IT-Sicherheit I - Zusammenfassung

February 1, 2014

1 Mathematische Grundlagen

Gruppe: (G,*), falls * assoziativ, $\exists$ neutrales Element, $\exists$ inverses $\forall$ g.
Ring: (G,+,*), falls (R,+) abelsche Gruppe, * distributiv, * hat Einselement, * Assoziativ. Ein Element heißt Einheit, falls es ein multiplikatives Inverses gibt.
Einheitengruppe: R* die Menge aller invertierbaren Elemente eines kommutativen Rings $(R, +, *)$ mit Einselement, dann ist (R*,$*$) eine abelsche Gruppe (Einheitengruppe).
Satz: Ist G eine Gruppe der Ordnung ($|G|$) n und neutrales Element e, so gilt: $g^n = e \forall g \in G$.
Schnelle Exponentation: Sei G eine Gruppe, $g \in G, n \in \mathbb{N}, g^n$ *gesucht*
Erstelle Tabelle mit $i = l = max(\lfloor log_2(n) \rfloor, 0), h = 1, k = g$
Solange $i \geq 0$:
Falls b(i) (Binärdarstellung von n an Stelle i) $\rightarrow$ h=k*h
$k = k^2$ **Berechnung von** $\Phi(n)$: $Phi(n) = (p_0 - 1) * p_0^{\alpha_0 - 1} * (p_1 - 1) * p_1^{\alpha_1 - 1} ...$ mit $p_i^{\alpha_i}$ Primfaktorzerlegung.
Erweiterter Euklidischer Algorithmus:e(a,b)
Bedingung: $a, b \in \mathbb{Z}, a \geq b \geq 0$
1. $a' = a, b' = b, x_0 = 1, y_0 = 0, x_1 = 0, y_1 = 1$
2. Schleife solange $b' \neq 0$
$q = a' \ div \ b', r = a' \ mod \ b', a' = b', b' = r$
$(x_0, y_0, x_1, y_1) = (x_1, y_1, x_0 - qx_1, y_0 - qy_1)$
y_1 nach dem letzten Schleifendurchlauf wo $b' \neq 0$ ist das Inverse von b in $\mathbb{Z}_a$.
$a' = ggt(a, b)$.

2 Grundlagen

2.1 Kryptosystem

Ein Kryptosystem ist ein Tupel $S = (X, K, Y, e, d)$ mit X: Klartextmenge, K: Schlüsselmenge, Y: Chiffretexte, e: Chiffrierfunktion, d: Dechiffrierfunktion. Folgende Eigenschaften müssen gegeben sein:

$$d(e(x, k), k) = x \ \forall x \in X, k \in K \tag{1}$$

$$Y = \{e(x, k) | x \in X, k \in K\} \tag{2}$$

2.2 Kryptoschema

Ein Kryptoschema ist ein Tupel der Form (X, K, E, D), wobei
$X \subseteq \{0, 1\}^*$ (möglicherweise Unendlich)
$K \subseteq \{0, 1\}^*$ endlich
E ein effizienter probabilistischer Verschlüsselungsalgorithmus
D ein effizienter deterministischer Entschlüsslungsalgorithmus ist.
Es soll die Dechiffrierbedingung gelten.

2.3 Asymmetrisches Kryptoschema

Ein asymmetrisches Kryptoschema ist ein Tupel $S = (X, G, K, E, D)$
mit einer Menge K von Schlüsselpaaren $(k, \hat{k})$ mit k als öffentlichen und $\hat{k}$ als privaten Schlüssel.
Ein effizienter propabilistischer Schlüsselgenerierungsalgorithmus G
Klartextraum X
Chiffrieralgorithmus $e(x, k)$
Dechiffrieralgorithmus $d(x, \hat{k})$
Dechiffrierbedingung muss gelten.

2.4 Sicherheitsspiel

2.4.1 Symmetrische Verschlüsselung

1. Eva sendet Charlie einige Klartexte x_i
2. Charlie verschlüsselt diese und sendet die Chiffren zurück
3. Eva wählt zwei Klartexte der selben Länge, Charlie verschlüsselt einen davon
4. Eva darf sich wieder einge Texte verschlüsseln lassen
5. Eva wählt einen aus den zwei Klartexten aus.

