Institut für Theoretische Informatik
Prof. Dr. Dennis Hofheinz
Übungsleiter: Thomas Agrikola

Stammvorlesung Sicherheit im Sommersemester 2017

Übungsblatt 4

Aufgabe 1. Wir instanziieren das ElGamal-Verschlüsselungsverfahren aus der Vorlesung über einer Untergruppe $\mathbb{G}$ der Gruppe $\mathbb{Z}_{59}^\times$ (die Gruppenoperation ist die Multiplikation modulo 59). $\mathbb{G}$ sei dabei die Untergruppe der Quadrate. Wir wählen als Erzeuger von $\mathbb{G}$ das Element $g := 27$ (Sie müssen nicht beweisen, dass g ein Erzeuger ist). Der Einfachheit halber sei der geheime Schlüssel sk in dieser Aufgabe direkt der geheime Exponent und der öffentliche Schlüssel $pk = g^{sk}$. (In der Vorlesung enthielten die Schlüssel jeweils noch die Gruppe $\mathbb{G}$ und g.)

(a) Berechnen Sie zum geheimen Schlüssel $sk := 20$ den öffentlichen Schlüssel pk.

(b) Berechnen Sie das Chiffrat c zur Nachricht $m := 22$ unter dem in (a) berechneten Schlüssel pk, wobei Sie $r := 12$ als Zufall verwenden.

(c) Es sei $c := (51, 8)$ ein Chiffrat. Verwenden Sie den geheimen Schlüssel aus (a), um die als Gruppenelement codierte Nachricht $m \in \mathbb{G}$ zu berechnen.

Hinweis: Versuchen Sie, sich die Rechenregeln der Modulo-Arithmetik und für endliche Gruppen zunutze zu machen. Sie sollten dann in der Lage sein, diese Aufgabe größtenteils „im Kopf“ zu lösen.

Zur Erinnerung: Die verwendete Gruppenoperation ist die Multiplikation modulo 59. $\mathbb{Z}_{59}^\times$ ist die Einheitengruppe von $\mathbb{Z}_{59}$ (d.h. die Gruppe der invertierbaren Elemente). Da 59 eine Primzahl ist, gilt $\mathbb{Z}_{59}^\times = \mathbb{Z}_{59} \setminus \{0\}$. Die Ordnung von $\mathbb{Z}_{59}^\times$ ist 58 (Warum?). Die Untergruppe der Quadrate von $\mathbb{Z}_{59}^\times$ besteht aus den Elementen $\{x^2 \bmod 59 \colon x \in \mathbb{Z}_{59}^\times\}$ von $\mathbb{Z}_{59}^\times$ und hat Ordnung 29. Die Gruppenoperation einer Untergruppe ist identisch zur Gruppenoperation der Obergruppe.

Lösungsvorschlag zu Aufgabe 1. Wir wollen uns zunächst allgemein zwei „nützliche Tricks“ zum Rechnen in Gruppen $\mathbb{Z}_N^\times$ für $N \in \mathbb{N}_{\geq 2}$ anschauen:

(i) Finden von handlichen Äquivalenzklassenvertretern, beispielsweise

$$51^2 \bmod 59 = (-8)^2 \bmod 59 = 64 \bmod 59 = 5$$

(ii) Reduzierung des Exponenten modulo der Gruppenordnung, beispielsweise

$$3^{60} \bmod 59 = 3^{60 \bmod 58} \bmod 59 = 3^2 \bmod 59 = 9.$$

Diese Technik geht zurück auf den Satz von Euler (der als Spezialfall den kleinen Satz von Fermat beinhaltet): Für $a, b, N \in \mathbb{N}$ mit $\mathsf{gcd}(a, N) = 1$ gilt $a^b \bmod N = a^{b \bmod \varphi(N)} \bmod N$. Hierbei ist $\varphi(N)$ die *eulersche Phi-Funktion*. Zwei häufige Fälle in der Kryptographie sind:

(a) N prim („ElGamal-Situation“ wie in dieser Aufgabe): Dann ist $\varphi(N) = N - 1$.

