Zusammenfassung DC

Paul Lödige
Matrikel: 15405036

SoSe 2020

Inhaltsverzeichnis

Kapitel 1

Substitionsverfahren

1.1 Skytale

Ein Streifen wird um einen Stock gewickelt und dann beschrieben. Nach dem abwickeln erhält man den entsprechenden Code. Durch aufwickeln auf einen Stock mit dem gleichen Umfang lässt sich die Nachricht wieder entschlüsseln.

Tipp:
Zum entschlüsseln der Nachricht am PC ist der Editor mit automatischen Zeilen-Wrap gut geeignet

1.2 Monoalphabetische Substitionsverfahren

Jeder Buchstabe wird bijektiv durch einen anderen Buchstaben des gleichen Alphabets ersetzt.

Alphabet:	$\mathscr{L} := \{A, B, ..., Z\}$
Schlüsselraum:	$\mathscr{K} := \{k : \mathscr{L} \to \mathscr{L} \mid k \text{ ist bijektiv}\}$
Verschlüsselung von $z \in \mathscr{L}$:	$k(z)$
Entschlüsselung:	$E(z) := k(z)$

1.2.1 Caesar-Verschlüsselung

Alle Buchstaben des Alphabets werden um einen konstanten Wert verschoben.
Beispiel:
Code: $n = 2$
Verschlüsselung: jeder Buchstabe wird durch den übernächsten Buchstaben im Alphabet ersetzt $(E(m) = (m + 3) \mod 26)$.
Entschlüsselung: jeder Buchstabe wird durch den vor-vorherigen Buchstaben ersetzt.

1.2.2 Häufigkeitsanalyse

Monoalphabetische Substitionsverfahren lassen sich mit moderner Technik sehr einfach durch die Verwendung von Häufigkeitsanalysen entschlüsseln. In jeder Sprache gibt es Buchstaben die deutlich häufiger vorkommen als andere. Durch einer Analyse der Häufigkeit der einzelnen Buchstaben im Geheimtext lassen sich diese den Ausgangsbuchstaben zuordnen.

1.3 Polyalphabetische Substitutionsverfahren

Damit sich ein Geheimtext nicht durch eine Häufigkeitsanalyse (1.2.2) entschlüsseln lässt wird bei polyalphabetischen Verschlüsselungsverfahren der Schlüssel regelmäßig gewechselt. Hierbei wird der Schlüsselwechsel meist selbst durch ein Schlüsselwort kodiert.

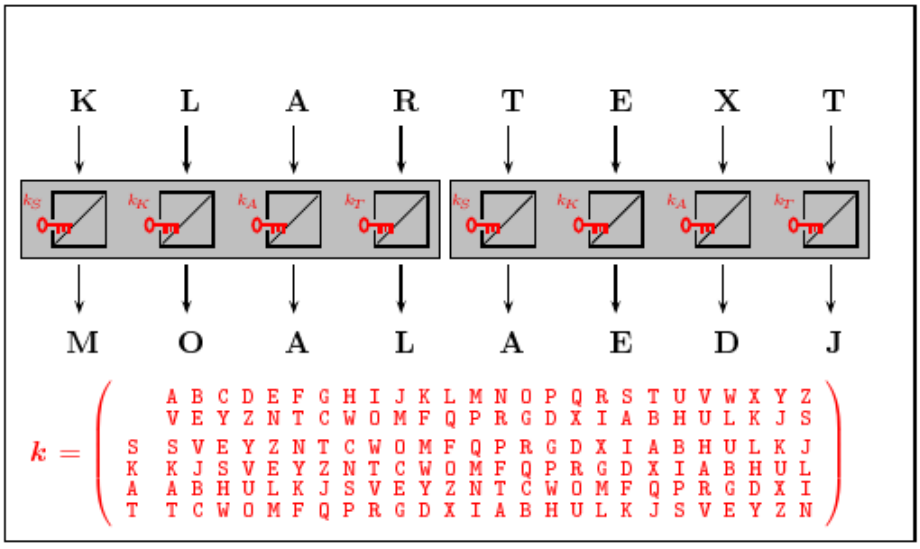
1.3.1 Vignère-Verfahren

Alphabet:	$\mathscr{L} := \{A, B, ..., Z\}$
Menge aller Wörter:	$\mathscr{W} = \mathscr{L}^{>0} := \bigcup_{n\in\mathbb{N}} \mathscr{L}^n = \{(m_1, m_2, ..., m_n) \vert m_i \in \mathscr{L}, n \in \mathbb{N}\}$
Schlüsselraum:	$\mathscr{K} := \Sigma_{\mathscr{L}} \times \mathscr{W}$