Sie gewinnt, falls der gewählte Klartext der ist, den Charlie verschlüsselt hat.
Für AES verschlüsselt Charlie in Punkt 3 nur ein zufälliges Bit.

2.4.2 Asymmetrische Verschlüsselungen

1. Charlie erstellt ein Schlüsselpaar und sendet den öffentlichen an k.
2. Eva wählt zwei Klartexte und sendet sie an Charlie
3. Charlie verschlüsselt einen dieser Texte
4. Eva wählt den verschlüsselten Klartext, sie gewinnt, wenn sie richtig liegt.

2.4.3 MAC

1. Charlie wählt einen zufälligen Schlüssel k
2. Eva darf sich eine Klartexte von Charlie etikettieren lassen.
3. Eva versucht ein gültiges Nachrichten-Etiketten-Paar (x,t) zu berechnen.

Falls $V(x, t, k) = ja$, hat Eva gewonnen.
Vorteil: adv(A)=Prob(A gewinnt)

2.4.4 Digitale Signaturen

1. Charlie wählt zufällig einen Schlüssel $(k, \hat{k})$
2. Eva darf sich einige Klartexte x_i wählen, die Charlie signiert.
3. Eva versucht ein gültiges Nachrichten-Etiketten-Paar (x,s) zu berechnen

Sie hat gewonnen, falls $V(x, s, k) = ja$
Sie hat einen Vorteil von $adv(A) = Prob(A\ gewinnt)$

2.5 Kasiski-Test

Sei y ein Chiffretext (durch Blockweises Vigenere verschlüsseln). Seien i_0, i_1 Positionen in y an denen ein häufig auftretendes Trigramm vorkommt. Man vermutet: $m|(i_1 - i_0)$. Um m zu bestimmen wählt man viele Positionen i_p, $m \approx ggT\{i_j - i'_j | j, j' \in \{0, ..., p\}\}$.

3 Verschlüsselungsverfahren

Bei einem symmetrischen Verfahren besitzen beide Seiten den selben Schlüssel, um die Nachricht zu entschlüsseln.

3.1 Vernam-Kryptosystem

Ein One-time-Pad Vernam Kryptosystem der Länge l sieht folgendermaßen aus:

$$S = (\{0,1\}^l, \{0,1\}^l, \{0,1\}^l, e, d) \tag{3}$$

mit $e(x,k) = x \oplus k \forall x, k \in \{0,1\}^l$ und $d(y,k) = y \oplus k \forall y, k \in \{0,1\}^l$

3.2 Possibilistische Sicherheit

Ein Kryptosystem heißt possibilistisch sicher, falls $\forall y \in Y, x \in X \exists k \in K : e(x,k) = y$ Jeder Klartext soll mit beliebigen Schlüsseln auf jede Chiffre abgebildet werden können.
Wenn ein Kryptosystem possibilistisch sicher ist gilt: $|K| \geq |Y| \geq |X|$ (Beweis: e muss wegen d injektiv sein $\rightarrow |Y| \geq |X|$, sei $e(x_0, k_y) = y$ und $y \neq y'$, dann muss $k_y \neq k_{y'}$ sein und $|K| \geq |Y|$).

3.3 Substitutionskryptosystem

Das Substitutionskryptosystem ist ein Tupel (X, P_X, X, e, d), wobei P_X die Menge aller Permutationen auf X (Menge aller Bijektionen von X nach X) ist. Außerdem gilt $e(x,\pi) = \pi(x)$, $d(y,\pi) = \pi^{-1}(y) \forall x, y \in X, \pi \in P_X$
π bezeichnet eine Ersetzung der Buchstaben aus X.

3.4 Verschiebekryptosystem

Das Verschiebekryptosystem mit Parameter $n > 0$ ist: $(\mathbb{Z}_n, \mathbb{Z}_n, \mathbb{Z}_n, e, d)$ mit $e(x,k) = x +_n k$ und $d(y,k) = y -_n k$. Das Verschiebekryptosystem ist für $n > 0$ possibilistisch sicher. Das Verschiebekryptosystem ist bei buchstabenweiser/blockweiser Verschlüsselung nicht sicher.