(b) $N = PQ$ für P, Q prim („RSA-Situation“): Dann ist $\varphi(N) = (P - 1)(Q - 1)$.

Wir lösen nun die Aufgabe.

(a) Für $g = 27, N = 59$ und $sk = 20$ erhalten wir $pk := g^{sk} \bmod N = 27^{20} \bmod 59 = 3^{60 \bmod 58} \bmod 59 = 3^2 \bmod 59 = 9$.

(b) Um ein Chiffrat $c := (c_1, c_2)$ für $m = 22$ und $r = 12$ zu erhalten, berechnen wir zuerst $g^r \bmod N$, um den Zufall zu „maskieren“: $g^r \bmod N = 27^{12} \bmod 59 = 3^{36} \bmod 59 \overset{(*)}{=} 3^7 \bmod 59 = 81 \cdot 3^3 \bmod 59 = 66 \cdot 9 \bmod 59 = 63 \bmod 59 = 4$. Dies ist der erste Teil des Chiffrats; somit setzen wir $c_1 := 4$. Anschließend berechnen wir den zweiten Teil des Chiffrat mittels $c_2 := pk^r \cdot m \bmod N = 9^{12} \cdot 22 \bmod 59 = 3^{24} \cdot 3^4 \bmod 59 = 3^{-1} \bmod 59 = 20$ und setzen $c := (4, 20)$.

(c) Mit Hilfe des geheimen Schlüssels $sk = 20$ berechnen wir zu einem Chiffrat $c = (c_1, c_2) = (51, 8)$ die Nachricht m als $m := (c_1^{sk})^{-1} \cdot c_2$. Konkret bestimmen wir zunächst $c_1^{sk} = 51^{20} \bmod 59 = -8^{20} \bmod 59 = -2^{60} \bmod 59 = -2^2 \bmod 59 = 4$. Wir invertieren c_1^{sk} mit dem erweiterten euklidischen Algorithmus EE (oder durch scharfes Hinsehen) und erhalten $(c_1^{sk})^{-1} = 15$. Schließlich berechnen wir $m := 15 \cdot 8 \bmod 59 = 2$.

Anmerkung: g aus der Aufgabenstellung ist kein Generator der Gruppe $\mathbb{G} := \mathbb{Z}_{59}^\times$, sondern ein Quadrat und ein Generator der Untergruppe der Quadrate von $\mathbb{Z}_{59}^\times$.

Aufgabe 2. Wir wollen uns damit beschäftigen, wie geeignet die in Aufgabe 1 verwendete Gruppe $\mathbb{Z}_{59}^\times$, beziehungsweise generell Gruppen der Art $\mathbb{Z}_p^\times$ für eine Primzahl $p > 2$, für ElGamal sind. Wir nennen

$$Q(\mathbb{G}) := \{x^2 : x \in \mathbb{G}\}$$

die *Menge der Quadrate* in der Gruppe $\mathbb{G}$. In Gruppen $\mathbb{Z}_p^\times$ lässt sich die Zugehörigkeit zu $Q(\mathbb{Z}_p^\times)$ effizient prüfen: Ist $y^{\frac{p-1}{2}} \bmod p = 1$, so ist $y \in Q(\mathbb{Z}_p^\times)$; andernfalls nicht.

Wie können wir uns die Eigenschaften von Quadraten in $\mathbb{Z}_p^\times$ zunutze machen, um einen erfolgreichen Angreifer auf die IND-CPA-Sicherheit von ElGamal über $\mathbb{Z}_p^\times$ zu konstruieren? Beschreiben Sie ein möglichst realistisches Szenario, in dem ein solcher Angriff nützlich ist.