Ein Schlüssel $k = (f, w) \in \mathscr{K}$ besteht aus einer Permutation $f \in \Sigma_{\mathscr{L}}$ (siehe 1.2) und einem Schlüsselwort w.
Das Vignère-Verfahren ist ein Blockverschlüsselungsverfahren, bei dem jeweils ein Block von Zeichen (im Beispiel 4) nach dem gleichen Verfahren verschlüsselt wird. Wenn die Blockgröße klein genug oder die Textlänge groß genug sind lässt sich ein solches Verfahren ebenfalls durch eine Häufigkeitsanalyse (siehe 1.2.2) knacken.

1.3.1.1 Verschlüsselung

Der Schlüssel f wird zyklisch mithilfe des Schlüsselwortes w verschoben.
Beispiel:
f = VEYZNTCWOMFQPRGDXIABHULKJS und w = SKAT



1.3.2 One-Time-Pad

Bei dem One-Time-Pad handelt es sich um ein absolut sicheres Substitionsverfahren (Nachrichten, bei denen sich Informationen von der Länge der Nachricht ableiten lassen müssen auf eine konstante Länge gebracht werden).

1.3.2.1 Verschlüsselung

Annahme:	Klar- und Geheimtext sind eine Folge von Zeichen der Menge $\mathscr{L} = \mathbb{Z}$ $(n \in \mathbb{N})$
Klartextnachricht:	$m = (m_1, m_2, ...m_l) \in {\mathbb{Z}_n}^l$
One-Time-Pad (zufällig):	$k = (k_1, k_2, ...k_l) \in {\mathbb{Z}_n}^l$
Geheimtext:	$E(m) := m + k := (m_1 + k_1, m_2 + k_2, ..., m_l + k_l)$

1.3.2.2 Perfekte Sicherheit

Wenn die Zeichen k_i des Schlüssels mit einem perfekten Zufallsgenerator erzeugt lassen sich **keine** Rückschlüsse auf die Klartextnachricht ziehen (außer Länge).
Allerdings darf ein One-Time-Pad nur ein einziges Mal für die Verschlüsselung verwendet werden. Andernfalls könnte durch eine Berechnung der Differenz der mit dem gleichen OTP verschlüsselten Nachrichten Informationen gewonnen werden. So kann eine Häufigkeitsanalyse (siehe 1.2.2) oder sogar eine komplette Entschlüsselung (falls Klartext einer Nachricht bekannt ist) möglich werden.

1.4 algebraische Substitionsverfahren

1.4.1 Hill-Verfahren

Das Hill-Verfahren ver- und entschlüsselt die Nachrichten mithilfe einer **invertierbaren** Matrix.

1.4.1.1 Verschlüsselung

Annahme:	$n, k \in \mathbb{N}$ sind vorgegeben
Klartextnachricht:	Menge von Blöcken $\mathscr{B} = {\mathbb{Z}_n}^k$
Schlüssel:	**invertierbare** (k, k)-Matrix K
Geheimtext:	$E_K(b) := K \cdot b \qquad (b \in \mathscr{B})$

1.4.1.2 Entschlüsselung

Annahme:	gleiche Bedingungen wie bei der Verschlüsselung
Geheimtextnachricht:	Menge von Blöcken $\mathscr{C} = {\mathbb{Z}_n}^k$
Klartext:	$E_K(c) := K^{-1} \cdot c \qquad (c \in \mathscr{C})$

Kapitel 2

Modulare Arithmetik

2.1 Exkurs: Division mit Rest

Für $a, b \in \mathbb{Z}, b \neq 0$ gibt es eindeutig bestimmte Element $q, r \in \mathbb{Z}, 0 \leq r < |b|$:

$$\begin{aligned} a &= b \cdot q + r \\ a/_{\mathbb{Z}} b &:= q \\ a \mod b &:= r \end{aligned}$$

