



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 (seit 1978)
 - Digitale Unterschriften
- Anwendungen: Electronic Banking, Sicherheitsinfrastrukturen



Kryptographie (geheim-schreiben)

Hauptziele (nach Wolfgang Ertel)

Vertraulichkeit / Zugriffsschutz: Nur berechtigte Personen können die Daten/Nachricht lesen.

Integrität / Änderungsschutz: Daten können nicht unbemerkt verändert werden.

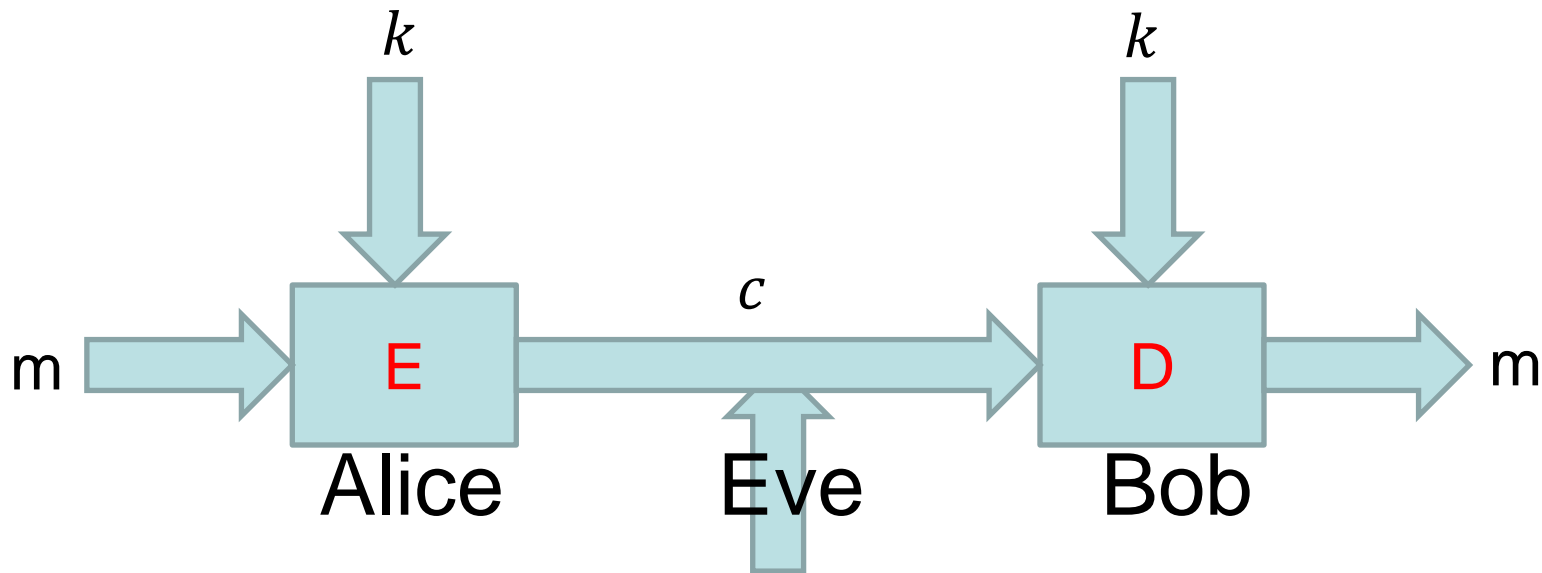
Authentizität/Fälschungsschutz: Der Urheber der Daten oder der Absender der Nachricht soll eindeutig identifizierbar sein

Verbindlichkeit/Nichtabstreitbarkeit: Urheberschaft sollte nachprüfbar und nicht abstreitbar sein.



Symmetrische Verschlüsselung

Alice und Bob verabreden einen geheimen Schlüssel k



Eve = Eavesdropper

Beispiel: Caesar

- D und E sind ein Gerät.
- Schlüssel k ist Drehwinkel, bzw. das Ziel von A.
- E liest von innen nach außen
D von außen nach innen.
- Einfach, aber sehr unsicher; nur 26 Schlüssel.



Nomenklatur

- m = Klartext, Nachricht, message
- c = Geheimtext, cyphertext
- E und D sind (allgemein bekannte) Geräte (heute meist Programme)
- Werden durch den Schlüssel k personalisiert
- Ohne Kenntnis von k soll es praktisch unmöglich (10min, 5h, 100 Jahre) sein, m aus c zu bestimmen
- Für alle m und k : $m = D(k, E(k,m))$



Symmetrische Kryptographie

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)

- Wie Caesar, aber für jeden Buchstaben des Texts benutzt man einen eigenen Schlüssel, d.h.
- Schlüssel ist ein zufälliger Text (jeder Buchstabe ist gewürfelt) mit der gleichen Länge wie die Nachricht.
- Absolut sicher, aber Schlüssel muss genauso lang wie Nachricht sein
- Schlüsselaustausch ist aufwendig



