Asymmetrische Chiffrierung

Christian Koch
ckoch@et.htwk-leipzig.de

22. Februar 2000

Protokoll

Ein Protokoll dient der Durchführung einer bestimmten Aufgabe und besteht aus einer Folge von Aktionen, an denen zwei oder mehr Parteien beteiligt sind.

Protokoll

Ein Protokoll dient der Durchführung einer bestimmten Aufgabe und besteht aus einer Folge von Aktionen, an denen zwei oder mehr Parteien beteiligt sind.

- Protokoll allen Beteiligten im voraus bekannt

Protokoll

Ein Protokoll dient der Durchführung einer bestimmten Aufgabe und besteht aus einer Folge von Aktionen, an denen zwei oder mehr Parteien beteiligt sind.

- Protokoll allen Beteiligten im voraus bekannt
- Beteiligte müssen sich an vereinbarte Regeln halten

Protokoll

Ein Protokoll dient der Durchführung einer bestimmten Aufgabe und besteht aus einer Folge von Aktionen, an denen zwei oder mehr Parteien beteiligt sind.

- Protokoll allen Beteiligten im voraus bekannt
- Beteiligte müssen sich an vereinbarte Regeln halten
- Protokoll eindeutig, klare Definition der Aktionen

Protokoll

Ein Protokoll dient der Durchführung einer bestimmten Aufgabe und besteht aus einer Folge von Aktionen, an denen zwei oder mehr Parteien beteiligt sind.

- Protokoll allen Beteiligten im voraus bekannt
- Beteiligte müssen sich an vereinbarte Regeln halten
- Protokoll eindeutig, klare Definition der Aktionen
- Vollständigkeit des Protokolls, jeder denkbaren Situation muß Aktion zugeordnet sein

Kryptosystem

$P \ldots$ Menge aller Klartexte m
$C \ldots$ Menge aller Geheimtexte c

Kryptosystem

$P \ldots$	Menge aller Klartexte m
$C \ldots$	Menge aller Geheimtexte c
$K_1 \ldots$	Menge aller Chiffrierschlüssel k_1
$K_2 \ldots$	Menge aller Dechiffrierschlüssel k_2

Kryptosystem

$P \ldots$	Menge aller Klartexte m
$C \ldots$	Menge aller Geheimtexte c
$K_1 \ldots$	Menge aller Chiffrierschlüssel k_1
$K_2 \ldots$	Menge aller Dechiffrierschlüssel k_2
$K \ldots$	Menge aller Schlüsselpaare (k_1, k_2)

Kryptosystem

$P \ldots$	Menge aller Klartexte m
$C \ldots$	Menge aller Geheimtexte c
$K_1 \ldots$	Menge aller Chiffrierschlüssel k_1
$K_2 \ldots$	Menge aller Dechiffrierschlüssel k_2
$K \ldots$	Menge aller Schlüsselpaare (k_1, k_2)
$E := \{E_{k_1} \mid k_1 \in K_1\} \ldots$	Familie von Chiffrieralgorithmen
$D := \{D_{k_2} \mid k_2 \in K_2\} \ldots$	Familie von Dechiffrieralgorithmen

