max planck institut
informatik

Kryptographie
Wie funktioniert Electronic Banking?

Kurt Mehlhorn

Adrian Neumann

Max-Planck-Institut für Informatik

Übersicht

- Zwecke der Kryptographie
- Techniken
  - Symmetrische Verschlüsselung(One-time Pad, Caesar, moderne Blockchiffres)
  - Asymmetrische Verschlüsselung, Public-Key Kryptographie (1978)
  - Digitale Unterschriften
- Anwendungen: Electronic Banking, Sicherheitsinfrastrukturen

Kryptographie (geheim-schreiben) Hauptziele (nach Wolfgang Ertel)

Vertraulichkeit / Zugriffsschutz: Nur dazu berechtigte Personen sollen in der Lage sein, die Daten oder die Nachricht zu lesen (auch teilweise).

Integrität / Änderungsschutz: Die Daten müssen nachweislich vollständig und unverändert sein.

Authentizität, Verbindlichkeit / Fälschungsschutz, Nichtabstreitbarkeit: Der Urheber der Daten oder der Absender der Nachricht soll eindeutig identifizierbar sein, und seine Urheberschaft sollte nachprüfbar und nicht abstreitbar sein.

Ver- und Entschlüsselung

- Σ Alphabet, a … z, A … Z 0 .. 9 ! ? …
- m, Nachricht, Klartext, message, Länge n
- Für jeden Nutzer X gibt es ein Paar (E_X, D_X) von Funktionen von Σ^n (Worte der Länge n) nach Σ^n mit
  - $D\big(E(x)\big) = x$ für alle x
  - E_X und D_X sind durch Programme realisiert. Sei m *eine beliebig e Nachricht und* $c = E(m)$. Ohne Kenntnis des Programms für D_X kann niemand aus c das m bestimmen. (zumindest nicht in fünf Minuten, einem Jahr, 100 Jahren)
  - E_X ist One-Way Function
  - E = encode D = decode

Ver- und Entschlüsselung

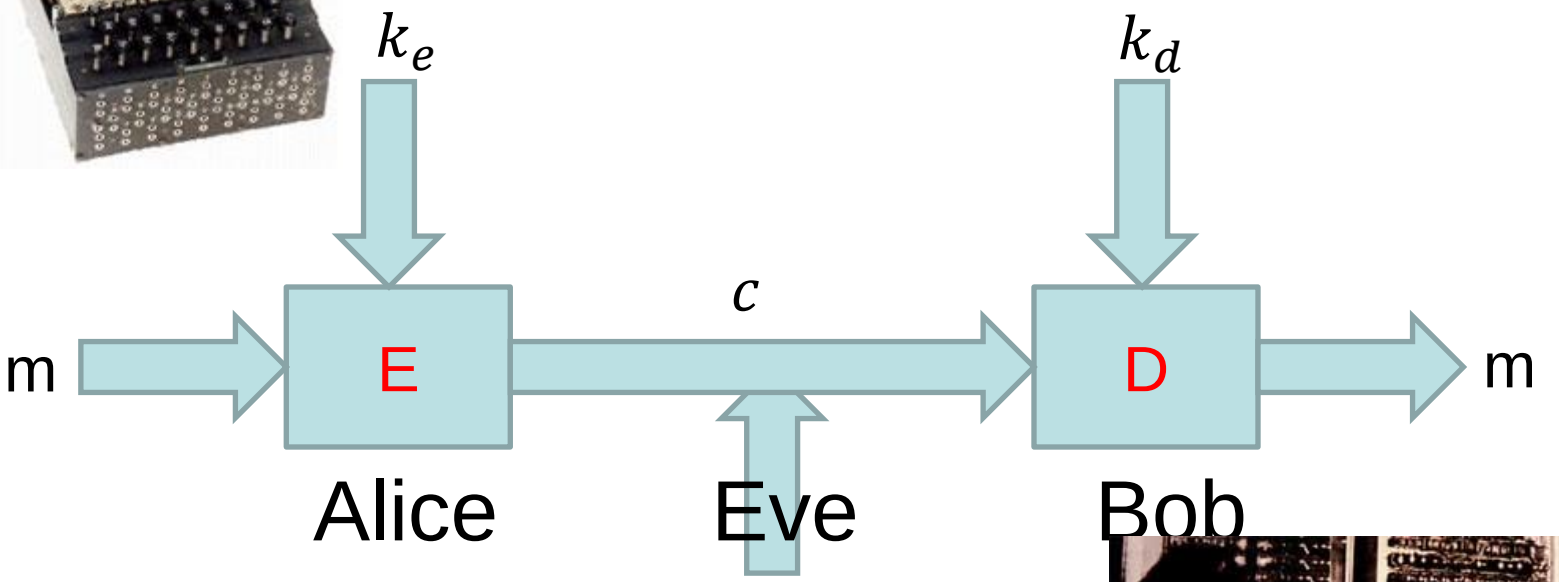
Für jeden Nutzer X gibt es ein Paar (E_X, D_X) von Funktionen von Σ^n nach Σ^n mit

