Kryptographie

Alexander Prosser

Ziele und Mittel

- **Vertraulichkeit** => Schutz gegen Abschöpfen und Mitlesen
- **Integrität** => Schutz gegen unbefugte Manipulation
- **Authentizität** von Daten und Zugriffen => Schutz gegen Identitätsdiebstahl

- Chiffrierung => Vertraulichkeit
- (Mathematische) Verschlüsselung im engeren Sinn => Vertraulichkeit
- Hash (Prüfwerte) => Integrität, Authentizität
- Digitale Signatur => Authentizität

Klassische Demonstration

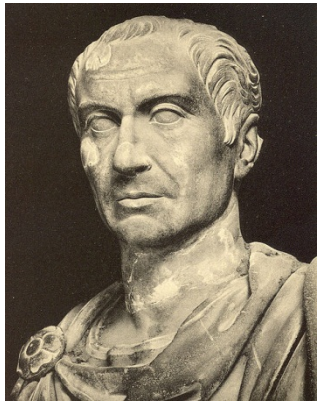


Photo: Public Domain

- Von CJ Caesar im Gallischen Krieg und römischen Bürgerkrieg verwendet.
- Jeder Buchstabe im Alphabet um n Positionen weitergeschoben
- Beispiel: n= 2
  Chiffrierung: gaius => (+2) => ickwu
  De-chiffrierung: ickwu => (-2) gaius
- Welche Angriffe auf dieses System wären möglich?

Chiffrierung: Block und Strom

Blockchiffre

- Blockchiffre: Substitutionen, Verschiebungen (s. Caesar), Permutationen etc. mit Schlüssel als Input

- 1971 Lucifer (IBM)
- 1977 DES (von NIST und NSA überarbeiteter Lucifer) 56 bit (Kompromiss zwischen IBM und NSA)
- Heute: DES in kurzer Zeit brechbar
- Triple DES mit 168 bit
- Problem der differenziellen Kryptoanalyse und des Meet-in-the-Middle Angriffs, wenn ein einziges Paar an Klartext/Chiffrat verfügbar ist.

Chiffrierung

Exkurs: Meet-in-the Middle (MIM-) Attacke auf Triple DES

- Linearer Angriff erfordert Abarbeitung von 2^168 Möglichkeiten

- Voraussetzung für MIM: ein Klartext/Chiffrat-Paar ist bekannt

a) Klartext wird zu allen Kombinationen des ersten Schritts verarbeitet: = 2^56 Interim-Chiffrate Stufe 1 (IC1)
b) Diese IC1 werden mit allen Kombinationen des zweiten Schlüssels verschlüsselt = 2^112 Kombinationen (IC2)
c) Alle Chiffrate der Stufe 3 werden mit dem drittes DES Schlüssel entschlüsselt (2^56 Kombinationen) und die so entstehenden IC2 …
d) … mit den IC2 aus Schritt b) verglichen. Resultat: der komplette 3DES Schlüssel ist bekannt.

- Statt 2^168 werden nur 2^112+2^56 Einzelschritte benötigt.

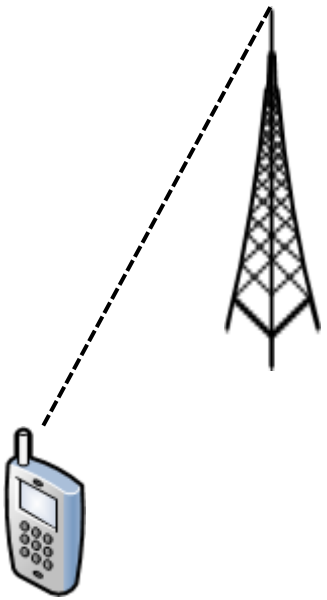
Chiffrierung: Block und Strom

Blockchiffre

- Heute: AES mit (typischerweise) 128 und 256 bit
- Substitutionen und Permutationen anhand des Schlüssels
- Sehr performant, gilt als sicher
- Animation:
  http://www.formaestudio.com/rijndaelinspector/archivos/Rijndael_Animation_v4_eng.swf
- Im Gegensatz zu RSA symmetrisch, d.h. der selbe Schlüssel wird zum Ver- und Entschlüsseln verwendet.
- Daher für öffentliche und verteilte Systeme nicht geeignet bzw. nur in Kombination mit RSA oder anderen asymmetrischen Methoden

Chiffrierung: Block und Strom

Stromchiffre



- Keine Sammlung von Daten zur Bildung eines Blocks
- Chiffrierung erfolgt pro Datenbyte (meist mit XOR)
- Wichtigster Standard: A5/1 (GSM-Telephonie)
- Leichte Kompromittierung wenn Klartext und Chiffrat vorhanden
- A5/2 als non-NATO „Einfachversion“ von A5/1 mit handelsüblichem Notebook de facto in Echtzeit brechbar
  http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2006/CS/CS-2006-07.pdf
- Nachfolger A5/3 (Blockchiffre) für UMTS gilt als sicher
  http://www.telekom.com/medien/konzern/209962

Symmetrische Verschlüsselung

Quasi Standard AES:

http://www.tools4noobs.com/online_tools/encrypt/

http://www.tools4noobs.com/online_tools/decrypt/

File encryption: winzip und rar

Übung: symmetrische Fileverschlüsselung per email

Verschlüsselung mit RSA

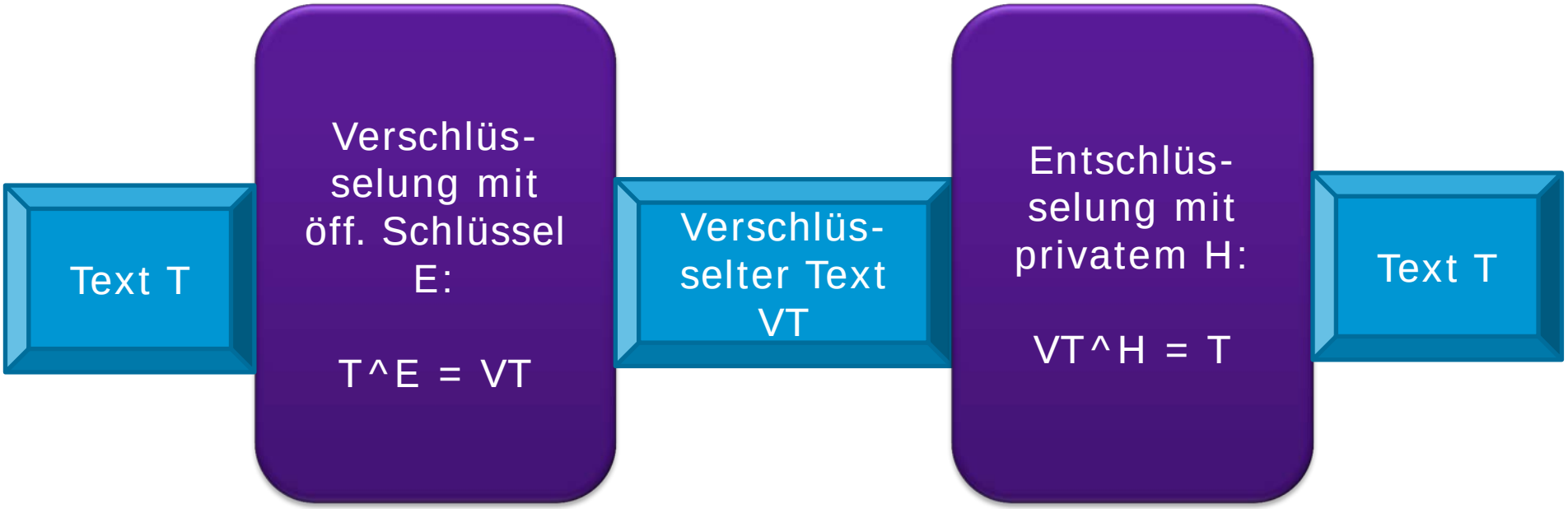
Nachteil symmetrische Verschlüsselung:
der Schlüssel/das Passwort muss geheim bleiben. Wenn bilateraler Austausch zwischen Personen, die sich vertrauen => ok
Multi-lateraler Austausch?

