zu Aufgabe 1.1

Sei $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ ein Verschlüsselungsschema mit einer Plaintext-Menge $\mathcal{M} = \{0,1\}^\ell$, $\ell \geq 1$.

Nehmen wir an, dieses Verschlüsselungsschema sei „völlig unsicher“ in folgendem Sinn: Es gibt einen Algorithmus $\boldsymbol{A}_0$, der zu gegebenem $\mathcal{E}_K(m)$ stets zuverlässig und schnell m ermittelt – und das für beliebige (unbekannte) mit $\mathcal{K}$ erzeugte Schlüssel K und beliebige Plaintexte $m \in \mathcal{M}$.

Beschreiben Sie (auf $\boldsymbol{A}_0$ zurückgreifend) möglichst erfolgreiche Angreifer

a. im LoR-OTCPA-Angriffsspiel,

b. im RoR-OTCPA-Angriffsspiel,

c. im RoR-CPA-Angriffsspiel, wobei der Angreifer das Verschlüsselungsorakel q mal verwendet für irgendeine ganze Zahl $q \geq 1$.

Welcher Vorteil lässt sich jeweils erreichen?

a. Ein Angreifer $\boldsymbol{A}$ im LoR-OTCPA-Angriffsspiel (Folie 2.11) kann wie folgt vorgehen.

   Erläuterungen zum Ablauf von $\boldsymbol{A}$ sind schräg gedruckt. Sie können weggelassen werden: Dann ist die Beschreibung von $\boldsymbol{A}$ als solche noch komplett (nur schlechter verständlich).

   $\boldsymbol{A}$ wählt irgendwelche voneinander verschiedene Plaintexte $m_1, m_0 \in \{0,1\}^\ell$, zum Beispiel $m_1 = 1\ldots1$ und $m_1 = 0\ldots0$, und sendet (m_1, m_0) an das Verschlüsselungsorakel. Das Verschlüsselungsorakel antwortet mit einem Ciphertext c.

   Im “left-or-right”-Angriffsspiel handelt es sich bei diesem Ciphertext im Fall “left” um ein Verschlüsselungsergebnis $\mathcal{E}_K(m_1)$, im Fall “right” um ein Verschlüsselungsergebnis $\mathcal{E}_K(m_0)$, jeweils unter einem unbekannten Schlüssel K, der mit $\mathcal{K}$ erzeugt wurde. („Ein“ und nicht „das“ Verschlüsselungsergebnis, denn der Algorithmus $\mathcal{E}$ ist probabilistisch, $\mathcal{E}_K$ kann also i. a. nicht als Abbildung aufgefasst werden!)

   Der Angreifer $\boldsymbol{A}$ greift auf $\boldsymbol{A}_0$ zurück, um diesen Ciphertext c zu entschlüsseln.

   Das leistet $\boldsymbol{A}_0$ laut Aufgabenstellung, und $\boldsymbol{A}$ erhält also im Fall “left” den Plaintext m_1 und im Fall “right” den Plaintext m_0 zurück.

   Das mit $\boldsymbol{A}_0$ gewonnene Entschlüsselungsresultat nennen wir m'. Ist nun $m' = m_1$, so gibt $\boldsymbol{A}$ das Bit 1 aus; ansonsten das Bit 0.

   Damit setzt $\boldsymbol{A}$ darauf, dass er es mit dem Fall “left” (1) bzw. “right” (0) zu tun hatte.

   Der Vorteil von $\boldsymbol{A}$ ist laut Definition

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

   wobei E_1 das Verschlüsselungsorakel für den Fall “left” bezeichnet und E_0 das Verschlüsselungsorakel für den Fall “right”. $\boldsymbol{A}$ wie oben beschrieben gibt im Fall “left” immer das Bit 1 aus, es gilt also

$$\Pr\left[\boldsymbol{A}^{E_1(\cdot,\cdot)} \Rightarrow 1\right] = 1;$$

   im Fall “right” immer das Bit 0, also

$$\Pr\left[\boldsymbol{A}^{E_0(\cdot,\cdot)} \Rightarrow 1\right] \;=\; 1 - \underbrace{\Pr\left[\boldsymbol{A}^{E_0(\cdot,\cdot)} \Rightarrow 0\right]}_{1} \;=\; 0.$$

Daraus folgt

$$\mathrm{Adv}^{\text{LoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} = 1.$$

b. Hier wählt $\boldsymbol{A}$ für das RoR-OTCPA-Angriffsspiel (Folie 2.7 f.) irgendeinen Plaintext $m \in \{0,1\}^\ell$, zum Beispiel $m = 0\ldots0$, und sendet diesen an das Verschlüsselungsorakel. Das Verschlüsselungsorakel antwortet mit einem Ciphertext c.

