

Asymmetrische Kryptologie am Beispiel RSA entdecken

Screenshots mit CrypTool 1.4.30



www.cryptool.com
www.cryptool.de
www.cryptool.org
www.cryptool.pl

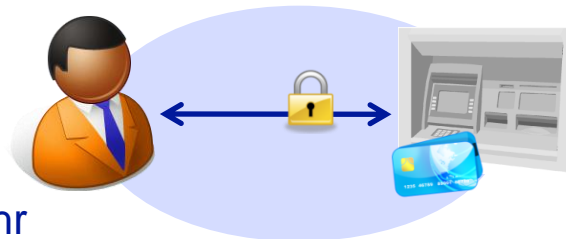
Übersicht – Aufbau der Folien

- Motivation und etwas Theorie: S. 3 ff.
- Aufbau des Workshops: S. 10 f.
- Komponenten der angewandten Kryptographie S. 12 ff.
 - Verschlüsselung, Schlüsselerzeugung
 - RSA
 - Hashverfahren
 - Digitale Signatur
 - Zertifikate, PKI
 - Hybride Verschlüsselung.

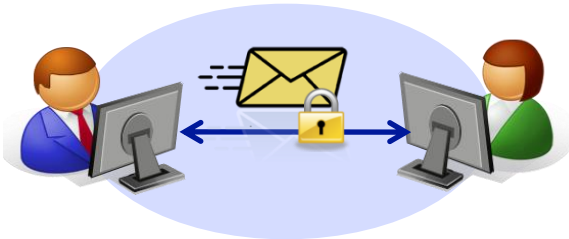
Kryptologie im Alltag

- Wo haben wir im Alltag mit verschlüsselten Daten zu tun?

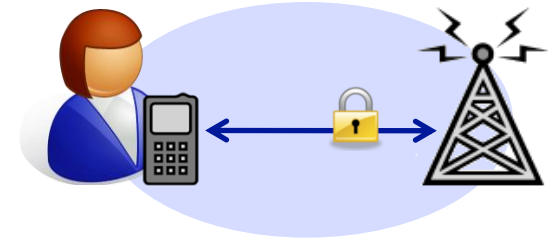
EC-Karten



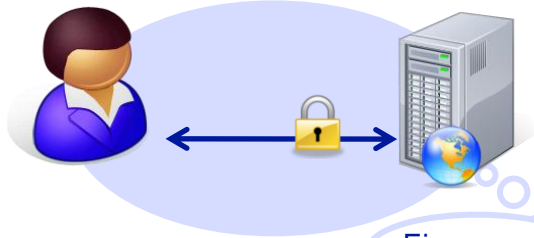
Verschlüsselter Email-Verkehr



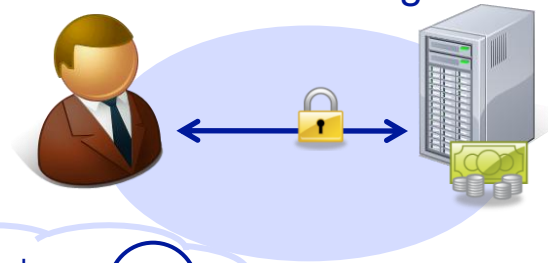
Handy-Verschlüsselung



Sichere Verbindung im Internet



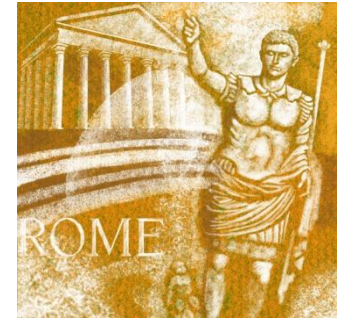
Online-Banking



Eine verschlüsselte Verbindung wird im Browser z.B. durch ein Schloss angezeigt.



Beispiel einer einfachen Verschlüsselung: Das Caesar-Verfahren



Die Caesar-Verschlüsselung gehört zu den Substitutionsverfahren. Alle Buchstaben im Klartext werden ersetzt, indem jeder Buchstabe im Klartext-Alphabet um eine bestimmte Stellenzahl verschoben wird. Der Schlüssel gibt an, um wie viele Stellen das Alphabet verschoben wird.

Geheimer Klartext

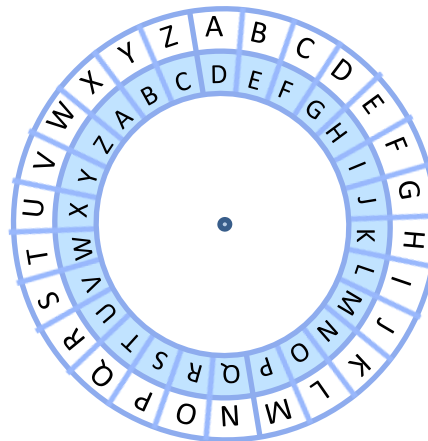


Verschlüsselung:

Die Buchstaben des Alphabets werden um **3** Stellen nach rechts verschoben und die Buchstaben im Text entsprechend ersetzt.

Der Schlüssel ist ein Buchstabe oder eine Zahl.

C = 3



Verschlüsselte Nachricht



Zur **Entschlüsselung** wird das Alphabet um **3** Stellen nach links verschoben.

Was ist ein Verschlüsselungsverfahren?

Verschlüsselungsverfahren:

Verfahren, bei dem ein Originaltext (der Klartext) mithilfe eines geheimen Schlüssels in einen Geheimtext (den Chiffretext) umgewandelt wird. Dies nennt man **Verschlüsselung**.

Umgekehrt wird das Verfahren auch zur **Entschlüsselung** verwendet. Das heißt, dass mit Kenntnis eines Schlüssels der Chiffretext in den Klartext zurückgewandelt wird. So lassen sich Nachrichten geheim übermitteln.

In der modernen Kryptographie werden Schlüssel häufig in Dateien gespeichert.



Beispiel

Caesar-Verfahren



Verschlüsselungsverfahren:

Verschiebung des Alphabets

Schlüssel:

Buchstabe oder Zahl, die angibt, um wie viele Stellen das Alphabet verschoben wird

Verschlüsselungsverfahren nutzen



- Ersetzen von Buchstaben des Alphabets
- Veränderung der Anordnung des Textes
- Mathematische Funktionen und Berechnungen
- u.v.m.

Schlüssel sind



- Buchstaben
- Wörter
- Zahlen
- u.v.m.

Grundsätzlich wird zwischen 2 Arten von Verschlüsselungsverfahren unterschieden: **symmetrisch** und **asymmetrisch**.

Arten von Verschlüsselungsverfahren (1)

1. Symmetrische Verschlüsselung

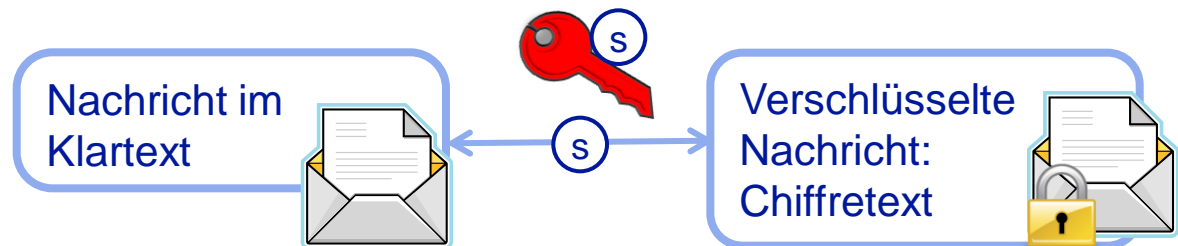
- Zur Ver- und Entschlüsselung wird derselbe Schlüssel verwendet.
(Im Folgenden wird ein Kleinbuchstabe als Abkürzung für einen Schlüssel verwendet.)
- Der symmetrische Schlüssel **s** ist dem Absender und dem Empfänger bekannt und muss vor Dritten geheim gehalten werden.

Es kann sein, dass bei der symmetrischen Verschlüsselung der Schlüssel zur Verschlüsselung nicht völlig identisch ist mit dem Schlüssel zur Entschlüsselung. Dann kann aber der eine Schlüssel mit Kenntnis des anderen leicht ermittelt werden.



Klartext- und
Geheimtextalphabet
z.B. ABC...XYZ

Ein Schlüssel **s** zum Verschlüsseln / Chiffrieren
und zum Entschlüsseln / Dechiffrieren.



Arten von Verschlüsselungsverfahren (2)

1. Symmetrische Verschlüsselung

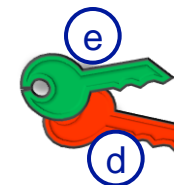
- Zur Ver- und Entschlüsselung wird derselbe Schlüssel verwendet.
(Im Folgenden wird ein Kleinbuchstabe als Abkürzung für einen Schlüssel verwendet.)
- Der symmetrische Schlüssel **s** ist dem Absender und dem Empfänger bekannt und muss vor Dritten geheim gehalten werden.

2. Asymmetrisch / Public Key

- Zur Ver- und Entschlüsselung werden verschiedene Schlüssel verwendet.
- Es gibt einen geheimen Schlüssel **d** und einen öffentlichen Schlüssel **e**.
- Auch mit Kenntnis von **e** kann **d** ohne weitere Information praktisch nicht berechnet werden.

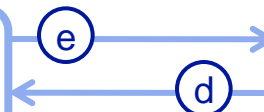
Klartext- und
Geheimtextalphabet
z.B. ABC...XYZ

Schlüssel **e** zum
Verschlüsseln /
Chiffrieren
(engl. encryption)



Schlüssel **d** zum
Entschlüsseln /
Dechiffrieren
(engl. decryption)

Nachricht im
Klartext



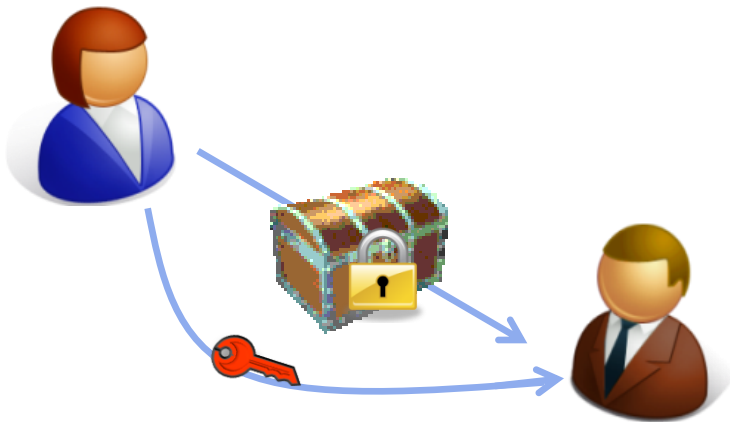
Verschlüsselte
Nachricht:
Chiffretext



Unterschied symmetrische und asymmetrische Verschlüsselung – übertragenes Beispiel

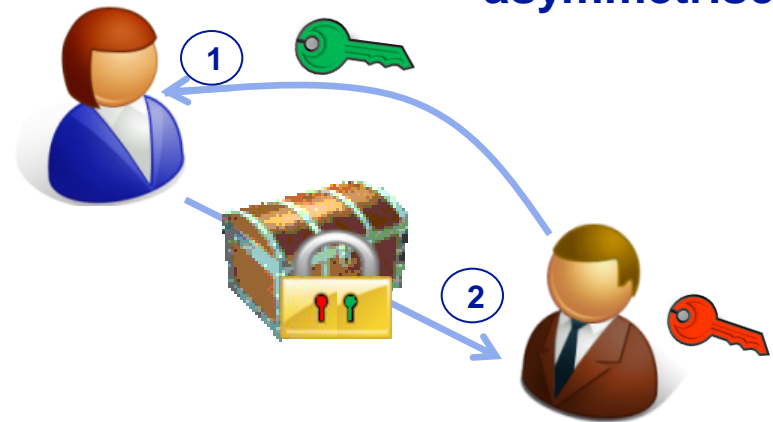
Alice möchte Bob ein Geheimnis schicken und packt es in eine Truhe.

symmetrisch



Idee: Es gibt nur einen Schlüssel. Mit diesem verschließt Alice die Truhe. Alice lässt Bob die verschlossene Truhe zukommen und separat auf sicherem Weg auch den Schlüssel zu dem Schloss. Mit diesem Schlüssel kann Bob die Truhe öffnen und das Geheimnis ansehen.

asymmetrisch

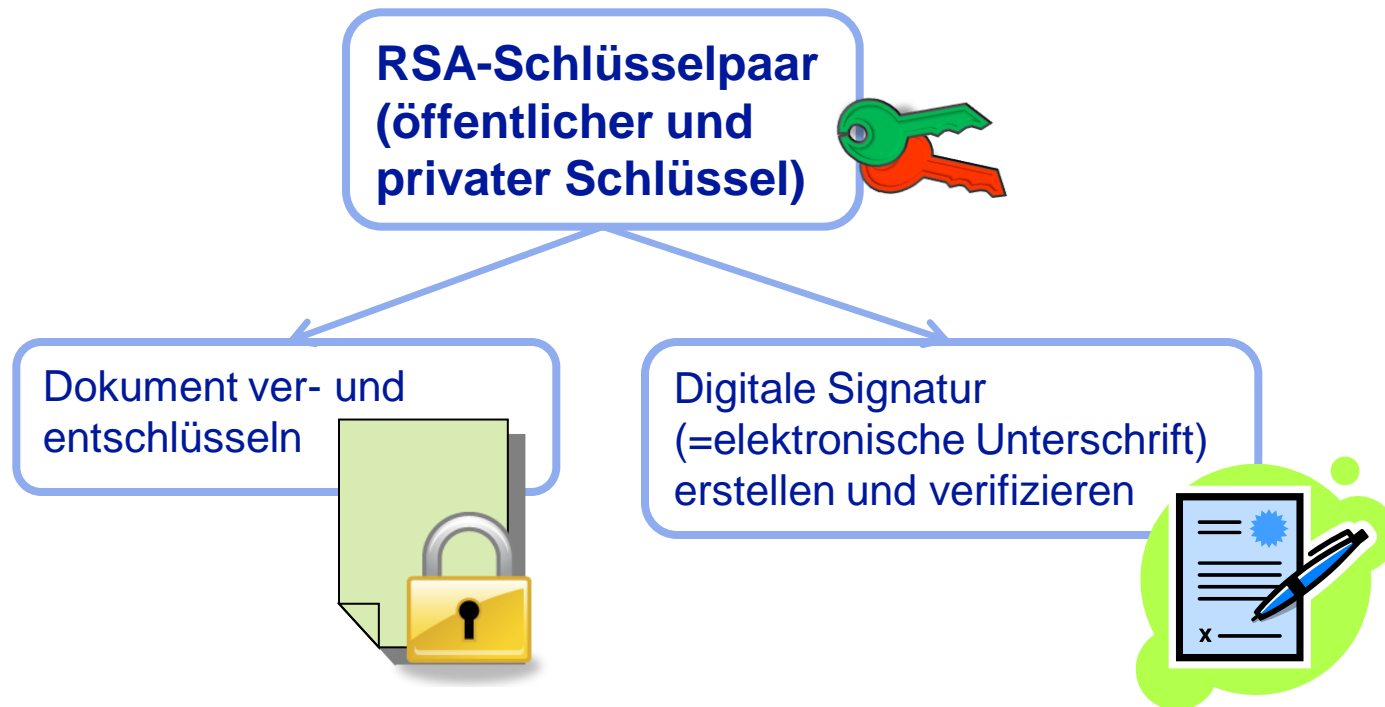


Idee: Es gibt zwei Schlüssel.




1. Alice besorgt Bobs öffentlichen Schlüssel. Mit diesem verschließt sie die Truhe.
2. Alice lässt Bob die verschlossene Truhe zukommen. Bob kann nun die Truhe mit seinem Schlüssel öffnen und das Geheimnis ansehen.

Allgemeine Informationen zu RSA

RSA ist ein asymmetrisches Verschlüsselungsverfahren, welches nach seinen Entwicklern Ronald **R**ivest, Adi **S**hamir und Leonard **A**dleman benannt wurde.



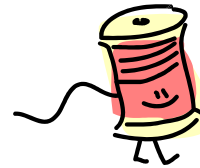
Hinweise zum Aufbau des Workshops

- Zunächst werden Aufgaben gestellt, die mit dem Lernprogramm  zu bearbeiten sind.
- Die **acht Aufgaben** bauen aufeinander auf und sollten nacheinander bearbeitet werden.
- Zu jedem Schritt werden Hinweise gegeben, worauf während der Bearbeitung der Aufgaben besonders geachtet werden sollte.
- Zu den meisten Aufgaben werden anschließend Verständnisfragen gestellt.
- Lösungsvorschläge der Aufgaben werden Schritt für Schritt anhand von Screenshots und Hinweisen gegeben.
- Für Interessierte gibt der „Besserwisser“ nähere Informationen zu mathematischen Hintergründen. 
- Hinweise und weitere Informationen werden mit  angezeigt.

In jeder Maske und zu jedem Menüpunkt stellt CrypTool ausführliche Online-Hilfe bereit: Einfach **F1** drücken.



Inhaltsübersicht zum Workshop



1. Schlüsselerzeugung S. 12

- Erzeugen eines eigenen Schlüsselpaars
- Wie ist ein Schlüsselpaar aufgebaut?

2. RSA-Demo S. 19

- Wie funktioniert die Ver- und Entschlüsselung?
(Beispiel anhand eines kleinen Textes)
- RSA knacken

3. Ver- und Entschlüsseln eines Dokuments S. 31

- Ver- und Entschlüsseln eines Dokuments zur Vertiefung

4. Hashverfahren S. 35

- Was ist ein Hashverfahren?
- Hash-Demo:
Wie funktioniert ein Hashverfahren?
- Worin bestehen Schwächen?

5. Digitale Signatur erstellen und verifizieren S. 42

- Was ist eine digitale Signatur?
- Signieren eines Dokumentes
- Prüfen einer Signatur auf Echtheit

6. Signaturdemo S. 49

- Schritt für Schritt durch das Signaturverfahren.
- Vertiefung des Wissens aus den vorherigen Aufgaben.

7. Digitales Zertifikat und PKI S. 53

- Klärung der Begriffe Digitales Zertifikat, PKI und CA
- Was ist Zertifizierung und wie wird es gemacht?