Hinweise: $Q(\mathbb{Z}_p^\times)$ ist eine Untergruppe von $\mathbb{Z}_p^\times$ und daher insbesondere abgeschlossen unter Multiplikation. Was lässt sich für $x \cdot y$ mit $x \in Q(\mathbb{Z}_p^\times)$ und $y \in (\mathbb{Z}_p^\times \setminus Q(\mathbb{Z}_p^\times))$ über die Zugehörigkeit zu $Q(\mathbb{Z}_p^\times)$ sagen? $Q(\mathbb{Z}_p^\times)$ enthält für $p > 2$ genau $|\mathbb{Z}_p^\times|/2 = (p-1)/2$ Elemente.

Lösungsvorschlag zu Aufgabe 2. Zunächst stellen wir fest, dass $x \cdot y$ mit $x \in Q(\mathbb{Z}_p^\times)$ und $y \in (\mathbb{Z}_p^\times \setminus Q(\mathbb{G}))$ kein Quadrat sein kann: Da x ein Quadrat ist, existiert ein $\overline{x}$ mit $x = \overline{x}^2 \bmod p$. Nehmen wir nun an, es gäbe ein z, so dass $xy = z^2$ gilt. Dann erhalten wir $xy = z^2 \Leftrightarrow y = z^2 x^{-1} \Leftrightarrow y = (z\overline{x}^{-1})^2$. y wäre also ein Quadrat entgegen der Annahme.

Sei nun $c = (a, b)$ ein ElGamal-Chiffrat, wobei $b = a^{sk} m$ für eine Nachricht $m \in \mathbb{Z}_p^\times$ ist. Dank unserer Feststellung, und da $Q(\mathbb{Z}_p^\times)$ eine (Unter-)Gruppe ist, wissen wir, dass b genau dann ein Quadrat ist, wenn a^{sk} und m Quadrate sind. Das Gruppenelement a^{sk} ist genau dann ein Quadrat, wenn a ist Quadrat oder g^{sk} ist Quadrat (dann sorgt nämlich potenzieren mit sk dafür, dass Nicht-Quadrate zu Quadraten werden). Dies kann ein Angreifer ausnutzen:

Wir beschreiben einen Angreifer $\mathcal{A}$ für das IND-CPA-Experiment. $\mathcal{A}$ erhält zunächst den öffentlichen Schlüssel $pk = (\mathbb{Z}_p^\times, g, g^x)$. Als Challenge wählt $\mathcal{A}$ ein nicht-Quadrat m_0 und ein Quadrat m_1 und schickt diese an das Experiment. Als Antwort erhält $\mathcal{A}$ ein Chiffrat $c^* = (a, b)$. Ist a^{sk} kein Quadrat, so rät $\mathcal{A}$ ein zufälliges Bit. Andernfalls rät $\mathcal{A}$ 1, wenn b ein Quadrat ist und 0, wenn b kein Quadrat ist.

Da $Q(\mathbb{Z}_p^\times)$ genau $|\mathbb{Z}_p^\times|/2$ Elemente enthält, wird das zufällige Element a in der Hälfte der Fälle ein Quadrat sein. Das gleiche gilt für das zufällige Element g^{sk}. Das Element a^{sk} ist also in $\frac{1}{4}$ der Fälle ein Nicht-Quadrat und in $\frac{3}{4}$ der Fälle ein Quadrat. Ist a^{sk} kein Quadrat, so wird $\mathcal{A}$ mit schlichtem Raten in der Hälfte der Fälle richtig liegen. Ist a^{sk} ein Quadrat, so liegt $\mathcal{A}$ immer richtig. Die Erfolgswahrscheinlichkeit von $\mathcal{A}$ ist also $\frac{1}{2} \cdot \frac{1}{4} + 1 \cdot \frac{3}{4} = \frac{7}{8}$ und liegt damit $\frac{3}{8}$ (nicht-vernachlässigbar) über schlichtem Raten.

In der Praxis kann ein Angreifer für Chiffrate, bei denen a^{sk} ein Quadrat ist, also ausschließen, dass sie gewisse Nachrichten enthalten (wenn b ein Quadrat ist, muss auch m ein Quadrat sein). Im schlimmsten Fall reicht dies, um die übermittelte Nachricht zu raten (beispielsweise wenn der Angreifer ein Chiffrat von „Ja“ oder „Nein“ erwartet und genau einer dieser Texte als Quadrat und einer als Nicht-Quadrat als Gruppenelement codiert wird).