2.2 Der Ring $\mathbb{Z}_n$

Ein Ring $\mathbb{Z}_n$ ist definiert durch:

$$\mathbb{Z}_n := 0, 1, ..., n-1$$

2.2.1 Addition und Multiplikation

$$\begin{aligned} a +_{\mathbb{Z}_n} b &:= (a + b) \mod n \\ a \cdot_{\mathbb{Z}_n} b &:= (a \cdot b) \mod n \end{aligned} \tag{2.1}$$

2.2.1.1 Inverse bezüglich der Addition

jedes $a \in \mathbb{Z}$ hat ein Inverses:

$$-a := \begin{cases} 0 & \text{für } a = 0 \\ n - a & \text{sonst} \end{cases}$$

2.2.1.2 Inverse bezüglich der Multiplikation

ein Element $a \in \mathbb{Z}_n$ ist *(multiplikativ) invertierbar*, falls es ein Element $b \in \mathbb{Z}_n$ gibt, für das gilt:

$$a \cdot b = 1$$

man schreibt auch:

$$a^{-1} := b$$

Die Menge der invertierbaren Elemente in $\mathbb{Z}_n$ wird als $\mathbb{Z}_n^*$ bezeichnet:

$$\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n \mid a \cdot b = 1 \text{ für ein } b \in \mathbb{Z}n\}$$

Zudem gilt, dass ein Element nur dann invertierbar ist, falls $ggT(a, n) = 1$:

$$\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n \mid ggT(a, n) = 1\}$$

2.2.2 Subtraktion

Eine Subtraktion entspricht einer Addition mit der Inverse:

$$a -_{\mathbb{Z}_n} b := a +_{\mathbb{Z}_n} (-b) \mod n$$

2.2.3 Teiler, Vielfache

$b \in \mathbb{Z}$ teilt $a \in \mathbb{Z}$ falls ein $q \in \mathbb{Z}$ existiert mit:

$$a = b \cdot q$$

man schreibt auch $b|a$

2.2.3.1 Teilerregeln

1. $a|0 \ \forall a \in \mathbb{Z}$
2. $a|b \Leftrightarrow a|(-b)$
3. $a|b$ und $a|c \Rightarrow a|(b+c)$

2.2.4 Kongruenz

$a, b \in \mathbb{Z}$ sind *kongruent modulo* n, falls $n \in \mathbb{N}|(a-b)$. Man schreibt auch $a \equiv b$

2.2.5 Matrizen

2.2.5.1 Determinantenberechnung

Die Determinante $\det(A)$ der (N,N)-Matrix $A = (a_{ij})_{1 \leq i,j \leq N}$ (mit ganzzahligen Einträgen) über $\mathbb{Z}_n$ wird definiert durch:

$$\det(A) \mod n = det((a_{i,j} \mod n)_{1 \leq i,j \leq N})$$

Zudem gilt für die Matrizen $A = (a_{ij})_{1 \leq i,j \leq N}$ und $B = (b_{ij})_{1 \leq i,j \leq N}$ (mit ganzzahligen Einträgen):

$$\begin{aligned} \det(A \cdot B) \mod n &= (det(A) \cdot \det(B)) \mod n \\ &= ((\det(A) \mod n) \cdot (\det(B) \mod n)) \mod n \\ &= (\det(A) \mod n) \cdot_{\mathbb{Z}_n} (\det(B) \mod n) \end{aligned}$$

2.2.5.2 Inverse Matrix

Die Inverse einer quadratischen Matrix A über $\mathbb{Z}_n$ lässt sich mithilfe der Adjunkten berechnen:

$$A^{-1} = (\det(A))^{-1} \cdot adj(A)$$

Die Adjunkte lässt sich über $\mathbb{Z}_n$ berechnen, da lediglich Summen und Differenzen von Produkten berechnet werden müssen.

2.3 Der erweiterte Euklid'sche Algorithmus

Der Euklid'sche Algorithmus ist ein sehr effizienter Weg den ggT zweier Zahlen zu ermitteln. Der Euklid'sche Algorithmus lässt sich auch über $\mathbb{Z}_n$ verwenden. Man spricht dann von dem erweiterten Euklid'schen Algorithmus.