8. Hybride Verschlüsselung S. 56

- Was ist hybride Verschlüsselung?
- Hybrid-Demo am Beispiel RSA und AES

Aufgabe 1 Schlüsselerzeugung

1.1 Erzeugen Sie sich unter dem Menüpunkt

„Digitale Signaturen/PKI ⇒ PKI ⇒ Schlüssel erzeugen/importieren“

Ihr eigenes RSA-Schlüsselpaar.

- Welche Daten werden erfasst?
- Was fällt Ihnen auf?

1.2 Sehen Sie sich Ihr Schlüsselpaar noch einmal an unter

„Digitale Signaturen/PKI ⇒ PKI ⇒ Schlüssel anzeigen/exportieren“: „Öffentliche Parameter“ sowie „Zertifikat anzeigen“.

- Was ist für andere Nutzer sichtbar?


1.3 Hätten Sie es gewusst?

- Wie ist ein RSA-Schlüsselpaar zusammen gesetzt?
- Was besagt die Länge des RSA-Moduls?
- Warum ist eine PIN-Eingabe bei der Schlüsselerzeugung erforderlich?

Aufgabe 1.1 Vor der Erzeugung des eigenen RSA-Schlüsselpaars

Die Erzeugung des Schlüsselpaars ist erst nach vollständiger Angabe der persönlichen Daten sowie einer PIN-Eingabe möglich.

Wahl des Verschlüsselungsverfahrens und der gewünschten Länge des RSA-Moduls

Mit PIN ist hier ein  Passwort gemeint. Es dürfen also Groß- und Kleinbuchstaben, Ziffern sowie Sonderzeichen eingegeben werden.

Erzeugung eines asymmetrischen Schlüsselpaars

Verfahren:

- ☒ RSA
Bitlänge des RSA-Moduls: 1024
- ☐ DSA
Bitlänge der DSA-Primzahl: 1024
- ☐ Elliptische Kurven
Bezeichner (Bitlänge und Kurvenparameter): prime239v1

Benutzerdaten:

Das erzeugte Schlüsselpaar wird in einer verschlüsselten Datei (PSE) abgelegt. Durch Ihren PIN-Code wird das Schlüsselpaar geschützt.

Name: Wonderland

Vorname: Alice

Schlüsselkennung: (Optional) AW

PIN-Code:

PIN-Verifikation:

Hier werden die Domain-Parameter der spezifizierten Elliptischen Kurve angezeigt.

Parameter	Wert des Parameters	Bitlänge
-----------	---------------------	----------

Zahlensystem der Parameterdarstellung:

☐ Oktal ☒ Dezimal ☐ Hexadezimal

☒ Anzeigen des erzeugten Schlüsselpaars

Neues Schlüsselpaar erzeugen ... PKCS #12-Import Schließen

Angabe Persönlicher Daten

PIN-Eingabe erforderlich
(Diese PIN wird während des gesamten Kurses beibehalten.)

Aufgabe 1.1 Erzeugung des eigenen RSA-Schlüsselpaars

Neues Schlüsselpaar erzeugen ...

SECURE Crypto Runtime - Zufallszahlen-Generator

Zufallszahlen-Generierung

Bewegen Sie die Maus und geben Sie Zeichen auf der Tastatur ein, bis ausreichend Zufallsmaterial gesammelt wurde.

Durch Mausbewegungen und Tastaturbedienung „entsteht“ Zufall.

„Zufall“ ist zur Generierung des Schlüsselpaars erforderlich.

Ausgabe der öffentlichen Parameter:
RSA-Modul **N** und Exponent **e**
(Näheres auf Folie 16.)

Öffentliche Parameter

Öffentliche Parameter von [Wonderland][Alice][RSA-1024][1256635339][AW] anzeigen.

Variable	Wert
Modul	1783576540593273983195324445832474068723695717953156867097...
Expon...	65537

Zahlensystem der Parameterdarstellung

☐ Oktal ☒ Dezimal ☐ Hexadezimal

Übernehmen Zurück

Wahl der Zahldarstellung der öffentlichen Parameter

Dieses Fenster erscheint nur, wenn das **erste** Schlüsselpaar erzeugt wird. Der Zufall wird „gesammelt“ und in einem Pseudozufallszahlengenerator verwendet. Die generierten Pseudozufallszahlen sollen nicht von „echten“ Zufallszahlen unterschieden werden können.

CrypTool

Die von ihnen gewählten Parameter und das erzeugte Schlüsselpaar wurden erfolgreich abgespeichert. Der zugewiesene Schlüsselbezeichner ist: [Wonderland][Alice][RSA-1024][1256635339][AW]

Zum Erzeugen des Schlüsselpaars benötigte Zeit: 4,766 Sekunden.

OK

- Was ist für andere Nutzer sichtbar?

Verfügbare asymmetrische Schlüsselpaare

Die folgende Liste zeigt die verfügbaren asymmetrischen Schlüsselpaare an.
Klicken Sie mit der linken Maustaste auf eine Zeile, um eine Auswahl vorzunehmen.

Name	Vorname	Schlüsseltyp	Schlüsselkennung	Erstellt am	Interne ID-Nr.
Bauer	Petra	RSA-1024	PIN=7435	22.10.2009 14:39:41	1256215181
HybridEncrypti...	Bob	EC-prime239v1	PIN=1234	09.05.2007 11:21:14	1178702474
Meyer	Hans	RSA-1024	PIN=5267	22.10.2009 13:48:54	1256212134
SideChannelAt...	Bob	RSA-512	PIN=1234	06.07.2006 11:51:34	1152179494
Wonderland	Alice	RSA-1024	AW	27.10.2009 10:22:19	1256635339

Ausgewählte Schlüsselpaare:

- ☒ RSA-Schlüssel
- ☒ DSA-Schlüssel
- ☒ EC-Schlüssel

Öffentliche Parameter anzeigen...

Zertifikat anzeigen

Eintrag löschen...

Alle Parameter anzeigen...

PSE exportieren (PKCS#12)

Schließen



Öffentlicher Parameter von: Alice Wonderland

Variable	Wert
Modul	178367654
Exponent	69537

Zahlensystem der Parameterdarstellung

☐ Oktal ☒ Dekimal ☐ Hexadezimal

Zurück

Persönliche
öffentliche
Parameter:

RSA-Modul **N**
Exponent **e**

Zertifikatsdaten	
Version:	2 (X.509v3-1996)
SubjectName:	CN=Alice Wonderland [1256635339], DC=cry
IssuerName:	CN=Cryptool CA 2, DC=cryptool, DC-org
SerialNumber:	C:FA:9E:09:AA:35:8B:2F
Validity:	NotBefore: Tue Oct 27 10:23:13 2009 (019027092313Z) NotAfter: Wed Oct 27 11:23:13 2010 (101027092313Z)
Public Key:	Issuerprint: 9221 DDB5 5C79 D127 D20C E20B 2275 908E
SubjectKey:	Algorithm name (OID 2.5.8.1.1), Keysize = 1024 Public modulus (no. of bits = 1024) 0 80F0E6F7 36D0F5D DFF6CD45 D21FE6A1 10 1813B92C E8DA5B07 C88732A AB5279D1 20 36C23A85 80785EC 6A9891A A3E8C3B 30 BA27A6A8 A87AE4F2 238C973F BE4E455 40 EE0E3F90 B2785F6F 98C61E32 324A4F59 50 C70E3E1F 5214919A 30B77C7C 423E95E 60 875BEBC7 79DA4FE6 630F3B94 04B593AB 70 5C8BA5C2 6B43CECF C637923 4A8FC75 Public exponent (no. of bits = 17) 0 010001
Schließen	
Zertifikatsdaten	
Public exponent (no. of bits = 17): 0 010001	
Certificate	0 0587E20
Private extensions:	OID 2.206.5.4.3.2
PrintableString:	[.Wonderland][Alice][RSA-1024][12] [56635339][AW]
SHA1 digest of DER code of this CertSigned:	1018666B B9B8FACA
Signature:	Algorithm sha1WithRSAEncryption (OID 1.3.6.1.2.1.1.1.1) 0 29568650 F74419D8 40F0F30B 03A8FC7D 10 E3A093B2 232530C 2640121D 05C0014A 20 F390D2DE AEE811B5 EE4F730B 14C25565 30 7210E357 9D45A19C A435D448 5E57B0C7 40 BFC627D7 CEEFD042 CFF6662D 73BD5D5A 50 29D62121 9E78A21A F81D12BF 77678983 60 856C3FAC 5C129920 79B070E0 9AC9F81B 70 31F0EAD0 9B4A19C 4435D448 5E57B0C7 80 6C2D32DD F0E78772 D20025D9 854DC4DA 90 F034DCD5 D4A19B63 43F35FEF 8112C0A3 A0 2ABE5F52 6C714A66 7BBE1A79 53E5EC88 B0 DC27F3A3 F84776E7 C07D0491 31533266 C0 4006C8B A7632710 E3FC3015 20F79A7C D0 3E743130 2608323D 8D36A0B2 24C8A29C E0 DC8619D0 697879E0 84BA2558 79F1A472 F0 144750AD E1C8277E 4497B699 55F01935
Certificate Fingerprint (MD5): OE F2 48 10 75: 86: 84: 42: 80: 03	
Schließen	

Persönliche Daten

Dauer der Gültigkeit

RSA-Modul

Öffentlicher Exponent

Signatur

(Näheres in Aufgabe 5)

Aufgabe 1.3 Hätten Sie es gewusst? (1)

- Wie ist ein RSA-Schlüsselpaar zusammen gesetzt?

Schlüsselpaar (e,d) 

- **e** öffentlicher Schlüssel (*e=encryption*)

 Dieser Schlüssel ist öffentlich, d.h. für jeden sichtbar.

- **d** privater Schlüssel (*d=decryption*)

 Dieser Schlüssel ist geheim, d.h. nur der Besitzer des Schlüssels kennt ihn.

Außerdem gehört zu jedem Schlüsselpaar der sogenannte RSA-Modul **N**.
N ist öffentlich.

Ohne ein Geheimnis zu kennen, ist es auch mit Kenntnis von N und e praktisch (mit Mitteln der heutigen Technologie) unmöglich, d zu berechnen.



- Der RSA-Modul N ist das Produkt $N=p \cdot q$; p und q sind zufällig erzeugte, etwa gleichlange Primzahlen. (Näheres zur Längenangabe auf Folie 17.)
- Zur Verschlüsselung wird eine „Falltürfunktion“ f verwendet.
D.h. $f(x) = y$ ist leicht zu berechnen, $f^{-1}(y)$ jedoch nicht, es sei denn, es ist eine Geheiminformation bekannt.

Aufgabe 1.3 Hätten Sie es gewusst? (2)

- Was besagt die Länge des RSA-Moduls?

Mit Länge ist hier die Bitlänge des RSA-Moduls gemeint.
Je länger der RSA-Modul, desto sicherer wird das Verfahren.

Anstatt von Länge des RSA-Moduls spricht man auch häufig von Länge des „RSA-Schlüssels“.



Ein **Bit** besteht aus einer 0 oder einer 1.
Die **Bitlänge** einer Zahl gibt an, aus wievielen Nullen und Einsen sie in ihrer Binärdarstellung besteht.

Beispiel

Die Binärzahl 10011001 hat eine Bitlänge von 8.
In Dezimaldarstellung: 153
Mit einer Bitlänge von 8 werden Zahlen im Bereich von 0 bis 255 dargestellt.




Aufgabe 1.3 Hätten Sie es gewusst? (3)

- Warum ist eine PIN-Eingabe erforderlich?

Durch die PIN wird der geheime Schlüssel geschützt, nur sein Besitzer hat darauf Zugriff. Der Schlüssel wird in einer verschlüsselten Datei gespeichert.

Ein gutes Passwort:

- besteht aus mindestens 8 (besser mehr) Zeichen und enthält Klein- und Großbuchstaben, Zahlen sowie Sonderzeichen. 
- enthält keine Namen und Geburtsdaten.
- enthält keine Begriffe aus Wörterbüchern.

Hinweis: *Passwort-Qualitätsmesser in CrypTool:
Einzelverfahren → Tools*

Beispiel

Schlechtes PW: Alice
Gutes PW: AW=1nM,smT!

Tipp: Merken Sie sich einen beliebigen Satz und benutzen die Anfangsbuchstaben und Satzzeichen als Passwort:
z.B. „Alice Wonderland = 1 nettes Mädchen, sie mag Tee!“ -> AW=1nM,smT!

Aufgabe 2 RSA-Demo

2.1 Erzeugen Sie einen 128 bit RSA-Schlüssel unter Menü

„[Einzelverfahren](#) ⇒ [RSA-Kryptosystem](#) ⇒ [RSA-Demo](#)“.

(Hinweis: Der RSA-Modul $N=p*q$ mit zwei etwa gleichlangen Primzahlen p und q .)

2.2 Dieser RSA-Schlüssel soll genutzt werden, um einen kleinen Text zu verschlüsseln.

Im nächsten Schritt soll der ausgegebene Text entschlüsselt werden.

- Welche Schritte werden bei der Ver- und Entschlüsselung jeweils vorgenommen?
- Welcher Schlüssel wurde jeweils zur Ver- und Entschlüsselung genutzt?
- Warum sollte die Blocklänge möglichst groß gewählt werden?
- Wie groß kann die Blocklänge sein?
- Falls nur die öffentlichen Parameter einsehbar sind: Was ist möglich?

2.3 RSA knacken

Versuchen Sie, den eben erzeugten RSA-Modul zu knacken .

Nutzen Sie dazu den Dialog „[RSA-Modul faktorisieren](#)“.

(Dazu muss der zweite Punkt in der Maske aktiviert sein.)

- Entschlüsseln Sie den verschlüsselten Text.
- Was heißt Faktorisierung?
- Was bedeutet es, wenn eine Faktorisierung des RSA-Moduls möglich ist?
- Wie kann der Möglichkeit der Faktorisierung vorgebeugt werden?

Aufgabe 2.1 Erzeugen eines 128 bit RSA-Schlüssels

RSA-Demo

RSA mit privatem und öffentlichem Schlüssel -- oder nur mit öffentlichem Schlüssel

- ☒ Wählen Sie 2 Primzahlen p und q. Die Zahl $N = pq$ ist der öffentliche RSA-Modul und $\phi(N) = (p-1)(q-1)$ ist die Eulersche Zahl. Der öffentliche Schlüssel e ist teilerfremd zu $\phi(N)$. Daraus wird der geheime Schlüssel $d = e^{-1} \pmod{\phi(N)}$ berechnet.
- ☐ Zur Verschlüsselung von Daten oder zur Verifikation einer Signatur genügt es, dass Sie die öffentlichen RSA-Parameter angeben: den RSA-Modul N und den öffentlichen Schlüssel e.

Primzahleingabe

Primzahl p

Primzahl q

Primzahlen generieren...

RSA mit privatem und öffentlichem Schlüssel -- oder nur mit öffentlichem Schlüssel

- ☒ Wählen Sie 2 Primzahlen p und q. Die Zahl $N = pq$ ist der öffentliche RSA-Modul und $\phi(N) = (p-1)(q-1)$ ist die Eulersche Zahl. Der öffentliche Schlüssel e ist teilerfremd zu $\phi(N)$. Daraus wird der geheime Schlüssel $d = e^{-1} \pmod{\phi(N)}$ berechnet.

Primzahltests:
Testen, ob eine gegebene Zahl eine Primzahl ist.

Vorgehen:
Aus dem gegebenen Wertebereich wird zufällig eine Zahl gewählt. Mit einem Primzahltest wird getestet, ob es sich um eine Primzahl handelt. Ist dies der Fall, wird sie als p oder q ausgegeben.

Primzahlen generieren

Primzahlen spielen in der modernen Kryptographie eine wichtige Rolle. Hier können Sie sich Primzahlen aus einem vorgegebenden Wertebereich [Untergrenze, Obergrenze] erzeugen.

Anzahl der zu generierenden Primzahlen

- ☒ Zwei Primzahlen zufällig aus dem Wertebereich (den Wertebereichen) generieren
- ☐ Alle Primzahlen in dem (für p vorgegebenen) Wertebereich generieren

Trennzeichen für die Ausgabe der Primzahlen:

Algorithmen zur Generierung

- ☒ Miller-Rabin-Test
- ☐ Solovay-Strassen-Test
- ☐ Fermat-Test

Wertebereich der Primzahlen p und q

- ☒ Unabhängig voneinander eingeben
- ☐ Beide gleich (nur einen eingeben)

Primzahl p

Untergrenze

Obergrenze

Ergebnis

Primzahl q

Untergrenze

Obergrenze

Ergebnis

Primzahlen generieren Primzahlen übernehmen Abbrechen

Mit Länge des RSA-Schlüssels ist die Länge des RSA-Moduls N gemeint.

Da $N=p \cdot q$ und p und q etwa gleichlang sind, müssen beide Zahlen etwa die „halbe Länge“ des RSA-Moduls haben.

Also hier: Um einen 128 bit langen RSA-Modul zu erzeugen, muss die Obergrenze des Wertebereichs 2^{64} betragen.

Aufgabe 2.2 Verschlüsselung eines kleinen Textes

RSA-Demo

RSA mit privatem und öffentlichem Schlüssel – oder nur mit öffentlichem Schlüssel

☒ Wählen Sie 2 Primzahlen p und q. Die Zahl $N = pq$ ist der öffentliche RSA-Modul und $\phi(N) = (p-1)(q-1)$ ist die Eulersche Zahl. Der öffentliche Schlüssel e ist teilerfremd zu $\phi(N)$. Daraus wird der geheime Schlüssel $d = e^{-1} \pmod{\phi(N)}$ berechnet.

☐ Zur Verschlüsselung von Daten oder zur Verifikation einer Signatur genügt es, dass Sie die öffentlichen RSA-Parameter angeben: den RSA-Modul N und den öffentlichen Schlüssel e.

Primzahleingabe

Primzahl p: 954881343166658021 Primzahlen generieren...

Primzahl q: 16678912785805852097

RSA-Parameter

RSA-Modul N: 159253826434699677652450041056204720037 (öffentlich)

$\phi(N) = (p-1)(q-1)$: 15926326434699677626222314838732209920 (geheim)

Öffentlicher Schlüssel e: 2¹⁶+1

Geheimer Schlüssel d: 45006119681563666778119799057224476673 Parameter aktualisieren

Operationen

☒ Verschlüsselung mit ☐ Entschlüsselung mit d

Eingabe als: ☒ Text ☐ Zahlen Optionen für Alphabet und Zahlensystem...

