Cheatsheet: Computersystemsicherheit

Version vom 10. September 2016

CBC-Mode

Nachricht in n Teile gliedern.
Schlüssel ⇒ Permutationsmatrix mit Einheitsvektoren

$$\pi = \begin{pmatrix} 1 & \cdots & n \\ a & \cdots & z \end{pmatrix} \Rightarrow P_\pi = \begin{pmatrix} e_a \\ \vdots \\ e_z \end{pmatrix}$$

Verschlüsselung:
$c_0 = IV$
$c_i = E(\pi,c_{i-1} \oplus m_i) = P_\pi \cdot (c_{i-1} \oplus m_i)$
$\Rightarrow \quad c_1,\ldots,c_n$
Entschlüsselung:
$m_i = D(\pi^{-1},c_i) \oplus c_{i-1} = (P_{\pi^{-1}} \cdot c_i) \oplus c_{i-1}$
mit $\pi^{-1} = \pi'$ (transponiert)

CFB-Mode

Verschlüsselung:
$c_i = E(\pi,c_{i-1}) \oplus m_i$
Entschlüsselung:
$m_i = E(\pi,c_{i-1}) \oplus c_i$

CTR-Mode

Verschlüsselung:
$c_i = E(\pi,IV + (i-1)) \oplus m_i$
(binäre Addition, Überträge verwerfen)
Entschlüsselung:
$m_i = E(\pi,IV + (i-1)) \oplus c_i$

Hashfunktion

Nicht injektive Abbildung, die Urbildbildbereich auf erheblich kleineren Bildbereich abbildet.
Speicherung von Passwörtern, Dateivalidierung

Message Authentication Code (MAC)

Hashfunktion mit geheimen Schlüssel zur Integritätsprüfung von Nachrichten. Ermöglicht kein non-repudiation, daher nicht als digitale Unterschrift geeignet

Kollisionsresistenz

Es ist schwierig zwei Werte x und y mit $H(x) = H(y)$ zu bestimmen

Schwache Kollisionsresistenz

Es ist schwierig zu geg. Wert x ein x' mit $H(x) = H(x')$ zu bestimmen

Shamir Secret-Sharing

t von n Stakeholdern sind nötig, um Geheimnis $k = f(0)$ zu entschlüsseln. Außerdem gegeben: Primzahl $p > n$,k und vom Dealer gewähltes Polynom f(x) vom Grad $t-1$
Schlüssel $s_i = f(i) \mod p$
Secret Recovery:
$k = f(0) = \sum s_i \cdot l_i(0) \quad l_i(0) := \left[\prod_{i=1,j\neq i} \frac{j}{j-i}\right] \mod p$

Erweiterter Euklidischer Algorithmus (EEA)

$ggT(a,b)$ und $x \cdot a + y \cdot b = d$

q	r	x	y	a	b	x_2	x_1	y_2	y_1
X	X	X	X	a	b	1	0	0	1
a/b	$a \mod b$	$x_2 - qx_1$	$y_2 - qy_1$	b	r	x_1	x	y_1	y
...									
?	0	?	?	**=d**	**0**	**=x**	?	**=y**	?

Inverses Element $(a)^{-1}$ berechnen

Es gilt: $\left[a \cdot (a)^{-1}\right] \mod k = 1$
$ggT(a,k)$ mit EEA durchführen.

$$(a)^{-1} = \begin{cases} (x+k) \mod k & \text{wenn } x < 0 \\ x \mod k & \text{sonst} \end{cases}$$

Remote Repositories

Änderungen veröffentlichen

Weitere praktische Kommandos

Beispielhafte Programmaufrufe

Platz für eigene Anmerkungen!