Blockchiffrierung

- Nachricht wird in Blöcke der Länge b zerlegt. Jeder Block wird getrennt kodiert.
- Alle mit dem gleichen Schlüssel.
- Typische Blocklänge 64, 128, 256 Bits
- Schlüssellänge ähnlich, 2^{128} verschiedene k
- Populäre Verfahren: DES (Data-Encryption-Standard), AES (Nachfolger)
- Sicherheit: 64Bit Versionen unsicher, 128Bit noch sicher,



Blockchiffrierung: Prinzip der Vorgehensweise

- Kodierung eines Blocks der Länge $b = 128$
- Verknüpfe mit dem Schlüssel (wie im One-Time Pad)
- Fasse Block als Folge von 16 Miniblöcken von je 8 Bit auf. 8 Bit = 0 .. 255. Ersetze Paare von Miniblöcken nach einem festen Schema, etwa 7, 113 \rightarrow 251, 33.
- Permutiere die Positionen
- Wiederhole 16 Mal.



Angriffe



- Caesar:
Buchstabenhäufigkeit
- DES 56: brute-force
mit Spezialhardware
- ENIGMA (Rätsel):
Alan Turing



Symmetrische Verfahren

Zusammenfassung

- Sender (Alice) und Empfänger (Bob) verabreden einen gemeinsamen Schlüssel k
- Dieser Schlüssel muss geheim bleiben
- Wie einigt man sich auf einen Schlüssel?
 - Früher: Treffen oder Bote
 - Heute: asymmetrisches Verfahren zum Schlüsselaustausch
- Beispiele: One-Time Pad, Caesar, AES128
- Sehr effiziente Ver- und Entschlüsselung
- Bei n Teilnehmern: $n \times n$ Schlüssel



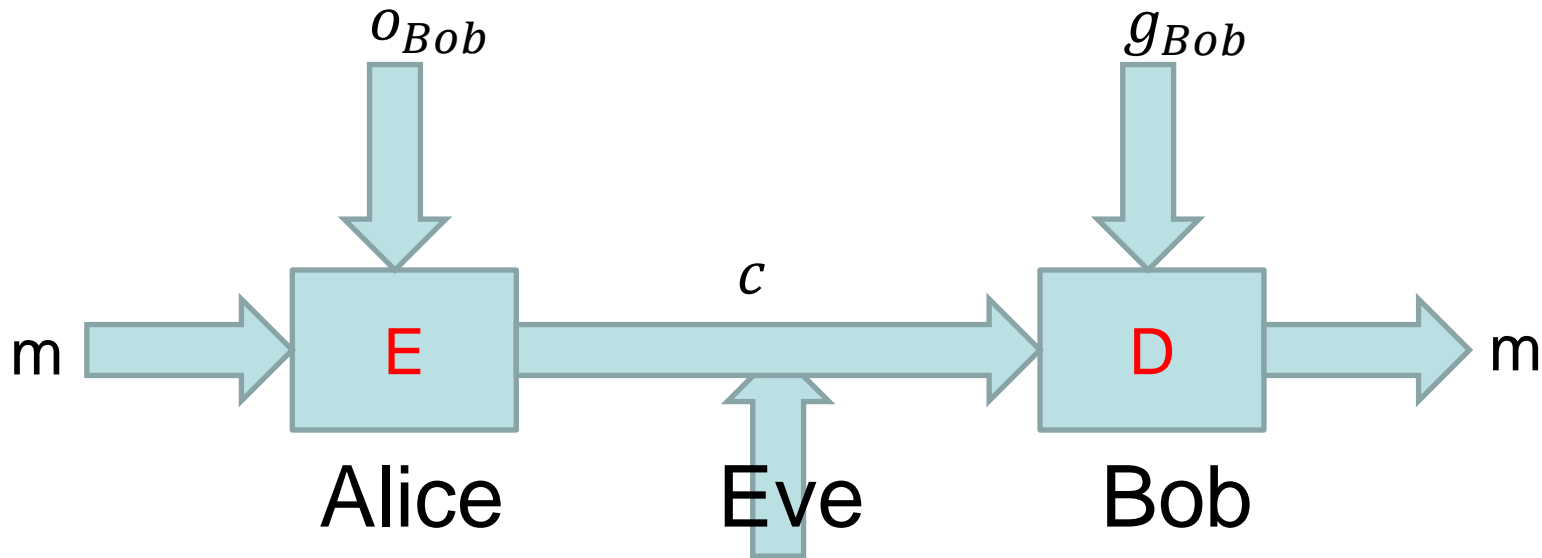
Asymmetrische Verfahren (seit 78)