2.3.1 Euklid'scher Algorithmus

gegeben: $a_0, b_0 \in \mathbb{Z}$

1. $a := a_0$ und $b := b_0$
2. falls $b = 0$ gebe $|a|$ aus und beende
3. $r := a \mod b$
4. $a := b$
5. $b := r$
6. goto 2.

2.3.2 erweiterter Euklid'scher Algorithmus

gegeben: $a_0, b_0 \in \mathbb{N}_0$
gesucht: $\alpha \cdot a_0 + \beta \cdot b_0 = g = ggT(a_0, b_0)$

1. $a := a_0$, $\alpha_a = 1$, $\beta_b = 0$, $b := b_0$, $\alpha_b := 0$, $\beta_b := 1$
2. falls $b = 0$ gebe $g := a$, $\alpha := \alpha_a$ und $\beta := \beta_a$ aus
3. $q := a/_{\mathbb{Z}} b$
4. $r := a - q \cdot b$,$\alpha_r := \alpha_a - q \cdot \alpha_b$, $\beta_r := \beta_a - q \cdot \beta_b$
5. $a := b$, $\alpha_a := \alpha_b$, $\beta_a := \beta_b$
6. $b := r$, $\alpha_b := \alpha_r$ $\beta_b := \beta_r$
7. goto 2.

2.3.2.1 Beispiel

Eingabe:

$$a_0 = 1224 \text{ und } b_0 = 156$$

Berechnung:

1224	156	a,b	q
1	0	1224	
0	1	156	7
1	-7	132	1
-1	8	24	5
6	-47	12	2
		0	

Ergebnis:

$$6 \cdot 1224 + (-47) \cdot 156 = 12$$

2.4 Euler'sche φ-Funktion

Die Euler'sche φ-Funktion bezeichnet die Anzahl invertierbarer Elemente in $\mathbb{Z}_n$

$$\varphi(n) := \begin{cases} |\mathbb{Z}_n^*| & \text{für } n \in \mathbb{N}, n \geq 2 \\ 1 & \text{für } n = 1 \end{cases}$$

Für $a, b \in \mathbb{N}$ mit $ggT(a, b) = 1$ gilt:

$$\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$$

Zudem gilt für ein $n \in \mathbb{N}$ dessen Primzahlzerlegung $n = p_1^{e_1} \cdots p_r^{e_r}$:

$$\varphi(n) = n \cdot \prod_{i=1}^{r} (1 - \frac{a}{p_i})$$

2.4.1 φ-Funktion und Primzahlen

für eine Primzahl p gilt:

$$\varphi(p) = p - 1$$

Für Primzahlpotenzen gilt zudem:

$$\varphi(p^e) = p^{e-1}(p - 1)$$

Kapitel 3

IT-Sicherheit: Gefährdungen und Maßnahmen

Im Folgenden wird auf mehrere Schutzziele eingegangen, welche für die IT-Sicherheit wichtig sind.

3.1 Vertraulichkeit

Bei dem Kriterium der Vertraulichkeit geht es darum, dass Daten nicht an unbefugte gelangt

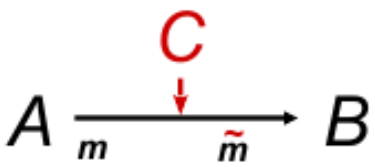
3.1.1 Schutzmaßnahmen: Verschlüsselungsverfahren

Durch die Verschlüsselung der Daten kann eine Gefährdungen der Vertraulichkeit verhindert werden. Allerdings muss hierbei auf folgende Punkte geachtet werden:

- Schlüsselerzeugung:
  Schlüssel müssen mit einem kryptographisch sicheren Zufallsgenerator erzeugt werden
- Schlüsselspeicherung:
  Schlüssel müssen sicher gespeichert sein
- Schlüsselaustausch:
  Damit zwei Systeme informationen austauschen können müssen zunächst Schlüssel ausgetauscht werden. Bei synchronen Verschlüsselungsverfahren muss dieser Austausch auf einem sicheren Weg geschehen.

3.2 Integrität

Bei dem Kriterium der Integrität geht es darum, dass die Daten, die verschickt werden auch unverändert empfangen werden.