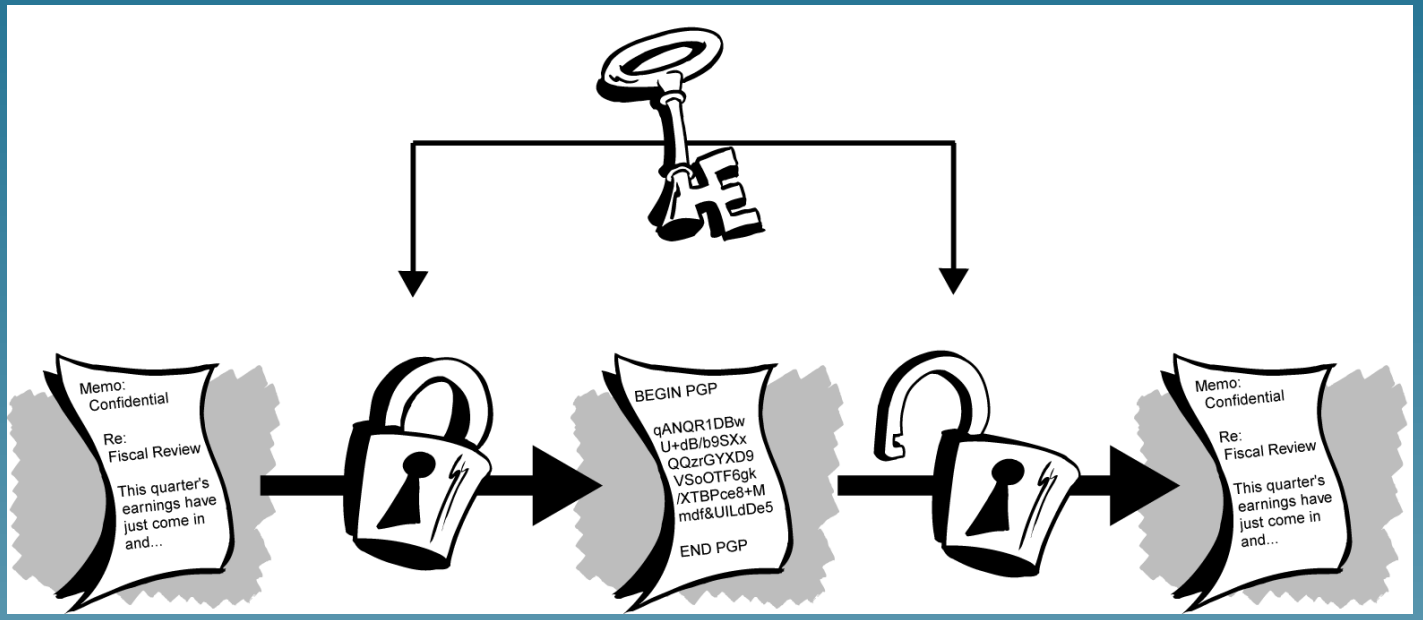
Kryptosystem

$P \ldots$	Menge aller Klartexte m
$C \ldots$	Menge aller Geheimtexte c
$K_1 \ldots$	Menge aller Chiffrierschlüssel k_1
$K_2 \ldots$	Menge aller Dechiffrierschlüssel k_2
$K \ldots$	Menge aller Schlüsselpaare (k_1, k_2)
$E := \{E_{k_1} \mid k_1 \in K_1\} \ldots$	Familie von Chiffrieralgorithmen
$D := \{D_{k_2} \mid k_2 \in K_2\} \ldots$	Familie von Dechiffrieralgorithmen

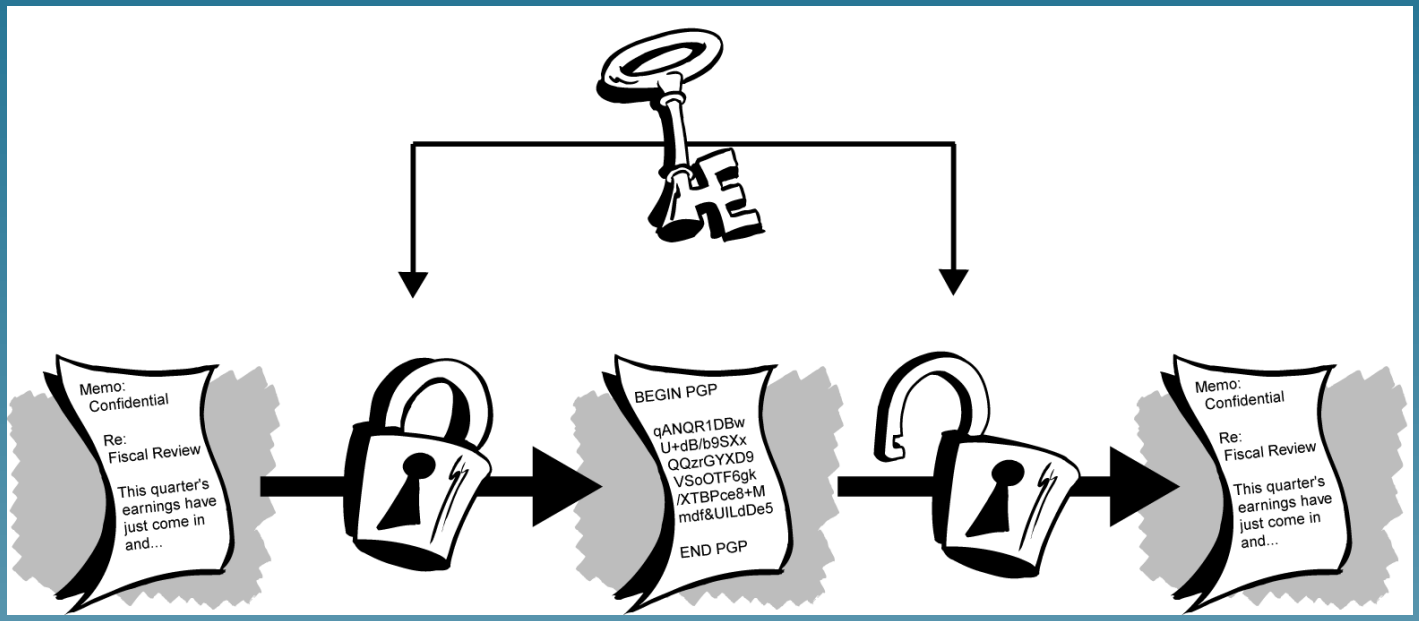
Für jedes $m \in P$ und jedes Paar $(k_1, k_2) \in K$ muß gelten:

$$D_{k_2}(E_{k_1}(m)) = m$$

Prinzip der symmetrischen Chiffrierung



Prinzip der symmetrischen Chiffrierung



Analogon: Tresor, in den nur diejenigen etwas hineinlegen oder herausholen können, die den passenden Schlüssel besitzen

Symmetrisches Kryptosystem

Ausgehend vom allgemeinen Modell bezeichnet man ein Kryptosystem, bei dem sich k_1 aus k_2 und umgekehrt direkt ableiten läßt, als symmetrisch. Meist ist sogar $k_1 = k_2$.

Symmetrisches Kryptosystem

Ausgehend vom allgemeinen Modell bezeichnet man ein Kryptosystem, bei dem sich k_1 aus k_2 und umgekehrt direkt ableiten läßt, als symmetrisch. Meist ist sogar $k_1 = k_2$.

Das Schlüsselpaar (k_1, k_2) muß über einen sicheren Kanal zwischen beiden Parteien übertragen und geheimgehalten werden. Dieses Paar kann nur zur Kommunikation zwischen diesen beiden Parteien benutzt werden. Sollen zwischen n Parteien sichere Kommunikationskanäle aufgebaut werden, so sind $n(n-1)/2$ Schlüsselpaare erforderlich $\rightarrow O(n^2)$.

Symmetrisches Kryptosystem

Ausgehend vom allgemeinen Modell bezeichnet man ein Kryptosystem, bei dem sich k_1 aus k_2 und umgekehrt direkt ableiten läßt, als symmetrisch. Meist ist sogar $k_1 = k_2$.

Das Schlüsselpaar (k_1, k_2) muß über einen sicheren Kanal zwischen beiden Parteien übertragen und geheimgehalten werden. Dieses Paar kann nur zur Kommunikation zwischen diesen beiden Parteien benutzt werden. Sollen zwischen n Parteien sichere Kommunikationskanäle aufgebaut werden, so sind $n(n-1)/2$ Schlüsselpaare erforderlich $\rightarrow O(n^2)$.

Bekanntestes Beispiel für die Anwendung dieser Art Chiffrierung ist die CAESAR-Chiffre.

Durchbruch

Durchbruch

Im Jahre 1976 haben Whitfield Diffie und Martin Hellman mit ihrer berühmten Arbeit „New Directions in Cryptography“ eine völlig neuartige Idee entwickelt, welche eine sichere Datenübermittlung ermöglicht, ohne daß ein geheimer Schlüssel ausgetauscht werden muß.

Durchbruch

Im Jahre 1976 haben Whitfield Diffie und Martin Hellman mit ihrer berühmten Arbeit „New Directions in Cryptography“ eine völlig neuartige Idee entwickelt, welche eine sichere Datenübermittlung ermöglicht, ohne daß ein geheimer Schlüssel ausgetauscht werden muß.

