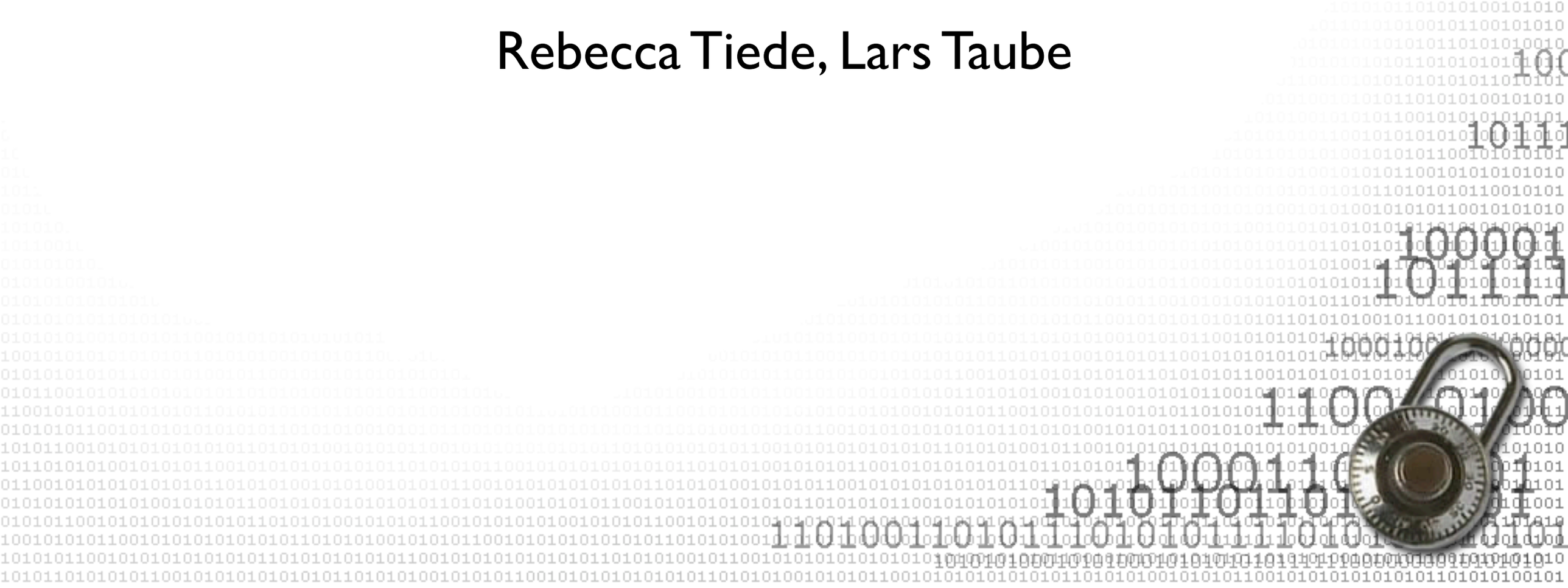
Kryptographie mit Anwendungen

Rebecca Tiede, Lars Taube



Inhalt

- Terminologie
- Authentizität
- Hashing
- Digitale Signatur
- Secret Sharing

Terminologie

Terminologie

Sender und Empfänger

Ein Sender möchte einem Empfänger eine Nachricht schicken, die sicher ankommt und nicht von anderen gelesen werden kann.

Nachricht (message M)

Verschlüsselung (encryption E)

Chiffretext (ciphertext C)

Entschlüsselung (decryption D)

Terminologie

- Kryptographie
  ist die Wissenschaft, die sich mit der Absicherung von Nachrichten beschäftigt.

- Kryptoanalyse
  ist die Kunst, Chiffretexte aufzubrechen und ihren geheimen Inhalt lesbar zu machen

- Kryptologie
  ist der Zweig der Mathematik, der Kryptographie und Kryptoanalyse umfasst

Terminologie

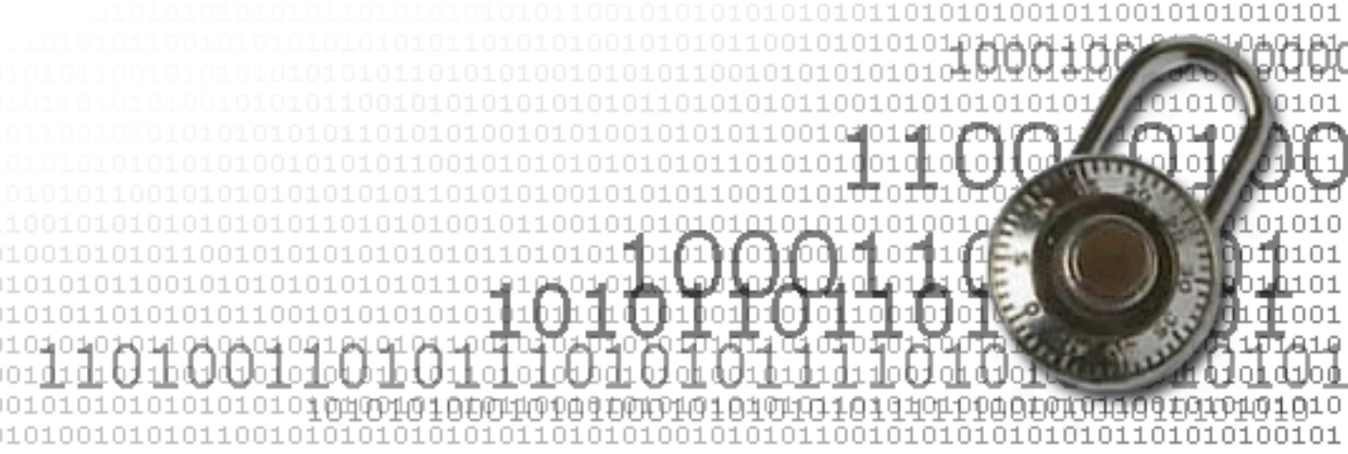
- kryptographischer Algorithmus / Chiffrierung
  ist die mathematische Funktion, die zur Ver- und Entschlüsselung verwendet wird.

- Eingeschränkter Algorithmus
  Wenn die Sicherheit eines Algorithmus von der Geheimhaltung seiner Arbeitsweise abhängt, spricht man von einem Eingeschränktem Algorithmus. Sobald ein Algorithmus „versehentlich“ bekannt wird, ein neuer benutzt werden muss.

Terminologie

- Um das zu vermeiden, führt man
  Schlüssel (key K)
  ein: Dieser kann aus einem sehr umfangreichen Wertebereich gewählt werden, dem

- Schlüsselraum
  Bereich aller möglichen Schlüssel



Terminologie

- Für die Verschlüsselungsfunktion ergibt sich

  $E_k(M) = C$

  $D_k(C) = M$

  -> $D_k (E_k (C)) = M$

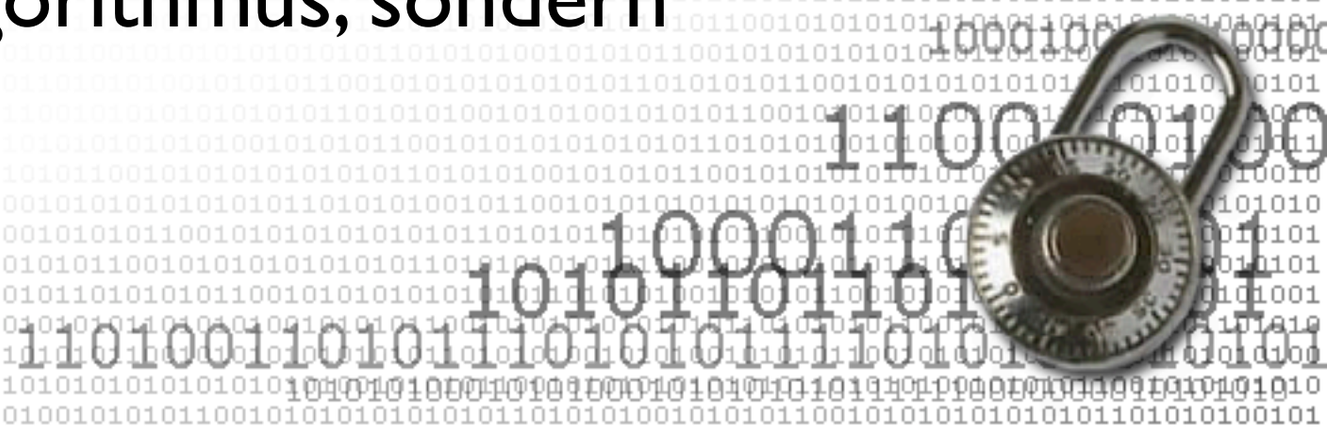
  Bei Verwendung von zwei unterschiedlichen Schlüsseln zur Chiffrierung und Dechiffrierung

  $E_{k1}(M) = C$

  $D_{k2}(C) = M$

  -> $D_{k2} (E_{k1} (C)) = M$

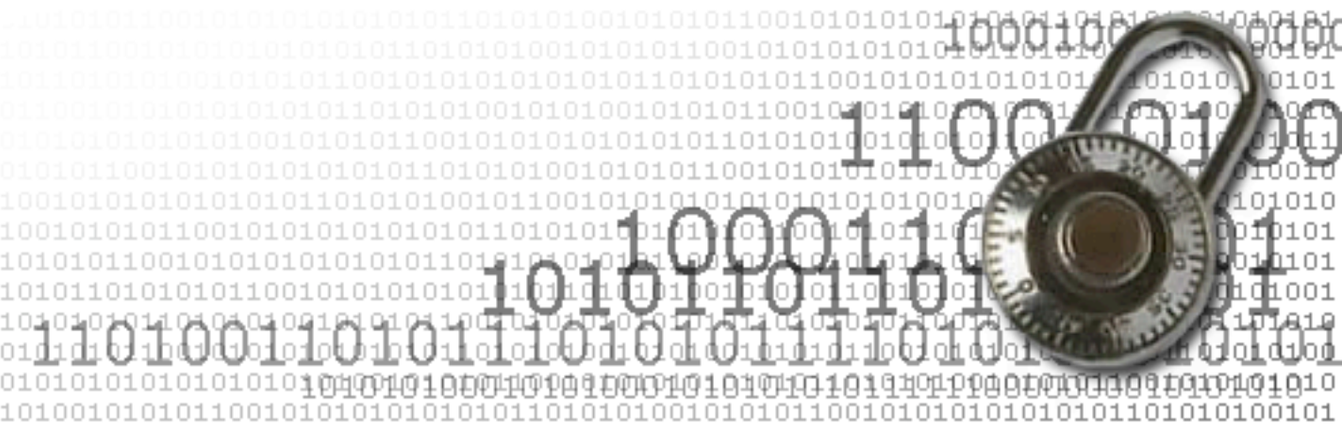
- Die Sicherheit beruht nun nicht mehr auf der Sicherheit des Algorithmus, sondern auf der der Schlüssel.