- (Empfänger) Bob erzeugt Schlüsselpaar g_{Bob} und o_{Bob} , hält g_{Bob} geheim, veröffentlicht o_{Bob}
- Jeder, der Bob eine Nachricht schicken will, benutzt o_{Bob} zum Verschlüsseln
- g_{Bob} kann aus o_{Bob} nach heutiger Kenntnis nicht in wenigen Jahren berechnen. Ohne Kenntnis von g_{Bob} kann man nicht entschlüsseln



Ver- und Entschlüsselung

$$m = D(g_{Bob}, E(o_{Bob}, m))$$



Eve = Eavesdropper

Asymmetrisches Verfahren

Eine Analogie

- Bob möchte, dass man ihm geheime Nachrichten schicken kann.
- Er kauft sich viele identische Bügelschlösser und hinterlegt die offenen Schlösser an öffentlichen Orten
- 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.



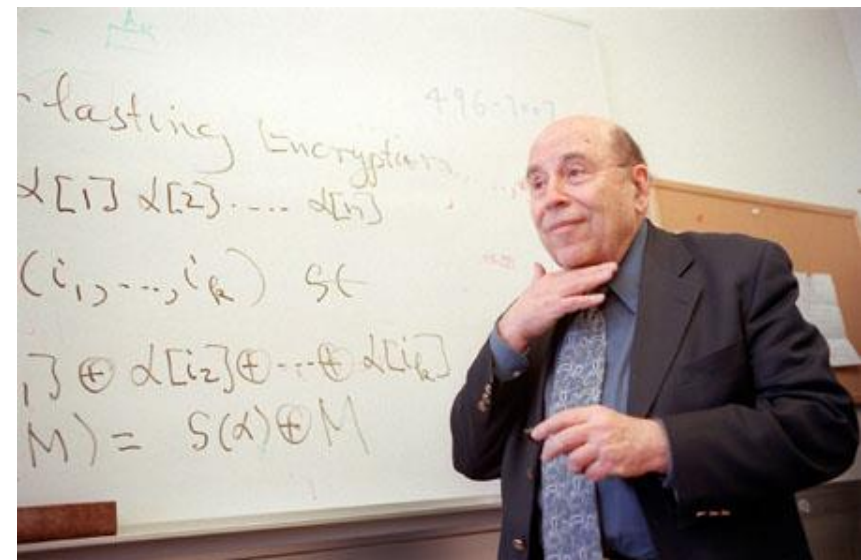
Nomenklatur

- E und D sind (allgemein bekannte) Geräte (heute meist Programme)
- Werden durch die Schlüssel personalisiert, also
 - $E_{Bob} = E$ mit Schlüssel o_{Bob} und
 - $D_{Bob} = D$ mit Schlüssel g_{Bob}
- E_{Bob} ist öffentlich, nur Bob kennt D_{Bob}



Erfinder

- RSA (Rivest-Shamir-Adleman, Turing Award), Rabin (Turing Award), 1978
- Später dann Verfahren von El Gamal und Elliptische Kurven



Sicherheit

- Sicherheit von RSA
 - Multiplizieren von 1000-stelligen Zahlen ist einfach, aber
 - sie aus ihrem Produkt zu berechnen, dauert nach heutigem Wissen 100 Jahre
- El Gamal: das gleiche gilt für den diskreten Logarithmus bezüglich 2000-stelliger Primzahl
- 1000-stellige Primzahlen findet man leicht



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 bleibt geheim)

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

- **Empfänger** berechnet

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

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



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 $s \times f$ und $N = m + s \times P$
- **Eve** kennt $s \times P$ nicht und weiß nur $m \in \{N, N - P, N - 2P, N - 3P, \dots\}$



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

$$4 \times 6 = 36 \equiv 1 \pmod{7}$$

$$3 + 4 \times 2 = 11 \equiv 4 \pmod{7}$$

- n prim, dann gibt es zu jedem $a \neq 0$ ein b so dass $a \times b \equiv 1 \pmod{n}$ 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) wobei $y = g^x \bmod p$
- Berechnung von y aus x ist leicht, aber von x aus y ist praktisch unmöglich
- Sender möchte m schicken, wählt s und schickt

$$(z = g^s \bmod p, N = m \times y^s \bmod p)$$



ElGamal

- Empfänger wählt Primzahl p , Erzeuger g und x , $2 \leq x \leq p - 1$ und veröffentlicht (p, g, y) wobei $y = g^x \bmod p$
- Sender möchte m senden, wählt s , sendet $(z = g^s \bmod p, N = m \times y^s \bmod p)$
- Eve kennt y^s und weiß nur $m \in \left\{ N, \frac{N}{y}, N/y^2, N/y^3, \dots \right\}$



ElGamal

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



Electronic Banking

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

- Problem: woher kenne ich 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 ihren geheimen Schlüssel kennt