Für die Instanziierung von ElGamal wird daher oft direkt die Gruppe $Q(\mathbb{Z}_p^\times)$ für eine Primzahl p verwendet, wobei man zusätzlich $p = 2q + 1$ für eine Primzahl q wählt (Primzahlen p dieser Art nennt man auch *safe primes*). So ist die Gruppe $Q(\mathbb{Z}_p^\times)$ von primer Ordnung q. Es wird derzeit angenommen, dass das DDH-Problem über $Q(\mathbb{Z}_p^\times)$ schwer ist.

Aufgabe 3. Es sei $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ ein EUF-CMA-sicheres Signaturverfahren. Wir verwenden die Algorithmen von Σ um zwei weitere Signaturverfahren zu konstruieren:

(1.) Betrachten Sie das Signaturverfahren Σ' mit den folgenden Algorithmen:

- $\mathsf{Gen}'(1^k) := \mathsf{Gen}(1^k)$,
- $\mathsf{Sig}'(sk, M) := (\mathsf{Sig}(sk, M \oplus r), r) = (\sigma_1, \sigma_2)$, wobei r ein zufällig gezogener Bitstring mit $|r| = |M|$ ist,
- $\mathsf{Ver}'(pk, M, \sigma) = \mathsf{Ver}'(pk, M, (\sigma_1, \sigma_2)) := \mathsf{Ver}(pk, M \oplus \sigma_2, \sigma_1)$.

(a) Zeigen Sie die Korrektheit von Σ', d.h. zeigen Sie, dass Ver' Signaturen die mit Sig' erstellt wurden, korrekt verifiziert.

(b) Zeigen Sie, dass Σ' nicht EUF-CMA-sicher ist.

(2.) Betrachten Sie das Signaturverfahren Σ^* mit den folgenden Algorithmen:

- $\mathsf{Gen}^*(1^k) := \mathsf{Gen}(1^k)$,
- $\mathsf{Sig}^*(sk, M) := \mathsf{Sig}(sk, \overline{M})$ (wobei $\overline{M}$ das bitweise Inverse von M ist),
- $\mathsf{Ver}^*(pk, M, \sigma) = \mathsf{Ver}(pk, \overline{M}, \sigma)$

(a) Zeigen Sie die Korrektheit von Σ^*.

(b) Zeigen Sie, dass Σ^* EUF-CMA-sicher ist.

Lösungsvorschlag zu Aufgabe 3.

(1.) (a) Sei M eine Nachricht. Signieren wir M mit dem Algorithmus Sig', so erhalten wir als Signatur

$$\sigma = (\sigma_1, \sigma_2) = (\mathsf{Sig}(sk, M \oplus r), r).$$

Setzen wir dies in den Verifikationsalgorithmus ein, so erhalten wir:

$$\mathsf{Ver}'(pk, M, \sigma) = \mathsf{Ver}(pk, M \oplus r, \sigma_1) = \mathsf{Ver}(pk, M \oplus r, \mathsf{Sig}(sk, M \oplus r)) = 1,$$

was aus der Korrektheit von Σ folgt. Somit folgt die Korrekheit von Σ'.

(b) Wir betrachten den folgenden EUF-CMA Angreifer $\mathcal{A}$. $\mathcal{A}$ erhält zu Beginn pk und schickt eine Nachricht M an sein Sig-Orakel. Er erhält eine gültige Signatur $\sigma = (\sigma_1, r)$ für M zurück. $\mathcal{A}$ wählt nun $M^* \neq M$ beliebig, setzt $r^* = M^* \oplus M \oplus r$ und $\sigma^* = (\sigma_1, r^*)$. Dies ist eine gültige Signatur für M^*, denn es gilt:

$$\begin{aligned}
\mathsf{Ver}'(pk, M^*, \sigma^*) &= \mathsf{Ver}'(pk, M^*, (\sigma_1, r^*)) \\
&= \mathsf{Ver}(pk, M^* \oplus r^*, \sigma_1) \\
&= \mathsf{Ver}(pk, M^* \oplus (M^* \oplus M \oplus r), \sigma_1) \\
&= \mathsf{Ver}(pk, M \oplus r, \sigma_1) \\
&= \mathsf{Ver}'(pk, M, (\sigma_1, r)) \\
&= \mathsf{Ver}'(pk, M, \sigma) \\
&= 1.
\end{aligned}$$

$\mathcal{A}$ hat polynomielle Laufzeit und Erfolgswahrscheinlichkeit 1. Somit kann Σ' nicht EUF-CMA-sicher sein.

(2.) (a) Sei M eine Nachricht. Signieren wir M mit dem Algorithmus Sig^*, so erhalten wir als Signatur

$$\sigma = \mathsf{Sig}^*(sk, M) = \mathsf{Sig}(sk, \overline{M}).$$

Setzen wir dies in den Verifikationsalgorithmus ein, so erhalten wir:

$$\mathsf{Ver}^*(pk, M, \sigma) = \mathsf{Ver}(pk, \overline{M}, \sigma) = \mathsf{Ver}(pk, \overline{M}, \mathsf{Sig}(sk, \overline{M})) = 1,$$

was aus der Korrektheit von Σ folgt. Somit folgt die Korrekheit von Σ^*.

(b) Wir nehmen zum Widerspruch an, dass Σ^* nicht EUF-CMA ist. D.h es exisitert ein effizienter PPT $\mathcal{A}$ mit nicht-vernachlässigbarer Erfolgswahrscheinlichkeit im EUF-CMA-Spiel gegen Σ^*. Wir konstruieren aus $\mathcal{A}$ einen Angreifer $\mathcal{B}$ auf die EUF-CMA-Sicherheit von Σ. $\mathcal{B}$ verwendet $\mathcal{A}$ als Subroutine, indem er für $\mathcal{A}$ das EUF-CMA-Spiel mit Σ^* simuliert.

$\mathcal{B}$ verhält sich in seinem EUF-CMA Spiel mit Σ folgendermaßen:

- $\mathcal{B}$ erhält vom Spiel pk und Zugriff auf ein $\mathsf{Sig}(sk, \cdot)$-Orakel. $\mathcal{B}$ startet $\mathcal{A}$ und gibt pk weiter an $\mathcal{A}$.
- $\mathcal{A}$ stellt im Laufe des Spiel Signaturorakelanfragen und erwartet Signaturen, die von Sig' erstellt wurden. $\mathcal{B}$ muss diese Anfragen mithilfe seines Sig-Orakels beantworten. Schickt $\mathcal{A}$ eine Anfrage für die Nachricht M, so stellt $\mathcal{B}$ eine Anfrage an sein Orakel mit der Nachricht $\overline{M}$, erhält eine Signatur $\sigma = \mathsf{Sig}(sk, \overline{M})$ und gibt σ an $\mathcal{A}$. Da

  $$\sigma = \mathsf{Sig}(sk, \overline{M}) = \mathsf{Sig}^*(sk, M)$$

  hat er damit die Anfrage von $\mathcal{A}$ korrekt beantwortet.
- $\mathcal{A}$ schickt irgendwann eine Fälschung (M^*, σ^*). $\mathcal{B}$ gibt $(\overline{M^*}, \sigma^*)$ als seine Fälschung aus. Dies ist eine gültige Fälschung, wenn (M^*, σ^*) eine gültige Fälschung für Sig^* war, denn es gilt:

  $$\mathsf{Ver}(pk, \overline{M^*}, \sigma^*) = \mathsf{Ver}^*(pk, M^*, \sigma^*) = 1.$$

  Hat $\mathcal{A}$ niemals eine Signaturanfrage für M^* gestellt, so stellt auch $\mathcal{B}$ nie eine Anfrage für $\overline{M^*}$ an sein Orakel. D.h. M^* ist „frisch“ und $\mathcal{B}$ kann $\overline{M^*}$ für seine Fälschung verwenden.