Terminologie

- Ein Kryptosystem
  besteht aus einem Algorithmus einschließlich aller möglichen Klartexte, Chiffretexte und Schlüssel. Neben der Geheimhaltung soll Kryptographie noch andere Ansprüche erfüllen:

- Authentizität
  Empfänger soll Herkunft einer Nachricht ermitteln können, ein Eindringling sollte sich nicht als andere Person ausgeben können



Terminologie

- Integrität
  Der Empfänger sollte überprüfen können, ob die Nachricht bei Übermittlung verändert wurde, ein Eindringling sollte die echte nicht durch eine falsche Nachricht ersetzen können.

- Verbindlichkeit
  Ein Sender sollte nachträglich nicht leugnen können, eine Nachricht gesendet zu haben.

Authentifizierung

Authentifizierung

Authentifizierung

- Zweck: Sicherheit über Inhalt und Quelle von übertragenen Nachrichten
- Authentifizierung nur berechtigten Teilnehmern durch einen geheimen Schlüssel möglich
- „Kommunikations – System“: Nachrichtenübertragung und – speicherung zusammengefasst.
- Besteht aus: Sender, Empfänger und einem potentiellen Angreifer, der Nachrichten abfangen oder verfälschen will.

Authentifizierung

In einem weiteren Szenario taucht noch eine vierte Person auf, ein sog. Vermittler, der Schutz vor dem Angreifer bieten soll.

Kommunikation über öffentlichen Kanal
=> Bedrohung der Authentifizierung, wenn Angreifer im System Nachrichten verfälschen oder falsche in den Kanal leiten kann

In geheimen Systemen: Angreifer passiv, „lauscht“
Authentifizierungssystem: Angreifer aktiv, verfälscht Nachrichten

Authentifizierung

Klassifizierung von Authentifizierungssystemen
Betrachtung zweier Kriterien:

- die Beziehung zwischen Authentifizierung und Sicherheit
  - MIT und OHNE Sicherheit
- der Rahmen / das System der Sicherheitsanalyse
  - „uneingeschränkte“ Sicherheit: Feind mit unendlicher Rechenpower
  - „berechenbare“ Sicherheit: Feind mit beschränkter Rechenpower

Authentifizierung

Authentifizierung mit Vermittler

Basismodell von Authentifizierung:

Sender und Empfänger vertrauenswürdig, teilen Schlüssel, kryptographisch nicht zu unterscheiden, Angreifer ein Außenseiter

Authentifizierung

Zur Unterscheidung: unterschiedliche Schlüssel, nicht vertrauenswürdig, vierte unvoreingenommene Person, der Vermittler, hat alle Informationen über die Schlüssel und vertrauenswürdig

„Schlüsselverteilungsphase“: Schlüsselwahl nach bestimmten Bedingungen, Sender verschlüsselt damit seinen Klartext und überträgt ihn.

Zweifel des Empfängers gegenüber dem Sender werden durch den Vermittler ausgeräumt.

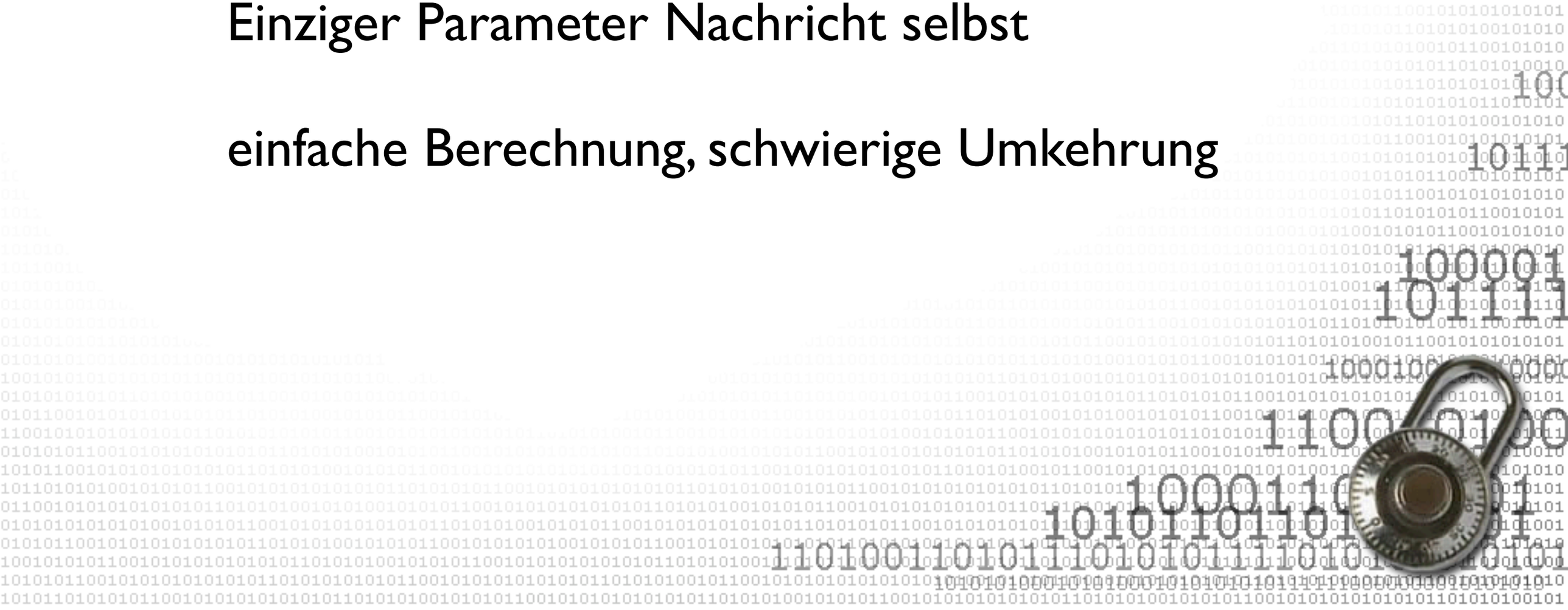
Hashing

Hashing

Eine Hashfunktion h ist eine Einweg-Funktion, die einen Eingabe-String variabler Länge in einen (i. d. R. kürzeren)Ausgabe-String (Hashwert) umwandelt.

Einziger Parameter Nachricht selbst

einfache Berechnung, schwierige Umkehrung



Hashing

Geforderte Eigenschaften:

- sie soll eine komplexe Funktion auf alle Bits der Nachricht sein
- unterschiedliche Nachrichten sollen durch sie auf unterschiedliche Werte abgebildet werden -> Kollisionsresistenz
- sie soll leicht zu berechnen, aber schwer zurückzurechnen sein

Hashing

Klassifizierung in zwei Klassen:

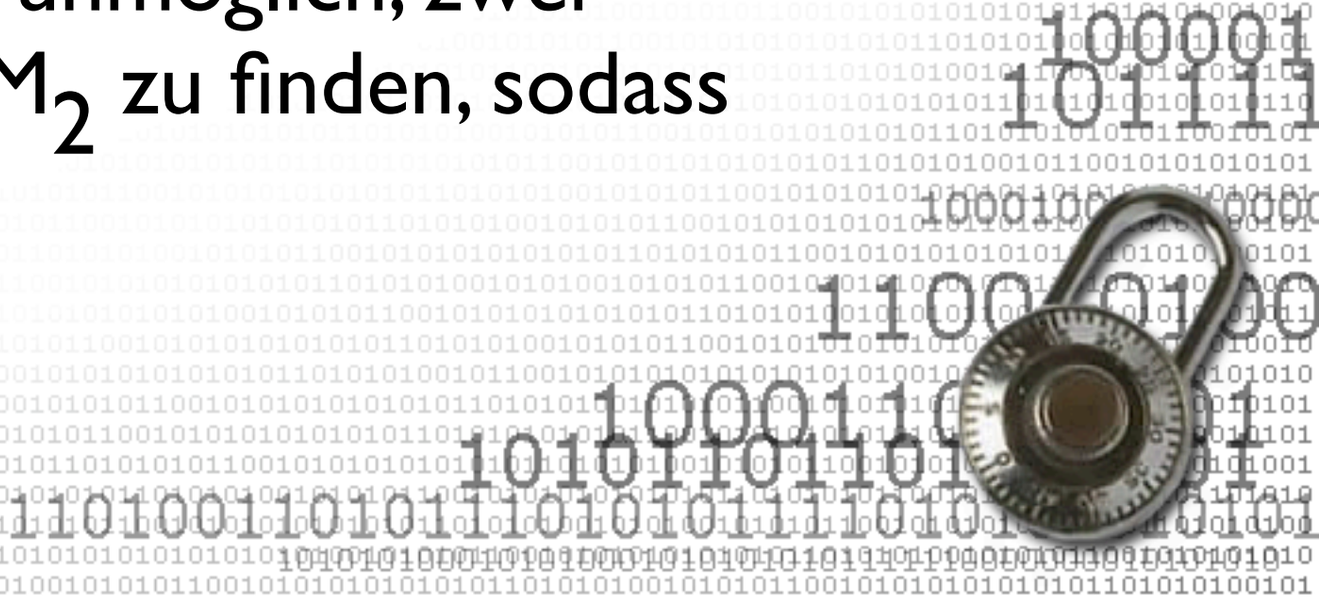
1. Starke Hashfunktion mit Eigenschaften

   i. h kann auf jede Nachricht jeder Größe angewandt werden

   ii. h gibt einen Wert fester Länge zurück

   iii. h(M) ist leicht anzuwenden

   iv. es ist rechnerisch unmöglich, zwei Nachrichten M_1, M_2 zu finden, sodass $h(M_1) = h(M_2)$



Hashing

2 Schwache Hashfunktion mit Eigenschaften

(i) – (iii)

(iv') es ist rechnerisch „schwer“ möglich, bei zufällig gewählter Nachricht M eine Nachricht M' zu finden, sodass h(M) = h(M')