<u>Was wird benötigt?</u>

Öffentlicher Teil zur Verschlüsselung => „wer mir etwas sendet soll das damit vercodieren“

Privater Teil zur Entschlüsselung => „aber nur ich kann das wieder lesbar machen“

Verschlüsselung mit RSA



Aus E kann H nicht abgeleitet werden.

Verschlüsselung mit RSA

- Beispiel RSA Verfahren „manuell“:

a) Wähle 2 Primzahlen, p und q, und bilde das Produkt m=p*q
p=3, q=11, m=33

b) Berechne z = (p-1)*(q-1) = 20

c) Wähle Primzahl E, die teilerfremd zu z ist.
Optionen: 3, 5, 7, 11, 13, 17, 19 => wir wählen 7

d) Dies ergibt den öffentlichen Schlüssel (E, m) = **(7, 33)**

e) Bilde geheimen Schlüssel H so dass E*H = 1 mod(z) bzw. 7*H=1 mod(20)
Einschub: Rechnen mod
(7*H) / 20 mit Rest 1 => erfüllt mit H=3 => (7*3) / 20 = 1 Rest 1

f) Geheimer Schlüssel (H, m) = (3, 33)

Verschlüsselung mit RSA

a) Verschlüssle Text T=5 mit (E, m) zu verschlüsseltem Text VT
(E, m) = (7, 33)

b) T^E = VT mod(m) bzw. 5^7 = VT mod(33)
d.h. (5^7) / 33 = ?? mit Rest VT

c) 33 ist in (5^7) 2.367 Mal enthalten, der Rest ist
5^7 – 33 * 2.367 = 78.125 – 78.111 = 14

d) VT = 14 abgespeichert oder versendet (je nach Anwendung)

Verschlüsselung mit RSA

a) Verschlüssle Text T=5 mit (E, m) zu verschlüsseltem Text VT
(E, m) = (7, 33)

b) T^E = VT mod(m) bzw. 5^7 = VT mod(33)
d.h. (5^7) / 33 = ?? mit Rest VT

c) 33 ist in (5^7) 2.367 Mal enthalten, der Rest ist
5^7 – 33 * 2.367 = 78.125 – 78.111 = 14

d) VT = 14 abgespeichert oder versendet (je nach Anwendung)

e) Entschlüsselung mit (H, m) = (3, 33)
VT^H = T mod(m) bzw. 14^3 = T mod(33)

f) 33 ist in (14^3) 83 Mal enthalten, der Rest ist
14^3 – 33*83 = 2744 – 2739 = 5

g) Resultat der Entschlüsselung T = 5

Verschlüsselung mit RSA

a) Verschlüssle Text T=5 mit (E, m) zu verschlüsseltem Text VT
(E, m) = (7, 33)

b) T^E = VT mod(m) bzw. 5^7 = VT mod(33)
d.h. (5^7) / 33 = ?? mit Rest VT

T < 33, d.h. längere Texte segmentiert oder in einem Hash-Verfahren „zusammengedampft“

c) 33 ist in (5^7) 2.367 Mal enthalten, der Rest ist
5^7 – 33 * 2.367 = 78.125 – 78.111 = 14

d) VT = 14 abgespeichert oder versendet (je nach Anwendung)

e) Entschlüsselung mit (H, m) = (3, 33)
VT^H = T mod(m) bzw. 14^3 = T mod(33)

Der „Schlüssel“ enthält immer E bzw. H und den Modulus m

f) 33 ist in (14^3) 83 Mal enthalten, der Rest ist
14^3 – 33*83 = 2744 – 2739 = 5

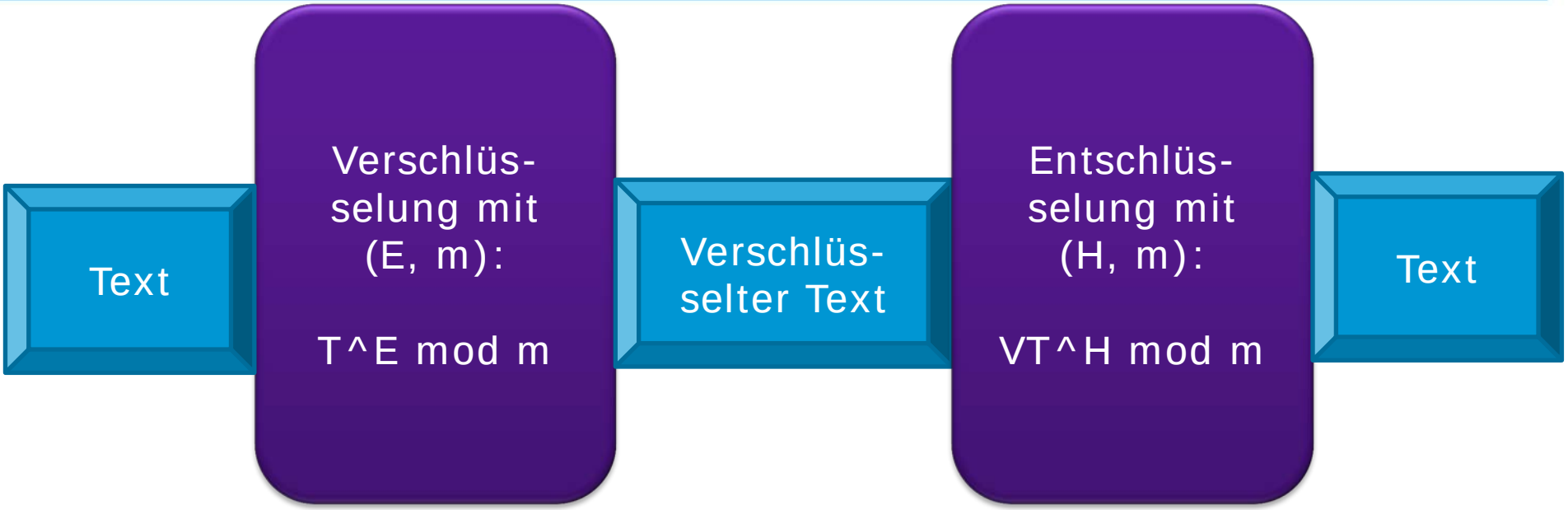
Verschlüsselung mit RSA

Übung 1: Wo greifen Sie das Verfahren an?