- $D\big(E(x)\big) = x$ für alle x
- Die Programme für E_X und D_X erhält man aus generischen Programmen E und D durch Wahl eines Schlüsselpaars (k_e, k_d).
- Jeder Nutzer hat sein eigenes Paar
- Dann $E_X(x) = E(k_e, x)$ und analog für D.

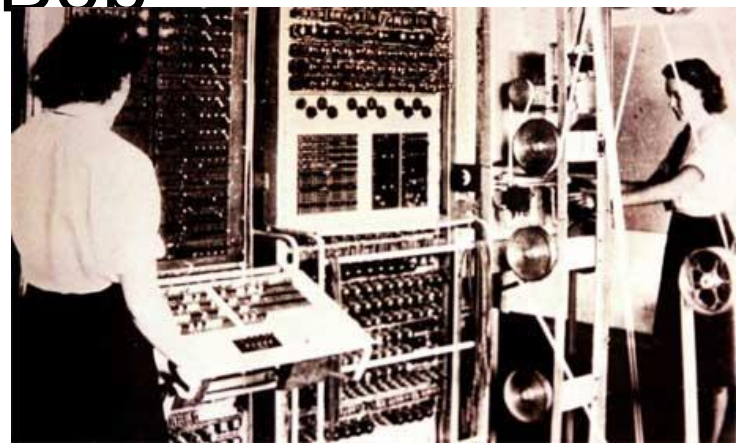
- E ist injektiv und damit bijektiv, D ist surjektiv und damit bijektiv
- Oft auch noch $D\big(E(x)\big) = x$ für alle x nützlich für digitale Unterschriften

Ver- und Entschlüsselung

$$m = D(k_d, E(k_e, m))$$



Eve = Eavesdropper



Symmetrische Verfahren

- Sender (Alice) und Empfänger (Bob) benutzen den **gleichen** Schlüssel
- Dieser Schlüssel muss geheim bleiben
- Wie einigt man sich auf einen Schlüssel?
  - Früher: physisches Treffen zum Schlüsselaustausch oder Bote
  - Heute: asymmetrisches Verfahren zum Schlüsselaustausch
- Beispiele: One-Time Pad, Caesar, …

Symmetrische Krypthographie
Eine Analogie

- Alice und Bob kaufen sich eine Kiste und ein Vorhängeschloss mit zwei identischen Schlüsseln. Jeder bekommt einen Schlüssel.
- Nachrichten kommen in die Kiste, die Kiste wird verschlossen, …
- Braucht ein Treffen oder einen vertrauenswürdigen Boten

One-Time Pad (Rotes Telefon)

- m ist Folge von Großbuchstabe + ZR
- k ist zufällige Folge über diesem Alphabet
- m_i, k_i , jeweilige i-te Buchstaben $\{0, \ldots, 26\}$
- $c_i = m_i + k_i \, modulo \, 27$
- Beispiel: 5 + 24 mod 27 = 2
- Decoding: $m_i = c_i - k_i \, mod \, 27$
- Absolut sicher, aber Schlüssel muss genauso lang wie Nachricht sein, rotes Telefon

Caesar

- m ist Folge von Großbuchstaben + ZR
- k ist Zahl zwischen 0 und 26
- Kodierung: Jeder Buchstabe von m wird ersetzt durch Buchstaben, der k später im Alphabet kommt
- Dekodierung: k früher
- Sehr unsicher, aber einfach
- und kurzer Schlüssel