Der Grundgedanke des neuen Konzeptes dabei war, daß Schlüssel paarweise, quasi komplementär, auftreten können, wobei es praktisch unmöglich ist, einen Schlüssel aus dem anderen zu bestimmen. Zur Chiffrierung werden sog. Falltürfunktionen mit Hintertür benutzt.

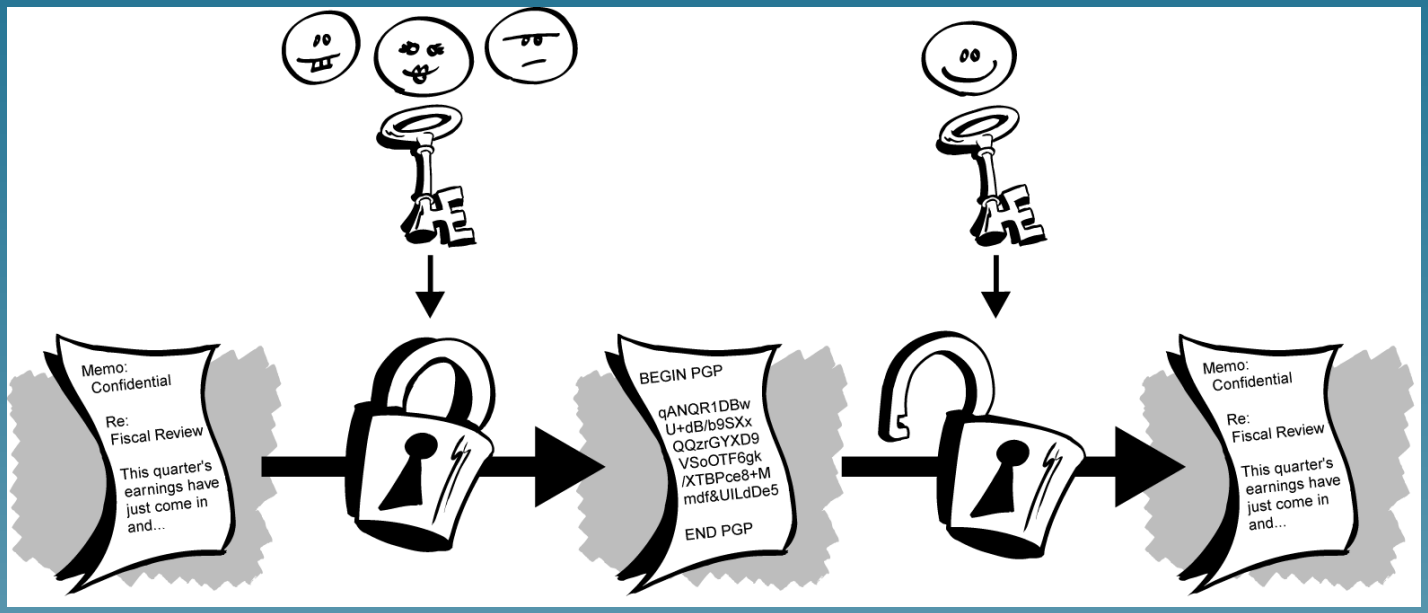
Durchbruch

Im Jahre 1976 haben Whitfield Diffie und Martin Hellman mit ihrer berühmten Arbeit „New Directions in Cryptography“ eine völlig neuartige Idee entwickelt, welche eine sichere Datenübermittlung ermöglicht, ohne daß ein geheimer Schlüssel ausgetauscht werden muß.