Im "real-or-random"-Angriffsspiel handelt es sich bei diesem Ciphertext im Fall "real" um ein Verschlüsselungsergebnis $\mathcal{E}_K(m)$, im Fall "random" um ein Verschlüsselungsergebnis $\mathcal{E}_K(m_0)$ für ein gleichverteiltes $m_0 \in \{0,1\}^\ell$; und das jeweils unter einem unbekannten Schlüssel K, der mit $\mathcal{K}$ erzeugt wurde.

Der Angreifer $\boldsymbol{A}$ greift auf $\boldsymbol{A}_0$ zurück, um diesen Ciphertext c zu entschlüsseln.

So erhält er im Fall "real" den Plaintext m zurück, im Fall "random" den zufälligen Plaintext m_0.

Das mit $\boldsymbol{A}_0$ gewonnene Entschlüsselungsresultat nennen wir m'. Ist nun $m' = m$, so gibt $\boldsymbol{A}$ das Bit 1 aus; ansonsten das Bit 0.

Damit setzt $\boldsymbol{A}$ darauf, dass er es mit dem Fall "real" bzw. "random" zu tun hatte.

Der Vorteil von $\boldsymbol{A}$ ist laut Definition

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

wobei E_1 das Verschlüsselungsorakel für den Fall "real" bezeichnet und E_0 das Verschlüsselungsorakel für den Fall "random". $\boldsymbol{A}$ wie oben beschrieben gibt im Fall "real" immer das Bit 1 aus, es gilt also

$$\Pr\left[\boldsymbol{A}^{E_1(\cdot)} \Rightarrow 1\right] = 1.$$

Im Fall "random" gibt $\boldsymbol{A}$ genau dann das Bit 1 aus, wenn zufällig $m_0 = m$ war, und somit ist

$$\Pr\left[\boldsymbol{A}^{E_0(\cdot)} \Rightarrow 1\right] = \frac{1}{2^\ell}.$$

Daraus folgt

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

Der Vorteil ist also sicherlich nicht verschwindend gering (sondern mindestens $1/2$), kann aber 1 nicht ganz erreichen, weil der "random"-Fall mit einer gewissen Wahrscheinlichkeit exakt wie der "real"-Fall abläuft.

c. $\boldsymbol{A}$ wählt für das RoR-CPA-Angriffsspiel (Folie 2.22) q-mal Plaintexte $m \in \{0,1\}^\ell$ und sendet diese jeweils an das Verschlüsselungsorakel, welches jedes Mal mit einem Ciphertext c antwortet. Der Angreifer $\boldsymbol{A}$ greift stets auf $\boldsymbol{A}_0$ zurück, um den jeweiligen Ciphertext c zu entschlüsseln, und erhält dazu ein Entschlüsselungsresultat m'. Ist durchgehend $m' = m$, so gibt $\boldsymbol{A}$ das Bit 1 aus; ansonsten (falls $m' \neq m$ war in mindestens einer der q „Runden") das Bit 0.

Bezeichne E_1 das Verschlüsselungsorakel für den Fall "real" und E_0 das Verschlüsselungsorakel für den Fall "random". $\boldsymbol{A}$ wie beschrieben gibt im Fall "real" immer das Bit 1 aus, es gilt also

$$\Pr\left[\boldsymbol{A}^{E_1(\cdot)} \Rightarrow 1\right] = 1.$$

Im Fall "random" gibt $\boldsymbol{A}$ genau dann das Bit 1 aus, wenn zufällig in allen q Runden $m_0 = m$ war, und es ist also

$$\Pr\left[\boldsymbol{A}^{E_0(\cdot)} \Rightarrow 1\right] \;=\; \frac{1}{2^{\ell \cdot q}}.$$