Da $\mathcal{B}$ alle Signaturanfragen von $\mathcal{A}$ korrekt beantwortet, simuliert $\mathcal{B}$ das EUF-CMA Spiel mit Σ^* für $\mathcal{A}$ perfekt. Da $\mathcal{B}$ eine gültige Fälschung ausgibt, wenn $\mathcal{A}$ eine gültige Fälschung ausgibt, ist somit die Erfolgswahrscheinlichkeit von $\mathcal{B}$ gleich der Erfolgswahrscheinlichkeit von $\mathcal{A}$. Diese war nach Voraussetzung nicht vernachlässigbar. Da $\mathcal{B}$ ebenfalls effizient ist (die Laufzeit entspricht ungefähr der von $\mathcal{A}$, mit kleinem Overhead), haben wir einen effizienten und erfolgreichen EUF-CMA Angreifer für Σ konstruiert. Dies steht im Widerspruch zur EUF-CMA-Sicherheit von Σ. Somit kann $\mathcal{A}$ nicht existieren und wir haben die EUF-CMA-Sicherheit von Σ^* gezeigt.

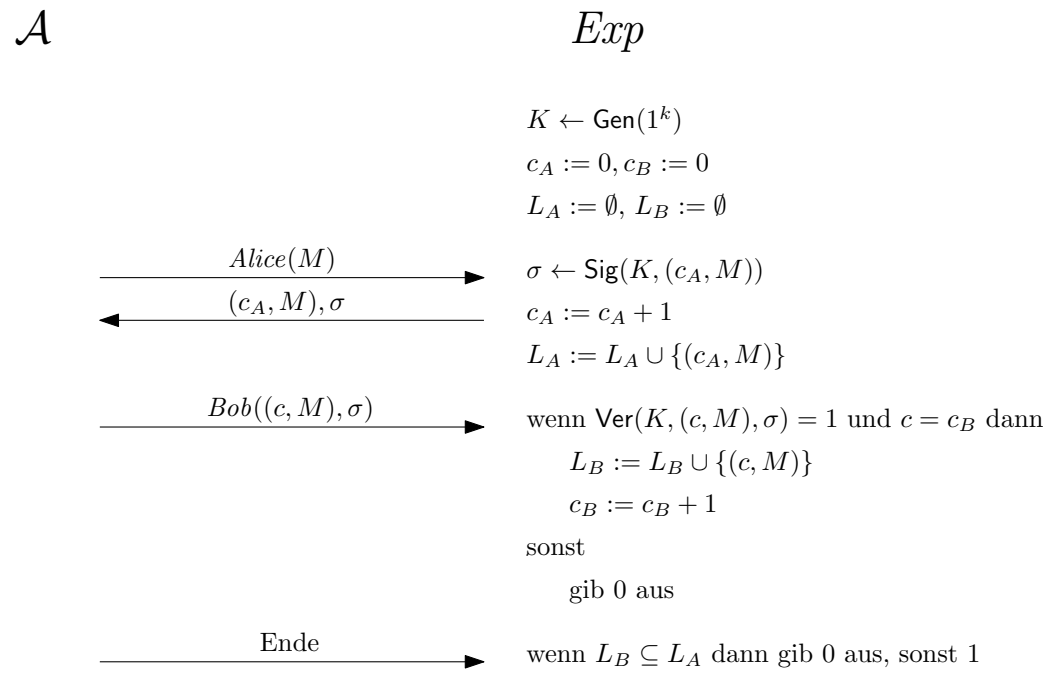

Abbildung 1: Sicherheitsexperiment für einen authentifizierten Kanal.