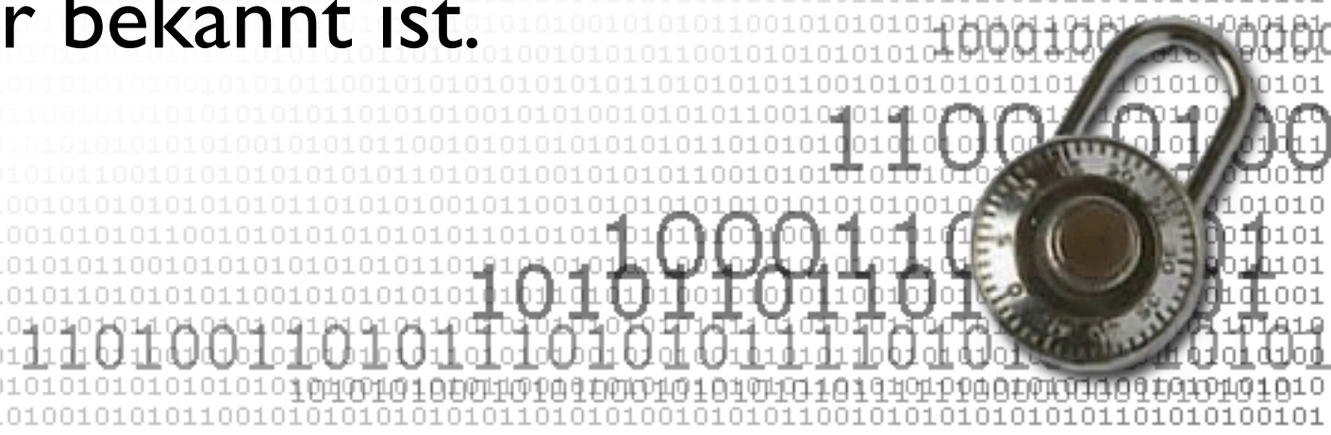
Hashing

Einsatz von Hashfunktionen

- Überprüfung der Echtheit eines Dokuments, anhand seines Hashwertes.

- Als Prüfsumme für Downloads, mit öffentlicher Hashfunktion.

- Zur Authentifizierung des Senders eines Dokumentes mit einer Hashfunktion, die nur Sender und Empfänger bekannt ist.



Hashing

Hashing mit Hilfe von Blockchiffren

Kurz: Was sind Blockchiffren?

Unter einer Blockchiffre versteht man das Verschlüsseln einer Nachricht durch Anwendung des Verschlüsselungsalgorithmus auf ihre einzelnen gleich großen Blöcke.

Hashing

Hashing mit Blockchiffren verläuft folgendermaßen:

- E ein willkürlich gewählter (Verschlüsselungs-) Algorithmus,
- M eine Nachricht,
- K ein Schlüssel und

Dann ist die Verschlüsselung von M mit dem Schlüssel K und den Algorithmus E benutzend:

E (K, M)

Beispiel: Rabins Scheme

Hashing

M wird in t gleichgroße Blöcke $M_1, M_2, \ldots M_t$

aufgeteilt, deren Länge der des Verschlüsselungs-Algorithmus entspricht.

Der Hashwert wird durch folgende Rechnungen ermittelt:

$H_0 = IV$ mit IV als Initialisierungsvektor

$H_i = E_k (M_i, H_{i-1}), i = 1, 2, ..., t$

$H (M) = H_t$

M_i ist ein Nachrichtenblock, H_i Zwischenergebnis der Hashfunktion (hier als Schlüssel für den jeweils nächsten Wert benutzt) und H (M) ist der Hashwert.

Hashing

Gegen Hashfunktionen gibt es zwei (Brute-Force-) Angriffe:

1. Finde bei gegebenem Hashwert einer Nachricht H (M) ein anderes Dokument M', sodass H (M) = H (M').

2. Finde zwei zufällige Nachrichten M und M', sodass H (M) = H (M').

Man spricht von „Geburtstagsangriff“ oder auch Geburtstags–Attacke

Hashing

Geburtstags–Attacke

Idee beruht auf bekanntem Problem der Wahrscheinlichkeitstheorie, dem Geburtstags-Paradoxon.

Frage 1: Wie viele Leute müssen in einem Raum sein, so dass mit hoher Wahrscheinlichkeit (>0,5) eine Person heute Geburtstag hat?

Frage 2: Wie viele Leute müssen in einem Raum sein, so dass mit hoher Wahrscheinlichkeit (>0,5) mindestens zwei Personen am gleichen Tag Geburtstag haben?

Hashing

- Zu Frage 1:
  Sei g_i der Geburtstag von Person i für i = 1, ..., n.

  $P(g_1 = x \vee g_2 = x \vee ... \vee g_n = x)$
  $= 1 - P(g_1 \neq x \wedge g_2 \neq x \wedge ... \wedge g_n \neq x)$

  $= 1 - (364/365)^n > 0.5$

  $\Rightarrow (364/365)^n > 0.5$
  $\Rightarrow n \geq 253$

Hashing

- Zu Frage 2:
  Gesucht ist die Zahl k der Personen, so dass im Mittel mindestens zwei Personen am gleichen Tag Geburtstag haben:

$$P(g_1 = g_2 \vee g_1 = g_2 \vee \ldots \vee g_1 = g_k)$$
$$= 1 - P(g_1 \neq g_2 \wedge g_1 \neq g_2 \wedge \ldots \wedge g_1 \neq g_k)$$
$$= 1 - (364/365)^{k(k-1)/2} > 0.5$$

Hashing

Setzt man nun k(k-1)/2 = n, so erhält man

$k = ½ + \sqrt{(1/4 + 2n)} = 22.98 \Rightarrow k \geq 23$

Auf unsere Hashfunktion bezogen bedeutet das:

Hashing

Zu 1.) $n = 2^{m-1}$, m ist die Länge des Hashwertes

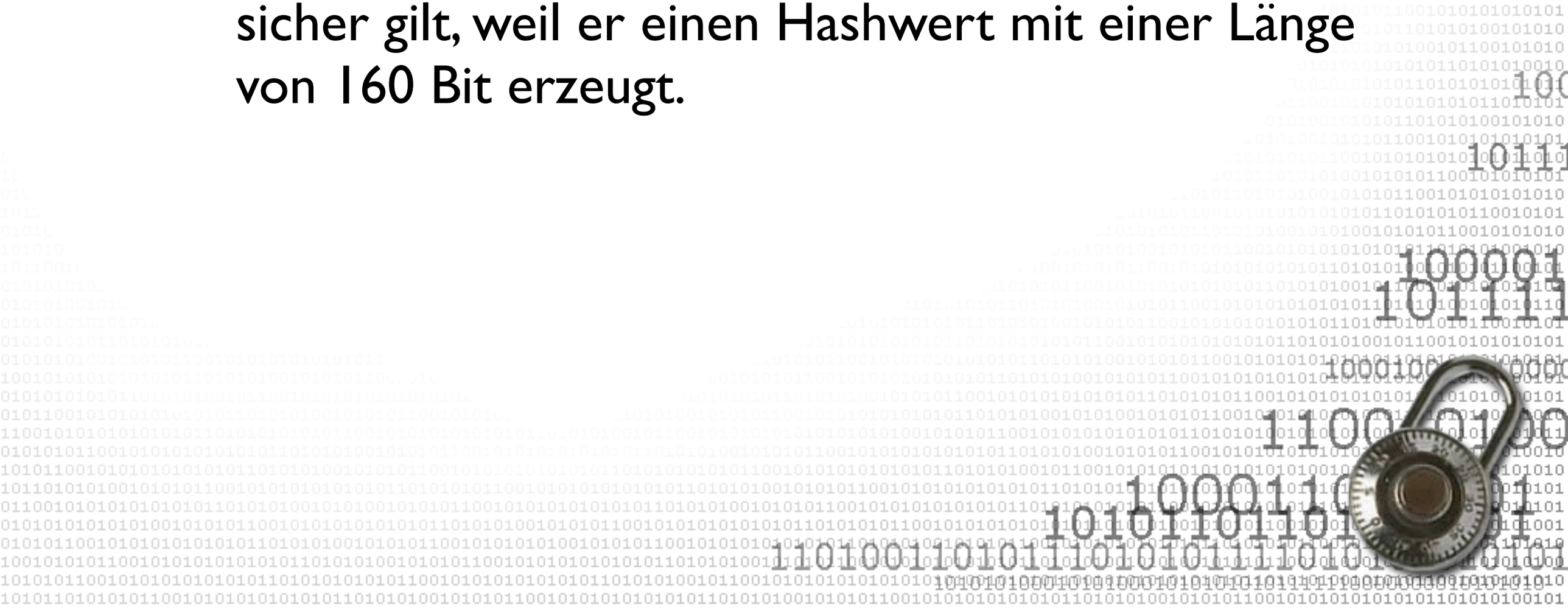
Zu 2.) $k \approx \sqrt{2n}$, d.h. $k \approx \sqrt{(2 \cdot 2^{m-1})} = \sqrt{2^m} = 2^{m/2}$

=> jeder Hashalgorithmus, der einen Wert der Länge 64 Bit erzeugt, gilt als unsicher, da die Zeitkomplexität der Geburtstags-Attacke hier bei 2^{32} liegt.

=> der erzeugte Hashwert einer Hashfunktion sollte eine Länge von ungefähr 128 Bit haben, um eine Geburtstags-Attacke zu vereiteln.

Hashing

Nun ein Hashalgorithmus, der in der Praxis für digitale Signaturen verwendet wird und der als sicher gilt, weil er einen Hashwert mit einer Länge von 160 Bit erzeugt.



Hashing

Secure Hash Algorithm (SHA)

- Hänge an die Nachricht eine 1 und dann so viele 0en, dass die Länge einem Vielfachen von 512 – 64 Bit entspricht.

- Hänge eine 64-Bit-Darstellung der Nachrichtenlänge (vor Auffüllen) an.

