verlangt nicht, dass die Plaintext-Länge versteckt wird!
  (In Ordnung, denn so etwas wäre allgemein nicht praktikabel.)
- Wird sonst alles „erkannt“, das als „unsichere Verschlüsselung“ gelten sollte ...?

LoR für Einmal-Verschlüsselung

Probieren wir eine neue Formalisierung der Sicherheit!

Ein *left-or-right*-Angriffsspiel für Verschlüsselungsschema $(\mathcal{K},\mathcal{E},\mathcal{D})$
(immer noch als Einmal-Verschlüsselung):

- Angreifer $\boldsymbol{A}$ sendet Plaintextpaar (m_1, m_0)
  ans Verschlüsselungsorakel $E(\cdot,\cdot)$,
  erhält von diesem dazu Ciphertext c.
  Hierbei muss gelten $|m_1| = |m_0|$ (gleiche Länge).
  $\boldsymbol{A}$ gibt schließlich ein Bit $\widetilde{b}$ aus.
- Fälle $b = 1$ (*left*) und $b = 0$ (*right*) mit Verschlüsselungsorakel $E_b(\cdot,\cdot)$:
  $K \xleftarrow{\$} \mathcal{K}$, Orakel gibt $\mathcal{E}_K(m_b)$ zurück.

LoR-OTCPA-Vorteil von $\boldsymbol{A}$ wie gewohnt:

$$\begin{aligned}\mathrm{Adv}^{\text{LoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} \;&=\; \Pr\big[\boldsymbol{A}^{E_1(\cdot,\cdot)} \Rightarrow 1\big] - \Pr\big[\boldsymbol{A}^{E_0(\cdot,\cdot)} \Rightarrow 1\big] \\ &=\; 2\Big(\Pr{}_{b\in_\$\{0,1\}}\big[\boldsymbol{A}^{E_b(\cdot,\cdot)} \Rightarrow b\big] - \frac{1}{2}\Big)\end{aligned}$$

RoR und LoR für Einmal-Verschlüsselung

- Zusammenhang RoR-OTCPA-Sicherheit und LoR-OTCPA-Sicherheit?
- Wir können eine Äquivalenz zeigen!
- Sei Angreifer $\boldsymbol{A}$ im RoR-OTCPA-Angriffsspiel gegeben;
  wir werden daraus Angreifer $\boldsymbol{B}$ im LoR-OTCPA-Angriffsspiel konstruieren.
- Sei Angreifer $\boldsymbol{A}$ im LoR-OTCPA-Angriffsspiel gegeben;
  wir werden daraus Angreifer $\boldsymbol{B}$ im RoR-OTCPA-Angriffsspiel konstruieren.

RoR und LoR für Einmal-Verschlüsselung (Forts.)

Sei $\boldsymbol{A}^{E'(\cdot)}$ ein Angreifer im RoR-OTCPA-Angriffsspiel.

Neuer Angreifer $\boldsymbol{B}^{E(\cdot,\cdot)}$ im LoR-OTCPA-Angriffsspiel:

- Lass den Algorithmus $\boldsymbol{A}$ ablaufen,
  beantworte dabei Anfrage m an dessen RoR-Verschlüsselungsorakel
  mit $E(m, m_0)$, wobei $m_0 \in_\$ \{0,1\}^{|m|}$
  (Verwendung des eigenen LoR-Verschlüsselungsorakels!)
- Gib das gleiche Bit zurück wie $\boldsymbol{A}$.

$$\begin{aligned}
\mathrm{Adv}^{\text{LoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{B}} &= \Pr\left[\boldsymbol{B}^{E_1(\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[\boldsymbol{B}^{E_0(\cdot,\cdot)} \Rightarrow 1\right] \\
&= \Pr\left[\boldsymbol{A}^{E'_1(\cdot)} \Rightarrow 1\right] - \Pr\left[\boldsymbol{A}^{E'_0(\cdot)} \Rightarrow 1\right] \\
&= \mathrm{Adv}^{\text{RoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}}
\end{aligned}$$

RoR und LoR für Einmal-Verschlüsselung (Forts.)

$\boldsymbol{B}$ hat das RoR-OTCPA-Angriffsspiel von $\boldsymbol{A}$ unverändert durchgeführt, deshalb gilt hier

$$\mathrm{Adv}^{\text{LoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{B}} = \mathrm{Adv}^{\text{RoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}}$$

Ein RoR-OTCPA-Angreifer ($\boldsymbol{A}$) lässt sich also in einen gleich erfolgreichen LoR-OTCPA-Angreifer ($\boldsymbol{B}$) umwandeln (mit im wesentlichen gleicher Laufzeit).