Eingabe der zu ver- oder entschlüsselnden Nachricht als Text oder als HexDump.

DIES IST EIN BEISPIELTEXT

Verschlüsseln Entschlüsseln Schließen

RSA-Parameter

Optionen für die RSA-Demo

Alphabetoptionen

☒ Alle 26 Zeichen Anzahl Zeichen: 27

☐ Alphabet vorgeben: ABCDEFGHIJKLMNOPQRSTUVWXYZ

RSA-Variante

☒ Normal ☐ Dialog der Schwestern

Methode, wie ein Block als Zahl codiert wird

☒ b-adisch ☐ Basissystem

Blocklänge

Die Anzahl der Zeichen, die pro RSA-Operation verschlüsselt werden. Die maximale Anzahl ist abhängig von der Bitlänge des RSA-Moduls N, der Anzahl der Zeichen im Alphabet und der Codierungsmethode der Nachricht.

Blocklänge / Zeichen: 5 (Maximale Blocklänge 26 Zeichen)

Zahlensystem

Die Zahlen der RSA-Ver-/Entschlüsselung werden in dem folgenden Zahlensystem dargestellt.

☒ Dezimal ☐ Binär ☐ Oktal ☐ Hexadezimal

OK Abbrechen

Wahl des Alphabets

(26 Großbuchstaben + ein Leerzeichen)

Wahl der Blocklänge

b-adisch:

b ist die Anzahl der Buchstaben im gewählten Alphabet (hier b=27). Die Codierung erfolgt zur Basis b.

Zahlensystem

Dezimal: Ziffern 0,...,9

Binär: 0 und 1

Oktal: 0,...,7

Hexadezimal: 0,...,9,A,B,...,F



Aufgabe 2.2 Arbeitsschritte der Verschlüsselung

RSA-Demo

RSA mit privatem und öffentlichem Schlüssel -- oder nur mit öffentlichem Schlüssel

☒ Wählen Sie 2 Primzahlen p und q. Die Zahl $N = pq$ ist der öffentliche RSA-Modul und $\phi(N) = (p-1)(q-1)$ ist die Eulersche Zahl. Der öffentliche Schlüssel e ist teilerfremd zu $\phi(N)$. Daraus wird der geheime Schlüssel $d = e^{-1} \pmod{\phi(N)}$ berechnet.

☐ Zur Verschlüsselung von Daten oder zur Verifikation einer Signatur genügt es, dass Sie die öffentlichen RSA-Parameter angeben: den RSA-Modul N und den öffentlichen Schlüssel e.

Primzahleingabe

Primzahl p: 954881343166658021 Primzahlen generieren...

Primzahl q: 16678912785805852097

RSA-Parameter

RSA-Modul N: 159263826434699677652450041056204720037 (öffentlich)

$\phi(N) = (p-1)(q-1)$: 1592638264346996776222314838732209920 (geheim)

Öffentlicher Schlüssel e: 2¹⁶+1

Geheimer Schlüssel d: 4500611968156366778119799057224476673 Parameter aktualisieren

RSA-Verschlüsselung mit e / Entschlüsselung mit d

Eingabe als: ☒ Text ☐ Zahlen Optionen für Alphabet und Zahlensystem...

Eingabetext: DIES IST EIN BEISPIELTEXT

Der Eingabetext wird in Blöcke der Länge 5 aufgeteilt. (das Symbol '#' dient als Trennzeichen).

DIES # IST E # IN BE # ISPIE # LTEXT

Zahlendarstellung der Eingabe zur Basis 10.

02307069 # 05171531 # 05058590 # 05168858 # 06775265

Verschlüsselung in den Chiffretext: $c[i] = m[i]^e \pmod{N}$

57291942290067526289946818532785760548 # 127751792569670123601713901668510982662 # 146581

Verschlüsseln Entschlüsseln Schließen

RSA mit privatem und öffentlichem Schlüssel -- oder nur mit öffentlichem Schlüssel

☒ Wählen Sie 2 Primzahlen p und q. Die Zahl $N = pq$ ist der öffentliche RSA-Modul und $\phi(N) = (p-1)(q-1)$ ist die Eulersche Zahl. Der öffentliche Schlüssel e ist teilerfremd zu $\phi(N)$. Daraus wird der geheime Schlüssel $d = e^{-1} \pmod{\phi(N)}$ berechnet.

Der Eingabetext wird in Blöcke der Länge 5 aufgeteilt

1. Einteilung des Klartextes in Blöcke einer festen Länge
2. Codierung der Buchstaben in Zahlen, z.B. A=01, B=02 etc.
3. Codierung der Zahlblöcke gemäß der gewählten Codierungsmethode (b-adisch oder Basissystem)
4. Verschlüsselung der einzelnen Blöcke
5. Ausgabe der verschlüsselten Blöcke

Beispiel

Verschlüsseln des 1. Blocks (der Länge 5)

Klartext: DIES_
 Codiert: 04 09 05 19 00
 27-adisch: $4 \cdot 27^4 + 9 \cdot 27^3 + \dots + 19 \cdot 27^1 + 0 \cdot 27^0$
 $= 2\,307\,069$
 $= m(1)$
 $c(1) = m(1)^e \pmod{N} = 57291942290067526289946818532785760548$



Chiffretext = Chifftrat
 Verschlüsselter Text = Geheimtext

Verschlüsselung in den Chiffretext: $c[i] = m[i]^e \pmod{N}$

Aufgabe 2.2 Arbeitsschritte der Entschlüsselung

[illegible]

1. Eingabe der verschlüsselten Blöcke als Zahlen
2. Entschlüsselung der Blöcke
3. Decodierung der entschlüsselten Blöcke → Ausgabertext
4. Zusammenfügen der Blöcke → Klartext

Beispiel

Entschlüsseln des 1. Blocks

Chiffretext in Zahlendarstellung:

$$c(1) = 57291942290067526289946818532785760548$$

Entschlüsseln und Decodieren:

$$\begin{aligned} m(1) &= c(1)^d \pmod{N} \\ &= 2307069 \\ &= 4 \cdot 27^4 + 9 \cdot 27^3 + 5 \cdot 27^2 + 19 \cdot 27^1 + 0 \cdot 27^0 \\ &= \text{DIES} \end{aligned}$$



Entschlüsselung in den Klartext $m[i] = c[i]^d \pmod{N}$

Ausgabertext aus der Entschlüsselung (in Blöcken der Länge 5:

Aufgabe 2.2 Hätten Sie es gewusst? (1)

- Warum sollte die Blocklänge möglichst groß sein?
 - Die Blocklänge gibt an, wie viele Zeichen pro RSA-Operation verschlüsselt werden.
 - Je größer die Blocklänge, umso mehr Informationen werden mit einer Operation verschlüsselt.
 - Bei einer größeren Blocklänge werden also weniger Operationen im Vergleich zu kleinen Blöcken benötigt.
 - Das Verfahren ist effizienter, aber man muss mit umso größeren Zahlen rechnen.
 - Bei einer kleinen Blocklänge ist es möglich, den Chiffretext durch eine Substitutionsanalyse zu knacken.
 - Je größer die Blocklänge, desto mehr mögliche Substitute gibt es.
 - Das Verfahren ist sicherer.

Aufgabe 2.2 Hätten Sie es gewusst? (2)

- Wie groß kann die Blocklänge sein?

Die maximal mögliche Blocklänge ist abhängig von

- der Bitlänge des RSA-Moduls N ,
- der Anzahl der Zeichen im Alphabet und
- der Codierungsmethode der Nachricht (b-adisch oder Basissystem).

Beispiel

zur Berechnung der maximalen Blocklänge

Es soll eine Nachricht m verschlüsselt werden. Dazu muss m in Blöcke m_i unterteilt werden, die jeweils kürzer sind, als der RSA-Modul N . Wie groß können die Blöcke maximal sein bzw. was ist die maximale Blocklänge x ?

Gegeben: N der Länge 128 bit, N ist also eine Zahl zwischen 0 und $2^{128}-1$.

Alphabet mit 27 Zeichen: ABC...XYZ und Leerzeichen

Die Zeichen haben folgende Codierung: Leerzeichen=00, A=01, B=02, ..., Z=26

Die Blöcke m_i des Klartextes werden als Zahlen dargestellt.

Dazu wird ein Basissystem gewählt. Hier wählen wir als Basis $b=27$ und haben damit das 27-adische Zahlssystem.

Mit der Blocklänge x können also die Blöcke m_i als Dezimalzahlen zwischen 0 und 27^x-1 dargestellt werden.



Berechnung von x :

Es gilt: $m_i < N$

Die Blöcke sind kleiner als der RSA-Modul.

d.h. $27^x - 1 < 2^{128} - 1$

$$\Leftrightarrow 27^x < 2^{128}$$

Diese Gleichung muss nun nach x umgestellt werden. Dazu wird der Logarithmus angewendet.

$$\Leftrightarrow \log 27^x < \log 2^{128}$$

$$\Leftrightarrow x \cdot \log 27 < 128 \cdot \log 2$$

$$\Leftrightarrow x < \frac{128 \cdot \log 2}{\log 27}$$

$$\frac{128 \cdot \log 2}{\log 27} \approx 26,92$$

x ist also kleiner als 26,92. Da x eine ganze Zahl ist und maximal sein soll, gilt $x=26$.

Die Blöcke m_i können also aus maximal 26 Zeichen des Alphabets bestehen.

z.B. WORT mit der Blocklänge 4 in Zahldarstellung:

W=23, O=15, R=18, T=20.

Damit: $\text{WORT} = 23 \cdot 27^3 + 15 \cdot 27^2 + 18 \cdot 27^1 + 20 \cdot 27^0 = 464.150$.

Aufgabe 2.2 Hätten Sie es gewusst? (3)

- Welcher Schlüssel wird jeweils zur Ver- und Entschlüsselung genutzt?

Verschlüsselung:

- Verwendung des öffentlichen Schlüssels e

Entschlüsselung:

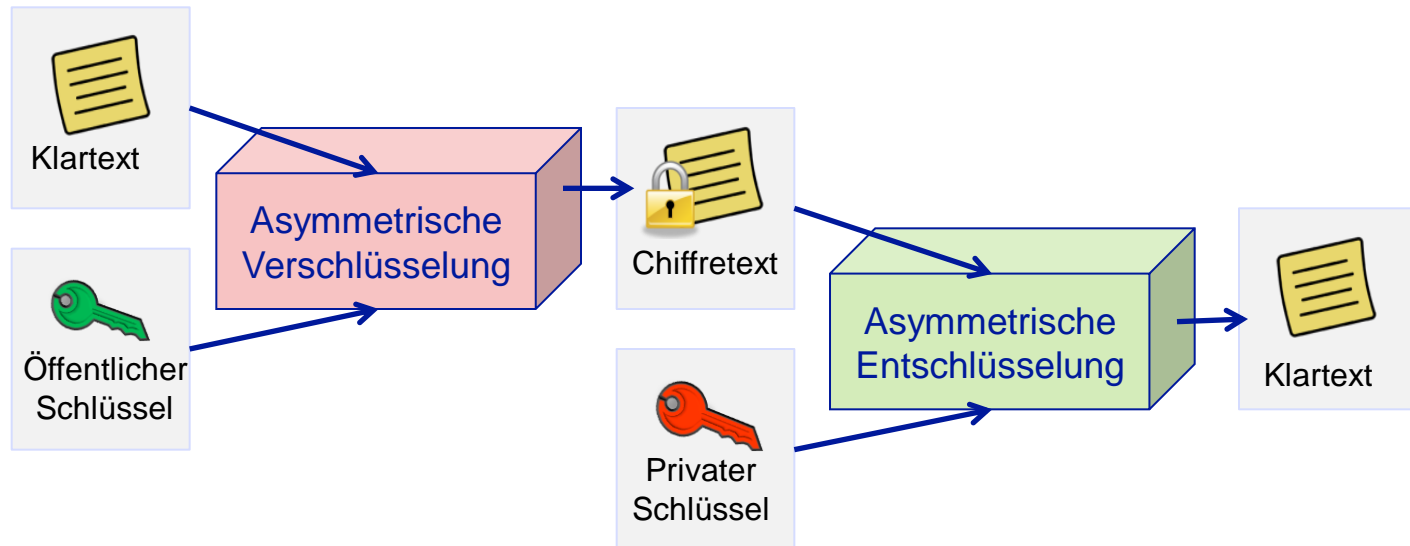
- Verwendung des privaten Schlüssels d

m = Nachricht im Klartext

c = Chiffretext

- Berechnung des Chiffretextes aus dem Klartext $c = m^e \pmod{N}$

- Berechnung des Klartextes aus dem Chiffretext $m = c^d \pmod{N}$



Aufgabe 2.2 Hätten Sie es gewusst? (4)

- Es sind nur die öffentlichen Parameter einsehbar. Was ist möglich?

- Zur Verschlüsselung von Daten oder zur Verifikation einer Signatur genügt es, dass Sie die öffentlichen RSA-Parameter angeben: den RSA-Modul N und den öffentlichen Schlüssel e .

geheime Parameter
nicht sichtbar

Möglich ist damit nur noch die
Verschlüsselung
(nicht mehr die Entschlüsselung)!

RSA-Demo

RSA mit privatem und öffentlichem Schlüssel -- oder nur mit öffentlichem Schlüssel

☐ Wählen Sie 2 Primzahlen p und q . Die Zahl $N = pq$ ist der öffentliche RSA-Modul und $\phi(N) = (p-1)(q-1)$ ist die Eulersche Zahl. Der öffentliche Schlüssel e ist teilerfremd zu $\phi(N)$. Daraus wird der geheime Schlüssel $d = e^{-1} \pmod{\phi(N)}$ berechnet.

☒ Zur Verschlüsselung von Daten oder zur Verifikation einer Signatur genügt es, dass Sie die öffentlichen RSA-Parameter angeben: den RSA-Modul N und den öffentlichen Schlüssel e .

Faktorisierungsangriff

Sie können mit Hilfe der Faktorisierung versuchen, den öffentlichen RSA-Modul N in seine Primfaktoren p und q zu faktorisieren.

RSA-Modul faktorisieren...

RSA-Parameter

RSA-Modul N 159263826434699677652450041056204720037 (öffentlich)

$\phi(N) = (p-1)(q-1)$ (geheim)

Öffentlicher Schlüssel e 2¹⁶+1

Geheimer Schlüssel d

Parameter aktualisieren

RSA-Verschlüsselung mit e / Entschlüsselung mit d

Eingabe als ☒ Text ☐ Zahlen Optionen für Alphabet und Zahlensystem...

Eingabe der zu ver- oder entschlüsselnden Nachricht als Text oder als HexDump.

DIES IST EIN BEISPIELTEXT

Verschlüsseln Entschlüsseln Schließen

Aufgabe 2.3 RSA knacken

Algorithmen, mit denen man eine Zahl faktorisieren kann.

RSA-Demo

RSA mit privatem und öffentlichem Schlüssel -- oder nur mit öffentlichem Schlüssel

☐ Wählen Sie 2 Primzahlen p und q. Die Zahl $N = pq$ ist der öffentliche RSA-Modul und $\phi(N) = (p-1)(q-1)$ ist die Eulersche Zahl. Der öffentliche Schlüssel e ist teilerfremd zu $\phi(N)$. Daraus wird der geheime Schlüssel $d = e^{-1} \pmod{\phi(N)}$ berechnet.

☒ Zur Verschlüsselung von Daten oder zur Verifikation einer Signatur genügt es, dass Sie die öffentlichen RSA-Parameter angeben: den RSA-Modul N und den öffentlichen Schlüssel e.

Faktorisierungsangriff

Sie können mit Hilfe der Faktorisierung versuchen, den öffentlichen RSA-Modul N in seine Primfaktoren p und q zu faktorisieren.

RSA-Modul faktorisieren...

RSA-Parameter

RSA-Modul N: 159263826434699677652450041056204720037 (öffentlich)

$\phi(N) = (p-1)(q-1)$: (geheim)

Öffentlicher Schlüssel e: $2^{16}+1$

Geheimer Schlüssel d:

Parameter aktualisieren

Faktorisieren einer Zahl

Algorithmen zur Faktorisierung

- ☒ Brute-Force
- ☒ Brent
- ☒ Pollard
- ☒ Williams
- ☒ Lenstra
- ☒ Quadratisches Sieb

Eingabe

Geben Sie die zu faktorisierende Zahl ein:

159263826434699677652450041056204720037

Faktorisierung (schrittweise)

Durch das Anklicken des Buttons "Weiter" wird initial die Zahl im Eingabefeld und dann jeweils die nächste zusammengesetzte Zahl im Feld "Produktdarstellung" in zwei Faktoren zerlegt.

Weiter

Faktorisierungsergebnis

Die Faktorisierung wird in dem Format $\langle z_1^{a_1} * z_2^{a_2} * \dots * z_n^{a_n} \rangle$ dargestellt. Zusammengesetzte Zahlen sind rot markiert.

Letzte Faktorisierung durch: Quadratisches Sieb 2 Faktoren gefunden in 2,265 Sekunden.

Produktdarstellung der Faktorisierung:

9548813431666658021 * 16678912785805852097

Details

Schließen

Jede natürliche Zahl kann eindeutig als Produkt von Primzahlen geschrieben werden.

