Dr. Dirk Frettlöh 19.6.2019

Übungen zur Vorlesung Kryptographie

Blatt 11

Aufgabe 40: (Buchstaben zu Punkten zu Buchstaben)
Wir benutzen die Koblitz-Kodierung aus der Vorlesung für die elliptische Kurve mit der Gleichung $y^2 = x^3 + x + 3$ über $\mathbb{F}_{17}$. Dabei ist wie üblich a=0, b=1, ... z=25. Also können wir jeden Buchstaben mit 6 Bit darstellen. Wir wählen hier also $d = 4$ und zerschneiden eine zu verschlüsselnde Botschaft (in Binärcodierung) in 2-Bit-Worte.

(a) Berechnen Sie die Koblitz-Kodierung des Buchstaben "s". Zeigen Sie Ihre Berechnung.

(b) Welches Wort ergibt das Ent-Kodieren der nach obigem Schema Koblitz-kodierten Nachricht

$$(2,8),(2,8),(8,8),(2,9),(2,8),(2,9),(6,2),(8,8),(2,9),(2,9),(8,9),(12,3),(2,9),(6,15),(2,8),(6,2),(8,8),(2,9)?$$

Aufgabe 41: (Einfache Hashfunktion)
Wir codieren Buchstaben als $a = 00, b = 01, c = 02 \ldots, z = 25$. Wir benutzen eine Merkle-Damgård-Konstruktion mit Startwert $s = x_0 = 23$. Die m_i sind die einzelnen Buchstaben $m_1, m_2, \ldots, m_n$ des zu hashenden Texts, gefolgt von der Länge $m_{n+1} := n$ der Nachricht; also $m = (m_1, m_2, \ldots, m_n, m_{n+1} = n)$, als zweistellige Zahlen gelesen. Die Kompressionsfunktion $x_i = f(x_{i-1}, m_i)$ $(i = 1, \ldots, n+1)$ funktioniert folgendermaßen:

(1) $y = 7 \cdot (m_i + x_{i-1}) \bmod 100$
(2) Vertausche die Ziffern von y, nenne diese neue Zahl z (Obacht: aus $y = 7 = 07$ wird 70)
(3) $x_i = x_{i-1} + z \bmod 100$

Was ist der Hashwert $h(m)$ des Wortes $m =$"passwort"? Finden Sie ein (sinnfreies) Wort mit demselben Hashwert $h(m)$. Was ist der Hashwert von $m' =$"a"? Was ist der Hashwert von $m'' =$"password"? Was ist der Hashwert des leeren Strings ""? Zeigen Sie die einzelnen Schritte der Berechnung, oder den Code.

Aufgabe 42: (Bessere Hashfunktion?)
Betrachten wir die Kompressionsfunktion $f : Z_{101} \times Z_{101} \to Z_{101},\ f(x,y) = 91^x \cdot 10^y \bmod 101$ und $g : Z_{107} \times Z_{107} \to Z_{107},\ g(x,y) = 2^x \cdot 5^y \mod 107$.

(a) Realisieren Sie die zugehörigen Hashfunktionen h zu f und h' zu g nach der Merkle-Damgård-Konstruktion (wie in Aufgabe 30, mit Startwert $s = x_0 = 41$ und mit Padding — also $m_{n+1} = n =$ Länge des Texts) und berechnen Sie jeweils die Hashwerte der Worte "bob", "eve", "sam", "sue". *(Gerne programmieren, als Abgabe reichen die korrekten Hashwerte.)*

(b) Bestimmen sie die Wertebereiche von h und h' — also die Menge aller Werte, die h und h' jeweils annehmen können. *(Als Abgabe reichen die korrekten Wertebereiche.)*

(c) Welche der beiden Funktionen ist die eindeutig ungeeignetere Hashfunktion? Erklären Sie, woran das liegt!

(d) Wir betrachten nur noch die Kompressionsfunktion g und die zugehörige Hashfunktion h' (mit Length Padding, und mit Startwert $x_0 = 41$). Finden Sie eine Kollision für g.

(e) Konstruieren Sie daraus eine Kollision von h' mit $h'(m) = h'(m')$, wobei $m = (m_1, m_2, m_3, m_4)$, $m' = (m'_1, m'_2, m'_3, m'_4)$ sowie $m'_1 \neq m_1$, $m'_2 \neq m_2$ und $m'_3 = m_3$, $m'_4 = m_4$.

(f) Wäre ihr Vorgehen in (e) auch durchführbar, wenn wir statt modulo 107 mit modulo p rechnen würden, wobei p 256 Binärstellen (also 256 bit) hat? Begründen Sie Ihre Antwort! Wie lange würde es dann auf einem heutigen Rechner dauern?

Aufgabe 43: (Wozu Padding?)
Eine naheliegende Variante der Merkle-Damgård-Konstruktion benutzt weder Startwert noch Padding. Ihre Aufgabe ist herauszufinden, warum das keine gute Idee ist. Sei der zu hashende Text $(m_0, m_1, \ldots, m_n)$. Betrachten Sie folgende Hashfunktion mit der Kompressionsfunktion g aus Aufgabe 42.

Setze $x_0 = m_0$. Für $i = 1, 2, \ldots, n$: Berechne $x_i = g(x_{i-1}, m_i)$. Ausgabe $h(m) = x_n$.

(a) Finden Sie zur Nachricht $m = (0, 11, 8, 2, 4)$ drei Kollisionen. Genauer: finden sie zu m drei weitere Urbilder m', m'', m''' der jeweiligen Länge 4, 3 und 2 mit $h(m) = h(m') = h(m'') = h(m''')$.

(b) Erläutern Sie allgemein, wie man hier zu einer Nachricht m der Länge $n + 1$ leicht eine Nachricht m' der Länge n findet mit $h(m) = h(m')$. Erläutern Sie auch, wie Padding dies verhindert, und wie das Nutzen eines Startwerts das verhindert.

(Teil (a) kann brute-force erledigt werden. Wer aber Teil (b) kann, kann Teil (a) ohne viel Aufwand erledigen.)

Wenn Sie Programmcode abgeben, bitte (a) auch das Ergebnis abgeben, (b) den Code selbst (kein Foto) an den Tutor schicken.

Abgabe: Mittwoch 26.6.2019 bis 14 Uhr in Postfach 2183 in V3, oder per Email an den Tutor.
Bitte auf jeder Abgabe das Tutorium angeben! (Di oder Do)

Dienstag	Philipp Braukmann	pbraukmann@techfak.uni-bielefeld.de
Donnerstag	Oliver Tautz	otautz@techfak.uni-bielefeld.de