3.5 Affines Kryptosystem

Das affine Kryptosystem mit Parameter $n > 1$ ist das Kryptosystem $(\mathbb{Z}_n, \mathbb{Z}_n^* \times \mathbb{Z}_n, \mathbb{Z}_n, e, d)$, mit $e(x,(a,b)) = a *_n x +_n b$ und $d(y,(a,b)) = a^{-1} *_n (y -_n b)$, wobei a^{-1} das Inverse von a modulo n ist.
Die affinen Kryptosysteme mit Parameter $n > 1$ sind possibilistisch sicher. Bei blockweiser Verschlüsselung für $l \geq 2$ ist es unsicher.

3.6 Vigenere Kryptosystem

Das Vigenere Kryptosystem mit Parameter $n > 0$ und Periode $m > 0$ ist $((\mathbb{Z}_n)^m, (\mathbb{Z}_n)^m, (\mathbb{Z}_n)^m, e, d)$ mit $e(x,k) = (x_0 +_n k_0)(x_1 +_n k_1).....(x_{m-1} +_n k_{m-1})$ und demnach Enstschlüsselung mit $y_i -_n k_i$.
Das Vigenere Kryptosystem ist für alle Parameter $n > 0, m > 0$ possibilistisch sicher.
Brechen der Verschlüsselung: 1. Bestimme m mittels Kasiski-Test 2. Bestimme k wie für normales Verschiebekryptosystem.

4 AES

AES ist definiert für die Schlüssellängen 128,192,256 und die Blocklängen 128,160,192,224,256. In der 128-128-Bit-Variante ist es wie folgt definiert: $AES = (\{0,1\}^{128}, \{0,1\}^{128}, \{0,1\}^{128}, e, d)$ Statt e(x,k) wird oft AES(x,k) geschrieben.
Ein Klartext x wird als 4x4 Matrix interpretiert (mit jeweils 8Bit pro Matrixfeld. Analog der Schlüssel.
Es gibt i Rundenschlüssel $K(k,i) \in \{0,1\}^{128}$ mit $K(k,0) = k$, die berechnet werden.
Ablauf AES:

1. initialer Weißschritt: $U = X \oplus K(k,0)$
2. 9 Runden (i=1 bis 9):
    a. Substituiere mithilfe der S-Box
    b. Rotiere Zeilen
    c. Durchmische Spalten

d. Addiere Schlüssel

3. a. Substitution per S-Box
   b. Rotiere Zeilen
   c. Addiere Schlüssel

5 Verschlüsselungsmodi

5.1 ECB - Electronic Code Book

Jeder Block wird durch den Schlüssel k unabhängig verschlüsselt. Am Beispiel AES:
Sei die Länge von x durch 128 teilbar.
Verschlüsselung:

Zerlege x in Blöcke der Länge 128 $x = x_0...x_{n-1}, x_i \in \{0,1\}^{128}$
$y_i = AES(x_i, k) \forall i \in \{0, ..., n-1\}$

5.2 (R-)CBC - Cipher Block Chaining

Jeder Block wird vor dem Verschlüsseln mit der vorherigen Chiffre per XOR verknüpft. Der erste Block wird mit einer Zufallszahl verknüpft.
Beispiel AES:

1. $x = x_0, ..., x_{n-1}$
2. y_{-1} =zufällig gewählter Bitstring der Länge 128
3. $y_i = AES(y_{i-1} \oplus x_i, k)$

5.3 (R-)CTR - Counter

Eine Zufallszahl, die mit jedem Block um 1 erhöht wird verschlüsselt und per XOR mit dem Klartext verknüpft.
Beispiel AES:

1. $x = x_0, ..., x_{n-1}$
2. y_{-1} =zufällig gewählter Bitstring der Länge 128
3. $y_i = AES(y_{-1} + i \ mod \ 2^{128}, k) \oplus x_i$

6 RSA

Schlüsselraum: $K = \{((n,e)(n,d))\}$ mit $p \neq q \in PRIME, p, q > 2, e * d \ mod \ m = 1, m = \Phi(n) = (p-1)(q-1)$
Der Schlüsselgenerierungsalgorithmus findet zwei Primzalen und ein $e \in \mathbb{Z}^*_{\Phi(n)}$ und berechnet $d = e^{-1} \ mod \ \Phi(n)$.
Chiffrieralgorithmus: $e(x,(n,e)) = x^e \ mod \ n$
Dechiffrieralgorithmus: $d(y,(n,d)) = y^d \ mod \ n$