Faktorisierung:
Zerlegung einer natürlichen Zahl in ihre Primfaktoren

Faktorisierungstimer

Zu faktorisierende Zahl: 159263826...204720037

Algorithmus	Iterationen	
Brent	3285	Abbrechen
Pollard	475749	Abbrechen
Williams	327338	Abbrechen
Lenstra	695	Abbrechen
Quadratisches Sieb	7	Abbrechen

Durch Faktorisierung gefundene Primfaktoren

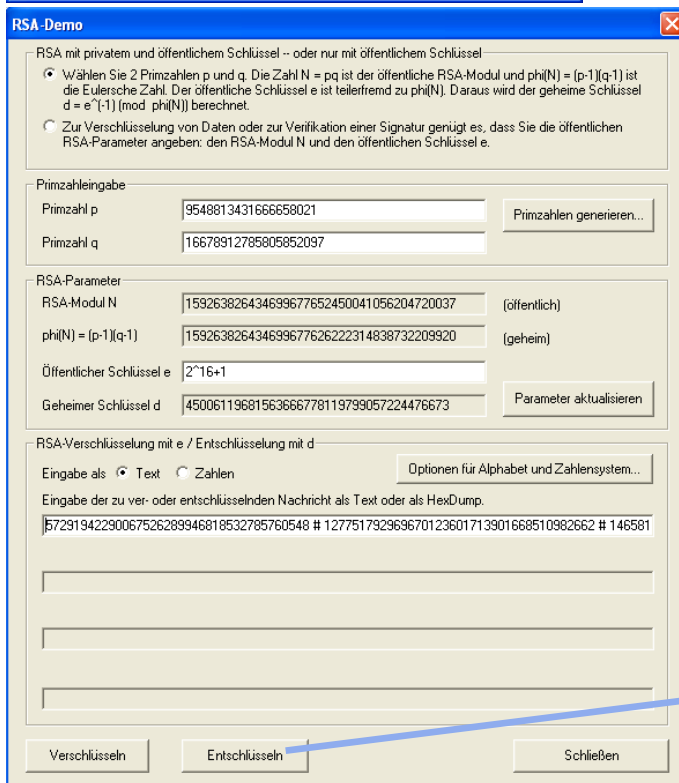
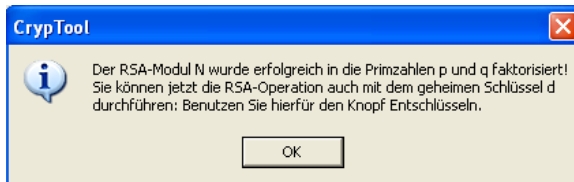
CrypTool

Der RSA-Modul N wurde erfolgreich in die Primzahlen p und q faktorisiert! Sie können jetzt die RSA-Operation auch mit dem geheimen Schlüssel d durchführen: Benutzen Sie hierfür den Knopf Entschlüsseln.

OK

Aufgabe 2.3 Hätten Sie es gewusst? (1)

- Was bedeutet es, wenn eine Faktorisierung des RSA-Moduls möglich ist?



Kann der RSA-Modul N in seine Primfaktoren zerlegt werden, dann sind die beiden geheimen Primzahlen p und q bekannt. D.h. die Primzahlen waren schlecht, z. B. zu klein gewählt.

Damit ist es möglich, den geheimen Schlüssel d zu berechnen.

Verschlüsselte Nachrichten können geknackt werden!



Aufgabe 2.3 Hätten Sie es gewusst? (2)

- Wie kann einer eventuellen Faktorisierung vorgebeugt werden?

Die Länge des RSA-Moduls sollte so groß gewählt werden, dass es technisch nicht möglich ist, die Primfaktorzerlegung in einer sinnvollen Zeit zu ermitteln.

Der aktuelle Rekord liegt in der Zerlegung der Zahl RSA-768.



In der Praxis sollte die Schlüssellänge nicht weniger als 1024 bit betragen!

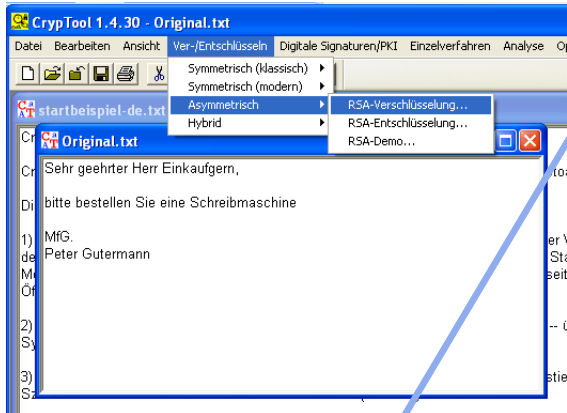
Aufgabe 3 Ver- und Entschlüsseln eines Dokuments

- 3.1 Öffnen Sie das Dokument „[Original.txt](#)“ (Datei → Öffnen, Unterordner „[Examples](#)“) und verschlüsseln Sie dieses Dokument mit Ihrem RSA-Schlüssel (Menü „[Ver-/Entschlüsseln](#) ⇒ [Asymmetrisch](#) ⇒ [RSA-Verschlüsselung](#)“).
- Wie ist das verschlüsselte Dokument aufgebaut?
- 3.2 Versetzen Sie sich in die Lage des Empfängers und entschlüsseln Sie das Dokument (Menü „[Ver-/Entschlüsseln](#) ⇒ [Asymmetrisch](#) ⇒ [RSA-Entschlüsselung](#)“).
- 3.3 Hätten Sie es gewusst?
- Welcher Teil des Schlüsselpaares wird vom Sender und welcher vom Empfänger verwendet?

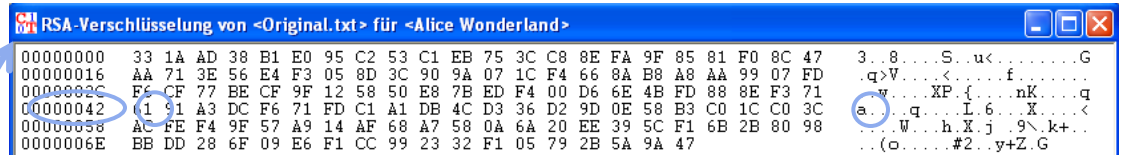
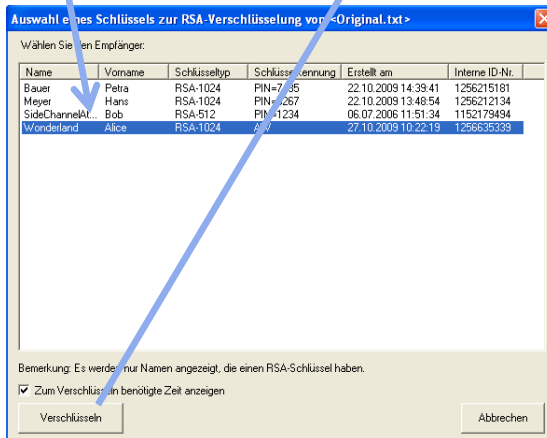


Aufgabe 3.1 Dokument verschlüsseln

- Wie ist das verschlüsselte Dokument aufgebaut?



Auswahl eines Empfängers, um dessen öffentlichen Schlüssel zu benutzen.



- Positionsanzeige (beginnend von 0)
- Gibt für jede Zeile der anderen beiden Spalten an, an welcher Position das erste Zeichen steht.
- Die Nummerierung erfolgt im hexadezimalen Zahlssystem.
- Z. B. steht an der Position 00000042 das Zeichen mit der Codierung 61 („a“). Die Stelle 42 hex = 66 dezimal.

- Hexadezimale Darstellung der Zeichen
- Ein Zeichen wird durch zwei aufeinanderfolgende Hexzeichen (0, 1, ..., 9, A, B, ..., F) dargestellt

Hinweis:
ASCII-
Tabelle
siehe Hilfe

- Darstellbare Zeichen gemäß ihrem ASCII-Code
- Nicht-anzeigbare Zeichen werden durch einen Punkt dargestellt.
- Der verschlüsselte Text enthält viele Zeichen, die nicht darstellbar sind. (ASCII-Werte > 128)

Aufgabe 3.2 Dokument entschlüsseln

```
RSA-Verschlüsselung von <Original.txt> für <Alice Wonderland>

00000000 33 1A AD 38 B1 E0 95 C2 53 C1 EB 75 3C C8 8E FA 9F 85 81 F0 8C 47 3..8...S...u<.....G
00000016 AA 71 3E 56 E4 F3 05 8D 3C 90 9A 07 1C F4 66 8A B8 A8 AA 99 07 FD .q>V...<...f.....
0000002C F6 CF 77 BE CF 9F 12 58 50 E8 7B ED F4 00 D6 6E 4B FD 88 8E F3 71 ..w...XP.f...nK...q
00000042 61 91 A3 DC F6 71 FD C1 A1 DB 4C D3 36 D2 9D 0E 58 B3 C0 1C C0 3C a...q...L.6...X...<
00000058 AC FE F4 9F 57 A9 14 AF 68 A7 58 0A 6A 20 EE 39 5C F1 6B 2B 80 98 ...U...h.X.j..9\..k+..
0000006E BB DD 28 6F 09 E6 F1 CC 99 23 32 F1 05 79 2B 5A 9A 47 ..(o...#2...y+Z.G
```

RSA-Entschlüsselung

Wählen Sie Ihren geheimen Schlüssel aus der Liste der PSE's aus:

Name	Vorname	Schlüsseltyp	Schlüsselkennung	Erstellt am	Interne ID-Nr.
Bauer	Petra	RSA-1024	PIN=7435	22.10.2009 14:39:41	1256215181
Meyer	Hans	RSA-1024	PIN=5267	22.10.2009 13:48:54	1256212134
SideChannel	Bob	RSA-512	PIN=1234	08.07.2008 11:51:34	1152179494
Wonderland	Alice	RSA-1024	Alw	27.10.2009 10:22:19	1256635339

Eingabe unvollständig

Bitte geben Sie den PIN-Code ein.

OK

Bemerkung: Gezeigt nur Namen mit einem RSA Schlüssel angezeigt. PIN-Code:

☒ Zum Entschlüsseln benötigte Zeit anzeigen

Entschlüsseln Abbrechen

Auswahl eines Empfängers, um dessen geheimen Schlüssel zu benutzen.

