



Lösungsskizzen zu Übung 1

Achtung: Die Lösungsskizzen sollen nur die Ideen veranschaulichen und sind daher gelegentlich sehr informell geschrieben; präzisere Lösungen sollten allerdings leicht daraus ableitbar sein.

Aufgabe 1 (One-Time-Pad)

Beide Verfahren bieten keinen Integritätsschutz, zumindest nicht gegen vorsätzliche Veränderungen. Der Angreifer kann in beiden Fällen seinen Wert δ ebenfalls in Blöcke aufteilen und dann die Summe $\Delta = \delta_1 \oplus \dots \oplus \delta_k$ bilden, und (C, M) zu $(C^*, M^*) = (C \oplus \delta, M \oplus \Delta)$ im ersten Fall sowie (C, V) zu $(C^*, V^*) = (C \oplus \delta, V \oplus \Delta)$ im zweiten Fall abändern. In beiden Fällen wird Bob $m^* = C^* \oplus k = m \oplus \delta$ dekodieren und dann gilt jeweils $M^* = M \oplus \Delta = m_1 \oplus \delta_1 \oplus \dots \oplus m_k \oplus \delta_k$. Bob bemerkt also auch hier keine Änderungen.

Die erste Lösung zerstört sogar die perfekte Geheimhaltung des ursprünglichen One-Time-Pad-Verfahrens, da der Angreifer nun Informationen über die Nachricht erhält. Besteht beispielsweise m aus zwei Blöcken und der zweite Block ist stets $0 \dots 0$, so kann der Angreifer die komplette Nachricht aus $M = m_1 \oplus 0 \dots 0 = m_1$ rekonstruieren.¹

Die zweite Lösung erhält zumindest noch die vom One-Time-Pad-Verfahren gewohnte Geheimhaltung, da (C, V) als One-Time-Pad-Verschlüsselung von (m, M) mit Schlüsseln (k, K) angesehen werden kann.

Aufgabe 2 (One-Way-Funktionen)

Es gilt: g_1 ist nicht one-way, g_2 ist one-way, g_3 ist nicht one-way.

- (a) Intuitiv ist die Funktion g_1 nicht one-way, da nur ein Bit der Eingabe geheim bleibt. Formal konstruieren wir folgenden Angreifer \mathcal{A}_1 gegen g_1 . Algorithmus \mathcal{A}_1 erhält als Eingabe 1^n und $f(b||0^{|y|})||y$, wobei y die Länge $n-1$ hat. Er schneidet zunächst y mit der bekannten Länge $n-1$ ab und erhält den Wert $z = f(b||0^{n-1})$. Er berechnet die Werte $z_0 = f(0^n)$ und $z_1 = f(1||0^{n-1})$ und gibt $a||y$ aus, wobei das Bit a so gewählt wird, dass der Wert z_a mit z übereinstimmt (stimmen beide Werte überein, wähle $a = 0$).

¹Mehr (Aufwand) kann in der Kryptographie also auch weniger (Sicherheit) bedeuten.

Man beachte, dass \mathcal{A}_1 effizient ist: die einfachen Bitoperationen lassen sich leicht effizient implementieren, und die beiden Auswertungen der Funktion f sind ebenfalls in Polynomzeit möglich (da One-Way-Funktionen effizient berechenbar sind). Ferner gilt, dass \mathcal{A}_1 stets ein Urbild findet, also mit nicht vernachlässigbarer Wahrscheinlichkeit erfolgreich ist.

- (b) Die Funktion g_2 sollte one-way sein, da man zum Invertieren auch f invertieren muss. Allerdings “enthüllt” g_2 ein Bit der Eingabe von f . Dieses Bit hätte man sich allerdings auch raten können.

Der Beweis erfolgt durch Widerspruch. Wir nehmen an, dass g_2 nicht one-way wäre. Dann gibt es einen effizienten Algorithmus \mathcal{A}_2 , der mit nicht vernachlässigbarer Wahrscheinlichkeit g_2 invertiert. Wir konstruieren daraus einen erfolgreichen Angreifer \mathcal{A}_f gegen f . Algorithmus \mathcal{A}_f erhält als Eingabe 1^n und einen Wert $z = f(x)$. Er wählt ein Bit b' zufällig und gibt $(1^n, b' || z)$ an \mathcal{A}_2 weiter. Dieser Algorithmus produziert eine (eventuell richtige) Ausgabe $b || y$, die \mathcal{A}_f einfach ausgibt.

Offensichtlich ist \mathcal{A}_f effizient (da er im wesentlichen nur den effizienten Algorithmus \mathcal{A}_2 als Unterprogramm ausführt). Für die Erfolgswahrscheinlichkeit beachte man, dass \mathcal{A}_f in der Hälfte der Fälle \mathcal{A}_2 auf einer “falschen” Eingabe laufen lässt (nämlich wenn das geratene Bit b' nicht mit dem ersten Bit von x übereinstimmt). Umgekehrt liegt \mathcal{A}_f mit Wahrscheinlichkeit $1/2$ aber auch richtig, und in diesem Fall liefert uns \mathcal{A}_2 mit nicht vernachlässigbarer Wahrscheinlichkeit $\epsilon(n)$ ein Urbild. Insgesamt hat \mathcal{A}_f daher die Erfolgswahrscheinlichkeit $\frac{1}{2} \cdot \epsilon(n)$, die ebenfalls nicht vernachlässigbar ist (entweder direkt überlegen oder Aufgabe 3b mit $q(n) = 2$ verwenden). Da wir vorausgesetzt haben, dass f one-way ist, dürfte es einen solchen Angreifer aber nicht geben. Folglich muss unsere Annahme, dass g_2 *nicht* one-way ist, falsch gewesen sein.

- (c) Folgt für g_3 analog zu g_1 indem man für den Angreifer gegen g_1 den öffentlichen Teil y statt $0^{|y|}$ verwende.

Aufgabe 3 (vernachlässigbare Funktionen)

- (a) Da $\nu(n)$ vernachlässigbar ist, gilt insbesondere $\nu(n) \leq \frac{1}{p^2(n)}$ für das spezielle Polynom $p^2(n)$ und alle $n \geq n_0$ für ein hinreichend großes n_0 . Da alle Werte nicht-negativ sind, gilt das folglich auch für die Wurzeln.
- (b) Angenommen, $\delta(n) := \epsilon(n)/q(n)$ wäre vernachlässigbar. Dann wäre wegen der Rechenregel auch $\epsilon(n) = q(n) \cdot \delta(n) \approx 0$, im Widerspruch zur Voraussetzung $\epsilon(n) \not\approx 0$.
- (c) Angenommen, $\delta(n) := \epsilon(n) - \nu(n)$ wäre vernachlässigbar. Dann wäre auch $\epsilon(n) = \delta(n) + \nu(n) \approx 0$, Widerspruch.