Hashing

- Fünf Variablen der Länge 32 Bit werden initialisiert:
  A=0x67452301
  B=0xefcdab89
  C=0x98badcfe
  D=0x10325476
  E=0xc3d2e1f0

a = A
b = B
c = C
d = D
e = E

Hashing

Die Hauptschleife besteht aus vier Runden mit je 20 Operationen.
Jede Operation führt mit dreien der Werte a, b, c, d, e eine nichtlineare Funktion durch und anschließend Verschiebung und Addition.
Die nichtlinearen Funktionen:

$f_t(X,Y,Z) = (X \wedge Y) \vee ((X) \wedge Z)$ für t = 0 bis 19

$f_t(X,Y,Z) = X \oplus Y \oplus Z$ für t = 20 bis 39

$f_t(X,Y,Z) = (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z)$ für t = 40 bis 59

$f_t(X,Y,Z) = X \oplus Y \oplus Z$ für t = 60 bis 79

Hashing

- Vier Konstanten werden benutzt:

K_t = 0x5a827999 für t = 0 bis 19 // $\frac{2^{\frac{1}{2}}}{4}$

K_t = 0x6ed9eba1 für t = 20 bis 39 // $\frac{3^{\frac{1}{2}}}{4}$

K_t = 0x8f1bbcdc für t = 40 bis 59 // $\frac{5^{\frac{1}{2}}}{4}$

K_t = 0xca62c1d6 für t = 60 bis 79 // $\frac{10^{\frac{1}{2}}}{4}$

Hashing

Nachrichtenblock wird von 16 32-Bit-Wörtern ($M_0 - M_{15}$) nun in 80 32-Bit-Wörter umgewandelt ($W_0 - W_{79}$):

$W_t = M_t$ für t = 0 bis15

$W_t = (W_{t-3} + W_{t-8} + W_{t-14} + W_{t-16})$ <<< 1für t = 16 bis 79

t sei Operationsnummer, die von 0 – 79 läuft

Wt stellt den t-ten Teilblock der expandierten Nachricht dar

<<< s steht für eine zirkuläre Linksverschiebung um s Bit

Hashing

Für t = 0 bis 79

TEMP = (a<<<5) + f_t (b, c, d) +e +W_t +K_t

e = d

d = c

c = b<<<30

b = a

a = TEMP

Schließlich werden a, b, c, d, e zu A, B, C, D, E addiert und man fährt mit dem nächsten Datenblock fort.

Die Ausgabe ist die Konkatenierung von A, B, C, D, E.

Digitale Signatur

Digitale Signatur

Was ist eine Digitale Signatur?

- Eine Digitale Signatur soll genau das selbe darstellen wie eine handschriftliche Signatur bzw, Unterschrift.
- Die Digitale Signatur sollte folgende Bestandteile umfassen:
  - Identität des Signierers
  - das Dokument
  - der Zeitpunkt der Signierung

Digitale Signatur

- Die digitale Signatur sollte folgendes leisten:
  - Einzigartig:
    Eine Signatur kann nur vom Besitzer erzeugt werden und reflektiert den Inhalt es Dokuments
  - Fälschungssicher
  - Einfach zu erstellen und zu verifizieren
  - Unleugbarkeit des Ursprungs der Signatur
- Der Algorithmus zur Verifikation muss "öffentlich" sein.

Digitale Signatur

Asymmetrische Verschlüsselung
Der Unterschied zu symmetrischer Verschlüsselung besteht darin, dass es anstelle eines Schlüssels ein Schlüsselpaar gibt: Einen privaten und einen öffentlichen (public) Schlüssel.

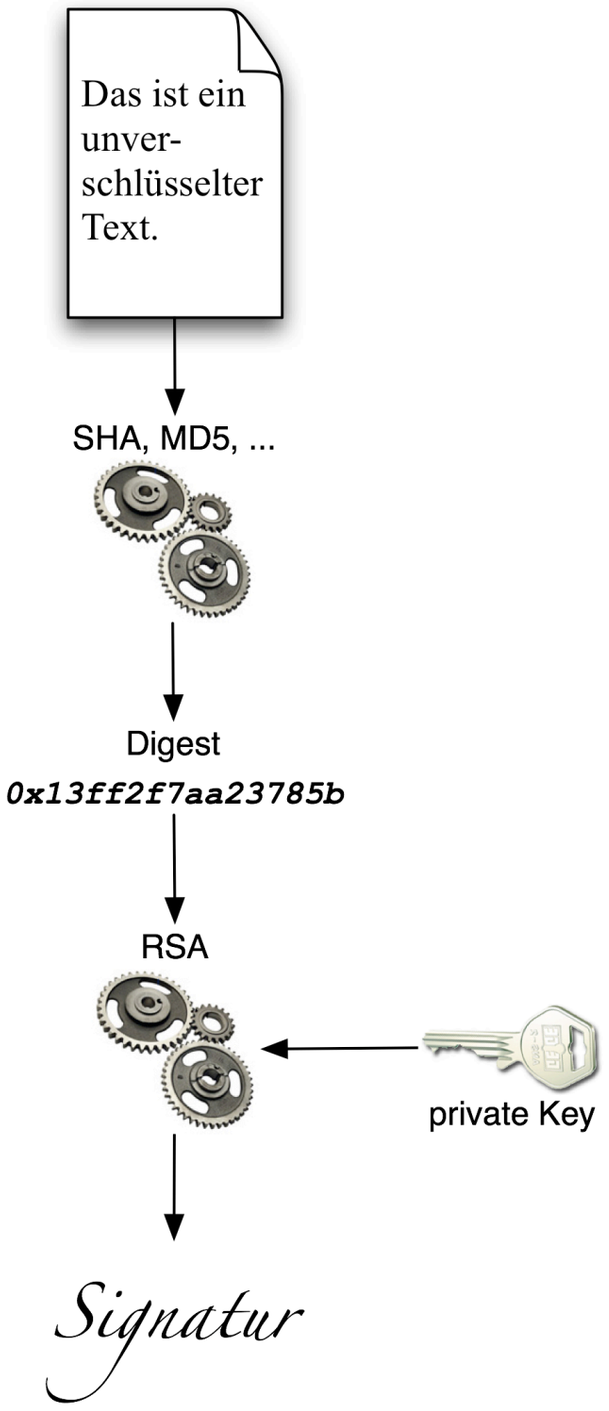
- Asymmetrische Verschlüsselung löst das Schlüsselverteilungsproblem.
- Bei symmetrischer Verschlüsselung ist es schwierig verschlüsselte Daten mit verschiedenen Leuten auszutauschen. Wenn sie denselben Schlüssel an mehrere Leute geben, wird das Verfahren unsicher.
- Asymmetrische Verschlüsselungsverfahren sind langsamer als symmetrische.

Digitale Signatur

Signatur erstellen

- Der Sender erstellt den Digest des Textes (SHA, MD5).
- Er verschlüsselt den Digest mit seinem privaten Schlüssel (RSA).
- Dann sendet er den Text und die Signatur.

Digitale Signatur

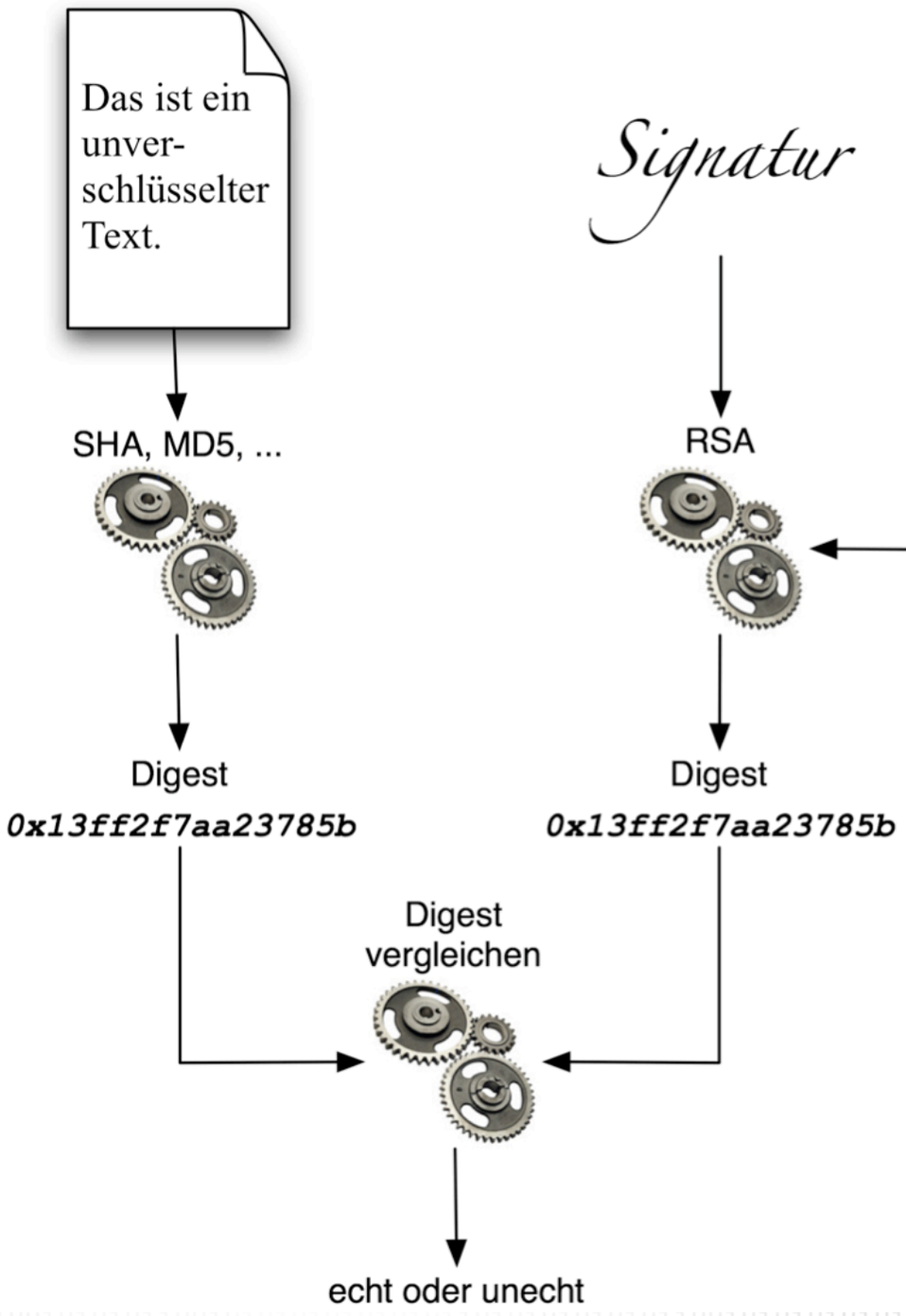


Digitale Signatur

Signatur verifizieren

- Der Empfänger bildet den Digest des Textes nach demselben Verfahren des Senders.
- Er entschlüsselt die Signatur mit dem öffentlichen Schlüssel des Senders.
- Er vergleicht den Digest, den er selbst erstellt hat mit dem aus der Signatur des Senders.