PIN-Eingabe erforderlich!

```
RSA-Entschlüsselung von <RSA-Verschlüsselung von <Original.txt> für <Alice Wonderland>>

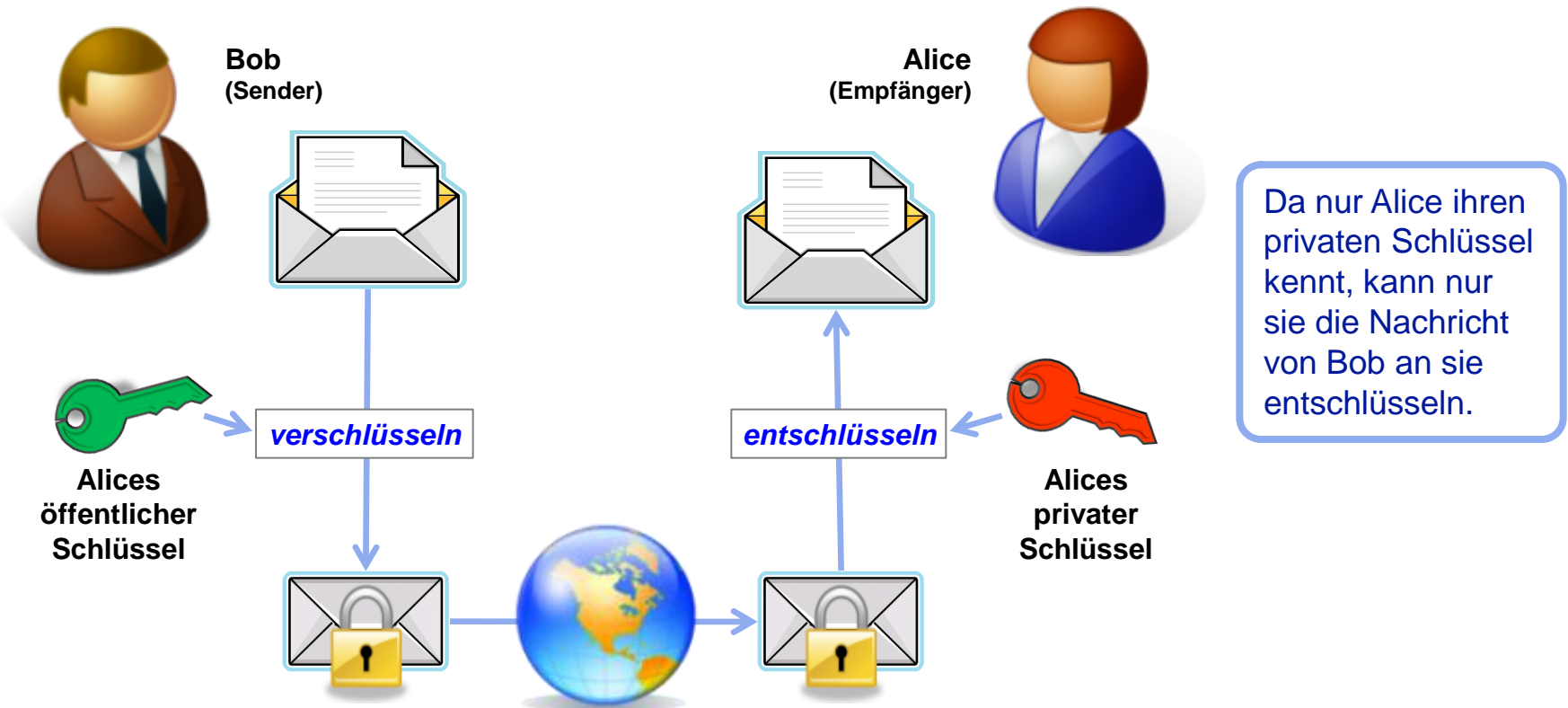
00000000 53 65 68 72 20 67 65 65 68 72 74 65 72 20 48 65 72 72 20 45 69 6E Sehr geehrter Herr Ein
00000016 6B 61 75 66 67 65 72 6E 2C 0D 0A 0D 0A 62 69 74 74 65 20 62 65 73 kaufgern,...bitte bes
0000002C 74 65 6C 6C 65 6E 20 53 69 65 20 65 69 6E 65 20 53 63 68 72 65 9 tellen Sie eine Schrei
00000042 62 6D 61 73 63 68 69 6E 65 0D 0A 0D 0A 4D 66 47 2E 0D 0A 50 65 4 bmaschine...MfG...Pet
00000058 65 72 20 47 75 74 65 72 6D 61 6E 6E 0D 0A 00 00 00 00 00 00 0 teller Gutermann.....
0000006E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Original.txt

Sehr geehrter Herr Einkaufern,
bitte bestellen Sie eine Schreibmaschine
MfG.
Peter Gutermann

Aufgabe 3.3 Hätten Sie es gewusst?

- Welcher Teil des Schlüsselpaares wird vom Sender und welcher vom Empfänger verwendet?



Aufgabe 4 Hashverfahren

4.1 Öffnen Sie das Dokument „[Original.txt](#)“ und rufen Sie unter Menü „[Einzelverfahren](#) ⇒ [Hashverfahren](#)“ die Hash-Demo auf.

- Nehmen Sie Veränderungen an dem Dokument vor und beobachten Sie, was mit dem Hash-Wert geschieht.

4.2 Nehmen Sie einen Angriff auf den Hash-Wert vor. Öffnen Sie „[Original.txt](#)“ als harmlose Datei und „[Faelschung.txt](#)“ als gefährliche Datei im Menü „[Analyse](#) ⇒ [Hash-Verfahren](#) ⇒ [Angriff auf den Hash-Wert](#)“.

Wählen Sie als signifikante Bitlänge verschiedene Werte, z.B. 16, 32, 64 und 128 bit und vergleichen Sie die verschiedenen Ergebnisse.
(*Sollte das Verfahren zu lange dauern, brechen Sie ab.*)

- Wie ist das Ergebnis zu bewerten?

4.3 Hätten Sie es gewusst?

- Welche Eigenschaften sollte eine Hashfunktion haben?
- Wozu dient ein Hashwert?

Aufgabe 4.1 Hash-Demo (1)

Hashfunktion:

Eine Funktion, die einer Datei einen Wert fester Länge (**Hashwert**) zuordnet.

Die Länge des Hashwertes ist normalerweise wesentlich kürzer als die Länge der Datei.

In der Praxis werden häufig die Hashfunktionen SHA-1 und MD5 verwendet.



Wahl der Hashfunktion

Text im Original

Hashwert der
Originaldatei

Hash-1 (160 Bit)-Hash für Original.txt

Beschreibung

- Wählen Sie ein Hashverfahren aus und editieren Sie dann unten die Kopie der Originaldatei (Textfeld "Aktuelles Dokument").
- Ganz unten sehen Sie, wie viele Bits sich im Hashwert ändern, wenn Sie das Dokument editieren.

Auswahl der Hashfunktion

SHA-1 (160 Bit)

Darstellung der Hashwerte

☒ hexadezimal ☐ dezimal ☐ binär

Aktuelles Dokument (Ihre Kopie der Originaldatei können Sie hier ändern)

Sehr geehrter Herr Einkaufsgrn,
bitte bestellen Sie eine Schreibmaschine
MfG.
Peter Gutermann

Hashwert der Originaldatei

8A 6E FD 12 AE C2 E7 4D 0A F9 1F 01 BD 1E 72 C3 F0 29

Hashwert der aktuellen Datei

8A 6E FD 12 AE C2 E7 4D 0A F9 1F 01 BD 1E 72 C3 F0 29

Unterschied zwischen dem Hashwert der Orignal- und dem Hashwert der aktuellen Datei

00000000#00000000#00000000#00000000#00000000#00000000#
00000000#00000000#00000000#00000000#00000000#00000000#
00000000#00000000#00000000#00000000#00000000#00000000#
00000000#00000000#
0.0% der Bits unterscheiden sich (0 von 160).
Längste unveränderte Bitfolge: Offset 0, Länge 160.

Dialog beenden

Aufgabe 4.1 Hash-Demo (2)

- Was passiert mit dem Hashwert, wenn das Dokument verändert wird?

Bei der kleinsten Veränderung an dem Originaldokument (z.B. Einfügen eines Leerzeichens) ändert sich der Hashwert des Dokuments.

Unterschied der Hashwerte in Prozent und absolut (Einsen zeigen unterschiedliche Stellen an.)

Vergleich der Hashwerte als Bitfolge: Stellen mit 1 zeigen Unterschied an.

Hash-Demo: SHA-1 (160 Bit)-Hash für Original.txt

Beschreibung

- Wählen Sie ein Hashverfahren aus und editieren Sie dann unten die Kopie der Originaldatei (Textfeld "Aktuelles Dokument").
- Ganz unten sehen Sie, wie viele Bits sich im Hashwert ändern, wenn Sie das Dokument editieren.

Auswahl der Hashfunktion

SHA-1 (160 Bit)

Darstellung der Hashwerte

☒ hexadezimal ☐ dezimal ☐ binär

Aktuelles Dokument (diese Kopie der Originaldatei können Sie hier ändern)

Sehr geehrter Herr Einkaufsgern,
bitte bestellen Sie zwei Schreibmaschine
MfG.
Peter Gutermann

Hashwert der Originaldatei

8A 6E FD 12 AE C2 E7 4D 0A F9 1F 01 BD 1E 72 C3 F0 29

Hashwert der aktuellen Datei

FD 34 A2 21 CF 7A 2E 64 41 01 BD 05 3B 51 DC 9A 4D F9

Unterschied zwischen dem Hashwert der Original- und dem Hashwert der aktuellen Datei

01110111#01011010#01011111#00110011#01100001#10111000#
11001001#00101001#01001011#11111000#10100010#00000100#
10000110#01001111#10101110#01011001#10111011#11010000#
00011001#10010000#
48.8% der Bits unterscheiden sich (78 von 160)
Längste unveränderte Bitfolge: Offset 140, Länge 7

Dialog beenden

Hashwert der veränderten Datei

Position und Länge der längsten unveränderten Folge von Bits

Aufgabe 4.2 Angriff auf den Hashwert (1)

Angriff auf den Hashwert der digitalen Signatur

Der hier implementierte Angriff auf die digitale Signatur beruht auf dem Versuch, zwei verschiedene Nachrichten mit gleichem Hashwert zu finden.

Default-Nachrichten benutzen

"Harmlose" Datei auswählen
Die "harmlose" Datei ist eine Nachricht, von der der Angreifer vermutet, dass der Unterzeichner sie digital signieren wird.

Suchen ...

"Gefährliche" Datei auswählen
Die "gefährliche" Datei ist eine Nachricht, von der der Angreifer nach erfolgreichem Angriff behaupten wird: "Diese Nachricht wurde vom Unterzeichner digital signiert."

Suchen ...

Suche starten / Optionen festlegen

Mit "Nachrichtenpaar suchen" starten Sie den Versuch, zu den oben eingestellten Nachrichten zwei Modifikationen zu finden, die denselben Hashwert haben.
Der Sinn (Semantik) der Nachrichten ändert sich während der Suche nicht, da zum Modifizieren nicht darstellbare bzw. Formatierungszeichen verwendet werden.

Unter "Optionen" können Sie das Hashverfahren, die Anzahl der für den Vergleich der Hashwerte herangezogenen Bits sowie verschiedene Modifikationsverfahren auswählen.

Wahl der Dokumente:
harmlos und gefährlich

Wahl der Hashfunktion

Anzahl der ersten Bits zweier Hashwerte, die übereinstimmen sollen, damit ein Angriff als erfolgreich angesehen wird (Wertebereich abhängig von der Hashfunktion).

Dies simuliert eine 16 bit Hashfunktion.

Hier ist ein generischer Angriff implementiert, der gegen jedes Hashverfahren gefahren werden kann.

Optionen für den Angriff auf den Hashwert der digitalen ...