Blockchiffrierung

- Nachricht wird in Blöcke der Länge b zerlegt. Jeder Block wird getrennt kodiert.
- Alle mit dem gleichen Schlüssel.
- Typisch Blocklänge 64, 128, 256 Bits
- Schlüssellänge ähnlich
- Populäre Verfahren: DES (Data-Encryption-Standard), AES (Nachfolger)
- Sicherheit: nicht gebrochen, aber …

Blockchiffrierung: Prinzip der Vorgehensweise

- Kodierung eines Blocks der Länge b
- Verknüpfe mit dem Schlüssel (wie im One-Time Pad)
- Wende invertierbare Substitution auf Paare benachbarter Buchstaben an
- Permutiere die Positionen
- Wiederhole 16 Mal.

Angriffe

- Caesar: Buchstabenhäufigkeit
- DES 56: brute-force mit Spezialhardware
- ENIGMA: Alan Turing und einer der ersten Computer
- Siehe Wikipedia: Cryptanalysis für weitere Beispiele
- AES 128 gilt als sicher für die nächsten 10 Jahre

Asymmetrisches Verfahren
Eine Analogie

- Bob möchte, dass man ihm geheime Nachrichten schicken kann.
- Er kauft sich ein Bügelschloss und hinterlegt das offene Bügelschloss an einem öffentlichen Ort
- Alice tut ihre Nachricht in eine Kiste, verschließt die Kiste mit dem Bügelschloss und schickt die Kiste an Bob
- Nur Bob kann die Kiste öffnen
- Vorteil: kein Treffen nötig
- Problem: aufwendiger, Authentifizierung, woher weiß Alice, dass das Schloss zu Bob gehört.

Asymmetrische Verfahren (seit 78)

- Sender (Alice) und Empfänger (Bob) benutzen verschiedene Schlüssel
- Bob erzeugt Schlüssel k_e und k_d, hält k_d geheim, veröffentlicht k_e
- Alice benutzt k_e zum Verschlüsseln
- Aus k_e kann man k_d nach heutiger mathematischer Kenntnis nicht in wenigen Jahren berechnen

Sicherheit

- RSA (Rivest-Shamir-Adleman, Turing Award), Rabin (Turing Award) : Faktorisierung von Zahlen mit 2000 Ziffern braucht nach Stand der Kunst Jahrzehnte (unter Nutzung aller Rechner)
- El Gamal: das gleiche gilt für diskreten Logarithmus bezüglich 2000 stelliger Primzahl
- Elliptische Kurven

Baby-Version von ElGamal

- Folge Bongartz/Unger (Alg der Woche)
- Annahme: Wir können multiplizieren und addieren, aber dividieren ist sehr sehr schwer, also
- Aus p und f kann man $P = p \times f$ berechnen, aber niemand kann aus f und $P = p \times f$ das p berechnen

Baby-Version von ElGamal

- **Empfänger** wählt p und f; veröffentlicht f und $P = p \times f$; p bleibt geheim
- **Sender** möchte m schicken, $m < P$
- Wählt eine zufällige Zahl s und schickt öffentlich

$$s \times f \text{ und } N = m + s \times P$$

Er hält s geheim.

- **Eve** kann s nicht berechnen und weiß nur

$$m \in \{N, N - P, N - 2P, N - 3P, \ldots\}$$

Baby-Version von ElGamal

- Empfänger wählt p und f und veröffentlicht f und $P = p \times f$
- Sender möchte m schicken, $m < P$
- Wählt eine Zahl s und schickt öffentlich

$$s \times f \quad \text{und} \quad N = m + s \times P$$

- **Empfänger** berechnet

$$p \times (s \times f) = s \times P$$

und dann $m = N - s \times P$

Die Details von ElGamal

- Die Details von ElGamal werde ich in der Vorlesung nicht behandeln, die Folien sind zum Nachlesen

Rechnen mod n

- Grundmenge = $\{0,1,\dots,n-1\}$, etwa $n = 7$
- Addition, Subtraktion, Multiplikation mod n