Übung 2: Verwendung am Beispiel eines Tools.

Übung 3: Tool verwenden

Verschlüsselung mit RSA



Verteilung öffentlicher Schlüssel?

Verschlüsselung mit RSA

Einfachste Methode: Download über eine Web Seite

Gesichert?

Standardisiert?

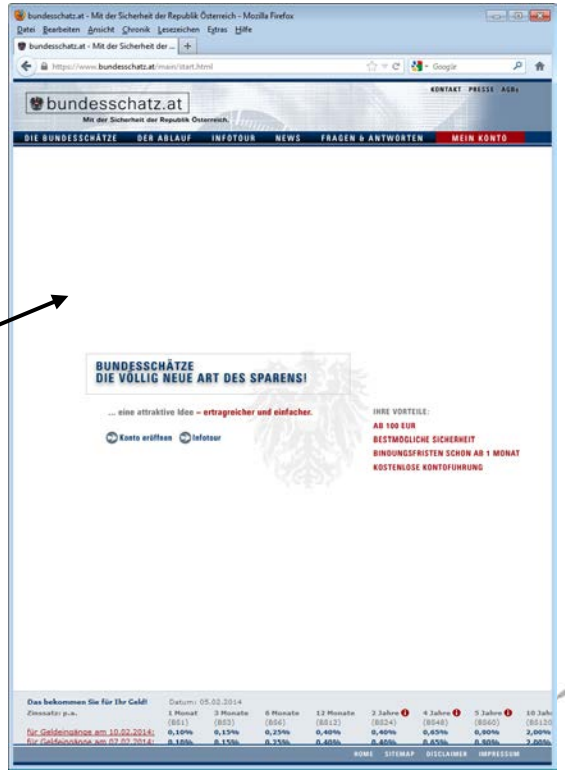
Was macht der User damit?

Lösung: Standardstrukturen, die von einschlägigen Programmen verarbeitet werden können => X.509

Info Organisation
Seriennummer
...
Verfahren (z.B. RSA)
Öffentlicher Schlüssel
...
Information Aussteller
...
Signaturverfahren
Signatur des Zertifikats

Signatur ist dem Browser bekannt …

oder vom Benutzer Installiert.



Zusammenspiel symmetrisch – asymmetrisch: SSL

- SSL Secure Socket Layer
- 1995 SSL 1.0/2.0 von Netscape (Mosaic Browser)
- 1996 SSL 3.0 => Basis für heutiges TLS (Transport Layer Security)
- 1999 TLS 1.0 ≈ SSL 3.0
- Aktuelle Version: 2008 TLS 1.2, wesentliche Sicherheitsupgrades http://tools.ietf.org/html/rfc5246
- Basis für „https://“
- Kombination von symmetrischen und asymmetrischen Schlüsseln

Zusammenspiel symmetrisch – asymmetrisch: SSL

1) Server teilt Browser sein Zertifikat mit => öffentlicher Schlüssel

2) Browser authentisiert Server
3) Browser wählt symmetrischen Schlüssel (typisch: AES)
4) Sendet ihn an Server RSA-verschlüsselt

5) Ab dann Kommunikation AES-verschlüsselt

Übung

Übung: Wie würden Sie das Verfahren angreifen ?

a) Als Krimineller ?

b) Als staatliche Organisation ?

Übung: Sie sind Gesetzgeber eines Staates, der diesen Zugriff will.
Welche rechtlichen Vorkehrungen brauchen Sie, um b) technisch zu realisieren ?

Übung

Prüfen Sie SSL-Zertifikat selbst.

Beispiel: https://www.bundesschatz.at/main/start.html

Further Reading

http://en.wikisource.org/wiki/Communications_Assistance_for_Law_Enforcement_Act_of_1994

http://en.wikisource.org/wiki/Foreign_Intelligence_Surveillance_Act_of_1978

http://en.wikisource.org/wiki/FISA_Amendments_Act_of_2008

Hash

Maßnahme: Bilden eines Prüfwertes über einen Datenbestand

Ziele: Sicherung der Integrität und Reduktion der Datenmenge für weitere Verarbeitung, z.B. digitale Signatur

Typisches Verfahren: SHA-1

Ältere Verfahren: MD4, MD5
Neuere Verfahren: SHA-256

Hash

Maßnahme: Bilden eines Prüfwertes über einen Datenbestand
Aus dem Hashwert ist der ursprüngliche Datenbestand nicht ableitbar.
Engl.: „trapdoor function“

Ziele: Sicherung der Integrität und Reduktion der Datenmenge für weitere Verarbeitung, z.B. digitale Signatur