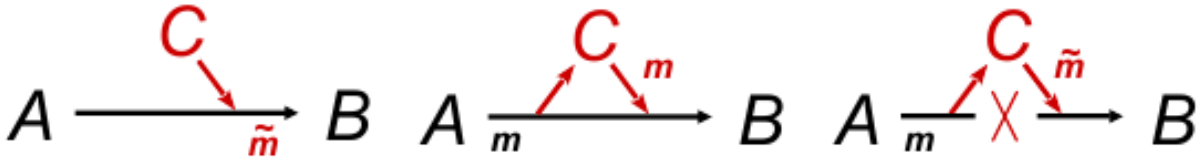


3.2.1 Schutzmaßnahme: Hashfunktionen, Whitelists

zur Sicherung der Integrität werden mithilfe von Hashfunktionen Prüfsummen errechnet und in einer Whitelist abgespeichert.

3.3 Authenzität der Daten

Bei dem Kriterium der Authenzität geht es darum, sicherzustellen, dass die empfangenen Daten auch tatsächlich vom angegebenen Absender stammen.



3.3.1 Schutzmaßnahme: Signaturen

Um die Authenzität sicherzustellen gibt es mehrer Möglichkeiten:

1. es wird ein zweiter Kommunikationsweg für die Authentifikation verwendet (2-Factor-Authentification)
2. **Signaturverfahren:**
   Eine Signatur wird (mit dem Signaturverfahren S) berechnet und mithilfe eines privaten Schlüssels k_{pri} verschlüsselt. Der Empfänger nutzt den öffentlichen Schlüssel k_{pub} und das zu S gehörigen Verifikationsverfahren V um die Nachricht zu authentifizieren.
3. **MAC-Verfahren:**
   Sender und Empfänger einigen sich auf einen geheimen Schlüssel k. Anschließend nutzt der Sender diesen Schlüssel um den MAC(Message Authentication Code)-Wert der Nachricht zu verschlüsseln. Wenn der Empfänger den MAC-Wert entschlüsselt kann er die Nachricht authentifizieren.

3.3.2 Schutz vor Replay-Angriffen

Um einen Replay-Angriff zu verhindern muss dafür gesorgt werden, dass jede Nachricht nur ein einziges Mal akzeptiert wird. Mögliche Verfahren hierfür sind Zählwerte, die mit jeder Nachricht inkrementiert werden oder Zeitstempel.

3.4 Authenzität von Nutzern

In vielen Fällen ist es nötig einen Nutzer zu authentifizieren (z.B. Anmeldung auf einer Webseite). Diese Authentifikation ist eine spezielle Form der Nachrichtenauthentifikation, bei der der Inhalt der übertragenen Daten nicht relevant ist.

3.4.1 Schutzmaßnahmen

- Nutzen eines gemeinsamen Geheimnisses (z.B. WLAN-Passwort)
- Nutzen eines Schlüssels, den nur eine Seite besitzt (siehe 3.3.1)
- Nutzen von einmaligen Eigenschaften (z.B. Fingerabdruck)

3.5 Zugriffskontrolle

Bei Systemen, die eine Aktion ausführen ist es wichtig abzusichern, dass nur erlaubte Aktionen angefragt werden können.

3.5.1 Schutzmaßname: Zugriffskontrollsystem

Es werden Listen (Access Control Lists) darüber geführt, welcher Nutzer welche Aktionen veranlassen darf. Hierbei werden die Rechte häufig in Form von Rollen vergeben (Role Based Access Control).

3.6 Nichtabstreitbarkeit, Verbindlichkeit

Eine Form der Authentifikation oder Authentifizierung, die auch gegenüber dritten unwiderlegbar ist. Dies ist vor allem für Kommunikationen wichtig, bei denen es für eine Partei vorteilhaft wäre sie abzustreiten (z.B. Verträge).

3.6.1 Schutzmaßname: Signaturen und PKI

Signaturen (siehe 3.3.1) können auch als Beweis für die Nichtabstreitbarkeit verwendet werden, falls der öffentliche Schlüssel der dritten Partei bekannt ist. Hierfür wird eine öffentliche Infrastruktur (Public Key Infrastructur), welche von Zertifizierungsstellen zur Verfügung gestellt wird (z.B. ITU (für X.509)).