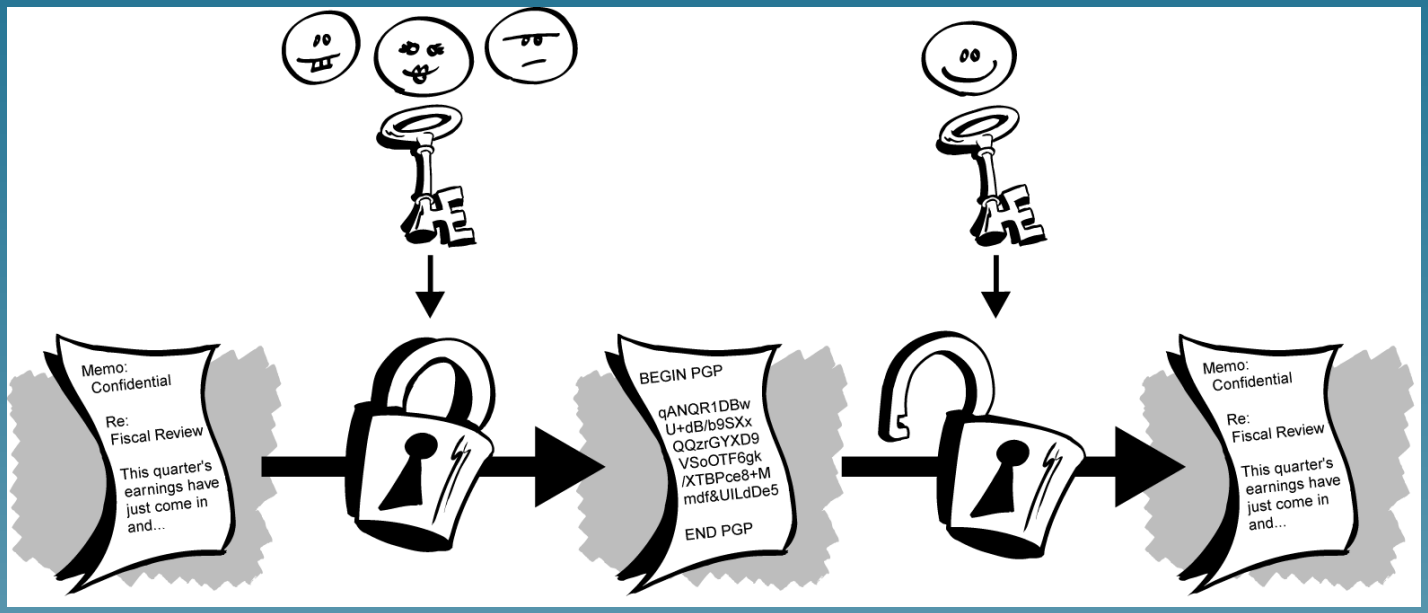
Der Grundgedanke des neuen Konzeptes dabei war, daß Schlüssel paarweise, quasi komplementär, auftreten können, wobei es praktisch unmöglich ist, einen Schlüssel aus dem anderen zu bestimmen. Zur Chiffrierung werden sog. Falltürfunktionen mit Hintertür benutzt.

$\Rightarrow$ asymmetrische Chiffrierung oder auch public-key cryptography

Prinzip der asymmetrischen Chiffrierung



Prinzip der asymmetrischen Chiffrierung



Analogon: Briefkasten, in den jeder etwas einwerfen kann, aber nur derjenige, der den richtigen Schlüssel besitzt, kann es wieder herausholen

Asymmetrisches Kryptosystem

Ausgehend vom allgemeinen Modell bezeichnet man ein Kryptosystem, bei dem sich k_2 aus k_1 praktisch nicht berechnen läßt, als asymmetrisch.

Asymmetrisches Kryptosystem

Ausgehend vom allgemeinen Modell bezeichnet man ein Kryptosystem, bei dem sich k_2 aus k_1 praktisch nicht berechnen läßt, als asymmetrisch.

Jede Partei besitzt nur ein Schlüsselpaar (k_1, k_2). k_1 ist der sog. öffentliche Schlüssel und jedem zugänglich. k_2 ist der private Schlüssel und ist nur dem Eigentümer bekannt.

Asymmetrisches Kryptosystem

Ausgehend vom allgemeinen Modell bezeichnet man ein Kryptosystem, bei dem sich k_2 aus k_1 praktisch nicht berechnen läßt, als asymmetrisch.

Jede Partei besitzt nur ein Schlüsselpaar (k_1, k_2). k_1 ist der sog. öffentliche Schlüssel und jedem zugänglich. k_2 ist der private Schlüssel und ist nur dem Eigentümer bekannt. Um zwischen n Parteien sichere Kommunikationskanäle aufzubauen, sind n Schlüsselpaare erforderlich $\rightarrow O(n)$.

Asymmetrisches Kryptosystem

Ausgehend vom allgemeinen Modell bezeichnet man ein Kryptosystem, bei dem sich k_2 aus k_1 praktisch nicht berechnen läßt, als asymmetrisch.