Hashfunktion

Wählen Sie eines der sechs Hashverfahren sowie die Anzahl der Bits, die für den Vergleich der Hashwerte herangezogen werden sollen.

☐ MD2 ☐ MD4 ☐ MD5
☐ SHA ☒ SHA-1 ☐ RIPEMD-160

Signifikante Bitlänge (Wertebereich: 1 - 160)

Optionen für die Nachrichtenmodifikation

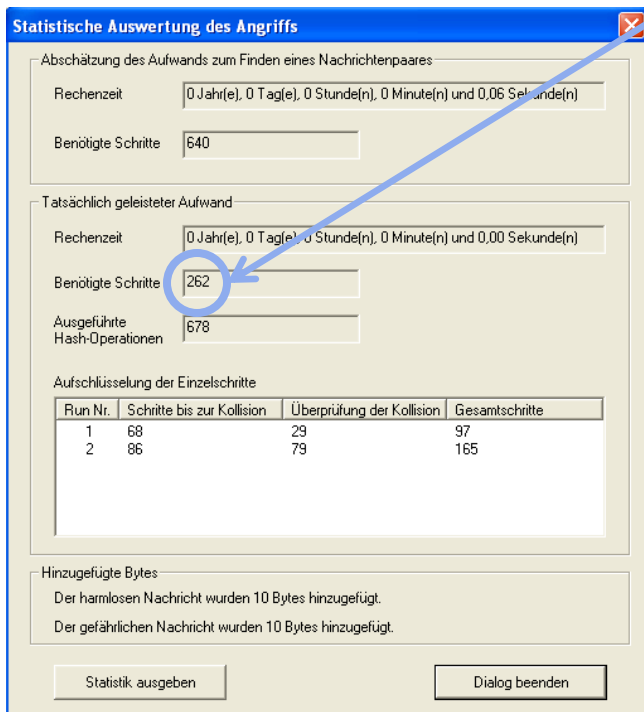
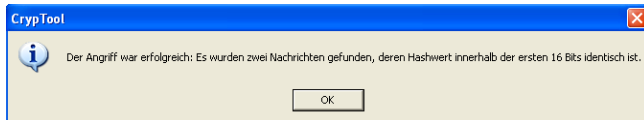
Entscheiden Sie, nach welchem Verfahren die Nachrichten modifiziert werden sollen.

☐ Leerzeichen einfügen ☒ Vor Zeilenende
☒ Leerzeichenverdoppelung
☒ Zeichen anhängen ☒ Druckbare Zeichen (zur Demonstration)
☐ Nicht druckbare Zeichen

Wahl der Art der Modifikation am Text

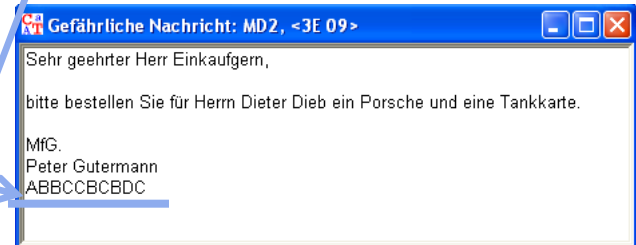
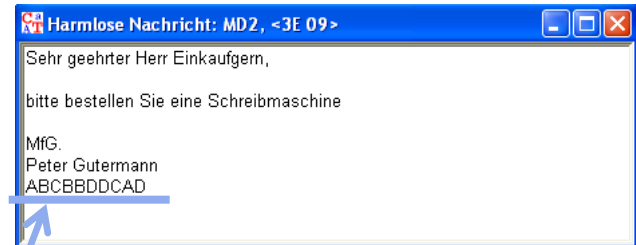
Aufgabe 4.2 Angriff auf den Hashwert (2)

- Signifikante Bitlänge von 16, 32, 64 und 128 bit



Anzahl der benötigten Schritte (Modifikationen an den Texten) zum Finden eines passenden Nachrichtenpaares

Hier sind die Veränderungen an den Texten sichtbar.



Wäre als Modifikationsoption „Leerzeichen einfügen“ gewählt worden, wären keine Veränderungen sichtbar!



Es handelt sich hierbei um einen sogenannten Geburtstagsangriff. Der Name leitet sich ab von dem Geburtstagsparadoxon, mit welchem das Verfahren eng verwandt ist.



Aufgabe 4.2 Angriff auf den Hashwert (3)

- Wie ist das Ergebnis zu bewerten?

Signifikante Bitlänge: 32

Statistische Auswertung des Angriffs

Schätzung des Aufwands zum Finden eines Nachrichtenpaares:

Rechenzeit: 0 Jahr(e), 0 Tag(e), 0 Stunde(n), 0 Minute(n) und 1,72 Sekunde(n)

Benötigte Schritte: 163.840

Tatsächlich geleisteter Aufwand:

Rechenzeit: 0 Jahr(e), 0 Tag(e), 0 Stunde(n), 0 Minute(n) und 0,29 Sekunde(n)

Benötigte Schritte: 199.392

Ausgeführte Hash-Operationen: 500.964

Aufschlüsselung der Einzelschritte

Run Nr.	Schritte bis zur Kollision	Überprüfung der Kollision	Gesamtschritte
1	107.180	92.212	199.392

Hinzugefügte Bytes:

Der harmlosen Nachricht wurden 18 Bytes hinzugefügt.

Der gefährlichen Nachricht wurden 18 Bytes hinzugefügt.

Statistik ausgeben Dialog beenden

Anzahl der benötigten Schritte zum Finden eines Nachrichtenpaares

In der Praxis übliche Bitlänge von Hashfunktionen: 128 - 160 bit

Ein Nachrichtenpaar wird gesucht ...

Run 1
Zyklussuche (64 Bit)
Fortschritt: 0% Restzeit: 05:22

Abbrechen

Signifikante Bitlänge: 64

Geschätzte Zeit zum Finden eines Nachrichtenpaares

Ein Nachrichtenpaar wird gesucht ...

Run 1
Zyklussuche (128 Bit)
Fortschritt: 0% Restzeit: 1.6e+094 Jahre

Abbrechen

Signifikante Bitlänge: 128

Ist die Länge der Hashfunktion zu kurz gewählt, sind Angriffe auf den Hashwert möglich!
Je länger die Hashfunktion, desto schwieriger ist es, zwei Dokumente mit demselben Hashwert zu finden.

→ Hashverfahren sind sicher, wenn sie korrekt angewendet werden!

Aufgabe 4.3 Hätten Sie es gewusst?

- Welche Eigenschaften sollte eine Hashfunktion haben?
- Wozu dient ein Hashwert?

Es sollte technisch unmöglich sein, zwei verschiedene Dateien zu finden, die denselben Hashwert haben.
→ Vermeidung von Kollisionen

Es gibt beliebig viele verschiedene Dokumente, die den gleichen Hashwert haben. Es sollte nur nicht möglich sein, sie zu finden.



Ein Hashwert eröffnet die Möglichkeit, schnell festzustellen, ob Veränderungen an einem Dokument vorgenommen wurden.

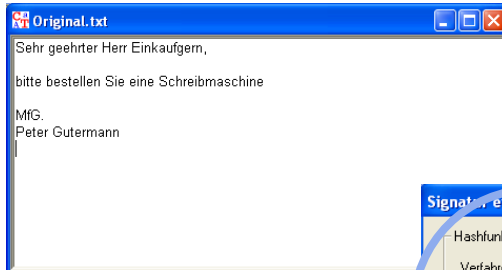
Ein Hashwert ist ein digitaler Fingerabdruck. Er wird z.B. bei der digitalen Signatur verwendet.



Aufgabe 5 Digitale Signatur

- 5.1 Öffnen Sie das Dokument „[Original.txt](#)“ (Unterordner „[Examples](#)“) und signieren Sie es mit Ihrem RSA-Schlüssel über den Menüpfad „[Digitale Signaturen/PKI](#) ⇒ [Dokument signieren](#)“.
- 5.2 Versetzen Sie sich in die Lage des Empfängers und verifizieren Sie die empfangene Signatur („[Digitale Signaturen/PKI](#) ⇒ [Signatur überprüfen](#)“). Lassen Sie sich dabei die Zwischenschritte anzeigen.
- 5.3 Hätten Sie es gewusst?
- Welcher Schlüssel wird zur Erstellung der Signatur verwendet?
 - Welcher Schlüssel wird zur Verifizierung der Signatur verwendet?
 - Warum ist vor der Signatur-Erstellung eine PIN-Eingabe notwendig?
 - Wie kann die Echtheit einer Signatur überprüft werden?
 - Was kann daraus geschlossen werden, wenn zwei Hashwerte gleich sind?
 - Welche Eigenschaften hat eine digitale Signatur?

Aufgabe 5.1 Signatur erstellen (1)



Digitale Signatur = digitale Unterschrift eines Dokuments

- dient der Authentifizierung, Unleugbarkeit und Datenintegrität

Der Hashwert einer Nachricht wird berechnet und anschließend signiert.

Signiert wird der Hashwert des Dokuments und nicht das Dokument selbst, da so weniger Rechenaufwand nötig ist.



h Hashwert eines Dokuments
d privater Schlüssel des Absenders

Signatur mit RSA:
$$S \equiv h^d \pmod{N}$$

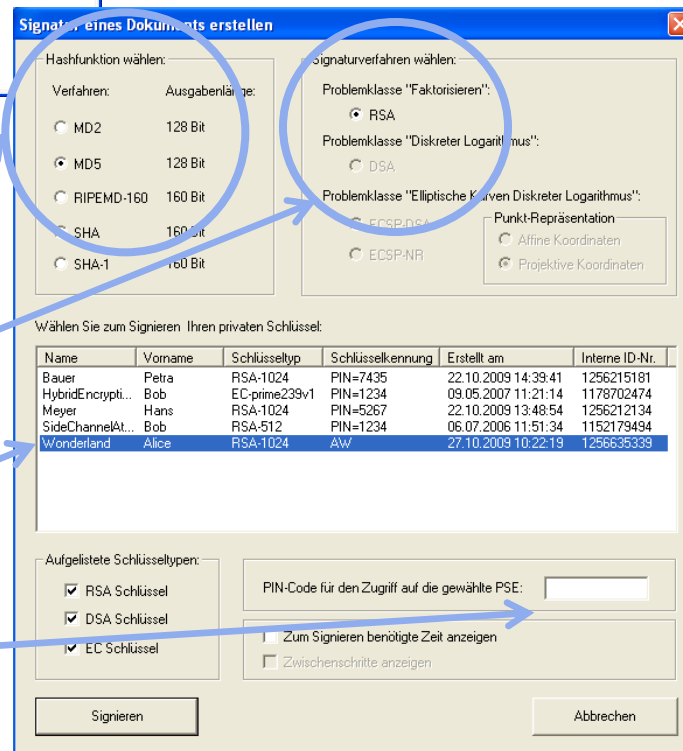
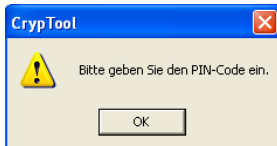


Wahl der Hashfunktion zur Berechnung des Hashwertes von Original.txt

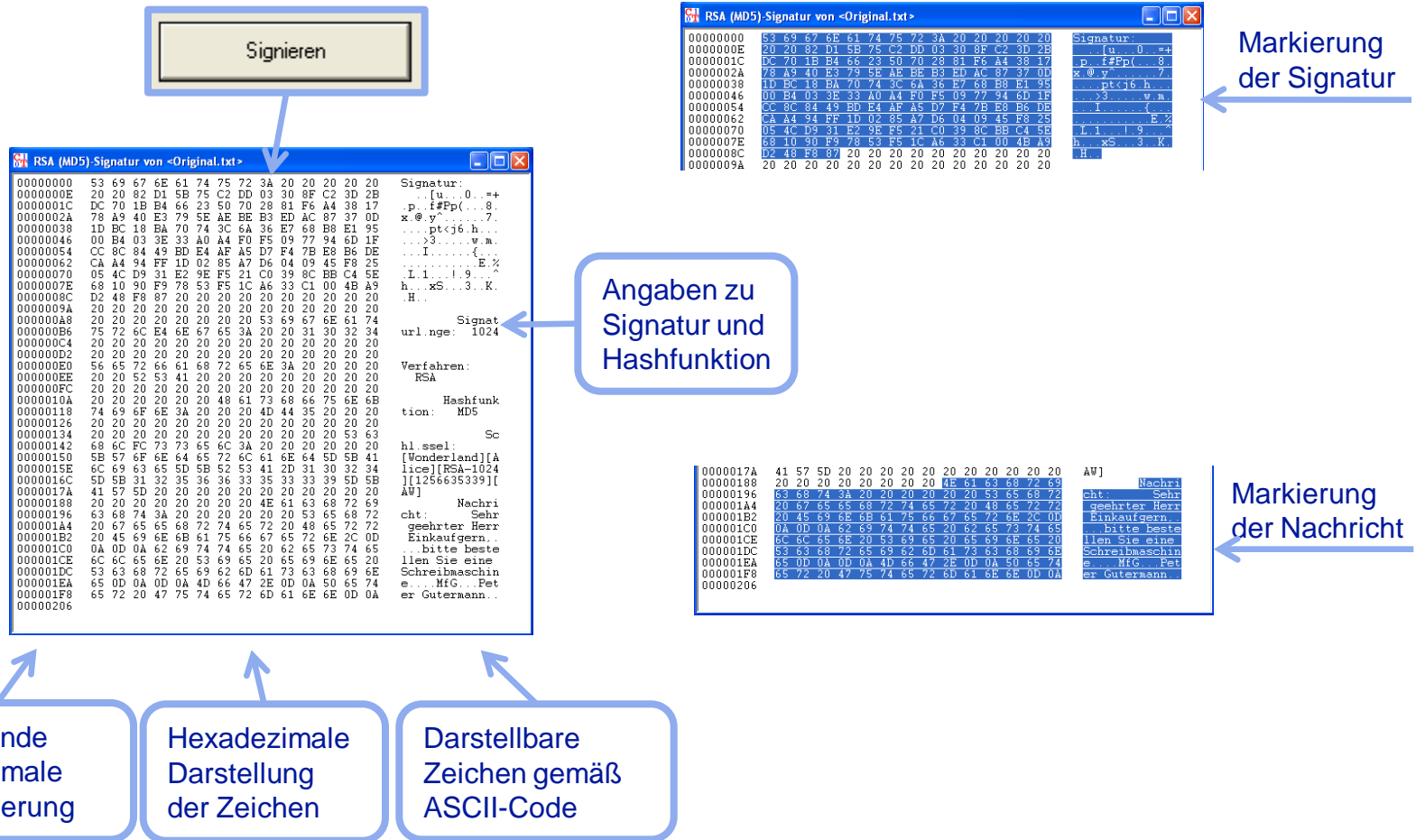
Wahl des Signaturverfahrens

Wahl des eigenen Schlüssels zur Signierung

PIN-Eingabe erforderlich!



Aufgabe 5.1 Signatur erstellen (2)



Ein von Alice signiertes Dokument wurde empfangen.

Auswahl des Absenders des signierten Dokuments

Zwischenschritte anzeigen lassen

Verifizieren einer Signatur

Wählen Sie aus der folgenden Liste den Signatursteller:

Name	Vorname	Signatur-Verfahren	Schlüssel-Verfahren	Erstellt am	Interne ID-Nr.
Bauer	Petra	RSA-1024	PIN=7435	22.10.2009 14:39:41	1256215181
HybridEncrypt...	Bob	EC prime239v1	PIN=1234	09.05.2007 11:21:14	1178702474
Meyer	Hans	RSA-1024	PIN=5387	22.10.2009 13:48:54	1256212134
SideChannelAt...	Bob	RSA-512	PIN=1234	06.07.2006 11:51:34	1152179494
Wonderland	Alice	RSA-1024	AW	27.10.2009 10:22:19	1256635339

Angabe der Daten

Signatur-Verfahren: RSA

Hashfunktion: MD5

Aufgelistete Schlüsselpaare:

☒ RSA-Schlüssel

☒ DSA-Schlüssel

☒ EC-Schlüssel

Verifikation mit Verfahren:

☐ ECSPDSA ☐ ECSPNR

Verifikation mit Hashfunktion:

☐ SHA-1 ☐ RIPEMD-160

Suche Schlüssel

☐ Zwischen Schritte benötigte Zeit anzeigen

☒ Zwischenschritte anzeigen

Repräsentation der EC-Punkte in:

☐ Affine Koord. ☐ Projektive Koord.

Signatur verifizieren

Abbrechen

Zwischenschritte anzeigen lassen

- ### Der Empfänger
1. berechnet den Hashwert der empfangenen Nachricht und
 2. vergleicht diesen mit dem signierten Hashwert.

Hashwert der
empfangenen Nachricht

Aufgabe 5.3 Hätten Sie es gewusst? (1)

- Welcher Schlüssel wird zur Erstellung der Signatur verwendet?