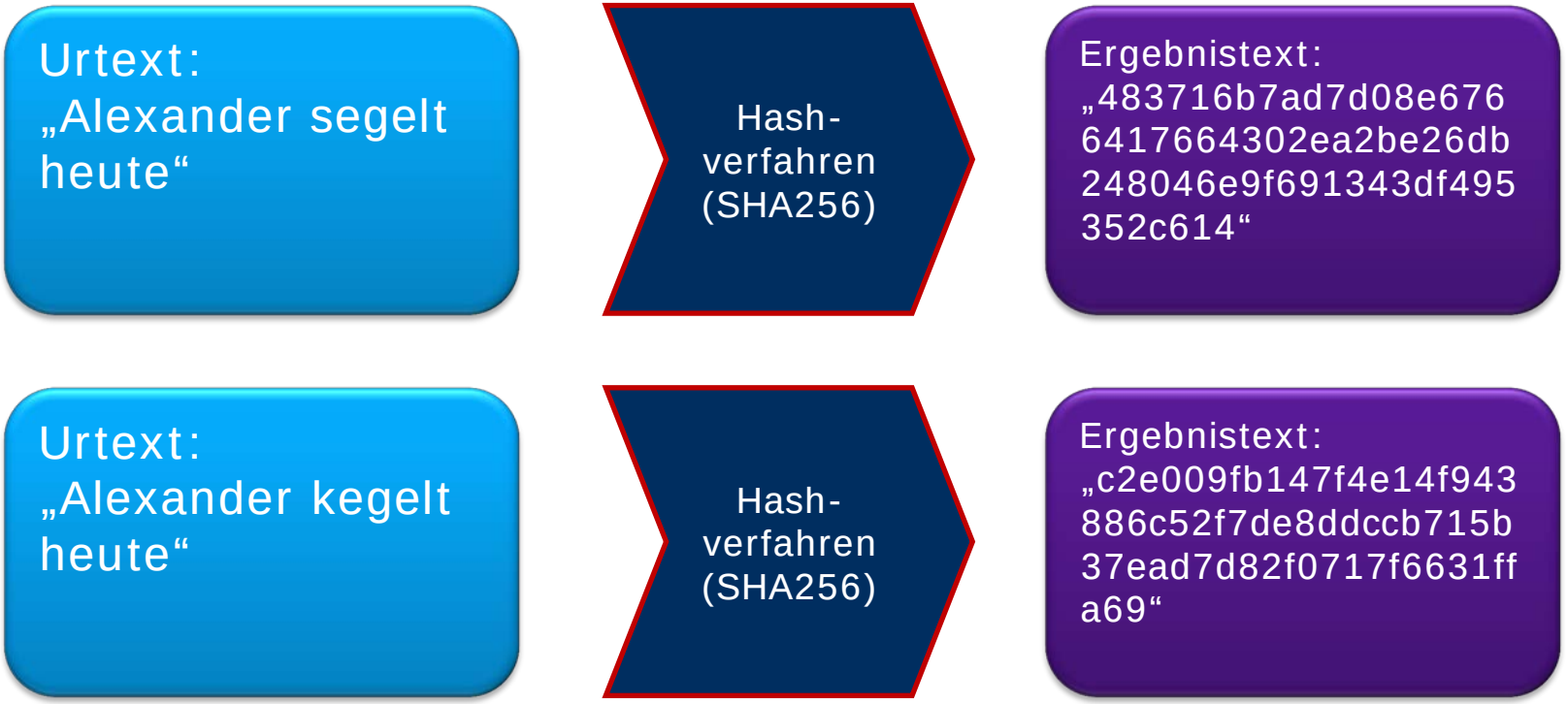
Typisches Verfahren: SHA-1

Ältere Verfahren: MD4, MD5
Neuere Verfahren: SHA-256

Methode: durch blockweise Registeroperationen wird eine beliebig große Nachricht (oder Datenmenge) zu einem 20-stelligen (=160 bit) „Digest“ komprimiert. SHA-1 ist daher keine eindeutige Abbildung.

Jede noch so kleine Änderung führt zu einer Änderung des SHA-1.

Hash


Urtext:
„Alexander segelt heute“
Hash-verfahren (SHA256)
Ergebnistext:
„483716b7ad7d08e6766417664302ea2be26db248046e9f691343df495352c614“
Urtext:
„Alexander kegelt heute“
Hash-verfahren (SHA256)
Ergebnistext:
„c2e009fb147f4e14f943886c52f7de8ddccb715b37ead7d82f0717f6631ffa69“

Hash - Anwendungen

- Prüfwerte, um Änderungen zu erkennen, Änderungen können manipulativ sein oder technischen Fehlern (z.B. Übertragungsfehlern) geschuldet sein.

- Basistechnologie der digitalen Signatur

- Speichern von Passworten
  Hinweis: eine Speicherung von Passworten in einem Informationssystem im Klartext ist ein Sicherheitsrisiko erster Ordnung und jur. vermutlich als grob fahrlässig einzuordnen.

- http://www.tools4noobs.com/online_tools/hash/

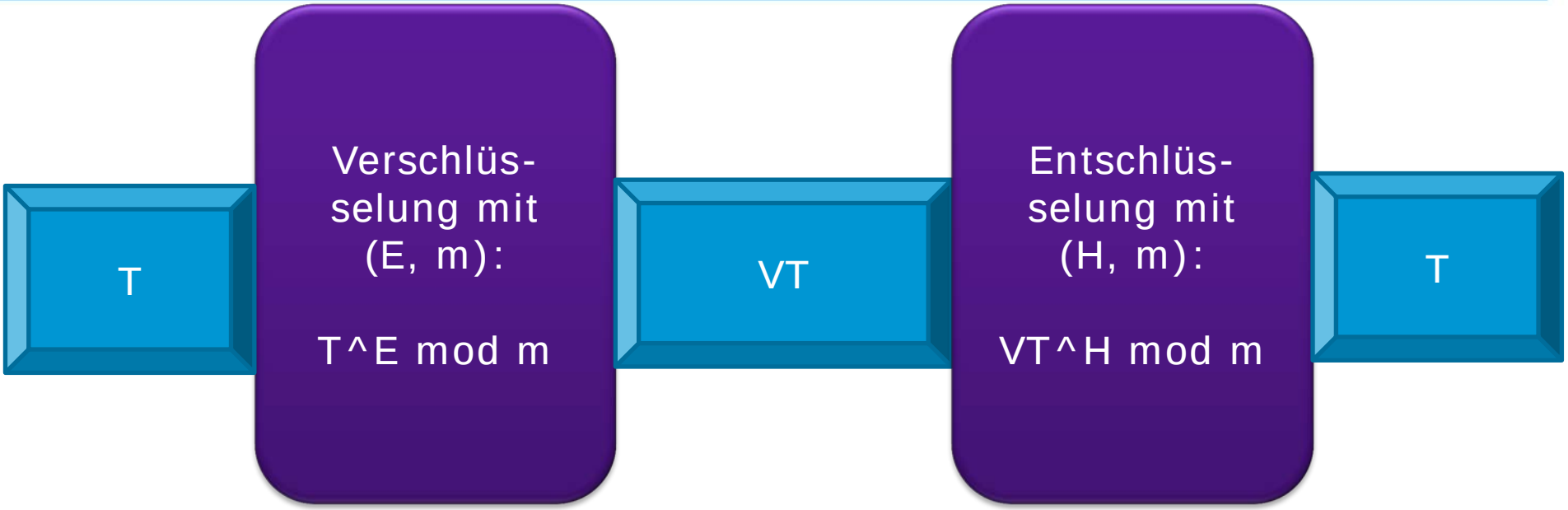
Hash für Passworte

Applikation (z.B. Webmaske)