Jede Partei besitzt nur ein Schlüsselpaar (k_1, k_2). k_1 ist der sog. öffentliche Schlüssel und jedem zugänglich. k_2 ist der private Schlüssel und ist nur dem Eigentümer bekannt. Um zwischen n Parteien sichere Kommunikationskanäle aufzubauen, sind n Schlüsselpaare erforderlich $\rightarrow O(n)$.

Ein Anwendungsbeispiel ist PGP, ein Programm, was hauptsächlich zur Verschlüsselung von E-Mail genutzt wird.

RSA

- RIVEST, SHAMIR und ADLEMAN: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. 1978.

RSA

- RIVEST, SHAMIR und ADLEMAN: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. 1978.
- beruht auf dem Problem der Faktorisierung großer Zahlen

RSA

- RIVEST, SHAMIR und ADLEMAN: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. 1978.
- beruht auf dem Problem der Faktorisierung großer Zahlen
- $p \neq q$ ... Primzahlen
- $n := pq$

RSA

- RIVEST, SHAMIR und ADLEMAN: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. 1978.
- beruht auf dem Problem der Faktorisierung großer Zahlen
- $p \neq q$ ... Primzahlen
- $n := pq$
- $\phi(n) = (p-1)(q-1)$ Eulersche ϕ-Funktion

RSA

- RIVEST, SHAMIR und ADLEMAN: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. 1978.
- beruht auf dem Problem der Faktorisierung großer Zahlen
- $p \neq q$ ... Primzahlen
- $n := pq$
- $\phi(n) = (p-1)(q-1)$ Eulersche ϕ-Funktion
- beliebige Zahl e mit $[1 < e < n] \wedge [\gcd(e, \phi(n)) = 1]$
- $d := e^{-1} \bmod \phi(n) \quad \Longleftrightarrow \quad de \equiv 1 \pmod{\phi(n)}$

RSA

- RIVEST, SHAMIR und ADLEMAN: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. 1978.
- beruht auf dem Problem der Faktorisierung großer Zahlen
- $p \neq q$ ... Primzahlen
- $n := pq$
- $\phi(n) = (p-1)(q-1)$ Eulersche ϕ-Funktion
- beliebige Zahl e mit $[1 < e < n] \wedge [\gcd(e, \phi(n)) = 1]$
- $d := e^{-1} \bmod \phi(n) \quad \Longleftrightarrow \quad de \equiv 1 \pmod{\phi(n)}$

- $K_1 := \{(n, e)\}$ und $K_2 := \{(n, d)\}$

- $K_1 := \{(n, e)\}$ und $K_2 := \{(n, d)\}$
- $0 \leq m < n$
- $E_{k_1}(m) := m^e \bmod n$

- $K_1 := \{(n, e)\}$ und $K_2 := \{(n, d)\}$
- $0 \leq m < n$
- $E_{k_1}(m) := m^e \bmod n$
- $D_{k_2}(c) := c^d \bmod n$

Praxis

- Modul n sollte Länge von mindestens 768 Bit aufweisen

Praxis

- Modul n sollte Länge von mindestens 768 Bit aufweisen
- square-and-multiply: Ausnutzung der Assoziativität der Modulo-Operation läßt Zwischenergebnisse dual maximal doppelt so lang wie der Modul werden

$$m^{17} \bmod n \quad = \quad m^{2^4+2^0} \bmod n$$

Praxis

- Modul n sollte Länge von mindestens 768 Bit aufweisen
- square-and-multiply: Ausnutzung der Assoziativität der Modulo-Operation läßt Zwischenergebnisse dual maximal doppelt so lang wie der Modul werden

$$\begin{aligned} m^{17} \bmod n &= m^{2^4+2^0} \bmod n \\ &= (((((m^2)^2)^2)^2) \cdot m) \bmod n \end{aligned}$$

Praxis

- Modul n sollte Länge von mindestens 768 Bit aufweisen
- square-and-multiply: Ausnutzung der Assoziativität der Modulo-Operation läßt Zwischenergebnisse dual maximal doppelt so lang wie der Modul werden

$$
\begin{aligned}
m^{17} \bmod n &= m^{2^4+2^0} \bmod n \\
&= (((((m^2)^2)^2)^2) \cdot m) \bmod n \\
&= (((((m^2 \bmod n)^2 \bmod n)^2 \bmod n)^2 \bmod n) \cdot m) \bmod n
\end{aligned}
$$

