Institut für Theoretische Informatik
Jun.-Prof. Dr. D. Hofheinz

Stammvorlesung Sicherheit im Sommersemester 2014

Übungsblatt 4

Hinweis: Übungsblätter können freiwillig bei Florian Böhl, Raum 255, Geb. 50.34 („Info-Bau“) bis zur Übung am 12.6.14 zur Korrektur abgegeben werden. Die Korrektur dient nur der Selbstkontrolle; es gibt keine Punkte und keinen Klausur-Bonus.

Aufgabe 1. Wir instanziieren das ElGamal-Verschlüsselungsverfahrens aus der Vorlesung über der Gruppe $\mathbb{G} := \mathbb{Z}_{59}^*$ und wählen $g := 27$ (g erzeugt nicht $\mathbb{G}$ sondern die Untergruppe der Quadrate[1] von $\mathbb{Z}_{59}^*$ der Ordnung 29). 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 zu dem geheimen Schlüssel $sk := 20$ den öffentlichen Schlüssel pk.

(b) Berechnen Sie das Chiffrat c zu der 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.

Lösungsvorschlag zu Aufgabe 1. Wir wollen uns zunächst allgemein zwei „nützliche Tricks“ zum Rechnen in Gruppen $\mathbb{Z}_N^*$ 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 $pk := g^{sk} \bmod N = 27^{20} \bmod 59 = 3^{60 \bmod 58} \bmod 59 = 3^2 \bmod 59 = 9$.

[1] Die Untergruppe der Quadrate von $\mathbb{Z}_{59}^*$ besteht aus den Elementen $\{x^2 \bmod 59 : x \in \mathbb{Z}_{59}^*\}$

(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 $m := 15 \cdot 8 \bmod 59 = 2$.

Hinweis: g aus der Aufgabenstellung ist kein Generator der Gruppe $\mathbb{G} := \mathbb{Z}_{59}^*$, sondern ein Quadrat und ein Generator der Untergruppe der Quadrate von $\mathbb{Z}_{59}^*$ (siehe Aufgabe 2). Warum eine Gruppe primer Ordnung ratsam ist betrachten wir in Aufgabe 4.

Aufgabe 2. Wir beschäftigen uns mit der Homomorphie des ElGamal-Verschlüsselungsverfahrens. Sei $\mathbb{G}$ eine öffentlich bekannte Gruppe primer Ordnung p mit Generator g. Weiterhin seien zwei Parteien A und B gegeben. A kennt ein Polynom

$$f(x) := \sum_{i=0}^{d} a_i x^i \in \mathbb{F}_p[X]$$

($\mathbb{F}_p$ ist hier der Körper mit p Elementen). B möchte für einige Elemente $E \subseteq \mathbb{F}_p$ testen, ob sie Nullstellen von f sind. A will f allerdings nicht einfach preisgeben (ist aber bereit B den Grad d von f mitzuteilen). Entwerfen Sie ein kryptographisches Protokoll auf Basis des ElGamal-Verschlüsselungsverfahrens, dass die gewünschte Funktionalität realisiert. Gehen Sie dabei davon aus, dass A, B beide kooperativ agieren (d.h. sie halten sich an das vorgegebene Protokoll), dabei aber versuchen, so viel wie möglich zu erfahren[2]. Sie dürfen außerdem annehmen, dass A und B bereits über einen sicheren Kanal kommunizieren; an dieser Stelle kommt also kein Angreifer ins Spiel. Was erfährt A bei Ihrem Protokoll günstigstenfalls über die Menge E? Was B günstigstenfalls über f? Könnte eine bessere Lösung existieren?

Lösungsvorschlag zu Aufgabe 2. TODO

[2] Dieses „honest but curious“-Verhalten ist in der IT-Sicherheit oft ein realistisches Szenario.