- Warum ist vor der Signaturerstellung eine PIN-Eingabe notwendig?

Der Absender nutzt zur Signierung seinen privaten Schlüssel.

$$S \equiv h^d \pmod{N}$$



- Welcher Schlüssel wird zur Verifizierung der Signatur verwendet?

Der Empfänger nutzt zur Verifizierung der Signatur den öffentlichen Schlüssel des Absenders.

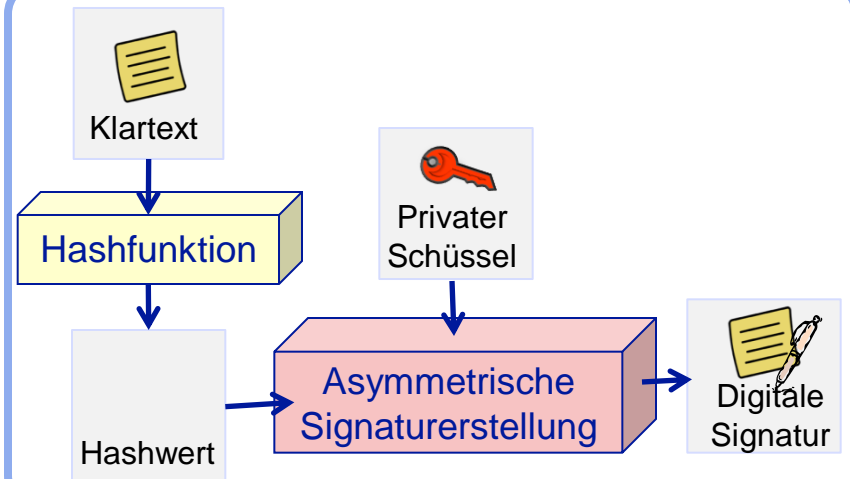


Dazu wird zuerst der Hashwert des Originaldokuments aus der Signatur berechnet.

$$S^e \equiv h^{ed} \equiv h \pmod{N}$$



Zur Signaturerstellung wird der private Schlüssel genutzt. Dieser wird durch eine PIN geschützt, die nur der Eigentümer des Schlüssels kennt. So kann sicher gestellt werden, dass auch nur der Eigentümer eine Signatur mit diesem Schlüssel erstellen kann.



Aufgabe 5.3 Hätten Sie es gewusst? (2)

- Wie kann die Echtheit einer Signatur überprüft werden?
- Was bedeutet es, wenn die Hashwerte gleich sind?

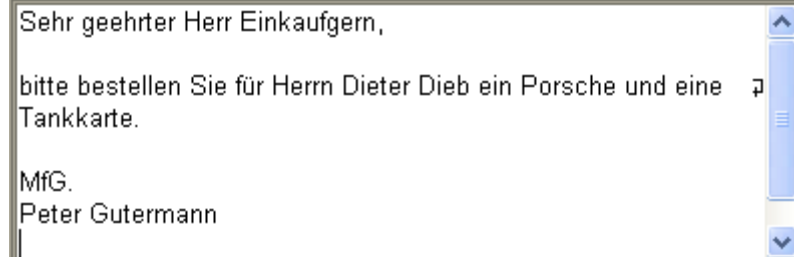
Der Hashwert wird zweimal berechnet:

- a) Von dem empfangenen Dokument mit der gleichen Hashfunktion wie bei der Signaturerstellung
 - b) Mithilfe des öffentlichen Schlüssels aus der empfangenen Signatur.
- Diese Werte werden verglichen.

Sind Originalhashwert und der Hashwert des empfangenen Dokumentes gleich, ist die Signatur gültig.
Mit anderen Worten, das Dokument wurde nicht verändert.

Beispiel

Bestellungen

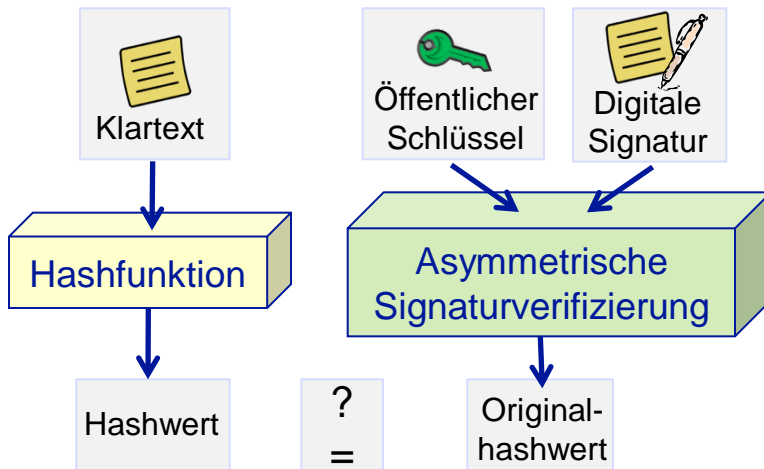


Sehr geehrter Herr Einkaufsfern,

bitte bestellen Sie für Herrn Dieter Dieb ein Porsche und eine Tankkarte.

MfG.
Peter Gutermann

Hat wirklich Herr Gutermann diese Bestellung verfasst?



Aufgabe 5.3 Hätten Sie es gewusst? (3)

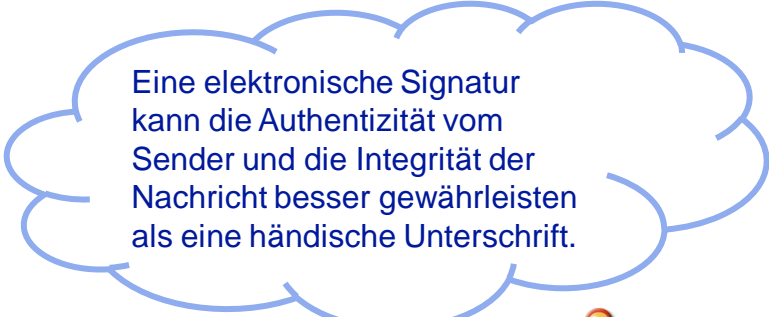
- Welche Eigenschaften hat eine digitale Signatur?

Die Signatur ist nicht nur abhängig von dem Absender, sondern auch von der zu übertragenden Nachricht.


→ Datenintegrität

Es ist sicher gestellt, dass das empfangene Dokument nur von einer bestimmten Person stammen kann.

→ Nachrichtenauthentizität



Eine elektronische Signatur kann die Authentizität vom Sender und die Integrität der Nachricht besser gewährleisten als eine händische Unterschrift.



Signierte Emails wären ein guter Schutz gegen Spam!







Aufgabe 6 Signaturdemo

Die Signaturerstellung soll nun Schritt für Schritt nachvollzogen werden (Menü „[Digitale Signaturen/PKI](#) ⇒ [Signaturdemo](#)“). Wählen Sie dabei im Schritt „Zertifikat bereit stellen“ wieder Ihren eigenen RSA-Schlüssel.

Erläuterungen zur Signaturdemo

Zur Visualisierung sind verschiedene Buttons miteinander verbunden, die farblich anzeigen, wie weit fortgeschritten der Arbeitsprozess ist. Die Buttons können per Mausklick aktiviert werden.

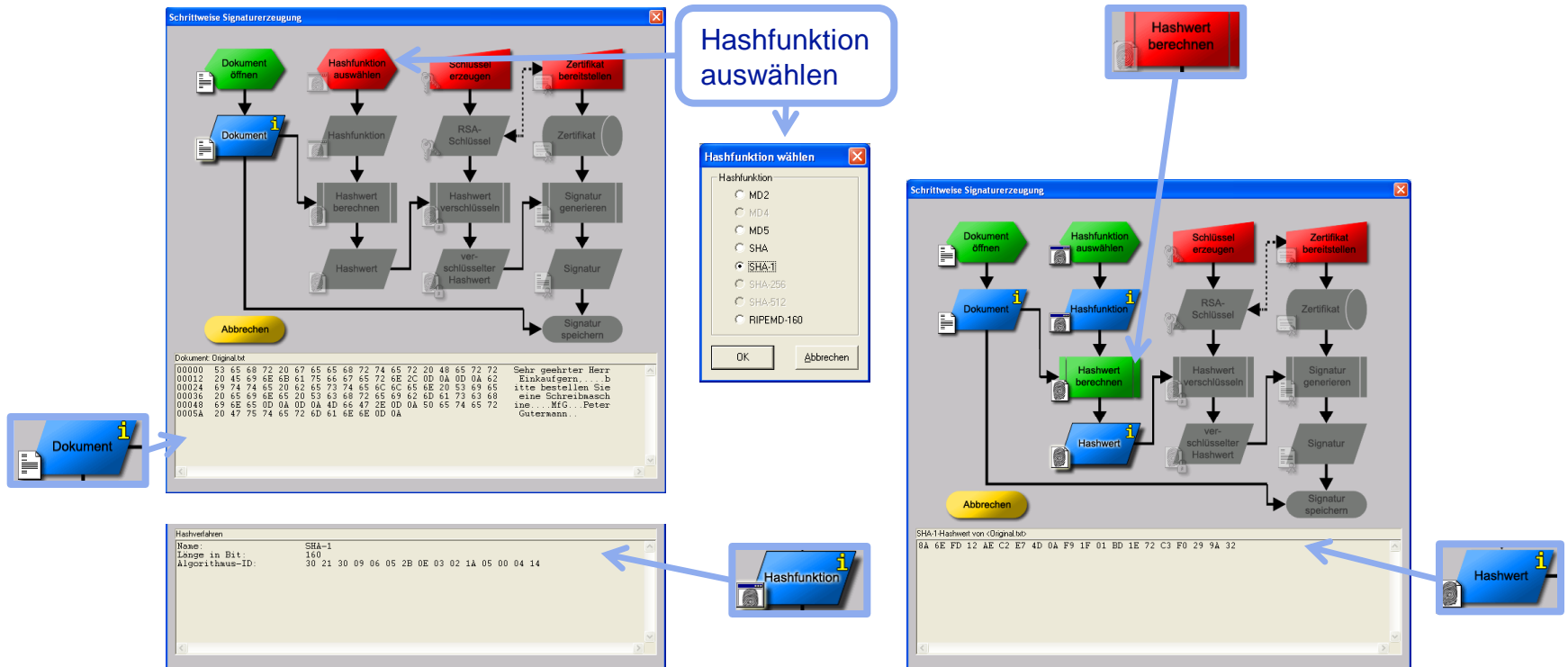
Als Buttons werden verschiedene Symbole verwendet:

-  Symbol für vorbereitende Schritte
-  Symbol für manuelle Dateneingabe
-  Symbol für Daten allgemein
-  Symbol für gespeicherte Daten
-  Symbol für eine Verarbeitungsfunktion
-  Symbol für das Ende eines Programmablaufs

Die Farben der Symbole geben Aufschluss über den Zustand der jeweiligen Arbeitsschritte:

- Grau** Inaktiv, fehlende Dateien oder noch nicht ausgeführter Prozess
- Rot** Zu erledigender Arbeitsschritt
- Grün** Bereits erledigter Arbeitsschritt
- Blau** Vorhandene Dateien

Aufgabe 6 Signaturdemo (1)



Im nächsten Schritt braucht man seinen privaten Schlüssel:

- Entweder erzeugt man sich ein neues Schlüsselpaar,
- Oder man wählt seines aus den schon erzeugten Schlüsselpaaren.

Aufgabe 6 Signaturdemo (2)



PSE = eine Art Tresor, der sowohl den privaten Schlüssel als auch das eigene Zertifikat enthält.

Vorhandene PSE auswählen

PIN Eingabe erforderlich!



Zertifikat und PSE erzeugen

Öffentliche RSA-Parameter:

Bitlänge:

RSA-Modul N:

Öffentlicher Schlüssel e:

Persönliche Daten für das Zertifikat:

Name:

Vorname:

Schlüsselbezeichner: (optional)

PIN-Code:

PIN-Verifikation:

Generierte Namen für PSE und Zertifikat:

User Key ID:

Distinguished Name:

Schlüssel und Zertifikat importieren

Wählen Sie Ihren geheimen Schlüssel aus der Liste der PSEs aus:

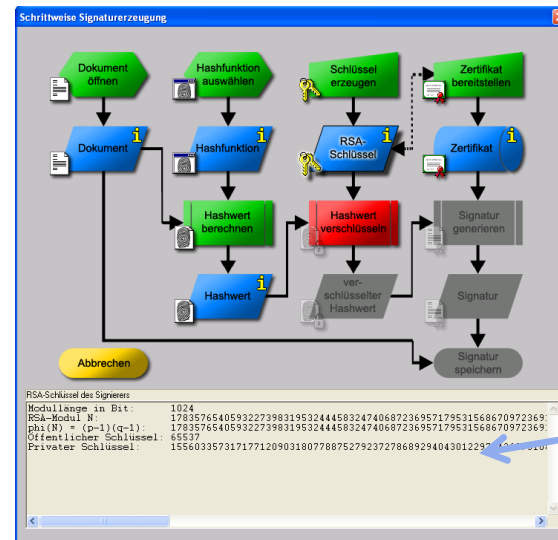
Name	Vorname	Schlüsseltyp	Schlüsselkennung	Erstellt am	Interne ID-Nr.
Bauer	Petra	RSA-1024	PIN=7435	22.10.2009 14:35:41	1256215181
Meyer	Hans	RSA-1024	PIN=5257	22.10.2009 13:45:54	1256212134
Schäfer	Bob	RSA-512	PIN=1234	06.07.2006 11:51:34	1153734834
Wunderland	Alice	RSA-1024	AW	27.10.2009 10:22:19	1256635339

Hinweis: Es werden nur Namen mit einem RSA Schlüssel angezeigt.

PIN-Code:



Zur Signaturerstellung wird der eigene private Schlüssel benutzt.



Zertifikat des Signiers (wird der Signatur zur Prüfung beigefügt)

Version: 2 (X.509v3-1996)

SubjectName: CN=Alice Wonderland [1256635339], DC=crpytool, DC=org

IssuerName: CN=CrypTool CA 2, DC=crpytool, DC=org

SerialNumber: SC=FA:8E:08:AA:25:B8:2F

Validity - NotBefore: Tue Oct 27 10:23:13 2009 (091027092313Z)

Validity - NotAfter: Wed Oct 27 11:23:13 2010 (101027092313Z)

Public Key Fingerprint: 9221 D0B8 5C79 6D17 D20C E20B 2275 988E

SubjectKey: Algorithm: rsa (OID 2.5.8.1.1), KeySize = 1024

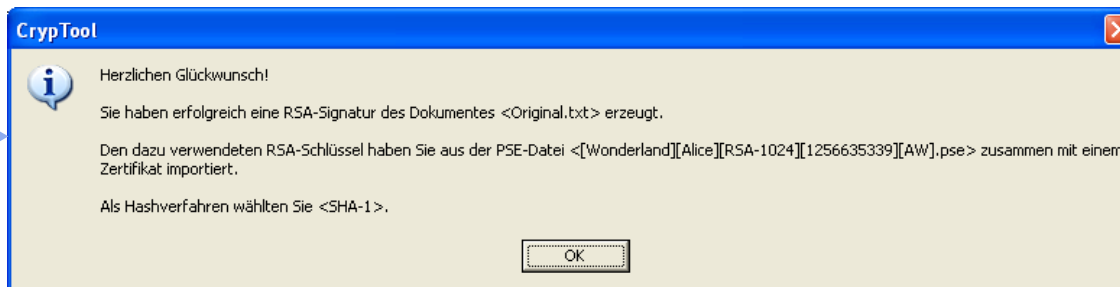
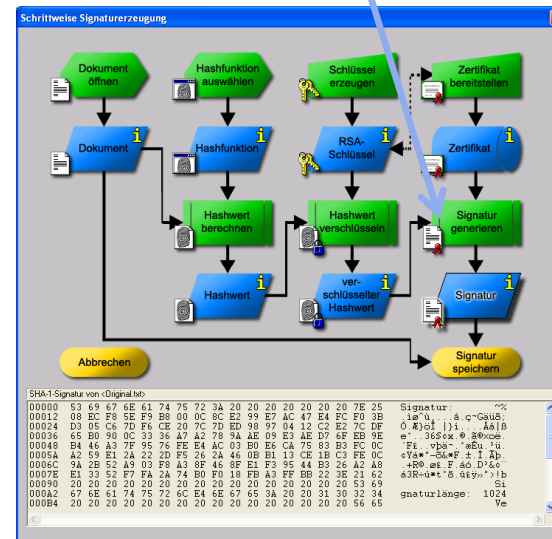
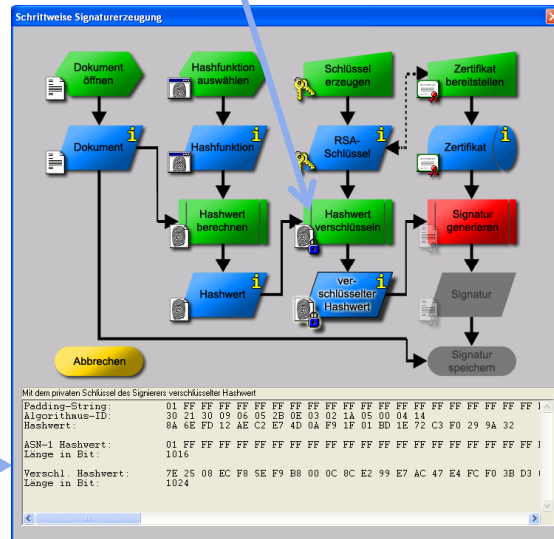
Public modulus (no. of bits = 1024):

0 F0F05E6 25D6F05D DFF6CD45 D21FE843

10 1813B93C E8DA5807 C86C732A AB2797D1



Aufgabe 6 Signaturdemo (3)



Aufgabe 7 PKI (1)



- Bob nutzt öffentliche Schlüssel von Kommunikationspartnern.

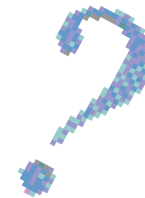
Bob möchte eine verschlüsselte Nachricht an Alice schicken. Dazu benutzt er Alices öffentlichen Schlüssel.



Bob empfängt ein von Alice signiertes Dokument. Die Verifizierung der Signatur führt er mit Alices öffentlichem Schlüssel durch.



Wie kann Bob sicher sein, dass es sich tatsächlich um Alices öffentlichen Schlüssel handelt und dass nicht jemand vorgibt, Alice zu sein?



Aufgabe 7 PKI (2)



- Wie kann Bob sicher sein, dass es sich tatsächlich um Alices öffentlichen Schlüssel handelt und dass nicht jemand vorgibt, Alice zu sein?

- Alice besitzt ein digitales Zertifikat, in dem ihr öffentlicher Schlüssel gespeichert ist.
- Bob kann in dem Zertifikat Alices öffentlichen Schlüssel einsehen.
- Das digitale Zertifikat wird von einer CA (*Certificate Authority*) ausgestellt.
- Das Zertifikat selbst ist durch die digitale Signatur der CA geschützt.
- Die CA benutzt für die Signatur ihren eigenen privaten Schlüssel.