Praxis

- Modul n sollte Länge von mindestens 768 Bit aufweisen
- square-and-multiply: Ausnutzung der Assoziativität der Modulo-Operation läßt Zwischenergebnisse dual maximal doppelt so lang wie der Modul werden

$$\begin{aligned} m^{17} \bmod n &= m^{2^4+2^0} \bmod n \\ &= (((((m^2)^2)^2)^2) \cdot m) \bmod n \\ &= (((((m^2 \bmod n)^2 \bmod n)^2 \bmod n)^2 \bmod n) \cdot m) \bmod n \end{aligned}$$

- Exponent e ist meist 3, 17 oder 65537 wegen geringer Anzahl von Multiplikationen bei der square-and-multiply-Methode; beeinflußt die Systemsicherheit nicht

Praxis

- Modul n sollte Länge von mindestens 768 Bit aufweisen
- square-and-multiply: Ausnutzung der Assoziativität der Modulo-Operation läßt Zwischenergebnisse dual maximal doppelt so lang wie der Modul werden

$$\begin{aligned} m^{17} \bmod n &= m^{2^4+2^0} \bmod n \\ &= (((((m^2)^2)^2)^2) \cdot m) \bmod n \\ &= (((((m^2 \bmod n)^2 \bmod n)^2 \bmod n)^2 \bmod n) \cdot m) \bmod n \end{aligned}$$

- Exponent e ist meist 3, 17 oder 65537 wegen geringer Anzahl von Multiplikationen bei der square-and-multiply-Methode; beeinflußt die Systemsicherheit nicht

Austausch einer geheimen Nachricht

Alice möchte Bob geheime Nachricht m mittels RSA übermitteln

Austausch einer geheimen Nachricht

Alice möchte Bob geheime Nachricht m mittels RSA übermitteln

1. Bob erzeugt sich ein Schlüsselpaar

Austausch einer geheimen Nachricht

Alice möchte Bob geheime Nachricht m mittels RSA übermitteln

1. Bob erzeugt sich ein Schlüsselpaar

2. Bob speichert seinen öffentlichen Schlüssel (n, e) in frei zugänglicher Datei, z. B. `http://wwwkeys.pgp.net/`, zusammen mit Benutzerkennung $\rightarrow$ $\{(\text{„Bob“}, n, e), \ldots\}$

Austausch einer geheimen Nachricht

Alice möchte Bob geheime Nachricht m mittels RSA übermitteln

1. Bob erzeugt sich ein Schlüsselpaar

2. Bob speichert seinen öffentlichen Schlüssel (n, e) in frei zugänglicher Datei, z. B. `http://wwwkeys.pgp.net/`, zusammen mit Benutzerkennung $\rightarrow$ $\{(\text{„Bob“}, n, e), \ldots\}$

3. Alice lädt den Schlüssel mit der Benutzerkennung „Bob“ aus der öffentlichen Datei und verschlüsselt mit diesem ihre Nachricht m

Austausch einer geheimen Nachricht

Alice möchte Bob geheime Nachricht m mittels RSA übermitteln

1. Bob erzeugt sich ein Schlüsselpaar

2. Bob speichert seinen öffentlichen Schlüssel (n, e) in frei zugänglicher Datei, z. B. `http://wwwkeys.pgp.net/`, zusammen mit Benutzerkennung $\rightarrow$ $\{(\text{„Bob“}, n, e), \ldots\}$

3. Alice lädt den Schlüssel mit der Benutzerkennung „Bob“ aus der öffentlichen Datei und verschlüsselt mit diesem ihre Nachricht m

4. Alice übermittelt den Chiffretext an Bob

Austausch einer geheimen Nachricht

Alice möchte Bob geheime Nachricht m mittels RSA übermitteln

1. Bob erzeugt sich ein Schlüsselpaar
2. Bob speichert seinen öffentlichen Schlüssel (n, e) in frei zugänglicher Datei, z. B. `http://wwwkeys.pgp.net/`, zusammen mit Benutzerkennung $\rightarrow \{(\text{„Bob“}, n, e), \ldots\}$
3. Alice lädt den Schlüssel mit der Benutzerkennung „Bob“ aus der öffentlichen Datei und verschlüsselt mit diesem ihre Nachricht m
4. Alice übermittelt den Chiffretext an Bob
5. Bob dechiffriert den empfangenen Text mit seinem privaten Schlüssel (n, d)