Aufgabe 4. Es sei $\mathsf{MAC} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ ein MAC-Verfahren. Wir wollen MAC benutzen, um einen authentifizierten Kanal zwischen zwei Parteien $A(lice)$ und $B(ob)$ zu realisieren. (Genau genommen betrachten wir der Einfachheit halber nur Nachrichten, die von A an B geschickt werden). Wir gehen davon aus, dass A und B bereits einen gemeinsamen Schlüssel K ausgetauscht haben. Die Sicherheit des Kanals beschreiben wir mit dem Spiel in Abbildung 1: Das Experiment generiert zunächst einen Schlüssel K und initialisiert Zähler c_A, c_B für die Parteien A und B sowie Listen der akzeptierten Nachrichten L_A und L_B. Der Angreifer kann nun (polynomiell oft) eine der folgenden Anfragen stellen:

- $Alice(M)$: Der Angreifer kann Alice eine Nachricht M schicken lassen. Alice signiert das Paar aus Nachricht und ihrem aktuellen Zähler. Das Zähler/Nachrichten-Paar wird der Liste L_A der von Alice geschickten Nachrichten hinzugefügt und der Zähler c_A hochgezählt.
- $Bob((c, M), \sigma)$. Der Angreifer kann Bob ein Zähler/Nachrichten-Paar (c, M) und eine Signatur σ empfangen lassen. Bob prüft Signatur und Zählerstand. Ist beides valide, fügt er das Zähler/Nachrichten-Paar der Liste seiner empfangenen Nachrichten hinzu und zählt seinen Zähler hoch. Ansonsten bricht das Experiment mit 0 ab (der Angreifer verliert).
- *Ende*: Der Angreifer beendet das Experiment. Sind die Nachrichten, die Bob akzeptiert hat, eine Untermenge der Nachrichten, die Alice geschickt hat, so verliert der Angreifer (0-Ausgabe). Ansonsten gewinnt er (1-Ausgabe).

Beweisen Sie, dass kein Angreifer das beschriebene Spiel für ein EUF-CMA-sicheres MAC-Verfahren MAC mit nicht-vernachlässigbarer Wahrscheinlichkeit gewinnen kann. Geben Sie hierzu an, wie Sie einen Angreifer $\mathcal{A}$ für das Spiel in Abbildung 1 benutzen können, um einen Angreifer $\mathcal{B}$ für das EUF-CMA-Spiel zu konstruieren. Hierbei sollte $\mathcal{B}$ nicht-vernachlässigbaren Erfolg haben, sofern $\mathcal{A}$ nicht-vernachlässigbaren Erfolg hat.

Diese Konstruktion eines authentifizierten Kanals aus einem MAC unter Zuhilfenahme von Zählern ist in der Praxis üblich und wird (sehr ähnlich) beispielsweise in Transport-Layer-Security-Verschlüsselungsprotokoll (TLS-Protokoll) verwendet. Überlegen Sie sich, warum ein Zähler eingeführt und mit der Nachricht authentifiziert wird.

Lösungsvorschlag zu Aufgabe 4. Es sei $\mathcal{A}$ ein Angreifer auf den durch das Spiel aus Abbildung 1 beschriebenen authentifizierten Kanal. Wir konstruieren einen Angreifer $\mathcal{B}$ auf die EUF-CMA-Sicherheit

von MAC, der $\mathcal{A}$ benutzt. $\mathcal{B}$ lässt $\mathcal{A}$ laufen und *simuliert* das Spiel aus Abbildung 1 für ihn. Zunächst initialisiert $\mathcal{B}$ Zähler und Listen wie in der Beschreibung des Experiments angegeben. Dann startet $\mathcal{B}$ den Angreifer $\mathcal{A}$ und reagiert auf seine Anfragen wie folgt:

- *Alice*(M): $\mathcal{B}$ erfragt eine Signatur σ für das Zähler/Nachrichten-Paar (c_A, M), inkrementiert c_A, fügt (c_A, M) der Liste L_A hinzu und schickt $(c_A, M), \sigma$ an $\mathcal{A}$.
- *Bob*$((c, M), \sigma)$: $\mathcal{B}$ prüft, ob $(c, M) \in L_A$. Falls nicht, schickt er $(c, M), \sigma$ als Fälschung an das EUF-CMA-Experiment. Andernfalls prüft $\mathcal{B}$, ob $c = c_B$. Falls nicht, bricht er das Spiel mit $\mathcal{A}$ ab. Andernfalls inkrementiert er c_B.
- *Ende*: $\mathcal{B}$ bricht das Spiel mit $\mathcal{A}$ ab.