Das heißt umgekehrt:

Gibt es keinen erfolgreichen LoR-OTCPA-Angreifer, kann es keinen erfolgreichen RoR-OTCPA-Angreifer geben;

LoR-OTCPA-Sicherheit impliziert RoR-OTCPA-Sicherheit!

RoR und LoR für Einmal-Verschlüsselung (Forts.)

Sei $\boldsymbol{A}^{E'(\cdot,\cdot)}$ ein Angreifer im LoR-OTCPA-Angriffsspiel.

Neuer Angreifer $\boldsymbol{B}^{E(\cdot)}$ im RoR-OTCPA-Angriffsspiel:

- Wähle ein Bit $x \in_{\$} \{0,1\}$.
- Lass den Algorithmus $\boldsymbol{A}$ ablaufen, beantworte dabei Anfrage (m_1, m_0) an dessen LoR-Verschlüsselungsorakel mit $E(m_x)$ (Verwendung des eigenen RoR-Verschlüsselungsorakels!)
- Gib 1 zurück, falls $\boldsymbol{A} \Rightarrow x$; sonst gib 0 zurück.

$$\begin{aligned}
\mathrm{Adv}^{\text{RoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{B}} &= \Pr\left[\boldsymbol{B}^{E_1(\cdot)} \Rightarrow 1\right] - \Pr\left[\boldsymbol{B}^{E_0(\cdot)} \Rightarrow 1\right] \\
&= \Pr\left[\boldsymbol{B}^{E_1(\cdot)} \Rightarrow 1\right] - \frac{1}{2} \\
&= \Pr_{x\in_{\$}\{0,1\}}\left[\boldsymbol{A}^{E'_x(\cdot,\cdot)} \Rightarrow x\right] - \frac{1}{2} \quad = \quad \frac{1}{2} \cdot \mathrm{Adv}^{\text{LoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}}
\end{aligned}$$

RoR und LoR für Einmal-Verschlüsselung (Forts.)

In $\boldsymbol{B}$ findet sich das LoR-OTCPA-Angriffsspiel von $\boldsymbol{A}$ etwas „aufgeweicht“ wieder, hier gilt

$$\mathrm{Adv}^{\text{RoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{B}} = \frac{1}{2} \cdot \mathrm{Adv}^{\text{LoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}}$$

Ein LoR-OTCPA-Angreifer ($\boldsymbol{A}$) lässt sich somit in einen *ähnlich* erfolgreichen RoR-OTCPA-Angreifer ($\boldsymbol{B}$) umwandeln (mit im wesentlichen gleicher Laufzeit).

Das heißt umgekehrt:

Gibt es keinen erfolgreichen RoR-OTCPA-Angreifer, kann es keinen (sehr) erfolgreichen LoR-OTCPA-Angreifer geben;

RoR-OTCPA-Sicherheit impliziert LoR-OTCPA-Sicherheit!

RoR und LoR für Einmal-Verschlüsselung (Forts.)

- Zum RoR-OTCPA-Angreifer $\boldsymbol{A}$ gibt es also einen „verwandten“ LoR-OTCPA-Angreifer $\boldsymbol{B}$ mit
$$\mathrm{Adv}^{\text{LoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{B}} = \mathrm{Adv}^{\text{RoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}}$$
- ... und zum LoR-OTCPA-Angreifer $\boldsymbol{A}$ gibt es einen „verwandten“ RoR-OTCPA-Angreifer $\boldsymbol{B}$ mit
$$\mathrm{Adv}^{\text{RoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{B}} = \frac{1}{2} \cdot \mathrm{Adv}^{\text{LoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}}$$
- D. h., *RoR-OTCPA-Sicherheit* und *LoR-OTCPA-Sicherheit* sind *vergleichbar*
- Indiz, dass diese Formalisierungen der Sicherheit sinnvoll sind. Aber beide wirken doch etwas willkürlich ...?
- *Später* wird direkter formalisiert, was Verschlüsselung „bedeutet“: Was der Angreifer einem Ciphertext ansehen kann, kann er auch ohne den Ciphertext (nur anhand der Länge) wissen. Dann spricht man von *semantischer Sicherheit,* z. B. *SEM-OTCPA-Sicherheit*
- Auch für SEM-OTCPA-Sicherheit werden wir sehen, dass sie vergleichbar ist mit RoR-OTCPA- und LoR-OTCPA-Sicherheit!