Das setzen wir in die Definition des RoR-CPA-Vorteils ein:

$$\mathrm{Adv}^{\text{RoR-CPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} \;=\; \Pr\left[\boldsymbol{A}^{E_1(\cdot)} \Rightarrow 1\right] - \Pr\left[\boldsymbol{A}^{E_0(\cdot)} \Rightarrow 1\right] \;=\; 1 - \frac{1}{2^{\ell \cdot q}}$$

zu Aufgabe 1.2 [Stream-Cipher-Verschlüsselung]

Sei g ein Pseudo-Random Generator mit Schlüsselmenge $\mathcal{K}$ und sei $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ das folgende Verschlüsselungsschema mit Plaintextmenge $\mathcal{M} = \{0,1\}^*$:

- Der Schlüsselgenerierungsalgorithmus $\mathcal{K}$ erzeugt eine Gleichverteilung auf der Menge $\mathcal{K}$.
- Der Verschlüsselungsalgorithmus berechnet $\mathcal{E}_K(m) \;=\; g_K\big(|m|\big) \oplus m$ (ist also deterministisch).
- Der Entschlüsselungsalgorithmus berechnet $\mathcal{D}_K(c) = g_K\big(|c|\big) \oplus c$.

Wir behaupten: Ist der Pseudo-Random Generator sicher, so ist das Verschlüsselungsverfahren sicher im Sinne von RoR-OTCPA.

Zeigen Sie dafür: Ist ein Angreifer $\boldsymbol{A}$ im RoR-OTCPA-Angriffsspiel gegeben, so lässt sich ein Angreifer $\boldsymbol{B}$ im PRG-Angriffsspiel konstruieren, der mit im wesentlichen der gleichen Laufzeit genau den gleichen Vorteil erreicht.

$\boldsymbol{B}$ als Angreifer im PRG-Angriffsspiel (siehe Vorlesungsfolie 2.19) hat Zugriff auf ein Orakel, das auf eine Anfrage $\ell \in \mathbb{N}$ genau ℓ Bits liefert. Wie bei vielen anderen Angriffsspielen sind hier zwei verschiedene Szenarien zu betrachten. Im einen Szenario hat der Angreifer $\boldsymbol{B}$ über das Orakel Zugriff auf einen konkreten PRG g_K mit einem gleichverteilt zufälligen Schlüssel $K \in \mathcal{K}$, also eine Abbildung $g_K\colon \mathbb{N} \to \{0,1\}^*$ mit folgenden Eigenschaften:

- $g_K(\ell) \in \{0,1\}^\ell$
- $g_K(\ell')$ ist Präfix von $g_K(\ell)$ für $\ell' \le \ell$

Im anderen Szenario hat $\boldsymbol{B}$ Zugriff auf eine Abbildung $S\colon 0, \ldots, L \to \{0,1\}^*$, die von einem gleichverteilt zufälligen Bitstring $S \in \{0,1\}^L$ herrührt und wie folgt definiert wird:

- $S(\ell) \in \{0,1\}^\ell$
- $S(\ell)$ ist Präfix von S

Parameter L muss hierbei genügend groß gewählt werden, nämlich entsprechend der Laufzeit, die Angreifern zugestanden wird. (Für uns sind stets nur praktisch denkbare Angreifer von Interesse, also Angreifer mit begrenzten Ressourcen. Setzen wir eine Schranke für die Laufzeit von Angreifern voraus, so beschränken wir damit auch die Anzahl der Bits, die die Angreifer von Orakeln anfordern können: Lange Bitstrings bei einem Orakelzugriff implizieren nämlich einen entsprechenden Zeitaufwand. Deshalb können wir davon ausgehen, dass ein PRG-Angreifer sich bei Anfragen an sein Orakel

schon bei der Wahl von ℓ einigermaßen zurückhält und irgendeine sehr große Grenze L nicht überschreitet.)

Nennen wir das Orakel allgemein $P(\cdot)$. Die Schreibweise $\boldsymbol{B}^{P(\cdot)}$ zeigt dann an, dass $\boldsymbol{B}$ mit diesem Orakel interagiert. Um einen PRG-Angreifer $\boldsymbol{B}^{P(\cdot)}$ zu konstruieren, greifen wir auf einen RoR-OTCPA-Angreifer $\boldsymbol{A}^{E(\cdot)}$ zurück, der das in der Aufgabe beschriebene Verschlüsselungsschema $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ angreift. Als Angreifer in einem Real-or-Random-Angriffsspiel erwartet $\boldsymbol{A}$ ein Orakel $E(\cdot)$, das entweder „echte“ Verschlüsselungen zurückliefert, also die Anfrage m beantwortet mit $g_K\bigl(|m|\bigr) \oplus m$, oder „zufällige“ Verschlüsselungen zurückliefert und die Anfrage m beantwortet mit $g_K\bigl(|m_0|\bigr) \oplus m_0$ für gleichverteilt zufälliges $m_0 \in \{0,1\}^{|m|}$ (das stochastisch unabhängig von m und K gewählt wird).