Bringe Ergebnis durch Restbildung wieder in die Grundmenge

$$4 \times 6 = 36 \equiv 1 \; mod \; 7$$

$$3 + 4 \times 2 = 11 \equiv 4 \; mod \; 7$$

- n prim, dann gibt es zu jedem $a \neq 0$ ein b so dass $a \times b \equiv 1 \; mod \; n$ und es gibt ein g so dass $\{g, g^2, \dots, g^{n-1}\} = \{1, \dots, n-1\}$

ElGamal

- Empfänger wählt Primzahl p, Erzeuger g und $x, 2 \leq x \leq p-1$ und veröffentlicht
$$(p, g, y = g^x \text{mod } p)$$
- Berechnung von y aus x ist leicht, aber von x aus y ist praktisch unmöglich
- Sender möchte m schicken, wählt s und schickt
$$(z = g^s \bmod p, N = m \times y^s \, mod \, p)$$

ElGamal

- Empfänger wählt Primzahl p, Erzeuger g und $x, 2 \leq x \leq p-1$ und veröffentlicht
$$(p, g, y = g^x \bmod p)$$
- Sender möchte m senden, wählt s, sendet
$$(z = g^s \bmod p, N = m \times y^s \, mod \, p)$$

- Eve kann s nicht berechnen und weiß nur $m \in \left\{N, \frac{N}{y}, N/y^2, N/y^3, \ldots\right\}$

ElGamal

- Empfänger wählt Primzahl p, Erzeuger g und $x, 2 \leq x \leq p-1$ und veröffentlicht
$$(p, g, y = g^x \text{mod } p)$$
- Sender möchte m senden, wählt s, sendet
$$(z = g^s \bmod p, N = m \times y^s \, mod \, p)$$
- Empfänger berechnet $z^x = g^{sx} = y^s$ und dann $m = N/y^s \; mod \, p$.

Electronic Banking

- Kunde kennt öffentlichen Schlüssel k_e der Bank
- Kunde erfindet geheimen Schlüssel k (256 Bit Zufallszahl) für symmetrisches Verf.
- Kunde verschlüsselt k mit k_e und schickt den verschlüsselten Schlüssel an die Bank
- Bank entschlüsselt mit Hilfe ihres privaten Schlüssels k_d
- Nun symmetrisches Verfahren mit k.

- Problem: woher kenne ich zuverlässig den öffentlichen Schlüssel meiner Bank?

Unterschriften

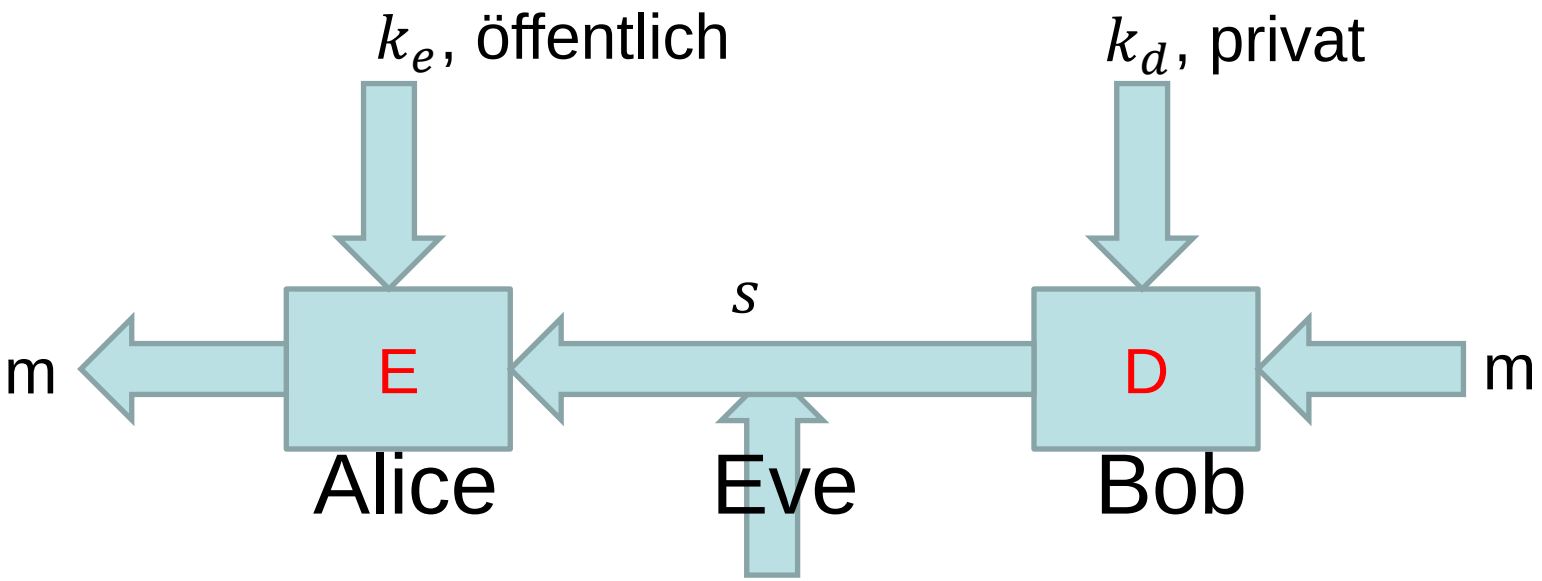
- Eigenschaft: Unterschreiber kann sie nicht abstreiten
- Zweck: Verbindlichkeit
- Wie : alles was nur der Unterschreiber kann:
  - Traditionnell: handschriftliche Unterschrift, Fingerabdruck,
  - Nun: Die Funktion D_X kann nur die Person X ausführen, weil nur sie den geheimen Schlüssel der Person X kennt