Speicherung in Datenbank

Es kann nicht rückgeschlossen werden

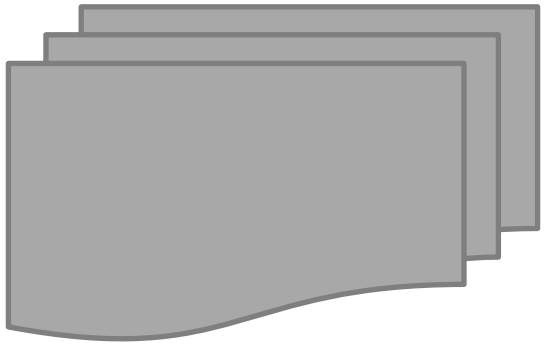
Wiederholung: RSA



Digitale Signatur

- Sicherstellen der Authentizität der Daten
- Vermeiden von Abstreitbarkeit
- Sekundär: Login-Funktion

Nachricht/en (auch verkettet) = T

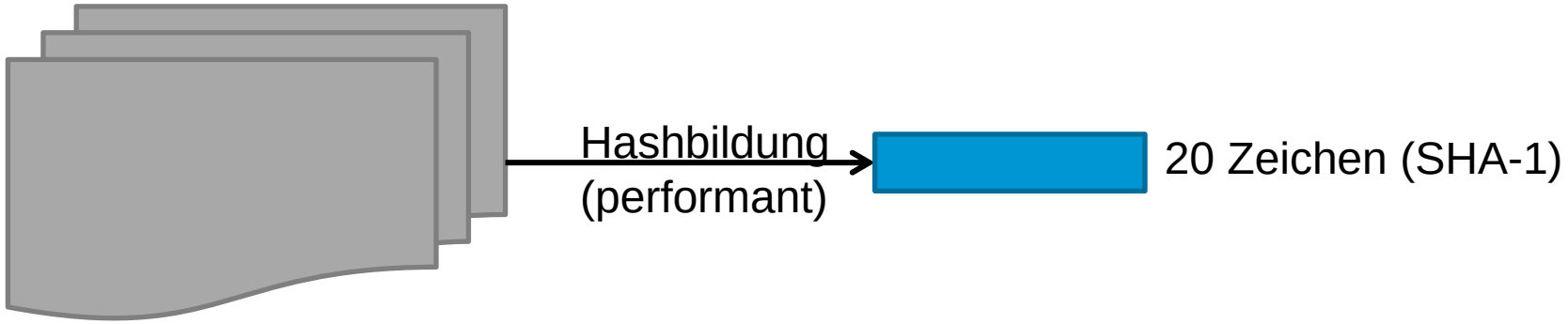


Problem: RSA relativ langsam
100 GB Text/Graphiken/…. ??
Wie groß wird das Signat ??

Digitale Signatur

- Sicherstellen der Authentizität der Daten
- Vermeiden von Abstreitbarkeit
- Sekundär: Login-Funktion

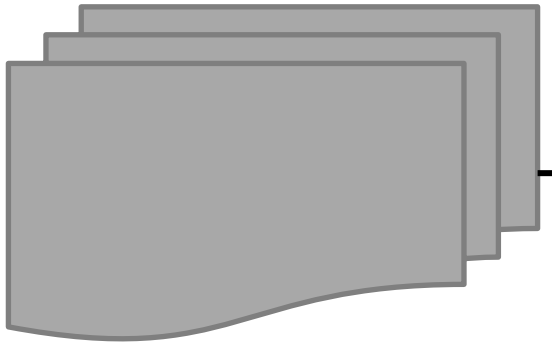
Nachricht/en (auch verkettet) = T



Digitale Signatur

- Sicherstellen der Authentizität der Daten
- Vermeiden von Abstreitbarkeit
- Sekundär: Login-Funktion

Nachricht/en (auch verkettet) = T



Hash = T

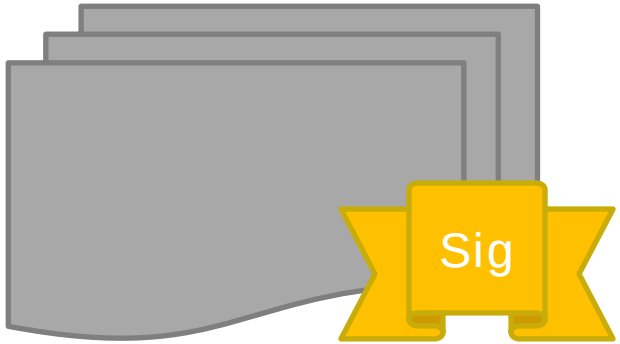
T^H mod(m)

Signat

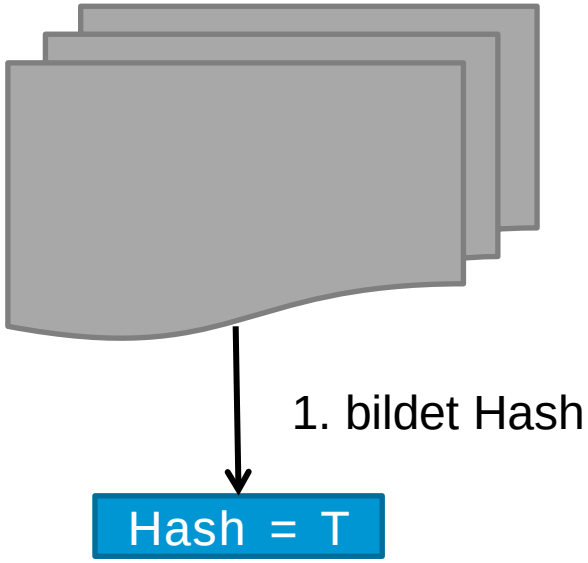
Sig

Digitale Signatur

Nachricht/en + Signat weitergegeben



Empfänger



2. bildet Sig^E mod m = X

3. vergleicht X = T?

Digitale Signatur

Man beachte die Abfolge:

Verschlüsselung:

Öffentlicher Schlüssel => Privater Schlüssel

Signatur:

Privater Schlüssel => Öffentlicher Schlüssel

https://www.bka.gv.at/DocView.axd?CobId=53919

Digitale Signatur - Realisierung

- Reine Softwarelösung
  - z.B. als Teil eines pdf Softwarepakets
  - Serversignatur
- Reine Hardwarelösung (Hardware Security Modul, HSM)
- Bürgerkarte oder eID http://www.buergerkarte.at/funktionsweise-karte.html
- Handysignatur http://www.buergerkarte.at/funktionsweise-handy.html

Übung: Was ist der wesentliche strukturelle Unterschied zwischen Karte und Handysignatur?