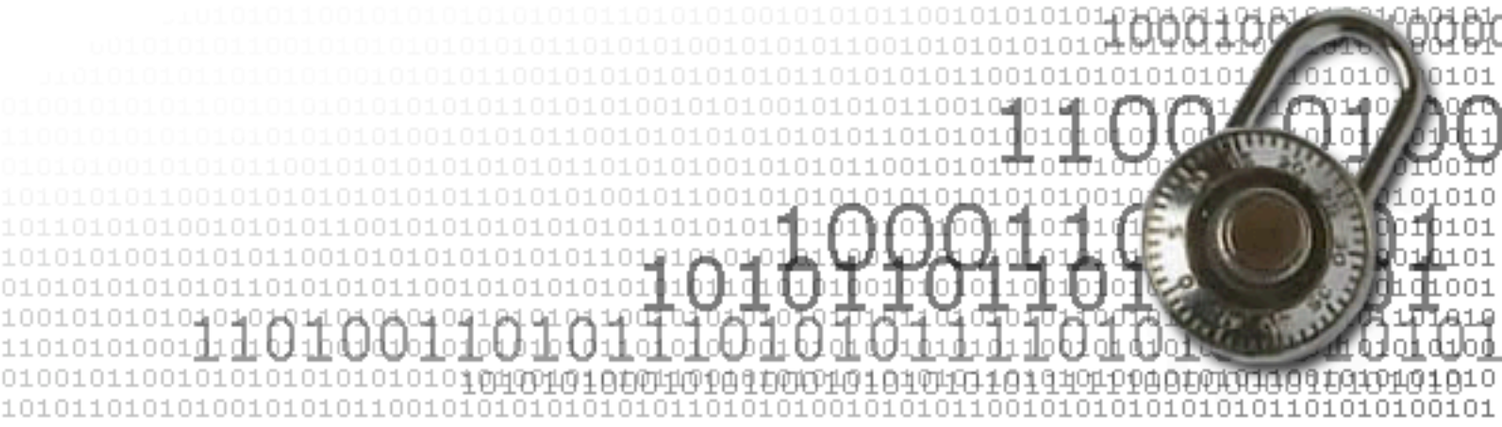
Digitale Signatur



Digitale Signatur

RSA

Dieser Algorithmus aus dem Jahr 1977 ist nach seinen Erfindern Ronald L. Rivest, Adi Shamir und Leonard Adleman benannt.



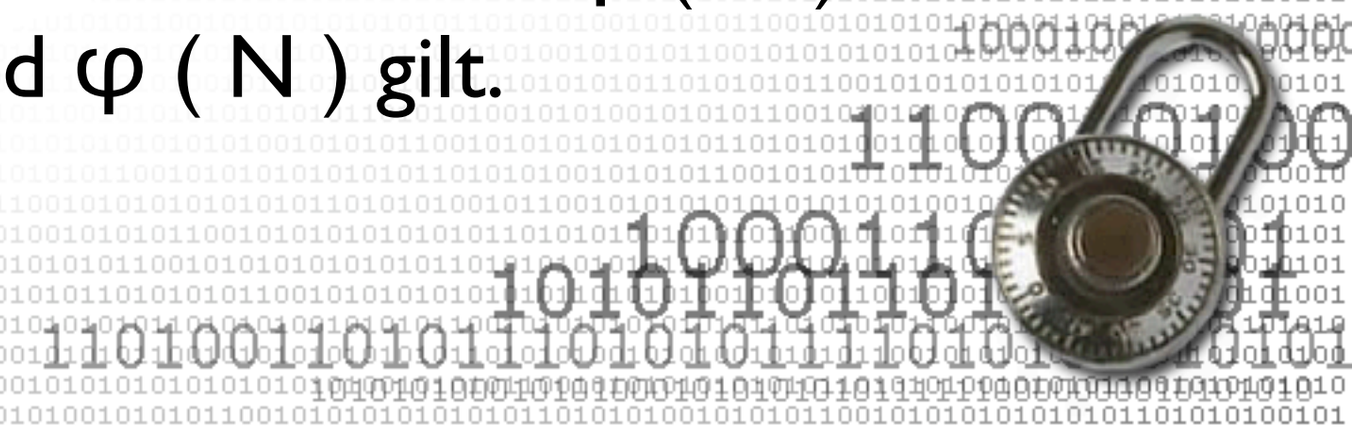
Digitale Signatur

Der Algorithmus basiert auf einer "Einwegfunktion" mit "Falltür". Unter einer "Einwegfunktion" versteht man eine Funktion die sich in eine Richtung einfach berechnen läst, in die andere Richtung aber nur sehr schwierig (z.B.: Zahnpasta aus der Tube drücken ;-). Man kann zwei Zahlen sehr leicht miteinander Multiplizieren und erhält ein Ergebnis. Dieses Ergebnis kann man aber nur sehr schwer wieder in seine Faktoren zerlegen. Mit Hilfe einer "Falltür" d.h. einer Zusatzinformation kann man die schwierige Richtung auch berechnen, z.B.: Brief in einen Briefkasten werfen (der Briefträger kann ihn leicht wieder herausholen).

Digitale Signatur

Schlüsselgenerierung:

- Wähle zufällig und stochastisch unabhängig zwei Primzahlen $p \neq q$, die etwa gleich lang sein sollten und berechne deren Produkt $N = p \cdot q$.
- Berechne $\varphi(N) = (p - 1) \cdot (q - 1)$, wobei φ für die Eulersche φ-Funktion steht.
- Wähle eine Zahl $e > 1$, die teilerfremd zu $\varphi(N)$ ist.
- Berechne die Zahl d so, dass das Produkt $e \cdot d$ kongruent 1 bezüglich des "Modulus" $\varphi(N)$ ist, dass also $e \cdot d \equiv 1 \bmod \varphi(N)$ gilt.



Digitale Signatur

- Die Zahlen N und e werden veröffentlicht (öffentlicher Schlüssel)
- d, p und q und damit auch φ (N) bilden den geheimen Schlüssel (secret key)

Digitale Signatur

Verschlüsseln von Nachrichten

Um eine Nachricht M zu verschlüsseln, verwendet der Absender die Formel

$$C \equiv M^e \bmod N$$

und erhält so aus dem Klartext M den Geheimtext C.

Digitale Signatur

Entschlüsseln von Nachrichten

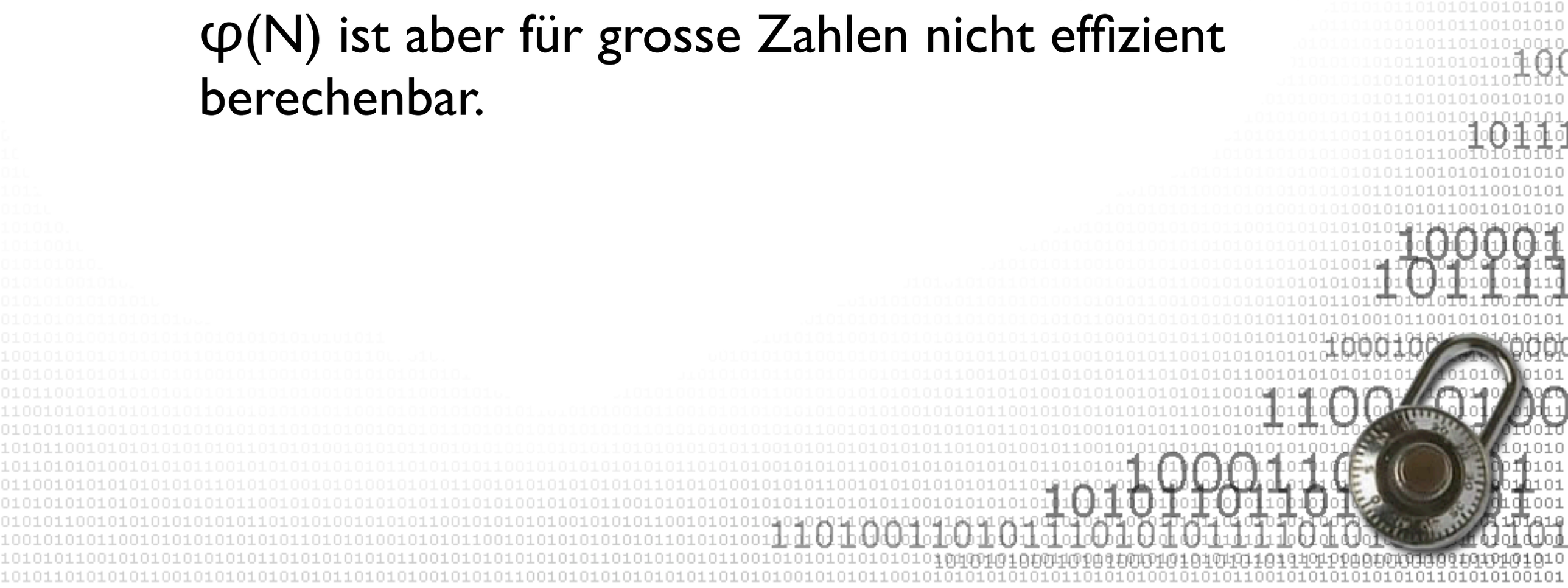
Der Geheimtext C kann durch modulare Exponentiation wieder entschlüsselt werden. Der Nachrichtenempfänger benutzt die Formel:

$$M \equiv C^d \bmod N$$

mit den nur ihm bekannten Werten d und N.

Digitale Signatur

Die Sicherheit basiert darauf, dass der Angreifer d nicht kennt. Um d zu berechnen benötigt er φ(N). φ(N) ist aber für grosse Zahlen nicht effizient berechenbar.



Secret Sharing

Secret Sharing

- Was ist Secret Sharing?

  Als Secret Sharing bezeichnet man das Verfahren, einen Schlüssel S in n Teilschlüssel s1,..,sn aufzuteilen.

- Wo wird Secret Sharing benutzt?

Secret Sharing

- Piraten haben ihre Schatzkarten zerschnitten und untereinander aufgeteilt, dadurch waren Sie nur gemeinsam in der Lage den Schatz wieder zu finden.