Der Angreifer $\boldsymbol{A}$ dient hier als „Baumaterial“. Unsere Überlegungen gelten für jeden denkbaren Angreifer $\boldsymbol{A}$; wir nehmen einfach an, dass irgendeine Beschreibung dieses probabilistischen Algorithmus vorliegt. Im zu beschreibenden Angreifer $\boldsymbol{B}$ können wir diesen Algorithmus Schritt für Schritt originalgetreu anwenden – mit einer Ausnahme: $\boldsymbol{A}$ erwartet ein Real-or-Random-Orakel $E(\cdot)$, beim Angreifer $\boldsymbol{B}$ sind wir aber in einem ganz anderen Szenario und haben ein Orakel $P(\cdot)$ wie oben beschrieben zur Verfügung. Das Real-or-Random-Orakel $E(\cdot)$ können wir jedoch unter Benutzung des eigenen Orakels $P(\cdot)$ nachzuempfinden versuchen und eine Antwort zur von $\boldsymbol{A}$ gestellten Anfrage m bestimmen: Wir probieren es mit der Antwort $E(m) = P\bigl(|m|\bigr) \oplus m$. Der Angreifer $\boldsymbol{B}$ führt also den „Programmcode“ von $\boldsymbol{A}$ so aus, als hätte dessen Orakel $E(\cdot)$ genau diese Antwort gegeben. Irgendwann erzeugt $\boldsymbol{A}$ schließlich ein Ausgabebit $\widetilde{b}$. Genau dieses Bit $\widetilde{b}$ lassen wir auch unseren neuen Angreifer $\boldsymbol{B}$ ausgeben. Der Laufzeitaufwand für den Angreifer $\boldsymbol{B}$ ist im wesentlichen so wie beim zugrundeliegenden $\boldsymbol{A}$ (nur beim Aufruf des jeweiligen Orakels gibt es minimale Unterschiede).

Sehen wir uns nun den Angreifer $\boldsymbol{B}^{P(\cdot)}$ in den verschiedenen Szenarien genauer an. Ist $P(\cdot) = g_K(\cdot)$ mit einem konkreten PRG g_K, so lautet die oben beschriebene Antwort für den Angreifer $\boldsymbol{A}$ auf dessen Orakel-Anfrage gerade $E(m) = g_K\bigl(|m|\bigr) \oplus m$, und es handelt sich dabei exakt um den “Real”-Fall des Real-or-Random-Verschlüsselungsorakels. Wenn E_1 dieses “Real”-Verschlüsselungsorakel bezeichnet, folgt daraus

$$\Pr_{K \in_\$ \mathcal{K}}\bigl[\boldsymbol{B}^{g_K(\cdot)} \Rightarrow 1\bigr] = \Pr_{K \leftarrow \mathcal{K}}\bigl[\boldsymbol{A}^{E_1(\cdot)} \Rightarrow 1\bigr].$$

Ist dagegen $P(\cdot) = S(\cdot)$ mit einem Bitstring S, so lautet die Antwort an $\boldsymbol{A}$ auf die Orakel-Anfrage statt dessen $E(m) = S\bigl(|m|\bigr) \oplus m$. Weil S gleichverteilt zufällig ist (und stochastisch unabhängig von m), ist das einfach ein gleichverteilt zufälliger Bitstring der gleichen Länge wie m: Jegliche etwaige Struktur von m außer der Länge geht bei der Exklusiv-Oder-Operation mit dem zufälligen $S\bigl(|m|\bigr)$ verloren. Ähnlich ist es auch im “Random”-Fall des Verschlüsselungsorakels im RoR-OTCPA-Angriffsspiel: Wie oben beschrieben antwortet es mit $g_K\bigl(|m_0|\bigr) \oplus m_0$ für ein zufälliges m_0, und auch das ist ein gleichverteilt zufälliger Bitstring der gleichen Länge wie m (jegliche etwaige Struktur von $g_K\bigl(|m_0|\bigr)$ außer der Länge geht bei der Exklusiv-Oder-Operation mit dem zufälligen m_0 verloren). Wenn E_0 das “Random”-Verschlüsselungsorakel im RoR-OTCPA-Angriffsspiel bezeichnet, folgt also