3.7 Verfügbarkeit

Bei dem Kriterium der Verfügbarkeit geht es darum, dass die Daten und IT-Systeme wie angedacht erreichbar sind. Mögliche Bedrohungsszenarien hierfür sind:

- Datenverlust durch defekte Daten
- Datenverlust durch Schadsoftware
- Nichterreichbarkeit von Diensten aufgrund von Netzwerkproblemen
- Nichterreichbarkeit von Webdiensten aufgrund erfolgreicher Denial-of-Service-Angriffen

3.7.1 Schutzmaßnahmen

- redundante örtlich verteilte Datenspeicherung
- Virenscanner und Paketfilter zum Schutz vor Malware
- Firewalls

3.8 Anonymität

Es gibt viele Gründe, wegen denen es sinnvoll ist, dass eine Datenübertragung sicher aber ohne den Versand persönlicher Daten funktioniert.

Kapitel 4

Verschlüsselungsverfahren

4.1 Das Kerckhoffs'sche Prinzip

„Ein Verschlüsselungssystem darf nicht der Geheimhaltung bedürfen und soll ohne Schaden in Feindeshand fallen können."

Folglich ist für die Entschlüsselung der Nachricht nicht die Kenntnis über das Verfahren, sondern der Schlüssel die relevante Information.

„Es soll nicht möglich sein, einen Geheimtext ohne Kenntnis des hierfür vorgesehenen Schlüssels **effizient** zu entschlüsseln."

4.2 Mathematische Modelierung von Verschlüsselungsverfahren

Klartextnachrichten und Geheimtextnachrichten sind Elemente einer Menge $\mathscr{M}$.
Schlüssel sind Elemente eines Menge $\mathscr{K}$, die als Schlüsselraum bezeichnet wird.
Die Verschlüsselungsverfahren bestehen aus einem Paar von Funktionen zur Verschlüsselung (E) und Entschlüsselung (D):

$$E : \mathscr{K} \times \mathscr{M} \to \mathscr{M}$$

$$D : \mathscr{K} \times \mathscr{M} \to \mathscr{M}$$

Hierbei sind die definierten Abbildungen bijektiv ($D_k := E_k^{-1}$):

$$E_k : \mathscr{M} \to \mathscr{M}, E_k(m) := E(k, m)$$

$$D_k : \mathscr{M} \to \mathscr{M}, D_k(m) := D(k, m)$$

Die Nachrichtenmenge $\mathscr{M}$ besteht in Realität aus einer endlichen Folgen (Tupeln) von Bit- oder Bytewerten. Diese können entweder die gleiche Länge besitzen (Blockverschlüsselung) oder von beliebiger Länge sein (Stromverschlüsselung).

$$\mathscr{M} = \mathscr{L}^n = \{(z_1, z_2, ..., z_n) \mid z_i \in \mathscr{L}\} \text{ für ein festes } n \in \mathbb{N} \text{ (Blockverschlüsselung)}$$

$$\mathscr{M} = \mathscr{L}^{>0} = \bigcup_{n \in \mathbb{N}} \mathscr{L}^n = \{(z_1, z_2, ..., z_n) \mid z_i \in \mathscr{L}, n \in \mathbb{N}\} \text{ (Stromverschlüsselung)}$$

4.3 Schlüsselaustausch

Um mithilfe eines symmetrischen Schlüssels Daten austauschen zu können muss der Schlüssel auf eine Sichere Art und Weise ausgetauscht werden. Dies ist zwar offline möglich, stellt allerdings kein praktikables Verfahren für die Kommunikation im Internet dar. Daher werden für einen sicheren Schlüsselaustausch über eine unsichere Infrastruktur asymmetrische Verschlüsselungsverfahren benötigt.

4.4 Angriffsszenarien

Da für eine effiziente Entschlüsselung einer Geheimtextnachrichten die Kenntnis des Schlüssels essentiell ist versuchen die meisten Angriffe diesen herauszufinden.