Primitive für die symmetrische Verschlüsselung

- Als Modellierung für eine Blockchiffre schon gesehen haben wir *pseudo-random permutation* mit PRP-Angriffsspiel, PRP-Vorteil:

$$\mathrm{Adv}^{\mathsf{PRP}}_{E,\boldsymbol{A}} \;=\; \Pr_{K\in_{\$}\mathcal{K}}\left[\boldsymbol{A}^{E_K(\cdot)} \Rightarrow 1\right] - \Pr_{\pi\in_{\$}\mathrm{Perm}(\{0,1\}^\ell)}\left[\boldsymbol{A}^{\pi(\cdot)} \Rightarrow 1\right]$$

  $\mathrm{Perm}(\{0,1\}^\ell)$ ist die Menge aller Bijektionen $\{0,1\}^\ell \to \{0,1\}^\ell$, $E_K \in \mathrm{Perm}(\{0,1\}^\ell)$ für jedes K.

- Ganz analog *pseudo-random function* mit *PRF-Angriffsspiel* und *PRF-Vorteil:*

$$\mathrm{Adv}^{\mathsf{PRF}}_{E,\boldsymbol{A}} \;=\; \Pr_{K\in_{\$}\mathcal{K}}\left[\boldsymbol{A}^{E_K(\cdot)} \Rightarrow 1\right] - \Pr_{f\in_{\$}\mathrm{Func}(\{0,1\}^\ell)}\left[\boldsymbol{A}^{f(\cdot)} \Rightarrow 1\right]$$

  $\mathrm{Func}(\{0,1\}^\ell)$ ist die Menge aller Abbildungen $\{0,1\}^\ell \to \{0,1\}^\ell$, $E_K \in \mathrm{Func}(\{0,1\}^\ell)$ für jedes K.

Primitive für die symmetrische Verschlüsselung (Forts.)

- *Pseudo-random generator* (PRG) ist ähnlich wie PRP und PRF, aber ohne einen Eingabewert wie dort.

  Statt dessen haben wir für $K \in \mathcal{K}$ eine Abbildung $g_K\colon \mathbb{N} \to \{0,1\}^*$ mit
  - $g_K(\ell) \in \{0,1\}^\ell$ für jedes ℓ,
  - $g_K(\ell')$ ist Präfix von $g_K(\ell)$ für $\ell' \le \ell$.

  g_K definiert quasi einen unendlichen Bitstring, von dem man die ersten Bits „bestellen“ kann.

- Analog definieren wir für einen Bitstring $S \in \{0,1\}^{\ell_{\max}}$ eine Abbildung $S\colon \{0,\ldots,\ell_{\max}\} \to \{0,1\}^*$ mit
  - $S(\ell) \in \{0,1\}^\ell$,
  - $S(\ell)$ ist Präfix von S.

- *PRG-Vorteil:* Die Laufzeit eines Angreifers $\boldsymbol{A}$ legt fest, wieviele Bits er höchstens ansehen kann; wähle $\ell_{\max}$ entsprechend.

$$\mathrm{Adv}^{\mathsf{PRG}}_{g,\boldsymbol{A}} \;=\; \Pr_{K\in_{\$}\mathcal{K}}\left[\boldsymbol{A}^{g_K(\cdot)} \Rightarrow 1\right] - \Pr_{S\in_{\$}\{0,1\}^{\ell_{\max}}}\left[\boldsymbol{A}^{S(\cdot)} \Rightarrow 1\right]$$

Primitive für die symmetrische Verschlüsselung (Forts.)

PRG g mit Schlüsselmenge $\mathcal{K}$ liefert ein Verschlüsselungsschema $(\mathcal{K}, \mathcal{E}, \mathcal{D})$:

- Schlüsselgenerierungs*algorithmus* $\mathcal{K}$: Gleichverteilung auf der *Menge* $\mathcal{K}$
- Verschlüsselungsalgorithmus:
$$\mathcal{E}_K(m) \Rightarrow g_K\big(|m|\big) \oplus m$$
- Entschlüsselungsalgorithmus:
$$\mathcal{D}_K(c) \Rightarrow g_K\big(|c|\big) \oplus c$$

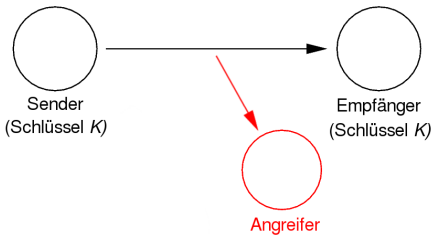
Von der *Korrektheit* überzeugt man sich leicht:

$$\mathcal{D}_K\big(\mathcal{E}_K(m)\big) = m$$

für alle K, m.