$$\Pr_{K \in_\$ \mathcal{K}}\bigl[\boldsymbol{B}^{S(\cdot)} \Rightarrow 1\bigr] = \Pr_{K \leftarrow \mathcal{K}}\bigl[\boldsymbol{A}^{E_0(\cdot)} \Rightarrow 1\bigr].$$

Aus den beiden Gleichungen folgt nun unmittelbar

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

wie behauptet, denn nach Definition ist

$$\mathrm{Adv}^{\mathsf{PRG}}_{g,\boldsymbol{B}} \;=\; \Pr_{K\in_\$\mathcal{K}}\bigl[\boldsymbol{B}^{g_K(\cdot)}\Rightarrow 1\bigr] - \Pr_{S\in_\$\{0,1\}^L}\bigl[\boldsymbol{B}^{S(\cdot)}\Rightarrow 1\bigr]$$

und

$$\mathrm{Adv}^{\mathsf{RoR\text{-}OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} \;=\; \Pr_{K\xleftarrow{\$}\mathcal{K}}\bigl[\boldsymbol{A}^{E_1(\cdot)}\Rightarrow 1\bigr] - \Pr_{K\xleftarrow{\$}\mathcal{K}}\bigl[\boldsymbol{A}^{E_0(\cdot)}\Rightarrow 1\bigr].$$

(*Hinweis:* Wir betreiben hier in den Formeln gleich doppelten „Notationsmissbrauch"! Das Symbol $\mathcal{K}$ bezeichnet einerseits eine Menge, z. B. in $K \in_\$ \mathcal{K}$; andererseits bezeichnet es einen probabilistischen Algorithmus, der gleichverteilt ein Element dieser Menge ausgibt, z. B. in $K \xleftarrow{\$} \mathcal{K}$. Das Symbol S bezeichnet einerseits einen Bitstring; andererseits eine Abbildung, die einen Präfix dieses Bitstrings ausgibt. Beide „Verwechslungen" sind für unsere Zwecke unproblematisch: $K \in_\$ \mathcal{K}$ und $K \xleftarrow{\$} \mathcal{K}$ etwa sind im Ergebnis nur verschiedene Schreibweisen für denselben Sachverhalt.)

zu Aufgabe 1.3 [One-Time-Pad-Verschlüsselung]

Sei $\mathcal{K} = \{0,1\}^k$ (mit einer ganzen Zahl $k \geq 1$) und sei $(\mathcal{K},\mathcal{E},\mathcal{D})$ das folgende Verschlüsselungsschema mit Plaintextmenge $\mathcal{M} = \{0,1\}^k$:

- Der Schlüsselgenerierungsalgorithmus $\mathcal{K}$ erzeugt eine Gleichverteilung auf der Menge $\mathcal{K}$.
- Der Verschlüsselungsalgorithmus berechnet $\mathcal{E}_K(m) = K \oplus m$.
- Der Entschlüsselungsalgorithmus berechnet $\mathcal{D}_K(c) = K \oplus c$.

Zeigen Sie: Dieses Verschlüsselungsschema ist sicher im Sinne von RoR-OTCPA.

Diese Aufgabe lässt sich durch eine Abkürzung lösen, die die Ähnlichkeit mit der Situation in Aufgabe 1.2 unterstreicht. Gegenüber Aufgabe 1.2 gibt es hier zwei Unterschiede: Statt eines PRG verwenden wir im Verschlüsselungsalgorithmus direkt den Schlüssel K; und das Verschlüsselungsschema ist (deshalb) nicht zum Verschlüsseln von Plaintexten quasi beliebiger Länge geeignet wie noch in Aufgabe 1.2. Wir können also versuchen zu zeigen, dass die Abbildung $K(\cdot)\colon \{0,\ldots,k\} \to \{0,1\}^*$ selbst auch eine Art PRG ist (lediglich mit einer starken Einschränkung der Bitanzahl). Das ist schnell erledigt: Für jeden PRG-Angreifer $\boldsymbol{A}$ ist