- Die Echtheit der Signatur kann mit dem öffentlichen Schlüssel der CA geprüft werden.

Ein **digitales Zertifikat** wird genutzt um zu bestätigen, dass ein öffentlicher Schlüssel zu einer Person gehört. Hierzu wird wiederum eine digitale Signatur verwendet.

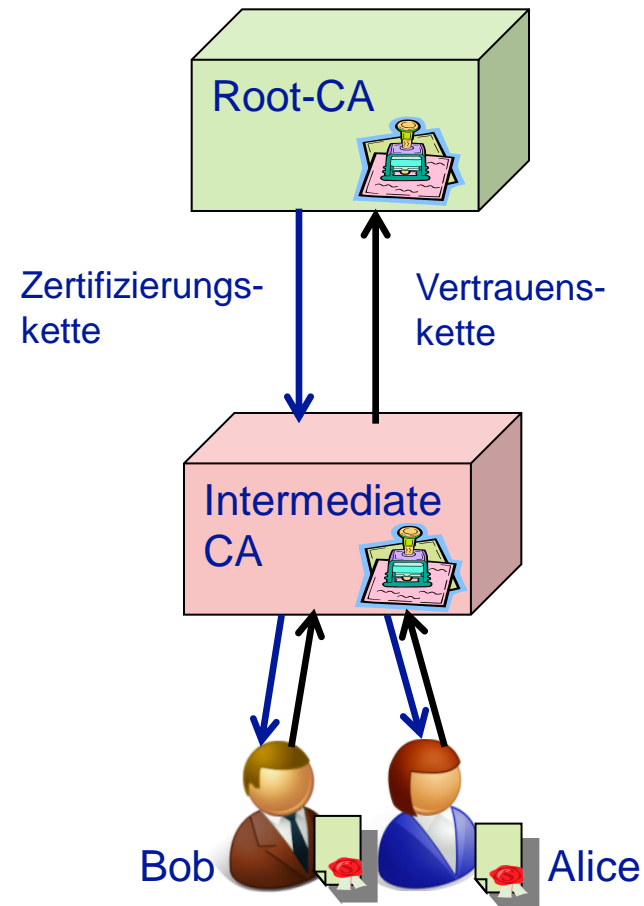
PKI *Public-Key Infrastructure* bezeichnet ein System, welches es ermöglicht, digitale Zertifikate auszustellen, zu verteilen und zu prüfen.

CA *Certificate Authority*
Zertifizierungsstelle;
Organisation, welche das CA-Zertifikat bereitstellt und beantragte Zertifikate signiert.

Aufgabe 7 PKI (3)

- Wie kann sicher gestellt werden, dass es sich bei dem öffentlichen Schlüssel um den der CA handelt?

- Es wird erneut ein digitales Zertifikat benötigt, das der CA ihren öffentlichen Schlüssel bestätigt.
- Dieses wird von einer höheren CA ausgestellt.
- Dadurch entsteht eine Zertifizierungskette mit mehreren sogenannten Intermediate (*Zwischen*-)CAs.
- An der Spitze dieser Kette steht eine sogenannte Root-CA, die als vertrauenswürdig angesehen wird.
- So entsteht eine Vertrauenskette von der Root-CA über die Intermediate CAs zu Bob und Alice.
- Der öffentliche Schlüssel der Root-CA muss jedem bekannt oder leicht und sicher zugänglich sein.



Aufgabe 8 Hybride Verschlüsselung

In der Praxis wird eine Kombination von symmetrischer und Public-Key-Verschlüsselung genutzt.



Hybride Verschlüsselung

Ein symmetrischer Schlüssel wird zur Ver- und Entschlüsselung von Daten verwendet (auch Sessionkey genannt). Der symmetrische Schlüssel wird dann mit dem Public-Key-Verfahren verschlüsselt und so zwischen Sender und Empfänger ausgetauscht.

Begründung für den Vorteil der Hybridverschlüsselung:
Der Sessionkey ist im Vergleich zum gesamten Klartext sehr kurz, weshalb seine asymmetrische Verschlüsselung sehr schnell geht!



Nur symmetrische Verschlüsselung:

Vorteil:

Der verschlüsselte Text kann verhältnismäßig schnell erstellt werden.

Problem:

Wie kann der Schlüsselaustausch zwischen Sender und Empfänger sicher vonstatten gehen?

Nur asymmetrische Verschlüsselung:

Vorteil:

Es werden Schlüsselpaare benutzt. So ist kein Austausch von privaten Schlüsseln nötig.

Problem:

Die Erstellung des gesamten Chiffretextes dauert deutlich länger.

Aufgabe 8 Hybrid-Verschlüsselung am Beispiel RSA und AES

8.1 Wählen Sie im Menü „**Ver-Entschlüsseln** ⇒ **Hybrid** ⇒ **RSA-AES-Verschlüsselung**“ und gehen Sie Schritt für Schritt die Demo durch, um die Hybrid-Verschlüsselung anzuwenden. (Beachten Sie, dass das Dokument an Sie gesendet werden soll.)

Achten Sie dabei darauf:

- In welchen Schritten wird welches Verfahren (symmetrisch oder asymmetrisch) benutzt?
- Welcher Schlüssel wird dabei jeweils verwendet?

8.2 Im nächsten Schritt entschlüsseln Sie das Dokument unter Menü:
„**Ver-Entschlüsseln** ⇒ **Hybrid** ⇒ **RSA-AES-Entschlüsselung**“

Achten Sie auch hier darauf:

- In welchen Schritten wird welches Verfahren (symmetrisch oder asymmetrisch) benutzt?
- Welcher Schlüssel wird dabei jeweils verwendet?

8.3 Hätten Sie es gewusst?

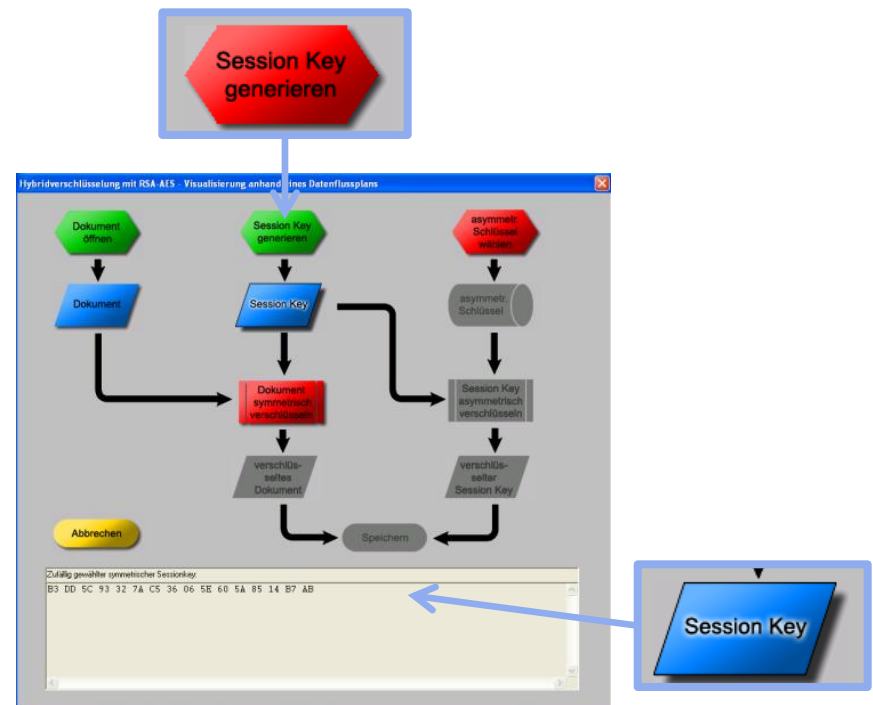
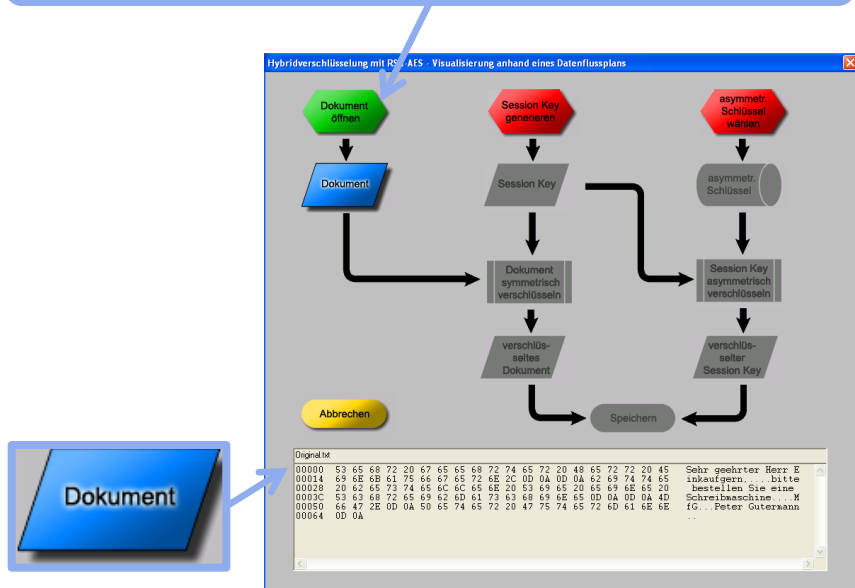
- Warum benutzt man hybride Verschlüsselung?

AES, *Advanced Encryption Standard*, ist der Standard für moderne symmetrische Verschlüsselung. AES wird auch Rijndael-Algorithmus genannt, nach seinen Erfindern V. Rijmen und J. Daemen.

Aufgabe 8.1 Hybrid-Verschlüsselung (1)

Das zu verschlüsselnde Dokument wird ausgewählt.

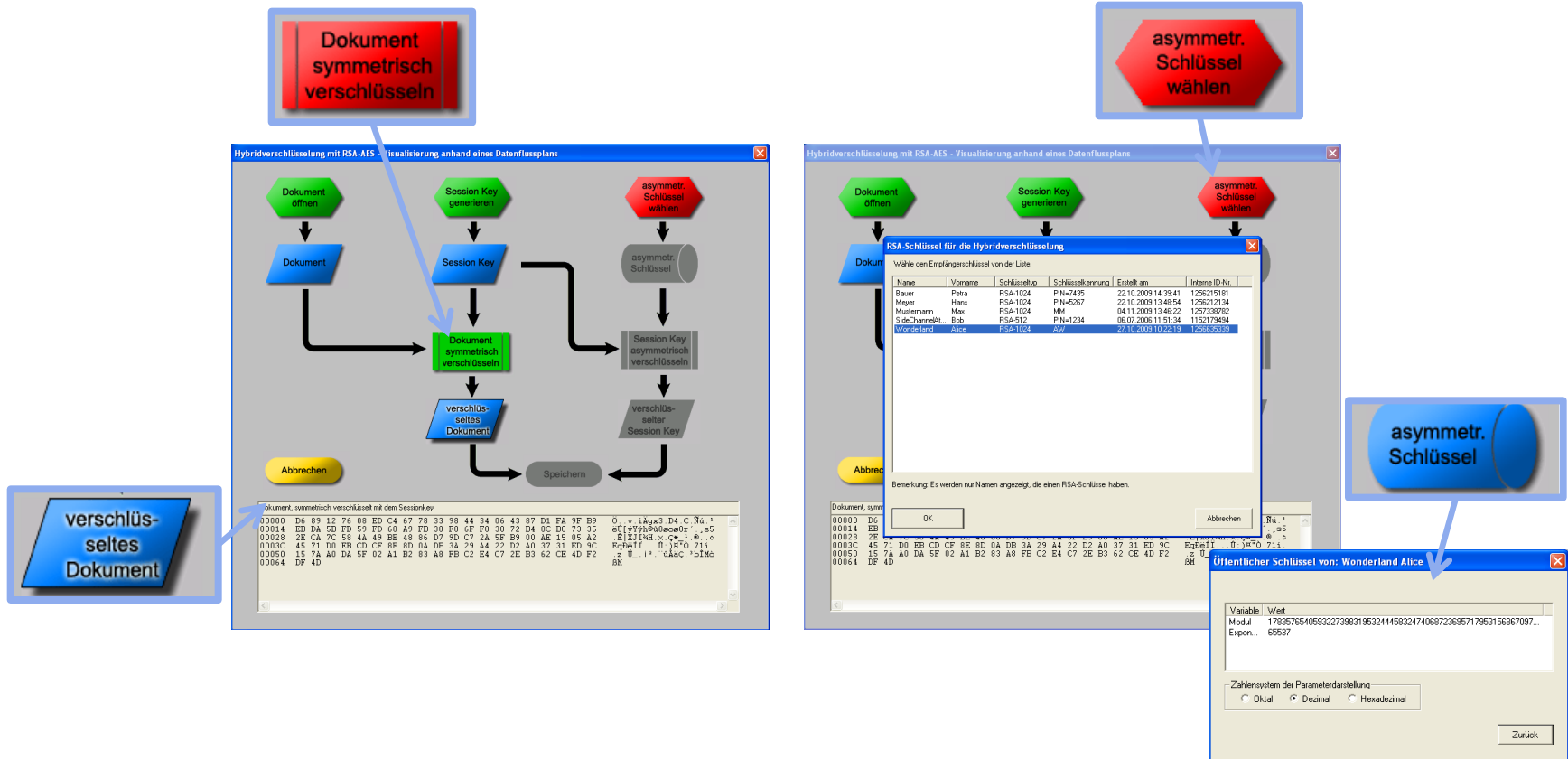
Per Zufall wird ein symmetrischer Schlüssel erzeugt:
der Sessionkey.
Mit diesem Schlüssel wird das Dokument verschlüsselt.



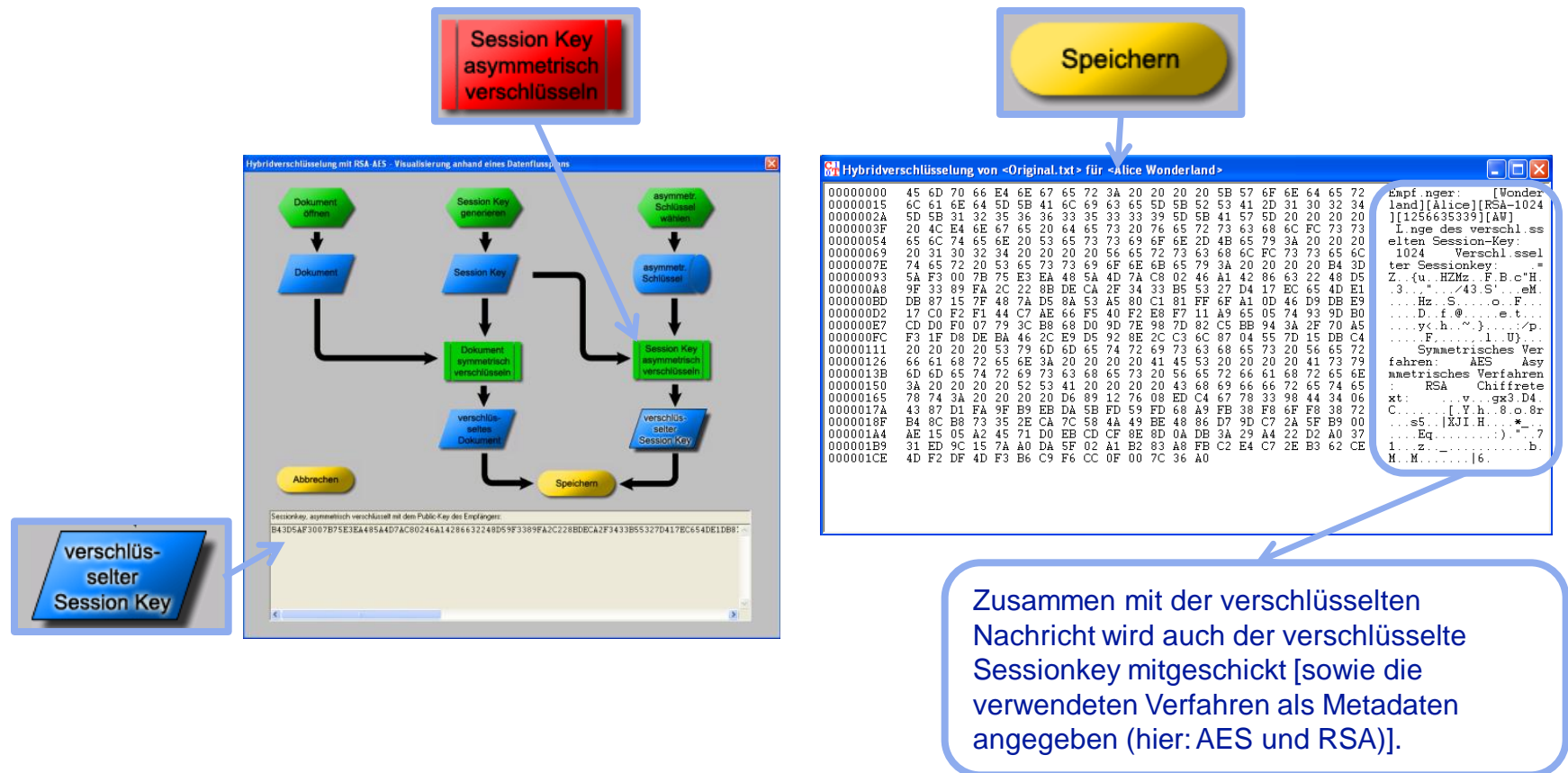
Aufgabe 8.1 Hybrid-Verschlüsselung (2)

Das Dokument wird mit dem Sessionkey symmetrisch verschlüsselt.

Empfängerschlüssel auswählen.
Der Sessionkey wird dann asymmetrisch mit dem öffentlichen Schlüssel des Empfängers verschlüsselt.

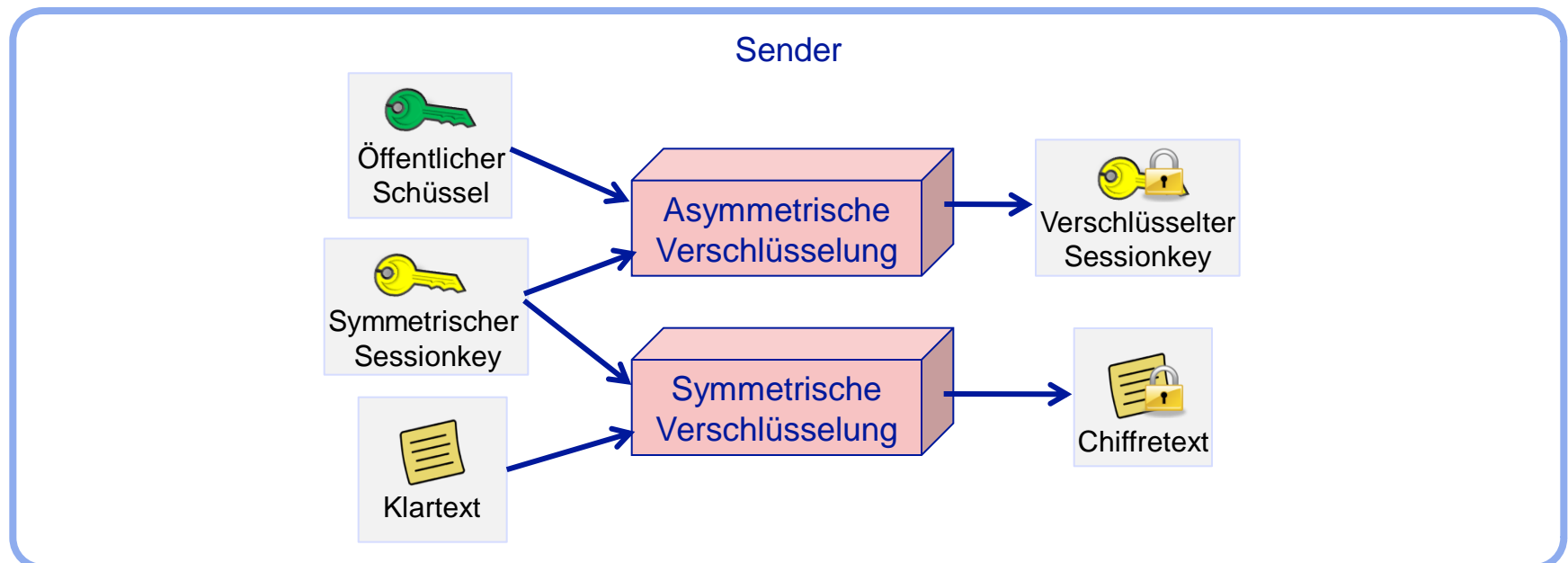


Aufgabe 8.1 Hybrid-Verschlüsselung (3)

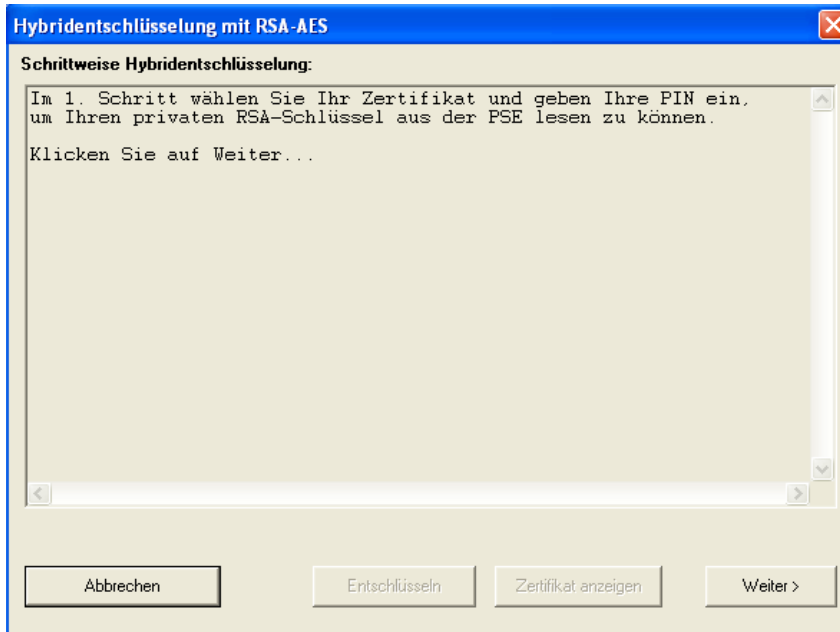


Aufgabe 8.1 Hätten Sie es gewusst?

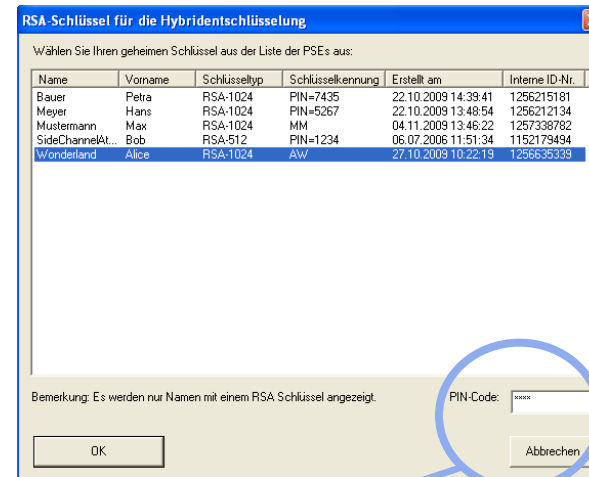
- In welchen Schritten wird welches Verfahren benutzt?
 - Welcher Schlüssel wird dabei jeweils verwendet?
1. Erzeugung eines symmetrischen Schlüssels: Sessionkey.
 2. Damit verschlüsselt der Sender die Nachricht. (symmetrisch)
 3. Der Sessionkey wird mit dem öffentlichen Schlüssel des Empfängers verschlüsselt. (asymmetrisch)
 4. Die verschlüsselte Nachricht sowie der verschlüsselte Sessionkey werden versendet.



Aufgabe 8.2 Hybrid-Entschlüsselung (1)



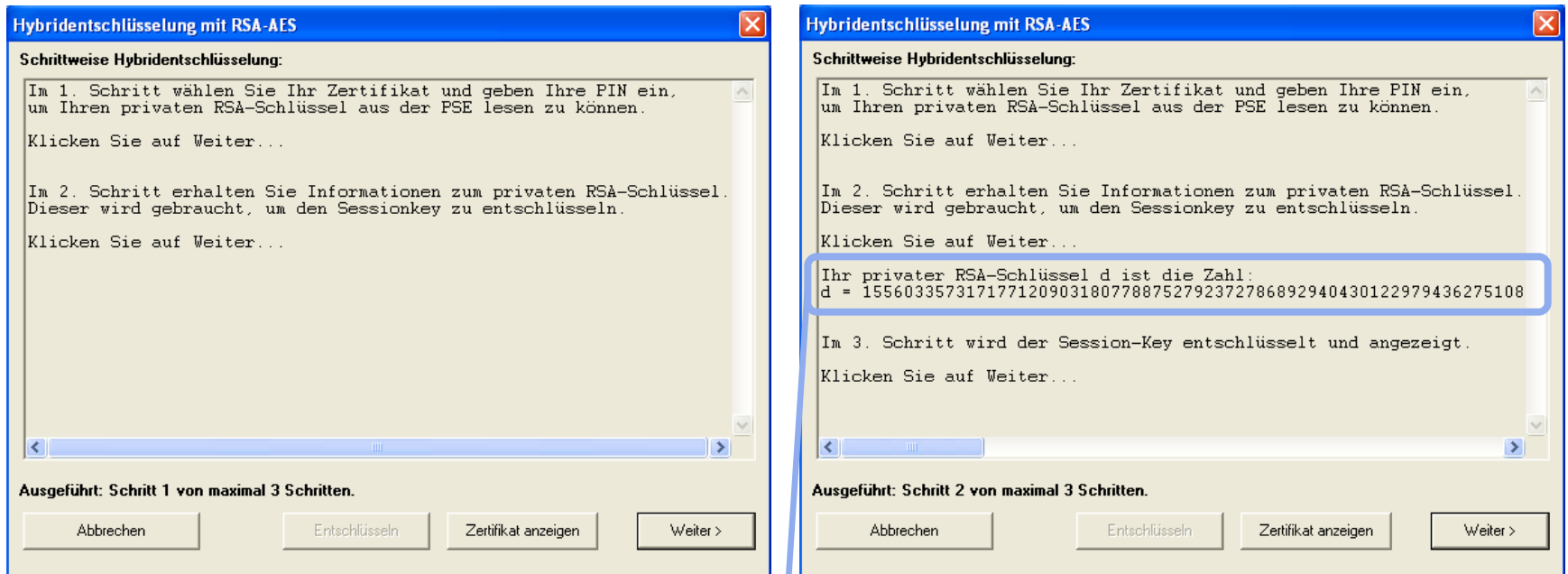
Um die verschlüsselte Nachricht lesen zu können, muss zunächst der symmetrische Sessionkey entschlüsselt werden. Dazu wird der private Schlüssel des Empfängers benötigt.



Um an den privaten Schlüssel zu kommen, muss man die PSE öffnen (PSE = eine Art Tresor für den privaten Schlüssel).

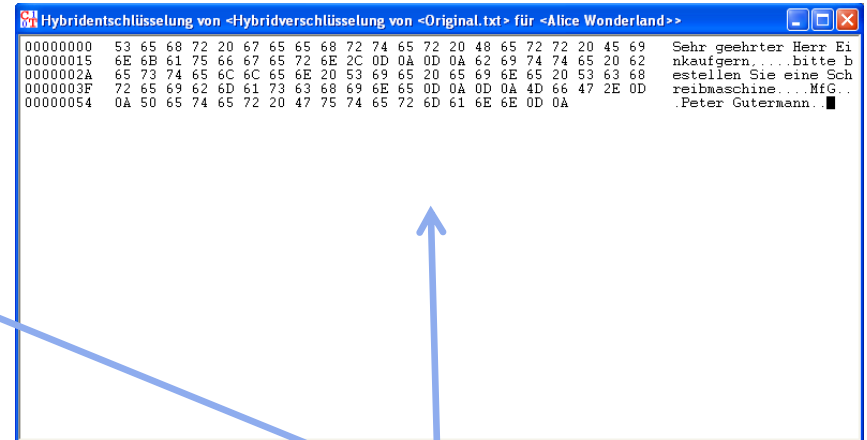
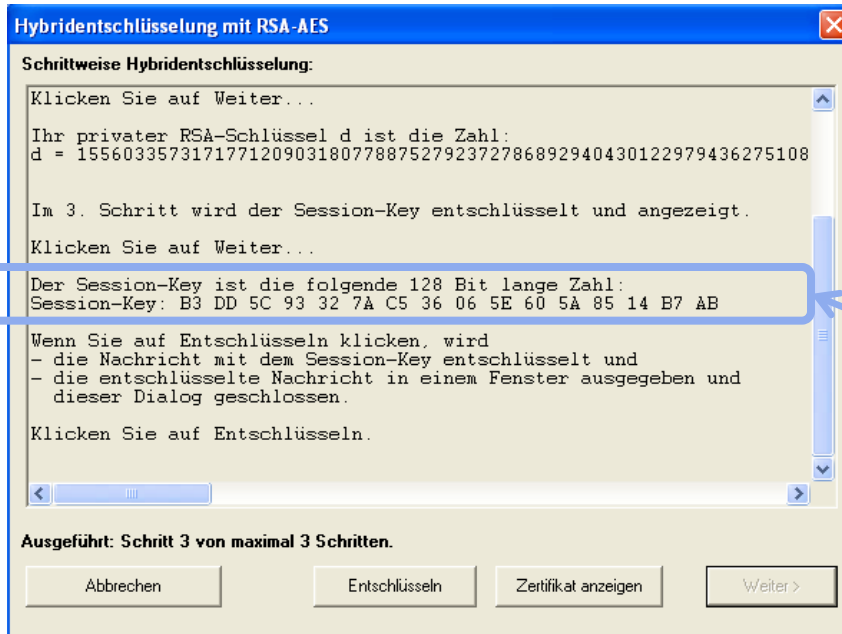
Um sicher zu stellen, dass nur der berechtigte Besitzer Zugriff auf seinen privaten Schlüssel hat, ist eine PIN-Eingabe erforderlich.

Aufgabe 8.2 Hybrid-Entschlüsselung (2)



Mit Hilfe des privaten Schlüssels wird der Sessionkey entschlüsselt.

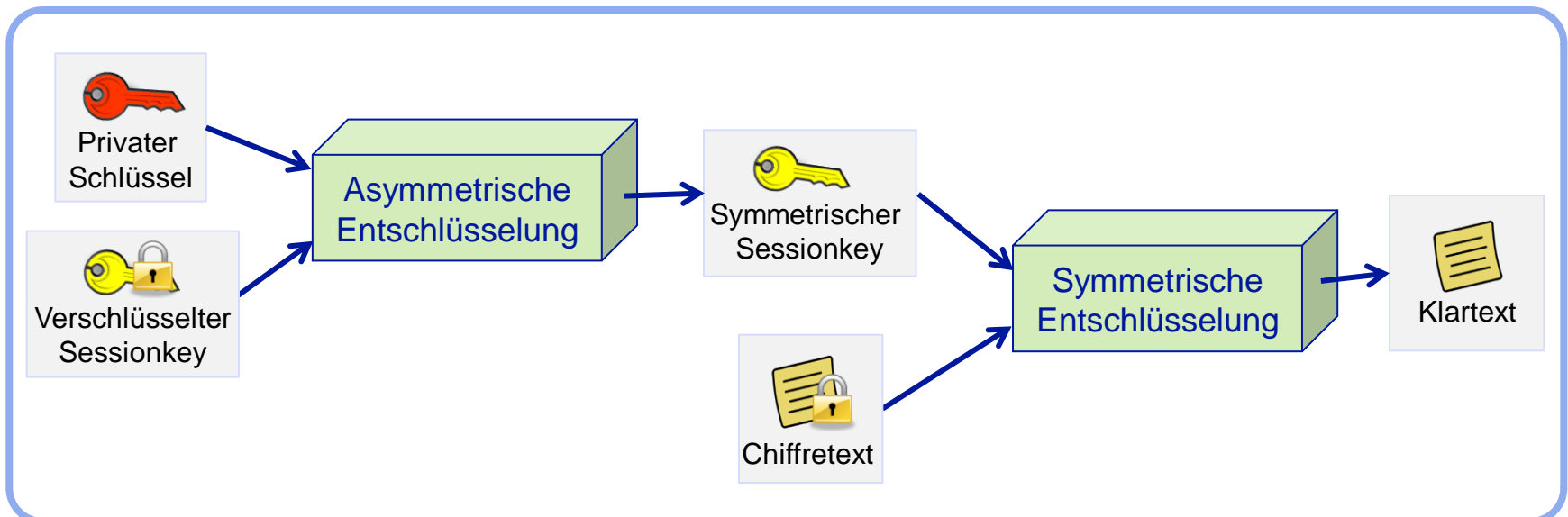
Aufgabe 8.2 Hybrid-Entschlüsselung (3)



Mit dem Sessionkey wird der verschlüsselte Text entschlüsselt und man erhält das ursprüngliche Dokument wieder.

Aufgabe 8.2 Hätten Sie es gewusst?

- Welche Schritte der Hybrid-Verschlüsselung nutzen welches Teil-Verfahren?
 - Welcher Schlüssel wird dabei jeweils verwendet?
1. Der Empfänger entschlüsselt den Sessionkey mit seinem privaten Schlüssel.
(asymmetrisch)
 2. Mit dem Sessionkey entschlüsselt der Empfänger den Chiffretext in den Klartext.
(symmetrisch)



Aufgabe 8.3 Hätten Sie es gewusst?

- Warum benutzt man hybride Verschlüsselung?

Die Vorteile der symmetrischen und der asymmetrischen Verschlüsselung werden vereint.

Sender und Empfänger können den geheimen Sessionkey, durch die asymmetrische Verschlüsselung geschützt, sicher austauschen.

Die Verschlüsselung der Daten erfolgt mit dem symmetrischen Sessionkey und ist dadurch sehr schnell.

Setzt man asymmetrische Kryptographie (Public-Key-Kryptographie) ein, muss man sich VORHER immer um den Aufbau einer Schlüsselinfrastruktur kümmern. Man hat also immer einen einmaligen initialen Mehraufwand, der aber danach den dauerhaften Betrieb deutlich vereinfacht.