Grundlagen der Informationssicherheit WS 2017/2018: Übungsblatt #3

Due on Dienstag, December 05, 2017

Gruppenabgabe

Lukas Baur, Felix Bühler, Marco Hildenbrand

Aufgabe 1

Table 1: Fast-Exponentiation

i	h	k
3	1	9
2	9	$81 \equiv 4$
1	36	16
0	36	$256 \equiv 25$
-1	$900 \equiv 53$	9

Das Ergebnis ist demnach 53.

Aufgabe 2

Problem 2a
Nein, es ist keine kollisions-resistente Hash-Funktion: Da jeder Eingabewerte denselben Hash-Wert ergibt, sobald die Anzahl der 1en identisch sind, ist es trivial eine zweite Nachricht erzeugen, die dieselbe Anzahl an Einsen hat. Falls x kein Pallindrom ist, reicht es zum Beispiel aus, die Eingabe zu spiegeln, andernfalls reich das Permutieren der Eingabe, damit ein $\tilde{x}$ ensteht, sodass $H(x) = H(\tilde{x})$ gilt.

Problem 2b
Nein, es ist keine kollisions-resistente Hash-Funktion, dazu berechnen wir zu einem x ein $\tilde{x}$, sodass $h(x) = h(\tilde{x})$ gilt:
Sei x_0 beliebig aber fest der Länge L-1. Dessen Hash-Wert ist dann $h(x_0)$ (mit der Länge l). Nun erstellen wir $x = h(x_0)||h(x_0)$. x ist der Länge $> L$ aber $< 2L$ und $H(x)$ wird dadurch gebildet mit $H(x) = h(x_0) \oplus h(x_0) = a \oplus a = 0^l$
Bestimme nun ein x_1 beliebig aber fest der Länge L-1. Dessen Hash-Wert ist dann $h(x_1)$ (mit der Länge l). Nun erstellen wir $\tilde{x} = h(x_1)||h(x_1)$. Analog zu dem ersten Teil haben wir für $H(\tilde{x})$ wieder 0^l. Damit wurde eine Kollision gefunden.□

Aufgabe 3

Im Folgenden zeigen wir, dass folgendes gilt:
h kollisionsresistent => h' kollisionsresistent

Wir können aus einer Kollision für h' eine Kollision für h berechnen. Ergebe x und x' eine Kollision für h': $h'(x) = x(0)|h(x) = x'(0)|h(x') = h'(x')$. daraus folgt ganz offensichtlich dass $h(x) = h(x')$ gilt. Also ist (x, x') auch eine Kollision unter h.

Sei h kollisionsresistent
Dann existiert kein effizientes Verfahren, mit dem zu einem x leicht ein x' berechnet werden kann, sodass

$h(x) = h(x')$ gilt. Wir nehmen im folgenden an, dass kein solches x' eine Kollision existiert, besser gesagt, dass der Angreifer keines findet. Das Finden einer Kollision in h' kann sehr leicht auf das Finden einer Kollision in h reduziert werden. Will man nun zu x eine Kollision mit x' in h' finden, muss folgendes gelten: $h'(x) = x(0)||h(x) = x'(0)||h(x') = h'(x')$. Damit ist das erste Bit für x' bereits bestimmt, es bleibt folgenden Problem zu lösen: h(x) = h(x'). Dieses Problem ist (hier) nicht lösbar, da gemäß der Annahme kein x' dazu berechnet werden kann. □

Aufgabe 4

Das Security-Spiel Besteht aus 2 Phasen:

Phase 1: Anfragen
Erhalte Mac für $x = x_0 = 0^l$ also $m_0 = E(x_0 \oplus v, k) = E(v, k)$
Erhalte Mac für $x' = x_0||m_0$ also $m_1 = E(m_0 \oplus m_0, k) = E(0^l, k)$
Erhalte Mac für $x'' = k$, k beliebiger Bit-String der Länger l, aber unterschiedlich zu 0^l. Der erhaltene Mac heißt m_2

Phase 2: Challenge
Sende nun als Challenge $(x''||m_2, m_1)$. Da x'' den verschlüsselten Wert m_2 hat, gibt später $E(x'' \oplus v, k) \oplus m_2 = 0^l$ gilt $x''||m_2$ ist selber Mac wie $m1$

Damit kann ein Angreifer verschiedene Eingaben x_i erstellen, die alle denselben Mac haben und davor noch nie an Alice zum mac-en gesendet worden sind. Die Wahrscheinlichkeit das Spiel zu gewinnen ist somit 1, also ist der Vorteil = 1. □

Aufgabe 5

Sei die darunterliegenden Hash-Funktion nicht kollisionsresistent und es kann demnach leicht ein x' zu einem x gefunden worden, sodass h(x) = h(x') gilt.
Jetzt gilt:
PKCS-sig$(x, (n, d))$ =PKCS-hash$(x)^d$ *mod* n und außerdem
PKCS-sig$(x', (n, d))$ =PKCS-hash$(x')^d$ *mod* n.
Da allerding PKCS-hash(x) *mod* n = PKCS-hash(x') *mod* n gilt, gilt auch
PKCS-hash$(x)^d$ *mod* n = PKCS-hash$(x')^d$ *mod* n sowie
PKCS-sig$(x, (n, d))$ = PKCS-sig$(x', (n, d))$. Also ist es leicht möglich, eine gleiche Signatur zu erzeugen, wenn man eine nicht-kollisionsresistente Funktion darunterliegend implementiert.

Sicherheitsspiel mit adavantage 1: Lasse ein x von Alice signieren. Erhalte s. Sende eine Challenge an Alice: (x', s). Der Angreifer gewinnt immer, da der Hash derselbe ist, da x' hier so gewählt ist, dass es eine Kollision mit x gibt, und demnach auch der exponentierte Hash derstelbe ist. Die Wahrscheinlichkeit ist demnach 100% und der Advantage demzufolge 1.