4.4.1 Ciphertext-only Angriffe

Vorraussetzung: ein oder mehrere Geheimtexte $c_i = E_k(m_i)$ bekannt
Angriffsziel: Bestimmung von m oder von k

4.4.2 Known-plaintext Angriffe

Vorraussetzung: Geheimtext $c = E_k(m)$ und eine Reihe von bekannten Paaren $(m_i, E_k(m_i))$ sind bekannt
Angriffsziel: Bestimmung von m oder von k

4.4.3 Chosen-plaintext Angriffe

Vorraussetzung: Geheimtext $c = E_k(m)$ und für eine Reihe von beliebig vorgegebenen Klartextnachrichten m_i kan
Angriffsziel: Bestimmung von m oder von k

4.5 Brute-Force Angriffe

Da die Vorraussetzung für ein gutes Verschlüsselungsverfahren ist, dass es nicht **effizient** entschlüsselt werden kann (siehe 4.1) muss die **Effizienz** definiert sein. An dieser Stelle setzen Brute-Force Angriffe an. Sie versuchen wie der Name schon sagt mit roher Rechenleistung den Schlüssel zu ermitteln.

4.5.1 Beispiel: Brute-Force Angriff auf k

Unter der Annahme, dass ein known-plaintext Angriff (siehe 4.4.2) vorliegt lässt sich der Schlüssel bestimmen, indem alle möglichen Schlüssel k des Schlüsselraums $\mathscr{K}$ „ausprobiert“ werden. Hierbei können für die einzelnen bekannten Klartextnachrichten mehrere Schlüssel passen:

$$\mathscr{K}_i := \{\tilde{k} \in \mathscr{K} \mid E_{\tilde{k}}(m_i) = c_i\}$$

Bei einer ausreichenden Menge bekannter Klartextnachrichten lässt sich der Schlüssel aus der Schnittmenge der möglichen Schlüssel bestimmen:

$$\bigcap_{i=1}^{N} \mathscr{K}_i = \{k\}$$

4.5.2 Beispiel: Brute-Force Angriff auf m

Liegen die Vorraussetzungen für einen chosen-plaintext Angriff vor, so kann die Klartextnachrichten herausgefunden werden, indem alle möglichen Klartextnachrichten $\tilde{m} \in \mathscr{M}$ „ausprobiert“ werden:

$$E_k(\tilde{m}) = c = E_k(m) \Longrightarrow \tilde{m} = m$$

4.5.3 Anforderungen zum Schutz vor Brute-Force

1. Es soll keinen Angriff auf den Schlüssel k geben, der durchschnittlich weniger als $\frac{|\mathscr{K}|}{2}$ Ver- oder Entschlüsselungsoperationen braucht.
2. Es soll keinen Angriff auf die Klartextnachricht m geben, der durchschnittlich weniger als $\min\left\{\frac{|\mathscr{K}|}{2}, \frac{|\mathscr{M}|}{2}\right\}$ Ver- oder Entschlüsselungsoperationen braucht.

4.6 Wörterbuchangriffe

Nachrichtenpaare (Geheim- und Klarttext), die mithilfe eines Schlüssels k verschlüsselt wurden können in einem Wörterbuch abgespeichert werden. Mithilfe des Wörterbuchs können diese Paare jederzeit entschlüsselt werden. Zudem kann für einen chosen-plaintext Angriff (siehe 4.4.3) ein Wörterbuch mit häufig vorkommenden Wörtern erstellt werden, sodass eine Entschlüsselung in wesentlich kürzerer Zeit möglich wird.

4.6.1 Schutz vor Wörterbuchangriffen

Um einem Wörterbuchangriff vorzubeugen ist es wichtig, dass sich der Geheimtext bei jeder Verschlüsselung des gleichen Klartextes unterscheidet. Dies wird meist dadurch erreicht, dass die Klartextnachricht vor Anwendung des Verschlüsselungsverfahrens abgeändert wird. Meist wird hierfür eine Nonce (Number used Once) verwendet. Die Nonce kann ein Zähler, ein Zeitstempel oder eine Zufallszahl sein. Der Nonce-Wert muss beiden Seiten bekannt sein. Hierbei kann er entweder mitverschlüsselt übertragen werden oder besteht aus einem Wert (z.B. Zähler), der beiden Seiten bekannt ist.

4.6.1.1 Nonce-Verschlüsselung

Ein Nonce-Wert $v \in \mathscr{M}$ wird dazu genutzt die Klartextnachricht $m \in \mathscr{M}$ abzuändern:

$$\tilde{m} = m + v$$

Anschließend wird die veränderte Nachricht verschlüsselt:

$$c = E_k(\tilde{m})$$

Für die Entschlüsselung wird der Schlüssel k und der Nonce-Wert v benötigt:

$$m = D_k(c) + v$$