$$\mathrm{Adv}^{\mathsf{PRG}}_{K(\cdot),\boldsymbol{A}} \;=\; \Pr_{K\in_\$\{0,1\}^k}\bigl[\boldsymbol{A}^{K(\cdot)}\Rightarrow 1\bigr] - \Pr_{S\in_\$\{0,1\}^k}\bigl[\boldsymbol{A}^{S(\cdot)}\Rightarrow 1\bigr] \;=\; 0,$$

denn in der Formel für den Vorteil steht hier zweimal genau die gleiche Wahrscheinlichkeit – die unterschiedliche Benennung ändert inhaltlich nichts. Mit dem Ergebnis von Aufgabe 1.2 folgt sofort $\mathrm{Adv}^{\mathsf{RoR\text{-}OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} = 0$.

Alternativ können wir die Aufgabe auch direkt lösen. Sei $\boldsymbol{A}$ nun also ein RoR-OTCPA-Angreifer. Das Verschlüsselungsorakel $E(\cdot)$ nennen wir wie gewohnt im "Real"-Fall $E_1(\cdot)$ und im "Random"-Fall $E_0(\cdot)$. Eine Orakelanfrage $E_1(m)$ liefert das Ergebnis $K \oplus m$; eine Orakelanfrage $E_0(m)$ ein Ergebnis $E_0(m) = K \oplus m_0$ mit gleichverteilt zufälligem $m_0 \in \{0,1\}^k$, also einen zufälligen Bitstring der Länge k. Nun ist der Angreifer im

RoR-OTCPA-Angriffsspiel auf eine einzige Orakelanfrage beschränkt, für einen gleichverteilt zufälligen (dem Angreifer nicht bekannten und deshalb von m stochastisch unabhängigen) Schlüssel $K \in \{0,1\}^k$ ist somit auch $K \oplus m$ ein zufälliger Bitstring. Deshalb ist

$$\Pr_{K \in_\$ \mathcal{K}}\left[\boldsymbol{A}^{E_1(\cdot)} \Rightarrow 1\right] \;=\; \Pr_{K \in_\$ \mathcal{K}}\left[\boldsymbol{A}^{E_0(\cdot)} \Rightarrow 1\right]$$

und somit

$$\mathrm{Adv}^{\text{RoR-OTCPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} \;=\; \Pr_{K \in_\$ \mathcal{K}}\left[\boldsymbol{A}^{E_1(\cdot)} \Rightarrow 1\right] - \Pr_{K \in_\$ \mathcal{K}}\left[\boldsymbol{A}^{E_0(\cdot)} \Rightarrow 1\right] \;=\; 0.$$

Das heißt, das Verschlüsselungsverfahren ist sicher im Sinne von RoR-OTCPA.

zu Aufgabe 1.4

> Zeigen Sie in der Situation von Aufgabe 1.3, dass das dortige Verschlüsselungsschema nicht sicher ist im Sinne von RoR-CPA. (RoR-OTCPA-Sicherheit impliziert also keine RoR-CPA-Sicherheit.)
>
> Beschreiben Sie dafür einen Angreifer, der mindestens den Vorteil $1/2$ erreicht.

Für das Resultat aus Aufgabe 1.3 war wesentlich, dass der Angreifer $\boldsymbol{A}$ auf eine einzige Anfrage ans Verschlüsselungsorakel beschränkt war (*one-time* chosen plaintext attack). Diese Einschränkung gilt beim Sicherheitsbegriff RoR-CPA nicht. Lassen wir den Angreifer also zweimal die gleiche Anfrage m an das Verschlüsselungsorakel senden. Dadurch bekommt er Antworten c_1 und c_2. Im "Real"-Fall berechnen diese sich bei diesem Verschlüsselungsschema beide zu $K \oplus m$, stimmen dann somit überein. Ist $c_1 = c_2$, so lassen wir den Angreifer $\boldsymbol{A}$ also das Bit 1 ausgeben und lassen ihn damit vermuten, dass er in der "Real"-Umgebung abläuft; andernfalls das Bit 0 für "Random". Liegt tatsächlich der "Random"-Fall vor, so werden die Antworten des Verschlüsselungsorakels bestimmt als $K \oplus m_0$, wobei m_0 jeweils ein gleichverteilt zufällig gewählter Bitstring der richtigen Länge ist. Das m_0 bei der ersten Anfrage stimmt nur mit Wahrscheinlichkeit 2^{-k} mit dem m_0 bei der zweiten Anfrage überein, und nur dann gilt $c_1 = c_2$. Im Fall $c_1 = c_2$ würde $\boldsymbol{A}$ auch hier das Bit 1 ausgeben. Wir haben also