7 Hashfunktionen

7.1 Definition

(L,l)-beschränkte Hashfunktion: $h : \{0,1\}^{\leq L} \rightarrow \{0,1\}^l$ mit $L > l > 1$
unbeschränkte Hashfunktion: $h : \{0,1\}^* \rightarrow \{0,1\}^l$ mit $l > 1$

l heißt Hashbreite, h(x) Hashwert von x

(l,b)-Kompressionsfunktion: $f : \{0,1\}^l \times \{0,1\}^b \rightarrow \{0,1\}^l$
Iterationsfunktion: $i^f : \{0,1\}^l \times \{0,1\}^{b*} \rightarrow \{0,1\}^l$

$i^f(u, \epsilon) = u$
$i^f(u, xv) = f(i^f(u,x), v)$ mit $x \in \{0,1\}^{b*}, v \in \{0,1\}^b$

Merkle-Damgard Füllfunktion: $p^{b,r}_{MD} : \{0,1\}^{<2^r} \rightarrow \{0,1\}^{b+}$

$p_{MD}^{b,r}(x) = x||1||0^s||(x)_2^r$, wobei $s \geq 0$ minimal mit $b - r = |x| + 1 + s \ mod \ b$
$(x)_2^r$ ist ein r-Bit-String der die Länge von x codiert.
MD-Hashfunktion: $h_u^{f,p} : \{0,1\}^{<2^r} \to \{0,1\}^l$ mit $h_u^{f,p}(x) = i^f(u, p(x))$
Wenn die Kompressionsfunktion kollisionsresistent ist, dann auch die iteriert Hashfunktion.

7.2 Rainbowtables

Es werden Ketten der Länge k gebildet, anfangend mit einem möglichen Passwort.
Für einen gegebenen Hashwert berechnet man nur abwechselnd R (die eine Funktion von Hashes in die Klartexte) und h, bis man mit weniger als k Schritten auf ein Kettenende gestoßen ist. Nun berechnet man die Kette mit dem zugehörigen Startwert, bis man auf das gewünschte Passwort stößt.
In den Ketten wird für jede Stufe der Kette eine andere Reduktionsfunktion gewählt.
Als Gegenmaßnahme verwendet man ein Salt, das Passwort wird gespeichert: $(Person, h(pw||s), s)$, womit zu jedem Passwort auch ein Zufallsstring kommt.

8 MAC

8.1 Definition

Symmetrisches Authentifizierungsverfahren: $M = (X, K, Y, T, V)$ mit
X: nicht leerer Klartextraum
Y: nicht leerer Etikettraum
K: Menge von Schlüsseln
$T : X \times K \to Y$ Etikettieralgorithmus
$V : X \times Y \times K \to \{ja, nein\}$
sodass: $V(x, T(x, k), k) = ja \forall x \in X, k \in K$
Gültiges Nachrichten-Etikett-Paar (x,t): Falls $V(x, t, k) = ja$
Falls (x,t) ein gültiges Nachrichten-Etikett-Paar, so ist t ein gültiges Etikett zu x.

8.2 Konstruktion aus Kryptosystem

Sei $B = (\{0,1\}^l, \{0,1\}^s, \{0,1\}^l, e, d)$ mit $l, s > 0$ ein Blockkryptosystem.
MAC: $M_B = (\{0,1\}^l, \{0,1\}^s, \{0,1\}^l, T, V)$ mit $T(x, k) = e(x, k)$, $V(x, t, k) = ja, \ falls \ d(t, k) = x, \ nein \ sonst$
Problem: Lässt sich nur auf Nachrichten der Länge l anwenden. Lösung mittels Zerlegung und Auffüllung der Nachrichtenblöcke.

8.3 CBC-MAC

Sei $B = (\{0,1\}^l, \{0,1\}^s, \{0,1\}^l, e, d)$ ein Blockkryptosystem mit $l, s > 0$
$v \in \{0,1\}^l$ beliebig.
Dadurch induzierte CBC-MAC:$M_{CBC} = (\{0,1\}^{l+}, \{0,1\}^s, \{0,1\}^l, T, V)$ mit T(x,k):

1. Zerlege x in Blöcke der Länge
2. Setze $y_{-1} = v$
3. Berechne $y_i = e(y_{i-1} \oplus x_i, k)$ für alle i
4. Gib y_{n-1} aus.