Digitale Signaturen

- Seien E_X und D_X die Funktionen von X und gelte $E_X(D_X(x)) = x$ für alle x.
- Um m zu signieren, erzeugt X den String $s = D_X(m)$
- *Das Paar (m,s) ist das unterschriebene m*
- Vertragspartner überprüft dass $E_X(s) = m$ gilt
- Da man nach Annahme s aus m ohne Kenntnis des Programms für D_X (also ohne Kenntnis von k_d) nicht berechnen kann, kann nur der Besitzer von k_d den Text s erzeugen. Also kann X die Unterschrift nicht abstreiten.

Digitale Signaturen
Signatur = etwas, das nur ich kann

$$m = E\big(k_e, D(k_d, m)\big)$$



Eve = Eavesdropper $\quad s =$ Signatur von m

Electronic Banking, Schritt 1

- Bank hinterlegt ihren öffentlichen Schlüssel k_e bei einem Trustcenter
- Kunde kennt (fest eingebaut im Browser) den öffentlichen Schlüssel des TC und fragt nach Schlüssel der Bank
- TC signiert k_e und schickt an Kunden
- Kunde verifiziert und benutzt dann k_e

Zusammenfassung

- Electronic Banking, Einkaufen im Netz nutzt symmetrische und asymmetrische Kryptographie
- Kommunikation mit der Bank ist damit geschützt https://my.hypovereinsbank.de/
- Aber Vorsicht: für die Qualität ihrer PIN und Passwörter sind sie selbst verantwortlich

Kryptographie (geheim-schreiben) Hauptziele (nach Wolfgang Ertel)

Vertraulichkeit / Zugriffsschutz: Nur dazu berechtigte Personen sollen in der Lage sein, die Daten oder die Nachricht zu lesen (auch teilweise).

Nachricht/Daten verschlüsseln

Integrität / Änderungsschutz: Die Daten müssen nachweislich vollständig und unverändert sein.

Nachricht/Daten verschlüsseln oder signieren

Authentizität, Verbindlichkeit / Fälschungsschutz, Nichtabstreitbarkeit: Der Urheber der Daten oder der Absender der Nachricht soll eindeutig identifizierbar sein, und seine Urheberschaft sollte nachprüfbar und nicht abstreitbar sein.

Nachricht/Daten signieren

Speicherung von Passwörtern

- $h =$ One-Way Funktion, z.B. Blockcypher
- Sei c = h(Passwort von KM)
- Speichere ungeschützt das Paar (KM, c)
- KM beweist seine Authentizität durch die Fähigkeit c erzeugen zu können
- Angriffe: brute-force, da Passworte oft kurz.
- Abhilfe
  - Maschine für h geht nach 3 inkorrekten Auswertungen kaputt
  - Oder automatische Verlängerung durch Zufallsstring speichere (KM, zufälliges s, h(Passwort von KM + s))