Wir wissen aus Abbildung 1, dass $\mathcal{A}$ das Sicherheitsexperiment für den authentifizierten Kanal gewinnt (1-Ausgabe), wenn $L_B \not\subseteq L_A$ (d. h. $L_B \setminus L_A \neq \emptyset$).

Wir zeigen nun, dass sich die Simulation von $\mathcal{B}$ für $\mathcal{A}$ von dem eigentlichen Sicherheitsexperiment nicht unterscheiden lässt, falls $\mathcal{A}$ gewinnt. In beiden Fällen wird die Funktion Gen benutzt, um einen Schlüssel K zu ziehen. Die Anfragen *Alice*(M) werden vollständig deckungsgleich beantwortet; d. h. $\mathcal{A}$ erhält eine Signatur unter K.

Schwieriger ist es bei Anfragen der Art *Bob*$((c, M), \sigma)$. Da $\mathcal{B}$ kein Verifikationsorakel hat, kann er nicht feststellen, ob die Signatur σ gültig ist. Daher richtet er sich nur nach der Zugehörigkeit zu L_A. Für $(c, M) \notin L_A$ endet das Spiel und die Interaktion mit $\mathcal{A}$. Für $(c, M) \in L_A$ unterscheiden wir die folgenden Fälle.

- σ nicht gültig: In diesem Fall hätte $\mathcal{A}$ das Spiel aus Abbildung 1 verloren. Die Simulation läuft zwar weiter, ist aber nun unterscheidbar vom Sicherheitsexperiment für einen authentifizierten Kanal. Wir müssen daher davon ausgehen, dass $\mathcal{A}$ nun keine Anfragen mehr stellen wird, die $\mathcal{B}$ helfen, eine gültige Fälschung zu finden.
- σ gültig: In diesem Fall ist die Verifikation im Spiel aus Abbildung 1 erfolgreich.

Sowohl in der Simulation wie auch im richtigen Spiel wird abgebrochen, genau dann wenn $c \neq c_B$.

Außerdem gewinnt $\mathcal{B}$ das EUF-CMA-Experiment, wenn $\mathcal{A}$ gewinnt: Sei (c, M) das Zähler/Nachrichten-Paar aus $L_B \setminus L_A$, das zuerst (im Experimentablauf) zu L_B hinzugefügt wurde. Da $\mathcal{A}$ gewinnt, muss die zu (c, M) übertragene Signatur σ gültig sein. Da $(c, M) \notin L_A$, sendet $\mathcal{B}$ $(c, M), \sigma$ als Fälschung an das EUF-CMA-Experiment und gewinnt, da σ gültig ist.

Also gewinnt $\mathcal{B}$, wenn $\mathcal{A}$ gewinnt. Insbesondere würde ein mit nicht-vernachlässigbarer Wahrscheinlichkeit erfolgreicher Angreifer $\mathcal{A}$ zu einem mit nicht-vernachlässigbarer Wahrscheinlichkeit erfolgreichem Angreifer $\mathcal{B}$ auf die EUF-CMA-Sicherheit von MAC führen. Die Existenz eines solchen Angreifers widerspricht unserer Annahme. □

Der Zähler in der Konstruktion wird eingeführt, um Replay-Attacken zu verhindern. Bei einer Replay-Attacke würde $\mathcal{A}$ eine zuvor belauschte und von *Alice* signierte Nachricht M mit der dazugehörigen Nachricht erneut an *Bob* schicken und ihn glauben lassen, *Alice* hätte M zweimal geschickt.