Verwundbarkeit

Staatliche Verwundbarkeit

- eGovernment (Manipulation, Vertraulichkeit und Verfügbarkeit)
- eBanking (Manipulation, Vertraulichkeit und Verfügbarkeit)
- „Internet of things“
  - Industrie, Handel und Logistik

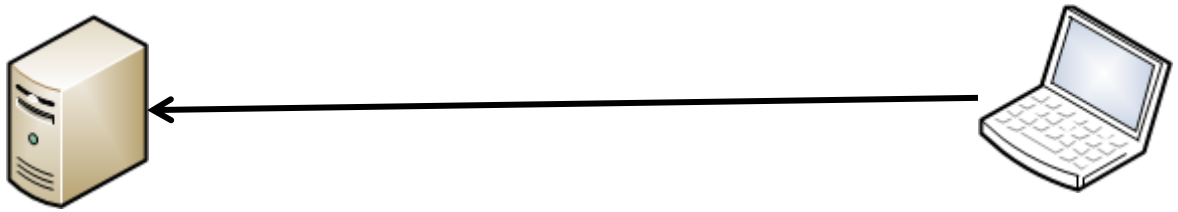
    http://www.digitalbond.com/blog/2012/01/31/langners-stuxnet-deep-dive-s4-video/
    https://www.schneier.com/cgi-bin/mt/mt-search.cgi?tag=Stuxnet
  - Haushalt
  - Infrastruktur, z.B. smart meter:

    http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:114:0064:0064:DE:PDF
    http://www.e-control.at/portal/page/portal/medienbibliothek/service-beratung/dokumente/pdfs/Intelligente-Messgeraete-Einfuehrungsverordnung%E2%80%93IME-VO.pdf

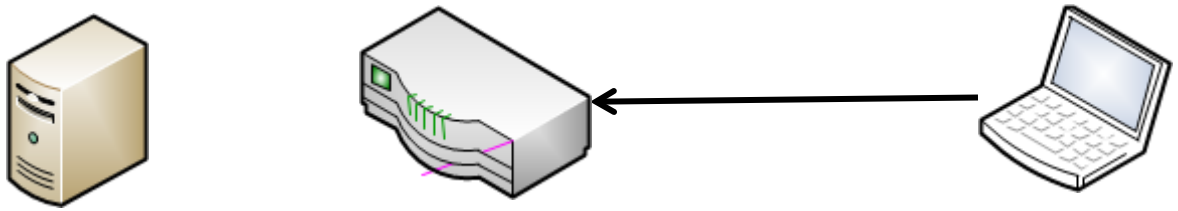
Problem ist nicht fehlerbedingte Verwundbarkeit („Bugs“ können behoben werden)
Problem sind systemische „built-in“ Verwundbarkeiten

(D)DOS?

Denial of Service Angriff

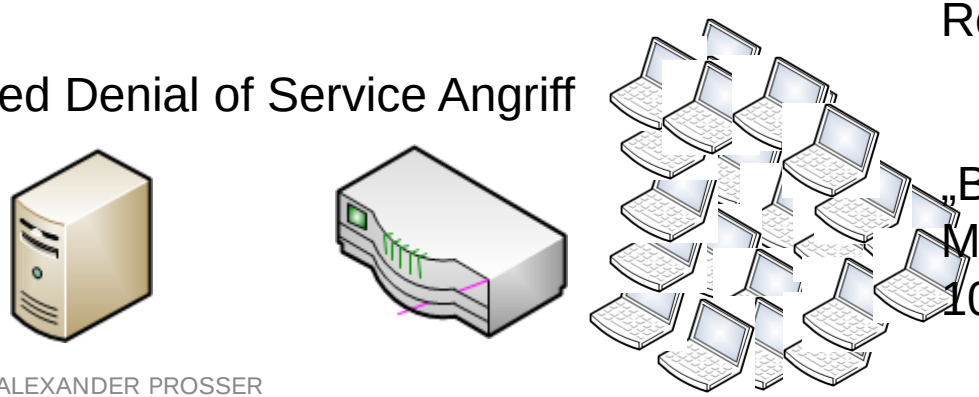


Request



Request

Distributed Denial of Service Angriff



„Botnet“
Miete einige 1000 bis
10.000 USD pro 24 Stunden

(D)DOS?

Verschiedene **Methoden**:

- Unvollständige IP-Pakete
- Systemnachrichten
- POST (Sende-) Nachrichten an den Web Server
- Öffnen (und nicht Schließen) einer großen Zahl von Sessions
- Provokation eines Puffer-Überlaufs
- …

Abwehr:

- Filterung des eingehenden Verkehrs (Firewall)
- Delayed Binding (Abwarten, ob korrekte Nachricht vom Partner kommt bis Verbindung zum Server aufgebaut wird)
- Erkennen von Angriffsmustern im Nachrichteninhalt
- …

In jedem Fall erfordert es professionelles RZ-Management

Übung

Methodik, Verwundbarkeit herauszufinden

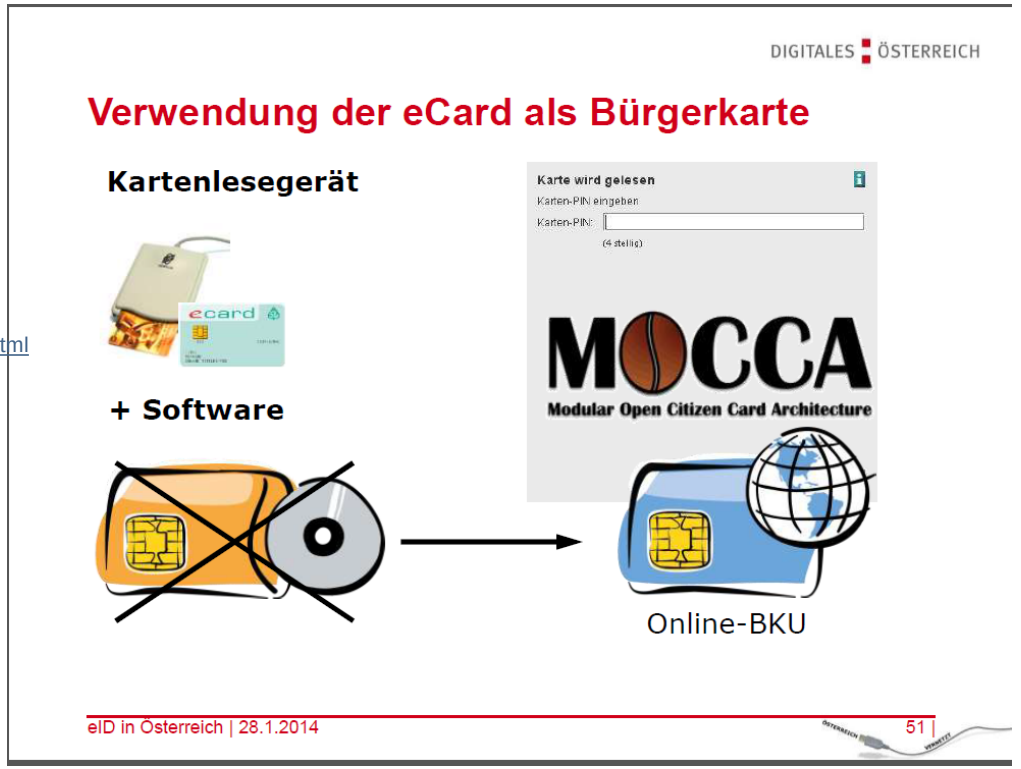
… am Beispiel Handysignatur

Bürgerkarte „klassisch“ vs. Handysignatur

Bürgerkarte:

- Hardware nötig
- Signaturerstellende Daten auf der Karte
- Geringe Verbreitung ~80.000 Karten

http://www.parlament.gv.at/PAKT/VHG/XXIV/AB/AB_15141/fnameorig_322471.html



Quelle: https://www.bka.gv.at/DocView.axd?CobId=53919

Bürgerkarte „klassisch“ vs. Handysignatur

Handysignatur:

- keine separate Hardware nötig
- signaturerstellende Daten am Server des Providers
- hohe (und steigende) Verbreitung

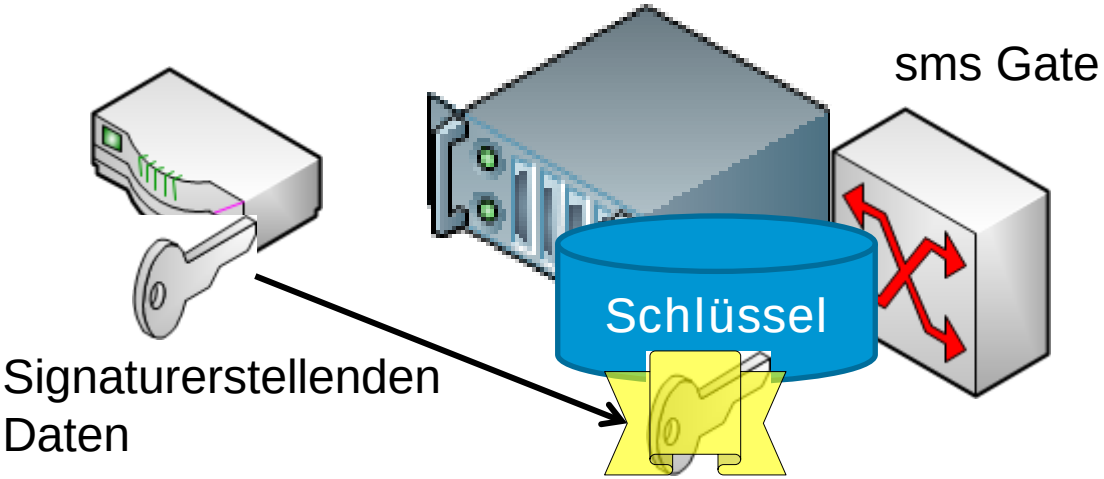
- Problem: österreichisches Handy
- Problem: sms Dauer

Beide Probleme aber im Inland gegenstandslos.