Digitale Signaturen

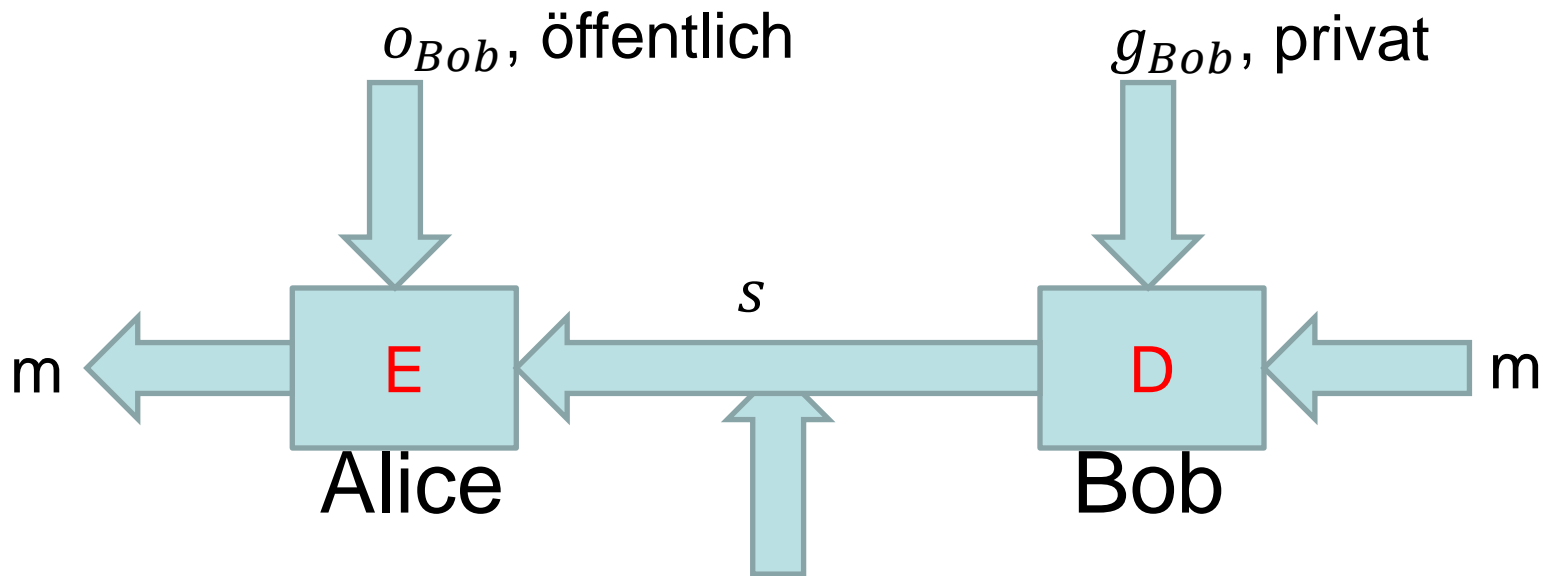
- Seien E_X und D_X die Funktionen von X und gelte $E_X(D_X(x)) = x$ für alle x .
- Um m zu signieren, berechnet X den String $s = D_X(m)$
- Das Paar (m,s) ist das unterschriebene m
- Vertragspartner überprüft dass $E_X(s) = m$ gilt
- **Nur X kann s aus m erzeugen. Also kann X die Unterschrift nicht abstreiten.**



Digitale Signaturen

Signatur = etwas, das nur ich kann

$$m = E\left(o_{Bob}, D(g_{Bob}, m)\right)$$



s = Signatur von m

**Bob möchte m signieren und dann
verschlüsselt an Alice schicken**



Electronic Banking, Schritt 1



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



Zusammenfassung

- Electronic Banking, Einkaufen im Netz nutzt symmetrische und asymmetrische Kryptographie
- Kommunikation mit der Bank ist damit geschützt <https://my.hypovereinsbank.de/>
- Aber Vorsicht: verschlüsselte Übertragung garantiert noch nicht Gesamtsicherheit, z.B. unsicheres Passwort
- Mehr dazu in der Vorlesung Sicherheit und Privatheit.



Kryptographie (geheim-schreiben)

Hauptziele (nach Wolfgang Ertel)

Vertraulichkeit / Zugriffsschutz: Nur dazu berechnigte 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 $c = h(\text{Passwort von KM})$
- 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 $(KM, \text{zufälliges } s, h(\text{Passwort von KM} + s))$



Ver- und Entschlüsselung

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

- $D_X(E_X(x)) = x$ 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 und D sind bijektiv
- Oft auch noch $D(E(x)) = x$ für alle x **nützlich für digitale Unterschriften**



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
 - $D_X(E_X(x)) = x$ für alle x
 - E_X und D_X sind durch Programme realisiert. Sei m eine beliebige Nachricht und $c = E_X(m)$. 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