Digitale Signatur

Alice möchte Nachricht m digital signieren und Bob diese verifizieren

Digitale Signatur

Alice möchte Nachricht m digital signieren und Bob diese verifizieren

1. Alice bildet den Hashwert $h(m)$, chiffriert diesen mit ihrem privaten Schlüssel und erhält Signatur $s := D(h(m))$

Digitale Signatur

Alice möchte Nachricht m digital signieren und Bob diese verifizieren

1. Alice bildet den Hashwert $h(m)$, chiffriert diesen mit ihrem privaten Schlüssel und erhält Signatur $s := D(h(m))$

2. Alice veröffentlicht („Alice“, m, s)

Digitale Signatur

Alice möchte Nachricht m digital signieren und Bob diese verifizieren

1. Alice bildet den Hashwert $h(m)$, chiffriert diesen mit ihrem privaten Schlüssel und erhält Signatur $s := D(h(m))$

2. Alice veröffentlicht („Alice“, m, s)

3. Bob lädt den Schlüssel „Alice“ aus der öffentlichen Datei

Digitale Signatur

Alice möchte Nachricht m digital signieren und Bob diese verifizieren

1. Alice bildet den Hashwert $h(m)$, chiffriert diesen mit ihrem privaten Schlüssel und erhält Signatur $s := D(h(m))$

2. Alice veröffentlicht („Alice“, m, s)

3. Bob lädt den Schlüssel „Alice“ aus der öffentlichen Datei

4. Bob dechiffriert s mit Hilfe des öffentlichen Schlüssels von Alice: $h'(m) := E(s)$

Digitale Signatur

Alice möchte Nachricht m digital signieren und Bob diese verifizieren

1. Alice bildet den Hashwert $h(m)$, chiffriert diesen mit ihrem privaten Schlüssel und erhält Signatur $s := D(h(m))$
2. Alice veröffentlicht („Alice“, m, s)
3. Bob lädt den Schlüssel „Alice“ aus der öffentlichen Datei
4. Bob dechiffriert s mit Hilfe des öffentlichen Schlüssels von Alice: $h'(m) := E(s)$
5. Bob bildet den Hashwert $h(m)$

Digitale Signatur

Alice möchte Nachricht m digital signieren und Bob diese verifizieren

1. Alice bildet den Hashwert $h(m)$, chiffriert diesen mit ihrem privaten Schlüssel und erhält Signatur $s := D(h(m))$
2. Alice veröffentlicht („Alice“, m, s)
3. Bob lädt den Schlüssel „Alice“ aus der öffentlichen Datei
4. Bob dechiffriert s mit Hilfe des öffentlichen Schlüssels von Alice: $h'(m) := E(s)$
5. Bob bildet den Hashwert $h(m)$
6. wenn gilt $h'(m) = h(m)$, so ist Unterschrift nicht gefälscht

Public-Key- vs. Secret-Key-Systeme

Vorteile

- für n Parteien nur n Schlüsselpaare nötig

Public-Key- vs. Secret-Key-Systeme

Vorteile

- für n Parteien nur n Schlüsselpaare nötig
- kein sicherer Kanal notwendig

Public-Key- vs. Secret-Key-Systeme

Vorteile

- für n Parteien nur n Schlüsselpaare nötig
- kein sicherer Kanal notwendig

Nachteile

- Problem der Authentizität der öffentlichen Schlüssel

Public-Key- vs. Secret-Key-Systeme

Vorteile

- für n Parteien nur n Schlüsselpaare nötig
- kein sicherer Kanal notwendig

Nachteile

- Problem der Authentizität der öffentlichen Schlüssel
- Schutz des geheimen Schlüssels

Public-Key- vs. Secret-Key-Systeme

Vorteile

- für n Parteien nur n Schlüsselpaare nötig
- kein sicherer Kanal notwendig

Nachteile

- Problem der Authentizität der öffentlichen Schlüssel
- Schutz des geheimen Schlüssels
- um Größenordnungen erhöhter Rechenaufwand als symmetrische Verfahren

Ich bedanke mich für die Aufmerksamkeit.