http://www.egiz.gv.at/de/schwerpunkte/11-buergerkarte#sub-handysignatur

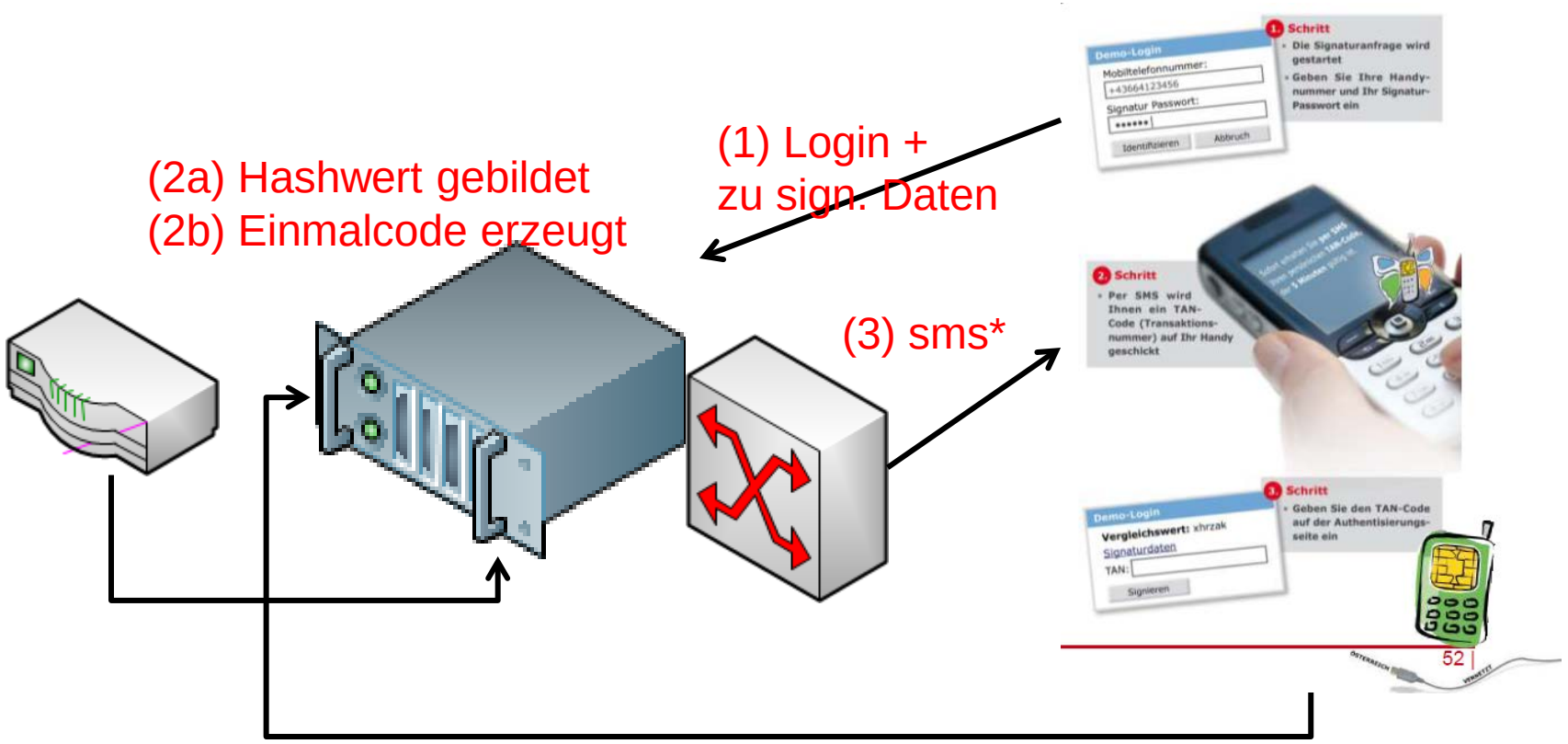
Handysignatur

Quelle: https://www.bka.gv.at/DocView.axd?CobId=53919



Handysignatur

Quelle: https://www.bka.gv.at/DocView.axd?CobId=53919



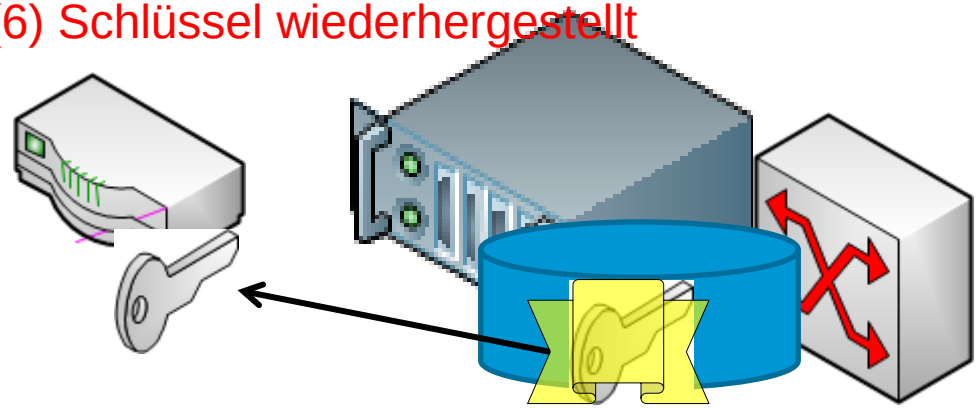
* enthält 2a und 2b

Handysignatur

Quelle: https://www.bka.gv.at/DocView.axd?CobId=53919

(5) Code geprüft

(6) Schlüssel wiederhergestellt



(7) Signatur über Hashwert

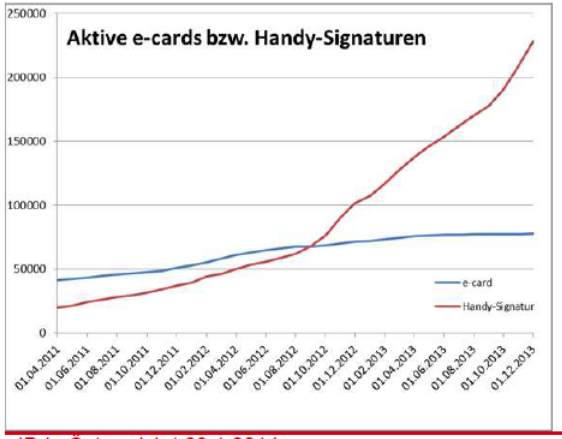
(8) Retournierung Signatur

Handysignatur

DIGITALES ÖSTERREICH

Handy-Signatur

- Derzeit sind ca. 270.000 aktive Handy-Signaturen im Einsatz.
- Jedes Monat steigt die Zahl um 10.000 – 25.000
- Monatlich wird mehr als 100.000 Mal mit dem Handy signiert



eID in Österreich | 28.1.2014 55 |

Wie kann diese Signaturform durch einen nachrichtendienstlichen Angreifer kompromittiert werden?

Zentral (Sever)

Dezentral (Übertragung/Nutzer)

Quelle: https://www.bka.gv.at/DocView.axd?CobId=53919

Voraussetzungen

Was ist nötig, um Handysignatur zu fälschen?

Wissen	Besitz
↓	↓
Logindaten	Einmalcode (SMS an Handy)
↓	↓
SSL (https)	Abhören SMS

Kann ein Nachrichtendienst das?

Maßstab: veröffentlichte Information über (mutmaßliche!) Fähigkeiten der NSA (http://www.nsa.gov/)

Angriff auf Logindaten

Variante 1: Keylogger bzw. Exploit
Gefahr der Entdeckung
http://en.wikipedia.org/w/index.php?title=File:Gchq-surveillance-the-documents.pdf&page=1

Variante 2: Angriff auf das SSL-Zertifikat
2a: „Cooperative Relationship“ mit Zertifikatserstellern

2b: Designschwächen
Beispiel RSA: Primzahlengeneratoren

http://www.theguardian.com/world/interactive/2013/sep/05/nsa-project-bullrun-classification-guide

SSL-Kompromitierung + Mitlesen bei Knoten/Kabel
= Fähigkeit Handshake bei SSL abzufangen

= Kenntnis des symmetrischen Session Keys in SSL
= Fähigkeit zum Mitlesen der Logindaten

Voraussetzungen

Was ist nötig, um Handysignatur zu fälschen?

Wissen

↓

Logindaten

↓

SSL (https)

Besitz

↓

Einmalcode (SMS an Handy)

↓

Abhören SMS

Angriff auf SMS

Es muss Inhalt der SMS abgeschöpft werden, nicht nur Metadaten.

Verschlüsselung in GSM:

- A5/1: Stromchiffre kann mutmaßlich abgehört werden
  http://apps.washingtonpost.com/g/page/world/how-the-nsa-pinpoints-a-mobile-device/645/#document/p1/a135574
- A5/2: „abgespeckte“ non-NATO Version, kann (de facto) online mitgehört werden.
- A5/3: Blockchiffre. Gilt als sicher. In UMTS Standard.

  Dt. Telekom hat das per Ende 2013 implementiert http://www.telekom.com/medien/konzern/209962

  Auch in anderen Ländern implementiert, z.B. CZ, MNE, MK

Einfachvariante durch „normale“ Kriminelle:

http://derstandard.at/1381369749828/Betrueger-nutzen-neue-Methode-zum-Betrug-bei-Onlinebanking

Installation Schadsoftware im Anhang einer gefälschten email

Abschöpfen Passwort/Login

Ersatz-SIM-Karte vom Handybetreiber

Mobile TAN SMS erhalten

Variante II: Zentraler Angriff auf Server durch Backdoors:

Generelle Problematik, die nicht nur digitale Signatur betrifft

Analyse von CALEA und FISA erforderlich

Nächstes Forschungsvorhaben

- A5/3 (64 bit) ist in Zeiten von mTAN, mGovernment etc. ein **MUSS**
  => mittelfristig A5/4 128 bit

- Forcierung inländischer/europäischer SSL Zertifikatsanbieter
  Sicherung deren ökonomischer/technischer Unabhängigkeit

- Inländisches/europäisches kryptographisches/sicherheitstechnisches Know-how
    - staatliche Stellen
    - Anbieter in der Wirtschaft
      Sicherung deren ökonomischer/technischer Unabhängigkeit

Dies ist Schlüsselinfrastruktur genauso wie Wasser- oder Stromversorgung.

VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

Department Informationsverarbeitung und Prozessmanagement

Augasse 2-6, 1090 Vienna, Austria

Univ.Prof. Dr. Alexander Prosser

T +43-1-313 36-5630
F +43-1-313 36-5610
prosser@wu.ac.at
www.wu.ac.at