8.3.1 Sicherheit

Der CBC-Mac ist sicher, falls das Blockkryptosystem sicher ist, und nur auch Nachrichten gleicher Länge angewandt wird.
Unischer, falls Nachrichten unterschiedlicher Länge
Unsicher, falls der Initialisierungsvektor teil des Etiketts ist.

8.4 HMAC

Sei f eine (l,b) Kompressionsfunktion mit $l, b > 0$ und $8|b$.
Sei $p = P_{MD}^{b,r}$ die Merkle-Damgard-Füllfunktion
Sei $u \in \{0,1\}^l, l \geq m \geqq b$ mit $8|m$
Sei $ipad = 00110110$ und $opad = 01011100$
$HMAC[f, p, u, m] = (\{0,1\}^{<2^r-b}, \{0,1\}^m, \{0,1\}^l, T, V)$
$T(x,k) = h(k_0||h(k_i||x))$, wobei $h = h_u^{f,p}$, $k_0 = k||0^{b-m} \oplus opad^{b/8}$, $k_i = k||0^{b-m} \oplus ipad^{b/8}$.

9 Digitale Signaturen

9.1 Signierschema

$S = (X, G, K, T, V)$
K: Schlüsselmenge, wobei jeder Schlüssel ein Paar $(k, \hat{k})$ ist, mit k Verifikationsschlüssel und $\hat{k}$ Signierschlüssel.
G: Effizienter propabilistischer Schlüsselgenerierungsalgorithmus
X: $(X_k)_{k \in K_{pub}}$ Familie von Nachrichtenräumen
T: Effizienter propabilistischer Signieralgorithmus T(x,$\hat{k}$)
V: Effizienter deterministischer Verifikationsalgorithmus V(x, y, k)
Es gilt: $V(x, T(x, \hat{k}), k) = ja \ \forall (k, \hat{k}) \in K, x \in X_k$
(x,s) heißt **gültiges Nachrichten-Signaturen-Paar**, falls $V(x, s, k) = ja$, s heißt dann gültige Signatur.

9.2 Signieren per RSA-Umkehrung

$S_{RSA} = (X, G, K, T, V)$ mit:
$K = \{((n,e),(n,d)) : n = pq$ mit Primzahlen $p \neq q, p > 2, q > 2, e * d \ mod \ m = 1, m = \Phi(n)\}$
G wählt zwei Primzahlen $p \neq q, p > 2, q > 2$ und ein Element $e \in Z_{\Phi(n)}^*$, berechnet $= e^{-1} \ mod \ \Phi(n)$ und gibt ((n,e),(n,d)) aus.
Nachrichtenraum Z_n
$T(x,(n,d)) = x^d \ mod \ n$, $(n,d) \in K_{priv}, x \in Z_n$
$V(x, s, (n,e)) =$ ja, falls $s^e \ mod \ n = x$, nein sonst.
Unsicher, da $(x^e \ mod \ n, x)$ gültiges Paar.
Ein Fälscher kann sich zu einem $r \in Z_n^*$ und $x * r^{-1}$ eine Signatur holen, dann ist $S(r) * S(x * r^{-1})$ eine Signatur für x.

9.3 Sicheres Signierschema

$S_{RSA,h} = (\{0,1\}^*, G, K, T', V')$
$T'(x,(n,d)) = h_n(x)^d \ mod \ n$
$V'(x, s, (n,e)) =$ ja, falls $h_n(x) = s^e \ mod \ n$
Mit einer idealen Hashfunktion h.

10 Public Key Infrastruktur

Schwache Schlüsselbindung: Kommunikationsteilnehmer behauptet er besäße den öffentlichen Schlüssel. Starke Schlüsselbindung: Der Kommunikationsteilnehmer besitzt den privaten Schlüssel.
Feststellen der starken Schlüsselbindung:

1. Alice muss den öffentlichen Schlüssel von Bob über einen authentischen Kanal bekommen
2. Proof of Posession: Alice muss überprüfen, ob Bob den privaten Schlüssel besitzt.

