Kryptographie

Vorlesung im WS 2006/7

http://www.math.tu-berlin.de/∼hess/krypto-ws2006

Florian Heß / Marcus Wagner
Technische Universität Berlin

Zielsetzungen in der Kryptographie

Sichere digitale Kommunikation in Anwesenheit von vertrauensunwürdigen Personen.

Hauptaufgaben:

- Verschlüsselung
- Digitale Signaturen

Spezieller:

- Authentizität, Authentifizierung, Datenintegrität
- Anonymität, Autorisierung, elektronisches Geld
- elektronische Wahlen
- ...

Terminologie

Kryptographie:

- Entwurf und Design von Kryptosystemen, also zum Beispiel Verfahren für die Verschlüsselung und digitale Signaturen.

Kryptoanalyse:

- Versucht, Kryptosysteme zu “knacken”.

Kryptologie:

- Zusammenfassender Begriff für Kryptographie und Kryptoanalyse.
- Wir verwenden “Kryptographie” synonym für “Kryptologie”.

Geschichte der Kryptographie

Alte Disziplin:

- Cäsar, Alexander der Große, ...
- Unsystematische Verwendung bis 20. Jahrhundert.

20. Jahrhundert:

- Systematisierung und Formalisierung.
- Wichtige Rolle im zweiten Weltkrieg (Enigma, A. Turing).
- Bis in die 60er Jahr hauptsächlich Verwendung durch Regierungen und Militär.
- Mit zunehmender Computerisierung verstärkte Verwendung und Anforderungen im wirtschaftlichen (privaten) Bereich.

Geschichte der Kryptographie

20. Jahrhundert (ctd):

- 1976: “New directions in cryptography” von Diffie und Hellman, Geburtsstunde der Kryptographie mit öffentlichem Schlüssel.
- 80er Jahre: Zero-knowledge Protokolle.
- 90er Jahre: Massenhafte Verbreitung kryptographischer Verfahren durch Computer, Internet, Geldautomaten, Mobilfunk, etc.

Kryptographie und andere Wissenschaften

Die Kryptographie ist ausgesprochen interdisziplinär:

- Theoretische und angewandte Informatik (Komplexitätstheorie, Informationstheorie, Protokolle, formale Methoden, Software Engineering, ...)
- Mathematik (Algebra, Zahlentheorie, ...)
- Elektrotechnik (Chipdesign, ...)
- Physik (Quantenphysik und -computer, ...)
- Gesellschaftspolitische und rechtliche Aspekte (DMCA, ...)

Informationssicherheit („Softwaresicherheit“) fällt nicht unter Kryptographie.

Algorithmen

Unter einem Algorithmus verstehen wir einen Text von elementaren Anweisungen, die beispielsweise auf einem Computer oder einer Turing Maschine ausgeführt werden können.

Ein Algorithmus erwartet eine Eingabe, tätigt eine Ausgabe und endet dann.

Ein Algorithmus heißt deterministisch, wenn die auszuführenden Anweisungen eindeutig durch die Eingabe bestimmt sind. Ein solcher Algorithmus verhält sich wie eine Funktion.

Ein Algorithmus heißt probabilistisch, wenn die auszuführenden Anweisungen neben der Eingabe auch von zufälligen Entscheidungen während des Laufs abhängen.

Algorithmen

Zu vorgegebener Eingabe ist die Laufzeit eines Algorithmus die Anzahl der bis zum Ende des Algorithmus ausgeführten Anweisungen, inklusive der Zufallsabfragen. Im Zusammenhang mit Computern werden häufig Bitoperationen gezählt.

Die Laufzeit und die Ausgabe eines probabilistischen Algorithmus hängen damit ebenfalls vom Zufall ab und stellen Zufallvariablen dar.

Ist A ein probabilistischer Algorithmus, so können wir daraus einen deterministischen Algorithmus A_D machen, indem wir die von A benutzte Zufallsquelle als Eingabe von A_D auffassen.

Ein Algorithmus ist polynomiell, wenn es ein Polynom $f \in \mathbb{Z}[t]$ gibt, so daß seine Laufzeit für die Eingabe x durch $f(|x|)$ beschränkt ist.

Algorithmen

Beispiel:
Aufgabe ist, eine 6 zu würfeln. Lösung ist, einfach wiederholt zu würfeln, bis eine 6 erscheint.

In vielen Fällen sind probabilistische Algorithmen deterministischen Algorithmen zur Lösung der gleichen Probleme überlegen (einfacher und schneller).

Alphabete und Worte

Ein Alphabet A ist eine nicht-leere Menge.
Die Wortmenge A^* über A ist $A^* = \cup_{i=0}^{\infty} A^i$.