Sicherheit: RoR-OTCPA-Vorteil durch PRG-Vorteil beschreiben
($\rightarrow$ *Übungsaufgabe!*)

Symmetrische Verschlüsselung



Bisher: Symmetrische *Einmal*-Verschlüsselung mit passivem Angreifer

Wir bleiben bei symmetrischer Verschlüsselung mit passivem Angreifer, aber ohne Beschränkung auf Einmal-Verschlüsselung!

Symmetrische Verschlüsselung (Forts.)

Real-or-random-Angriffsspiel für Verschlüsselungsschema $(\mathcal{K}, \mathcal{E}, \mathcal{D})$
analog wie bei Einmal-Verschlüsselung:

- Angreifer $\boldsymbol{A}$ sendet Plaintexte m an Verschlüsselungsorakel E,
  erhält dazu jeweils einen Ciphertext c.
  $\boldsymbol{A}$ gibt schließlich ein Bit $\widetilde{b}$ aus.
- Im Fall $b = 1$ (*real*): $K \xleftarrow{\$} \mathcal{K}$ wird einmal festgelegt.
  Das Verschlüsselungsorakel $E_1(\cdot)$ gibt jeweils $\mathcal{E}_K(m)$ zurück.
- Im Fall $b = 0$ (*random*): $K \xleftarrow{\$} \mathcal{K}$ wird einmal festgelegt.
  Das Verschlüsselungsorakel $E_0(\cdot)$ gibt jeweils $\mathcal{E}_K(m_0)$ zurück
  für m_0 mit $|m_0| = |m|$, m_0 gleichverteilt zufällig.

RoR-CPA-Vorteil:

$$\mathrm{Adv}^{\text{RoR-CPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} = \Pr\left[\boldsymbol{A}^{E_1(\cdot)} \Rightarrow 1\right] - \Pr\left[\boldsymbol{A}^{E_0(\cdot)} \Rightarrow 1\right]$$

Symmetrische Verschlüsselung (Forts.)

Left-or-right-Angriffsspiel für Verschlüsselungsschema $(\mathcal{K}, \mathcal{E}, \mathcal{D})$
ebenfalls analog wie bei Einmal-Verschlüsselung:

- Angreifer $\boldsymbol{A}$ sendet Plaintextpaare (m_1, m_0) an Verschlüsselungsorakel E,
  erhält dazu jeweils einen Ciphertext c.
  $\boldsymbol{A}$ gibt schließlich ein Bit $\widetilde{b}$ aus.
- Fälle $b = 1$ (*left*) und $b = 0$ (*right*) mit Verschlüsselungsorakel $E_b(\cdot, \cdot)$:
  $K \xleftarrow{\$} \mathcal{K}$ wird nur einmal festgelegt.
  Orakel gibt zur Anfrage (m_1, m_0) jeweils $\mathcal{E}_K(m_b)$ zurück.

LoR-CPA-Vorteil:

$$\mathrm{Adv}^{\text{LoR-CPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} = \Pr\left[\boldsymbol{A}^{E_1(\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[\boldsymbol{A}^{E_0(\cdot,\cdot)} \Rightarrow 1\right]$$

Symmetrische Verschlüsselung (Forts.)

- RoR-CPA-Sicherheit impliziert RoR-OTCPA-Sicherheit
  (denn das RoR-OTCPA-Angriffsspiel ist ein Sonderfall von RoR-CPA)
- LoR-CPA-Sicherheit impliziert LoR-OTCPA-Sicherheit
  (denn das LoR-OTCPA-Angriffsspiel ist ein Sonderfall von LoR-CPA)
- Analog zu OTCPA: LoR-CPA-Sicherheit und RoR-CPA-Sicherheit sind vergleichbar.
  ($\rightarrow$ *Übungsaufgabe!*)
- RoR-OTCPA-Sicherheit impliziert *nicht* RoR-CPA-Sicherheit!
  LoR-OTCPA-Sicherheit impliziert *nicht* LoR-CPA-Sicherheit!
  ($\rightarrow$ *Übungsaufgabe!*)