10.1 Challenge-Response-Protokoll

PoP per Ver- und Entschlüsselung:
Alice schickt Bob einen durch den öffentlichen Schlüssel Verschlüsselten Text. Falls Bob den richtigen Klartext zurückschickt, besitzt er den privaten Schlüssel.
Dabei muss sichergestellt werden, dass kein Man-In-The-Middle Angriff besteht.
PoP per digitalen Signaturen:
Alice schickt Bob einen Klartext. Bob schickt eine gültige Signatur zurück. Falls Alice die Signatur verifizieren kann, besitzt Bob den privaten Schlüssel.

10.2 Zero-Knowledge-Beweis

Sudoku:
Zeile/Spalte/Quadrat wählen. Die darin enthaltenen Zahlen werden entnommen, gemischt und vorgelegt.
Man kann nun nachvollziehen, dass überall nur die Zahlen 1-9 enthalten sind, aber nicht, wo sich welche Zahl befindet.
3-Färbbarkeit eines Graphen:
Kante zum öffnen wählen.
Farben werden gezeigt, nach zudecken werden die Farben neu durchmischt

10.3 PKI, Zertifikate

Bindungsproblem wird an zentrale Zertifizierungsstellen überlassen.
Ein Zertifikat ist ein Dokument, was den Eigentümer eines öffentlichen Schlüssel beglaubigt.
Ein Zertifikat, welches den öffentlichen Schlüssel k der Person Z zuordnet und von Z ausgestellt wurde: $Zert_{\hat{k}_Z}(Y, k)$ mit $\hat{k}_Z$ dem privaten Schlüssel der CA.
Probleme: 1.Von wem erhält Alice $Zert_{\hat{k}_Z}(Bob, k)$? 2.Warum vertraut Alice Z? 3.Gehört der öffentliche Schlüssel von Z wirklich Z?
1. Mail, Website, Zertifikatserver
2. Bekanntheit von Z
3. Eine CA (unmöglich), mehrere CA (jeweils Vertrauen entscheiden), Hierarchien (Vertrauen vererbt)

10.4 Web-Of-Trust

Nutzer vertrauen Nutzern und verifizieren diese.
PGP - Pretty good privacy
Jeder nutzer besitzt zwei Schlüsselringe:
PubRing: Ein Netz öffentlicher Schlüssel (selbstsigniert).
Entält öffentliche Schlüssel anderer Nutzer
PrivRing: Private Schlüssel des Besitzer
Darstellung PubRing Anhang.
1. Bestimmung der Vertrauensgrade: vertrauenswürdig, eingeschränkt, nicht
2. Bestimmung der Schlüsselbindung:
Zuerst ist nur der Besitzer mit gültiger Schlüsselbindung. (A)
   1. Falls N' ein Zertifikat von einem vertrauenswürdigen Nutzer mit gültiger Bindung ausgestellt, dann ist N' mit gültiger Bindung.
   2. Wurden für N' von mindestens zwei eingeschränkt vertr. Nutzern mit gültiger Bindung Zertifikate ausgestellt, so hat N' eine gültige Bindung.
   3. Wurde für N' von genau einem eingeschränkt vertr. Nutzer mit gültiger Bindung ein Zertifikat ausgestellt, so hat N' eine eingeschr. gültige Bindung.

10.5 Kryptographische Protokolle

Ermöglichung sicherer Kommunikation durch Verschlüsselung und Signierung.
Einfaches Beispiel:
Alice sendet Bob $E(N_A, K)$ mit N_A Zufallszahl und K ein gemeinsamer Schlüssel.
Bob sendet N_A zurück. Damit meint Alice sicher mit Bob zu reden.
Das Verfahren ist unsicher, da ein Angreifer I, der von Alice $E(N_A, K)$ abfängt, das selbe zurücksenden Kann und von Alice automatisch die Lösung N_A bekommt.

10.5.1 Needham-Schroeder

1. A to B: $E(N_A||A, K_B)$
2. B to A: $E(N_A||N_B, K_A)$
3. A to B: $E(N_B, K_B)$

Unsicher:
A to I: $E(N_A||A, K_I)$ I to B: $E(N_A||A, K_B)$
B to I: $E(N_A||N_B, K_A)$ I to A: $E(N_A||N_B, K_A)$
A to I: $E(N_B, K_I)$ I to B: $E(N_B, K_B)$
Alice redet mit I, aber Bob glaubt mit Alice zu reden.
Verbesserung:
2.: B to A: $E(N_A||N_B||B, K_A)$