Beispiel:

- $A = \{\mathsf{A}, \ldots, \mathsf{Z}\}$, HALLO $\in A^*$
- $A = \{0, 1\}$, $101101 \in A^*$.
- $1^k = 11 \cdots 1$, k Einsen.

Leeres Wort: ε.
Aneinanderhängen von Worten: $m_1, m_2 \in A^* \rightarrow m_1 m_2 \in A^*$.
Länge $|m|$ eines Worts $m \in A^*$: $m \in A^{|m|}$.

Schreibweise in einigen Programmiersprachen: ’A’, “HALLO“, ““

Konzepte der Verschlüsselung

Klartexte sind aus Klartextraum: $m \in M \subseteq A_1^*$.
Chiffretexte sind aus Chiffretextraum: $c \in C \subseteq A_2^*$.
Schlüssel sind aus Schlüsselräumen: $e \in K_1 \subseteq A_3^*$, $d \in K_2 \subseteq A_4^*$.

Verschlüsselungsverfahren $\mathcal{E} : K_1 \times M \rightsquigarrow C$.
Entschlüsselungsverfahren $\mathcal{D} : K_2 \times C \rightarrow M$.

Verschlüsselung von Klartext m mit Schlüssel e: $c \leftarrow \mathcal{E}(e, m)$.
Entschlüsselung von Chiffretext c mit Schlüssel d: $m \leftarrow \mathcal{D}(d, c)$.

$\mathcal{E}$ und $\mathcal{D}$ sind effiziente Verfahren (z.B. Programme).
$\mathcal{E}$ und $\mathcal{D}$ dürfen probabilistisch sein (dürfen den Zufall verwenden).
$\mathcal{E}$ kann mehrdeutig sein ($\rightsquigarrow$).

Beispiele: Affine-lineare Blockchiffren

$R = \mathbb{Z}/m\mathbb{Z}$ und $M = C = R^n$, $K_1 = K_2 = \mathrm{GL}(n, R) \times R^n$.

Für $A \in \mathrm{GL}(n, R)$ und $b \in R^n$ seien

- $C = \mathcal{E}((A, b), M) = AM + b$,
- $M = \mathcal{D}((A, b), M) = A^{-1}(C - b)$.

Chiffre von Cäsar:

- $m = 26$, $n = 1$, $A = (1)$, $b = 3$.

Chiffre von Vignère:

- $m = 26$, $A = I_n$, $b \in R^n$.
- Nicht jeder Buchstabe wird gleich verschlüsselt.

Chiffre von Hill:

- $m = 26$, A Permutationsmatrix, $b = 0$

Beispiele: Affine-lineare Blockchiffren

Kryptoanalyse:

- Angriff durch Untersuchung der Häufigkeit des Auftretens von Buchstaben in natürlicher Sprache.
- Angriff falls $n+1$ Klartext- und Chiffretextpaare (M_i, C_i) vorliegen.

Dann $C_i - C_0 = A(M_i - M_0)$. Ist U Matrix mit den Spalten $C_i - C_0$ und V Matrix mit den Spalten $M_i - M_0$, so gilt $U = AV$. Ist V invertierbar, folgt $A = UV^{-1}$ und $b = C_0 - AM_0$.

Beispiele: Affine-lineare Blockchiffren

Vernam's One Time Pad:

- Im Chiffre von Vignère den Vektor b gleichverteilt zufällig aus R^n wählen und nur einmal verwenden.
- Dann C ebenfalls völlig zufällig, keine Verbindung zu Klartext. Daher unmöglich, aus C Informationen über M zu bekommen.
- Chiffre aber inpraktikabel (wie b zu Kommunikationspartner bringen?)

Symmetrisch, asymmetrisch

Symmetrisches Verschlüsselungssystem (secret key):

- $d = e$ (oder d leicht aus e berechenbar).
- Sender und Empfänger haben e als gemeinsames Geheimnis.

Asymmetrisches Verschlüsselungssystem (public key):

- d geheim, e öffentlich.
- d schwer oder gar nicht aus e berechenbar.
- Empfänger besitzt e und d.
- Sender verwendet e zum Verschlüsseln.
- Empfänger verwendet d zum Entschlüsseln.

Prinzip von Kerkhoff

Grundlegende Anforderung:
Sicherheit eines Systems sollte sich nicht aus der Geheimhaltung der Algorithmen, sondern nur aus den geheimen Schlüsseln ableiten.