$$\mathrm{Adv}^{\text{RoR-CPA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} \;=\; \Pr_{K \in_\$ \mathcal{K}}\left[\boldsymbol{A}^{E_1(\cdot)} \Rightarrow 1\right] - \Pr_{K \in_\$ \mathcal{K}}\left[\boldsymbol{A}^{E_0(\cdot)} \Rightarrow 1\right] \;=\; 1 - \frac{1}{2^k} \;\geq\; \frac{1}{2}.$$

Bemerkung. Eine ähnliche Beobachtung gilt allgemein: Ein Verschlüsselungsschema mit *deterministischer* Verschlüsselung kann nicht sicher sein im Sinn von RoR-CPA oder LoR-CPA, es kann nur für Einmalverschlüsselung Sicherheit bieten (RoR-OTCPA oder LoR-OTCPA). Strebt man RoR-CPA-Sicherheit an, so muss das Verschlüsselungsschema echt *probabilistisch* sein.

zu Aufgabe 1.5

Geben Sie analog zu Folien 2.12 ff. (RoR-OTCPA und LoR-OTCPA) Formeln an, die die folgenden Aussagen ausdrücken:

a. LoR-CPA-Sicherheit impliziert RoR-CPA-Sicherheit.

b. RoR-CPA-Sicherheit impliziert LoR-CPA-Sicherheit.

Was ist die quantitativ stärkere Anforderung: LoR-CPA-Sicherheit oder RoR-CPA-Sicherheit?

(Ansatz: „Sicherheit" in einem bestimmten Sinn heißt, der Vorteil jedes denkbaren Angreifers liegt unter einer Grenze ϵ. Ist ein Schema denkbar, das nur für einen der Sicherheitsbegriffe an der Grenze scheitert?)

Die Überlegungen von Folie 2.12 ff. lassen sich direkt übertragen: Dass die Einschränkung auf "One-Time"-Verschlüsselung wegfällt, ändert dort nichts, wie man anhand von Folien 2.13 und 2.15 nachvollziehen kann. Das bedeutet:

a. Zu einem Angreifer $\boldsymbol{A}$ im RoR-CPA-Angriffsspiel lässt sich ein Angreifer $\boldsymbol{B}$ im LoR-CPA-Angriffsspiel konstruieren mit

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

   Dabei hat $\boldsymbol{B}$ im wesentlichen den gleichen Ressourcenbedarf wie $\boldsymbol{A}$.

b. Zu einem Angreifer $\boldsymbol{A}$ im LoR-CPA-Angriffsspiel lässt sich ein Angreifer $\boldsymbol{B}$ im RoR-CPA-Angriffsspiel konstruieren mit

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

   Dabei hat $\boldsymbol{B}$ im wesentlichen den gleichen Ressourcenbedarf wie $\boldsymbol{A}$.

Nehmen wir jetzt an, eine bestimmte Ressourcenbeschränkung für Angreifer sei festgelegt. Nennen wir ein Verschlüsselungsschema „ϵ-LoR-CPA-sicher" bzw. „ϵ-RoR-CPA-sicher", wenn der LoR-CPA-Vorteil bzw. der RoR-CPA-Vorteil jedes solchen Angreifers unterhalb einer Schranke ϵ bleibt. Ist ein Verschlüsselungsschema nun ϵ-LoR-CPA-sicher, so ist es (laut a oben) auch ϵ-RoR-CPA-sicher. Ist ein Verschlüsselungsschema aber ϵ-RoR-CPA-sicher, so ist es zwar (laut b oben) (2ϵ)-LoR-CPA-sicher, aber unter Umständen nicht ϵ-LoR-CPA-sicher. (Das haben wir konkreter in Aufgabe 1.1 gesehen.) LoR-CPA-Sicherheit ist also eine quantitativ etwas stärkere Anforderung.