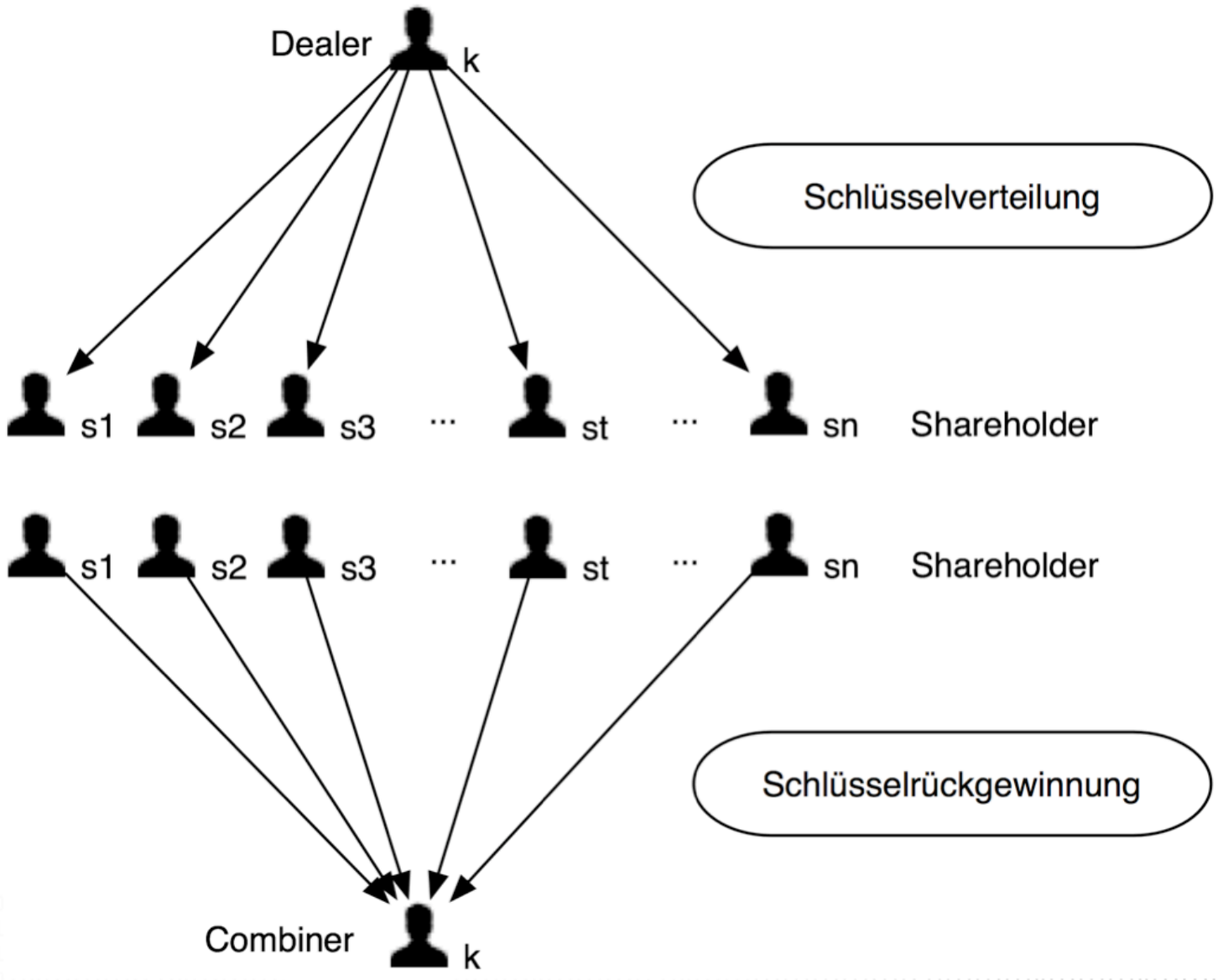
Secret Sharing

weiter Beispiele:

- Eine Atombombe kann nur von mehreren Leuten gleichzeitig gestartet werden.
- Einen wichtigen Tresor in einer Bank können nur zwei Angestellte gleichzeitig öffnen.

Secret Sharing

Modell:



Secret Sharing

Es gelten folgende Voraussetzungen:

- Der Dealer ist eine vertrauenswürdige Institution.
- Jeder Teilnehmer hat einen sicheren und geheimen Kommunikationsweg, über diesen gelangen die Keys vom Dealer zu den Shareholdern.

Secret Sharing

Es gibt Verfahren mit folgenden Eigenschaften:

- perfektes Secret Sharing:
  Als perfektes Secret Sharing bezeichnet man ein Verfahren, bei dem es nicht möglich ist mit weniger als t-Teilschlüsseln auf den Schlüssel S zu schließen.
- robustes Secret Sharing:
  Unter robust versteht man ein Verfahren, welches selbst durch verfälschte oder zurückgehaltene Teilschlüssel noch sicher bleibt.
- demokratisches oder autokratisches Verfahren:
  Damit bezeichnet man ein Verfahren ohne oder mit Geber.

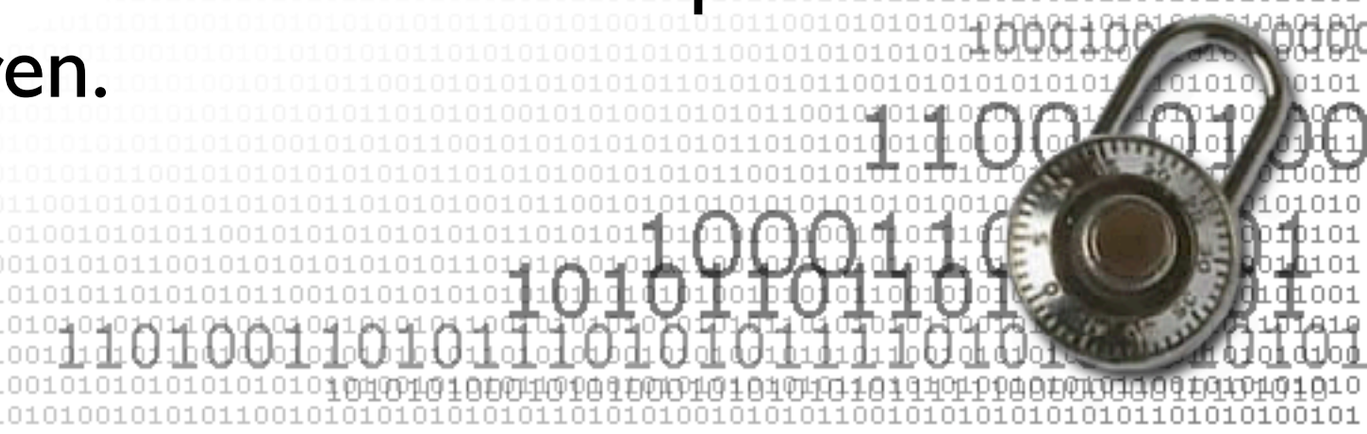
Secret Sharing

Threshold Secret Sharing bzw. Schwellwert Geheimnis Aufteilung

- Der Schwellwert bezeichnet die Anzahl an Shareholdern, ab der man den Schlüssel wieder herstellen kann.

- Es gibt Verfahren bei denen man alle Teilschlüssel braucht. Diese nennt man (n,n)-Verfahren.

- Es gibt aber auch Verfahren, bei denen ein kleinerer Schwellwert t ausreicht. Hier spricht man von (t,n)-Verfahren.



Secret Sharing

Ein einfaches Beispiel für ein (n,n)-Verfahren

- Der Schlüssel ist eine Summe und die Teilschlüssel bestehen aus den Summanden.
- Man kann den Schlüssel wieder herstellen, indem man die Summanden addiert.

Secret Sharing

- Dieses Verfahren ist perfekt, weil man mit t-1 Summanden nicht auf die richtige Summe kommt.

- Das Verfahren ist aber nicht robust, weil einer der Shareholder einen falschen Summanden angeben kann. Dadurch kommt man zu einem falschen Schlüssel. Für den betrügerischen Shareholder ist es nun ein Leichtes, aus der falschen Summe die richtige zu berechnen.

Secret Sharing

Aus diesem (n,n)-Verfahren kann man leicht ein (t,n)-Verfahren machen.

Die einzelnen Shareholder teilen ihren Summanden in j weitere Summanden auf und verteilen diese j Summanden auf j Shareholder, denen Sie vertrauen.

Secret Sharing

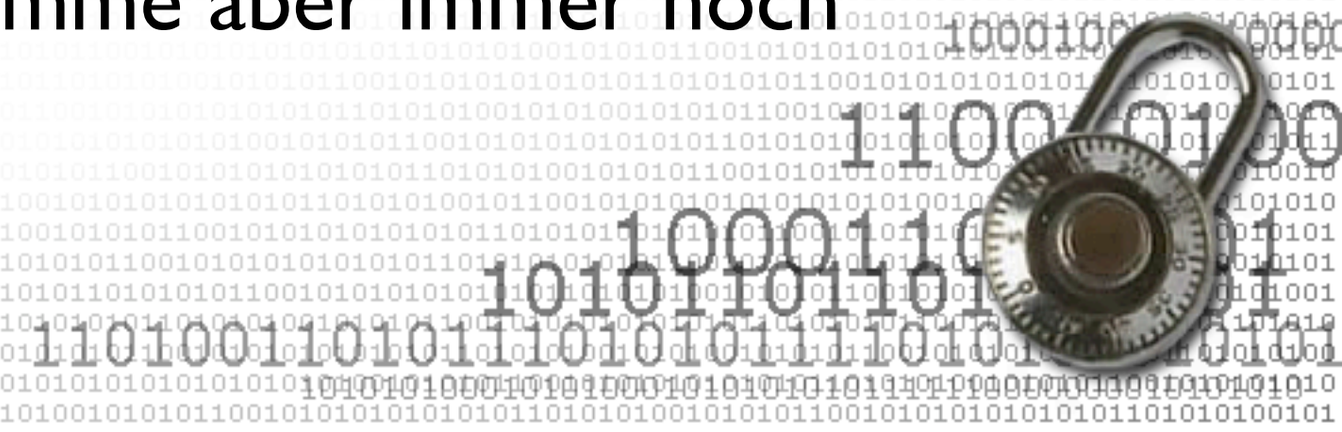
Beispiel für ein (2,3)-Verfahren:

S=30, s1=8, s2=13, s3=9 => 8+13+9=30

Der Shareholder mit dem Teilschlüssel s2 teilt seinen Summanden in 6 und 7 auf und gibt jeweils einen von den beiden an die anderen beiden Shareholder. Diese können nun durch die Addition ihrer beiden Schlüssel und den Schlüsseln von Shareholder 2 die Summe bilden:

8+9+6+7=30

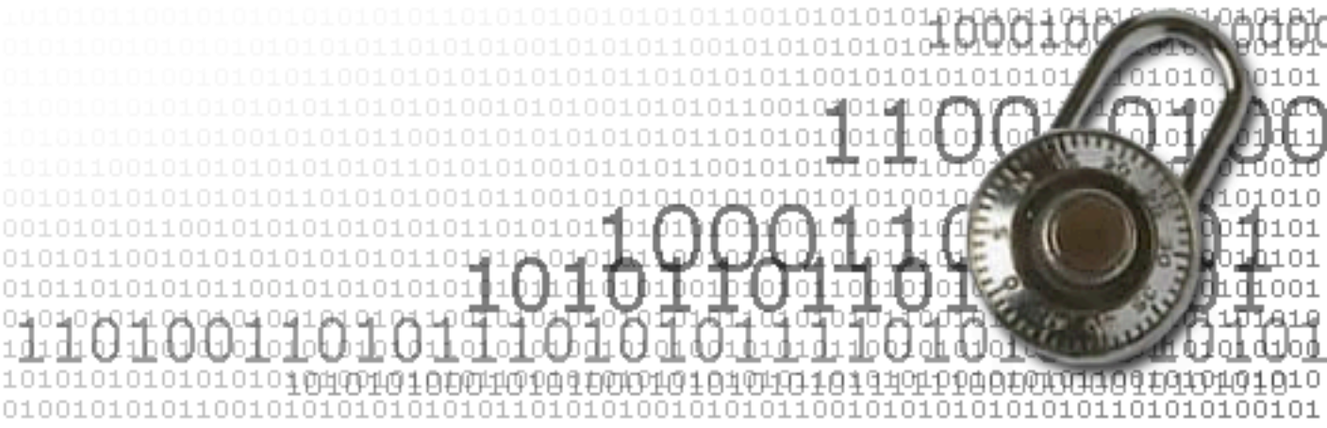
Einer alleine kann die Summe aber immer noch nicht bilden.



Secret Sharing

Das Blakley Schema

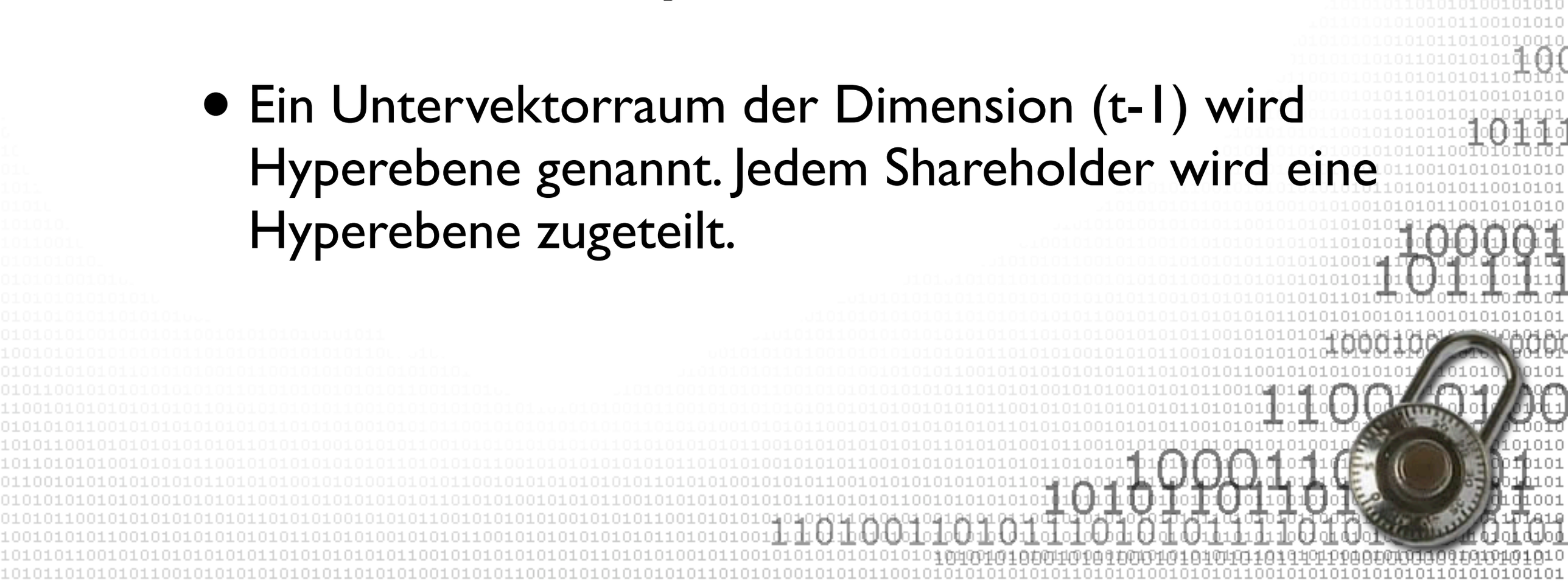
Blakley's im Jahre 1979 vorgestelltes Schwellwert Schema beruht auf geometrischer Konstruktion und ist ein (t,n) Schema.



Secret Sharing

Schlüsselverteilung:

- Der Dealer wählt einen zufälligen Punkt in einem Vektorraum der Dimension t über dem Ganzzahlkörper $\mathbb{Z}_P$.
- Ein Untervektorraum der Dimension (t-1) wird Hyperebene genannt. Jedem Shareholder wird eine Hyperebene zugeteilt.

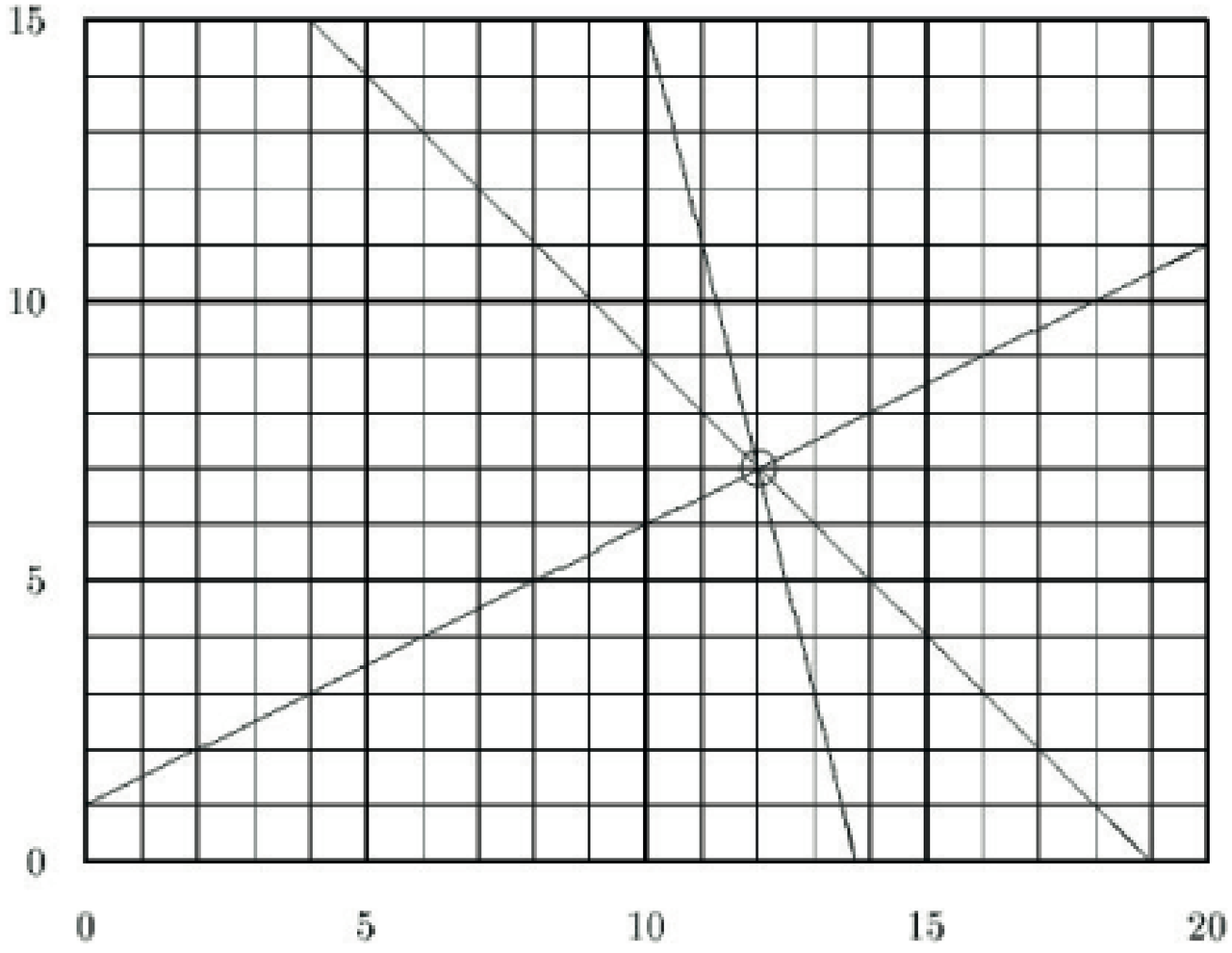


Secret Sharing

- Man braucht zur Rekonstruktion des Schlüssels genau t Hyperebenen.
- Dieses Verfahren ist nicht perfekt. Man kann zwar nicht mit t-1 Teilschlüsseln auf das Ergebnis schließen, aber man kann es enorm einschränken, denn der Punkt muss irgendwo auf den t-1 Hyperebenen liegen.