Gründe dafür:

- Geheime Schlüssel lassen sich schneller und häufiger austauschen, größere Flexibilität.
- Geheimhaltung funktionert nicht lange (siehe COMP128 im GSM Mobilfunk, RC4 von RSA), Umstellung auf neue Algorithmen teuer.
- Öffentliche Algorithmen können von unabhängiger Seite untersucht werden. Dies eliminiert Designfehler und baut Vertrauen bei Benutzern bzw. Kunden auf (siehe GSMK Cryptophone).

Prinzip von Kerkhoff

Gründe dagegen:

- Firmen wollen Geschäftsgeheimnisse bewahren.
- Staatliche Behörden wollen einen verbreiteten Gebrauch (z.B. durch Terroristen) des für eigene Zwecke entwickelten und benutzten, vielleicht sehr sicheren Kryptosystems verhindern.

Im allgemeinen wird die Meinung vertreten, daß das Prinzip von Kerkhoff befolgt werden sollte.

Angriffe und Sicherheitsmodelle

Es werden Angreifer (Algorithmen, Programme, Menschen etc.) betrachtet.

Im wesentlichen drei Aspekte:

- Was ist das Ziel eines Angriffs (wann erfolgreich)?
- Welche Informationen stehen dem Angreifer zur Verfügung?
- Welche Rechenleistung besitzt der Angreifer?

Trifft man für diese Aspekte eine Wahl, so legt man sich auf ein Sicherheitsmodell fest.

Gibt es unter den Aspekten keinen Angreifer, so ist das Kryptosystem sicher (im gewählten Modell).

Maximale Sicherheit, wenn Ziele minimal und Informationen maximal.

Angriffe und Sicherheitsmodelle

Ziele eines Angreifers:

- Chiffretext entschlüsseln.
- Chiffretexte bekannten Nachrichten zuordnen.
- Gegebenen Chiffretext in neuen Chiffretext umwandeln, so daß neuer Klartext mit alten Klartext sinnvoll verbunden ist.
- Irgendeinen Chiffretext erzeugen (ohne Klartext zu kennen).

Die korrespondierenden (negierten) Eigenschaften des Verschlüsselungssystems:

- Einweg-Eigenschaft (onewayness, OW).
- Semantische Sicherheit, Nichtunterscheidbarkeit (indistinguishability, IND).
- Nicht-Modifizierbarkeit (non-malleability, NM).
- Plaintext-awareness (PA).

Angriffe und Sicherheitsmodelle

Informationen (Fähigkeiten) des Angreifers in aufsteigender Reihenfolge:

- Ciphertext-only Angriff: Der Angreifer erhält nur Chiffretexte.
- Known-plaintext Angriff: Der Angreifer erhält Klartexte und die zugehörigen Chiffretexte.
- Chosen-plaintext Angriff (CPA): Der Angreifer kann sich die Klartexte aussuchen und erhält die zugehörigen Chiffretexte.

Zusätzlich bei Public-Key Verschlüsselungssystemen:

- Chosen-ciphertext Angriff (CCA1): Der Angreifer kann sich Chiffretexte aussuchen und erhält die zugehörigen Klartexte.

Angriffe und Sicherheitsmodelle

Chosen-plaintext und Chosen-ciphertext Angriffe gibt es auch in adaptiver und kombinierter Form:

Der Angreifer darf Klar- und Chiffretexte in Abhängigkeit zuvor erhaltener Chiffre- bzw. Klartexte und den Ergebnissen von Zwischenrechnungen wählen.

CCA1 wird dann zu CCA2.

Angriffe und Sicherheitsmodelle

Rechenleistung eines Angreifers:

- Vergleichbar der von $\mathcal{E}$ und $\mathcal{D}$ (polynomiell).
- Unbeschränkt.

Die korrespondierenden (negierten) Eigenschaften des Verschlüsselungssystems:

- Komplexitätstheoretische Sicherheit (computational security).
- Perfekte Sicherheit (unconditional security, perfect secrecy).

Im allgemeinen betrachtet man nur komplexitätstheoretische Sicherheit.

Angriffe und Sicherheitsmodelle

Beispiel:

Das für Public-Key Systeme stärkste Sicherheitsmodell ist

Nichtunterscheidbarkeit unter einem adaptiven Chosen-ciphertext Angriff (IND-CCA2).

Hier versucht ein Angreifer die Chiffretexte zweier von ihm vorgegebener Nachrichten den Nachrichten zuzuordnen.

Gelingt dies mit Wahrscheinlichkeit signifikant besser als 1/2, so gilt der Angriff als erfolgreich.

NM ist sicherer als IND ist sicherer als OW.

CCA2 ist sicherer als CCA1 ist sicherer als CPA.

NM-CCA2 = IND-CCA2

(In spezieller Situation: PA ist sicherer als IND-CCA2, NM-CCA2)