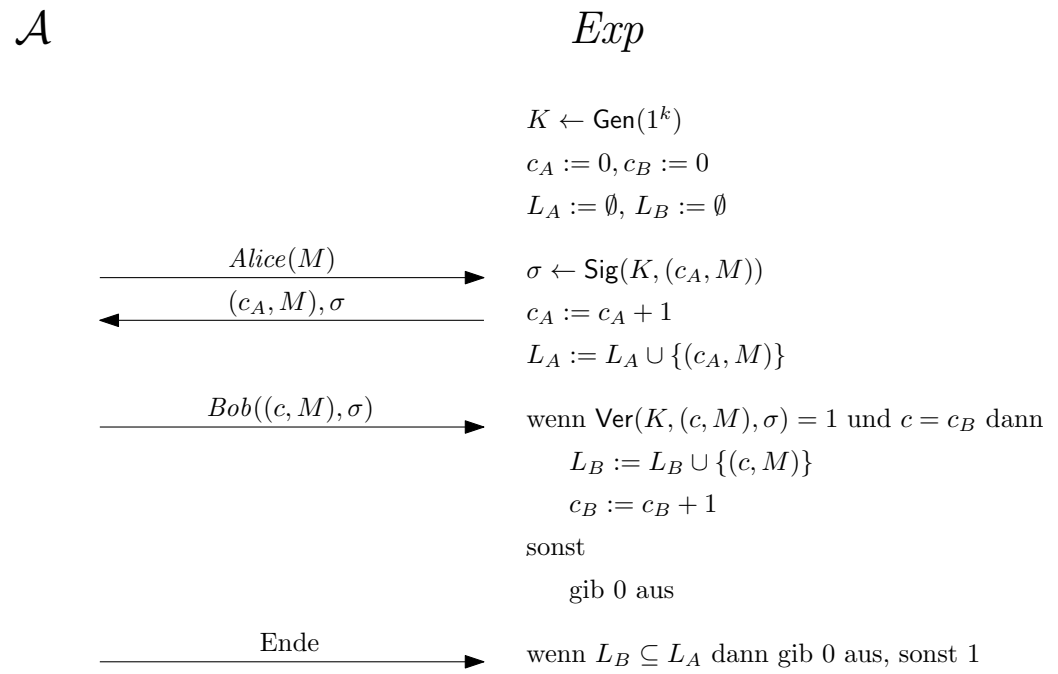

Abbildung 1: Sicherheitsexperiment für einen authentifizierten Kanal.

Aufgabe 3. 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 für die Parteien A und B sowie Listen der akzeptierten Nachrichten L_A und L_B. Der Angreifer kann nun beliebig 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.
- *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. 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 Signaturverfahren 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 3. 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 diesen Fall ist die Verifikation im Spiel aus Abbildung 1 erfolgreich.

Sowohl in der Simulation wir 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.

Aufgabe 4. Der ebenso geniale wie durchtriebene Wissenschaftler und Superbösewicht Doktor Meta ist beunruhigt. Erneut schweift sein Blick über eine Anordnung von Monitoren, die die Videos der über seine geheime Liegenschaft verteilten Überwachungskameras direkt im Kontrollraum anzeigen. Er lehnt sich zurück. Sieben Tage ist es nun her, dass er Annette Gui, die Geliebte seines Gegenspielers Max Security, in seine Gewalt bringen konnte. Seitdem hat er sein Anwesen keine wache Sekunde aus dem Auge gelassen. Wird er Anettes Aufenthaltsort noch lange genug vor Max Security geheim halten können, um seinen gewieften Plan realisieren zu können? Eigentlich kann sie niemand hier erwarten. Doktor Meta seufzt und blickt auf den Kalender. Schon am kommenden Montag beginnt SuperCon – das ultimative Jahrestreffen der Superbösewichte, das er sich auf keinen Fall entgehen lassen will. Die Überwachung des Anwesens muss er in dieser Zeit seinen Schergen überlassen. Sorgenvoll wendet er sich seiner Tastatur zu und beginnt zu tippen.

Max Security reibt sich die Augen. Sein normalerweise ruhiger tiefer Schlaf ist seicht und von Alpträumen durchzogen seit Anette verschwunden ist. Obwohl ihre Beziehung wahrlich nicht als sorgenfrei beschrieben werden kann, wünscht er sich nichts sehnlicher als Anette wieder an seiner Seite zu haben. Ohne sie fühlt er sich nicht komplett. Max' Bewusstsein nimmt erst jetzt das Piepen und die grüne Kontrollleuchte seiner Superbösewichtüberwachungsanlage wahr. Noch schlaftrunken dreht er sich aus dem Bett und wendet sich einem Monitor zu. Zahlenkolonnen rasen über das Display – eine abgefangene Nachricht von Doktor Meta. Sofort ist Max hellwach. Wird er endlich einen Hinweis auf Anettes Aufenthaltsort finden?

Doktor Meta verwendet das ElGamal-Verschlüsselungsverfahren. Auf der Vorlesungswebseite finden Sie zwei Dateien

- `Sicherheit_UE04_A4_Oeffentlich.txt` enthält einen primen Modulus p, einen Generator g für die multiplikative Gruppe $\mathbb{Z}_p^*$ ($\langle g \rangle = \mathbb{Z}_p^*$) und den verwendeten öffentlichen Schlüssel pk.
- `Sicherheit_UE04_A4_Chiffrat.txt` enthält die abgefangene Nachricht. Aus technischen Gründen muss Doktor Meta seine Nachricht zeichenweise in ASCII-Codierung (d.h. `A` $\mapsto$ 65) verschlüsseln und verwendet nur Großbuchstaben (keine Umlaute, kein scharfes S).

Lösungsvorschlag zu Aufgabe 4. TODO