Secret Sharing

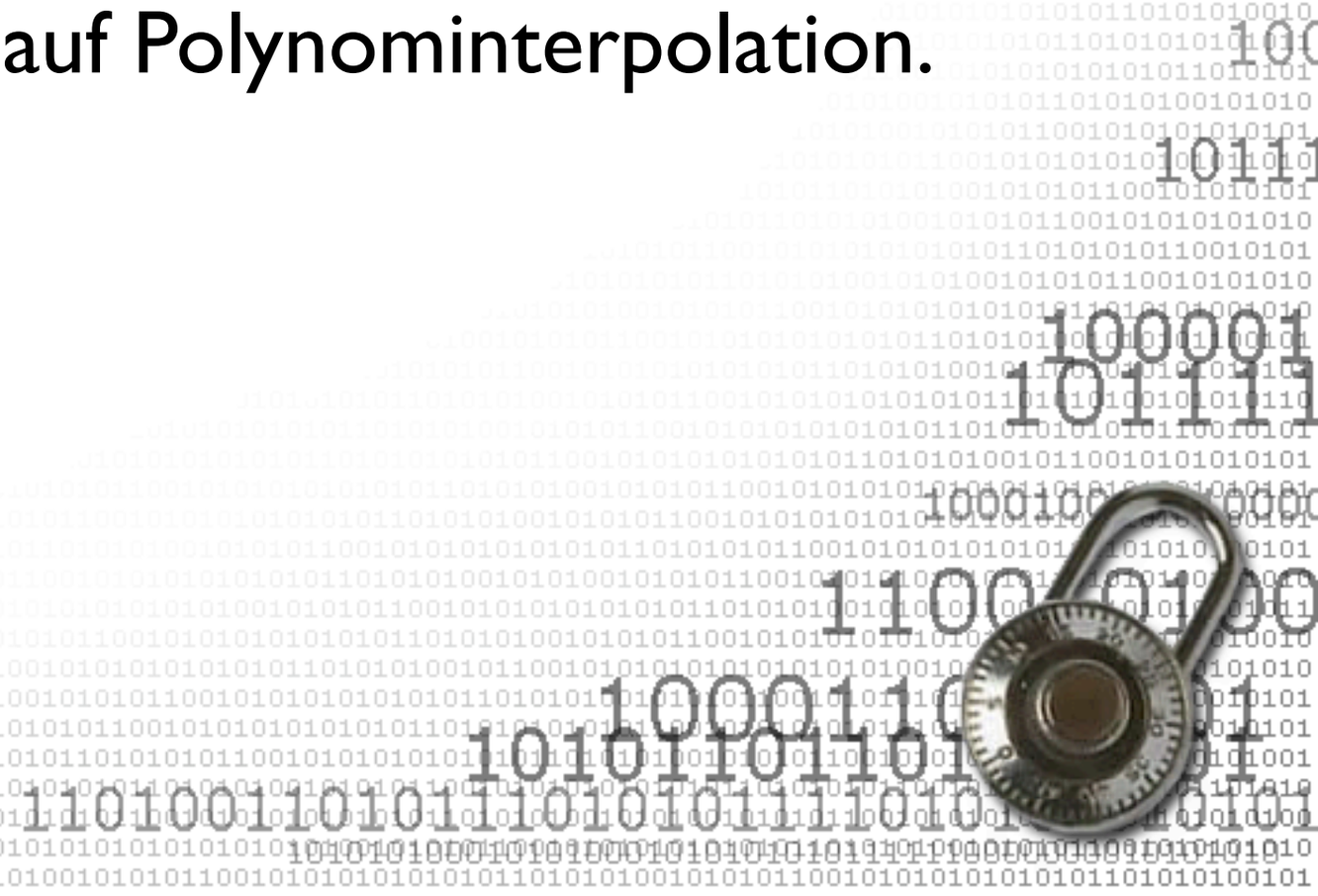
Beispiel für ein (2, 3) Blakley Schema:
Das Geheimnis ist der Punkt (12,7). Zur Rekonstruktion des Geheimnisses sind mindestens 2 Geraden welche sich in diesem Punkt schneiden notwendig.



Secret Sharing

Das Shamir Schema

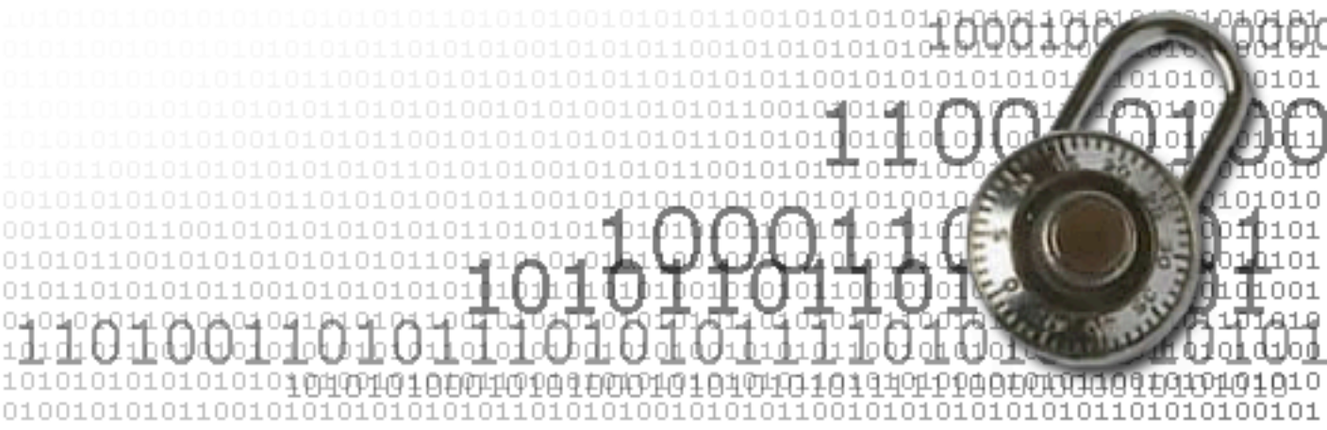
Das 1979 von Adi Shamir vorgestellte Schema ist ein (t,n) Schema und basiert auf Polynominterpolation.



Secret Sharing

Schlüsselverteilung:

- Der Dealer wählt eine Primzahl p für die gilt p>k und p>n
- Der Dealer wählt n verschiedene Punkte x_i
- Der Dealer bestimmt ein Polynom P(x) vom Grad t-1 mit Koeffizienten a_i aus Z
- Das Geheimnis ist k=P(0)
  Die Shares sind $s_i=P(x_i)$ und werden an die n Shareholder verteilt.



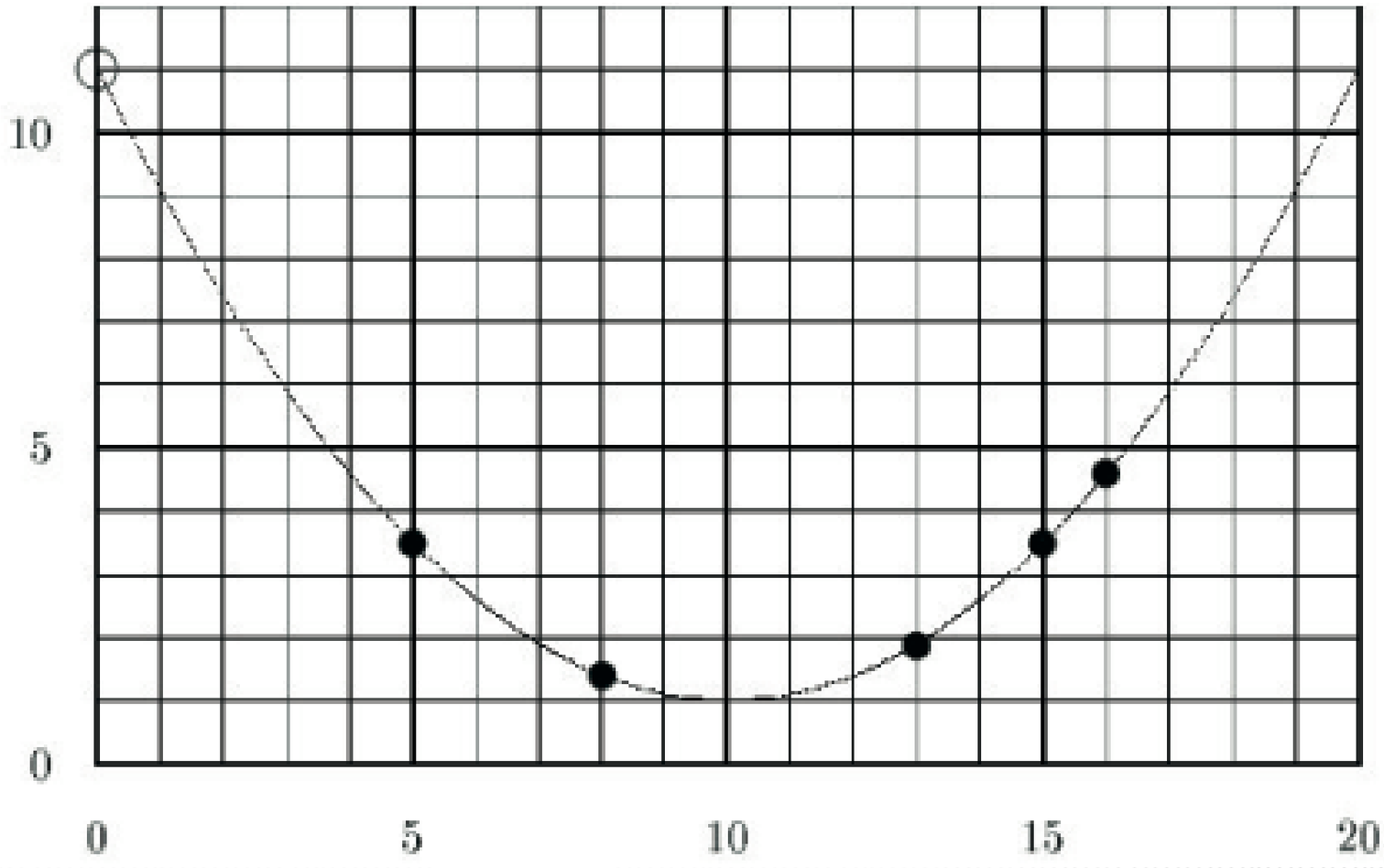
Secret Sharing

Schlüsselrückgewinnung:

- t der n Shareholder geben ihre Shares dem Combiner
- Der Combiner versucht das Polynom wieder herzustellen

Secret Sharing

Mit Hilfe der Lagrange Interpolation läst sich k aus t Punkten bei einem Polynom vom Grad t-1 berechnen.



Secret Sharing

- Wenn man t oder mehr Shareholder hat, kann man den Schlüssel wieder rekonstruieren, daher gilt das Verfahren als perfekt.

- Wenn ein Shareholder aber einen falschen Punkt angibt, dann kommt es zu einem falschen Schlüssel, das Schema ist also nicht robust.
Um das Verfahren robust zu machen kann man den einzelnen Shareholdern noch einen Checkvektor zum überprüfen der Punkte der anderen Shareholder mitgeben. Dieser darf aber nicht auf den richtigen Schlüssel schließen lassen.

ENDE