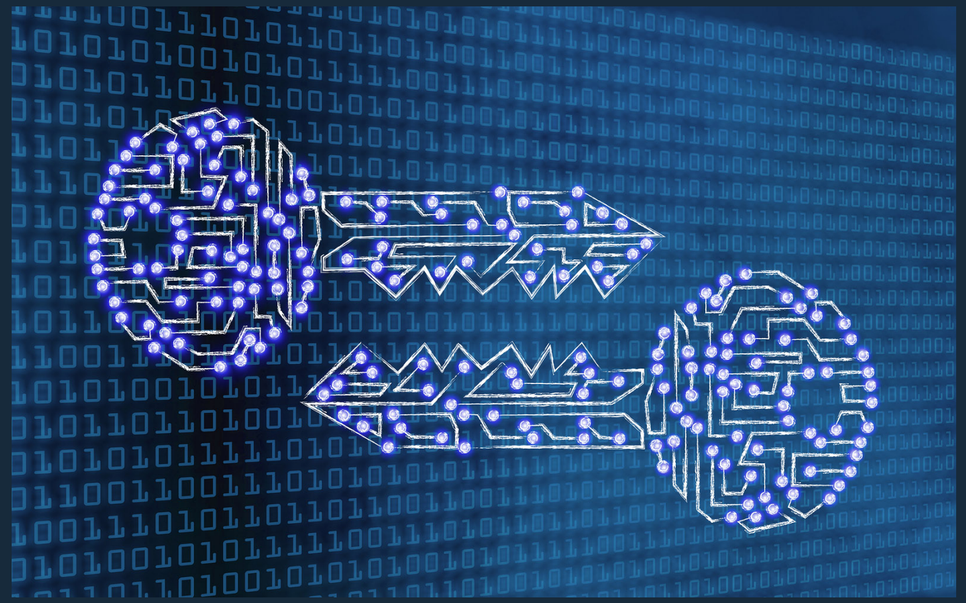
Kryptografie

Wissenstransfer

Julia Bremer - Univention GMBH

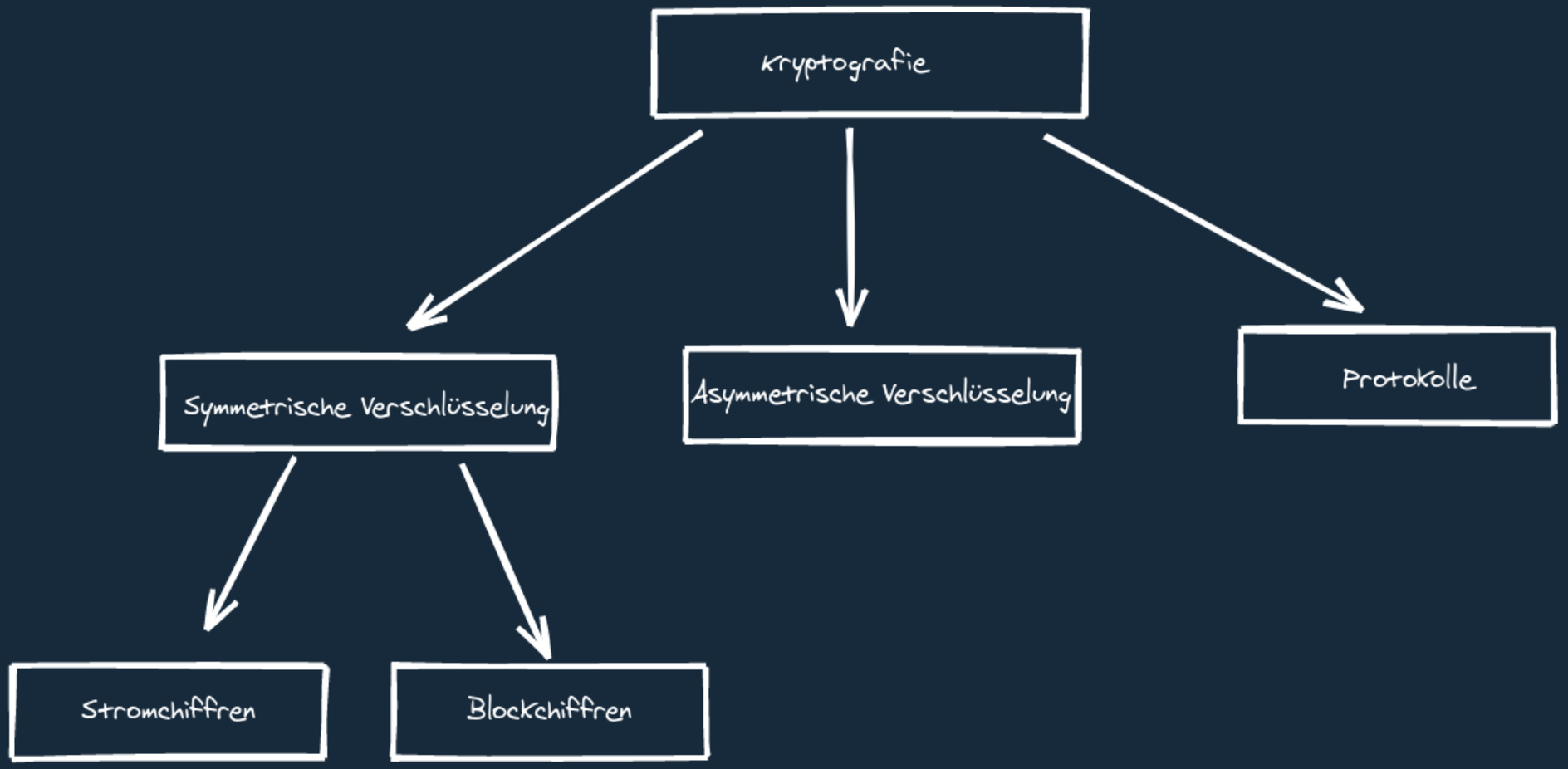
Agenda

- Kryptografie
- Schutzziele
- Obfuscation / Hashing / Encryption
- Block cipher / Stream cipher
- **Symmetrische** Verschlüsselung
- **Asymmetrische** Verschlüsselung
- **Hybrid** - Diffie-Hellman
- Algorithmen
- Signaturen



Kryptografie

Die Wissenschaft der **Verschlüsselung**



Schutzziele

- **CIA**
- **C**onfidentiality (Vertraulichkeit)
  - Zugriff auf Informationen nur auf die von dem Besitzer beschränkte Personengruppe möglich
- **I**ntegrity (Integrität)
  - Unautorisierte Personen dürfen Daten ohne die Erlaubnis des Besitzers nicht modifizieren dürfen
- **A**vailability (Verfügbarkeit)
  - Das System darf nicht so sehr gestört werden können, dass es nur eingeschränkt oder **nicht** mehr benutzt werden kann
  - Denial of Service **DOS** Attacke

- Authenticity (Authentizität)
- Commitment (Nichtabstreitbarkeit)
  - in ITSYS wollen sie auch diese beiden wissen, zusätzlich zu **CIA**
  - **Commitment** und **Authenticity** also vielleicht **CCIAA** als Eselsbrücke?
- Attributability (Zurechenbarkeit)
- Privacy (Datenschutz)

Obfuscation

- **Security by Obscurity**
- Bei Obfuscation geht es nur darum Dinge **unverständlicher** zu machen
- Für Computer leicht drüber hinwegzusehen, für Menschen schwer
- Keine gute Sicherheitsmaßnahme
- Beispiel: AD supplementalCredentials werden obfuscated übertragen, damit in network traces nicht klar erkennbar ist, dass es Credentials sind.

Hash

- Im Gegensatz zur Verschlüsselung eine mathematische **Einwegsfunktion**
  - Das heißt, man kann aus einem Hash **nicht** die originalen Daten rekonstruieren
- Verwendungszweck meist in Kombination mit Verschlüsselung
- Problem: Wurde ein Passwort zu einem Passworthash ermittelt, sind dann alle aufgeflogen die dieses Passwort verwenden?
- **Rainbowtables** mappen Worte zu Hashes, die häufigsten sind also bekannt
- Damit diese Rainbowtables nicht funktionieren **saltet** man
- Hashfunktionen können **Kollisionen** haben
- Kollision heißt es exisieren mehrere Wörter, die den gleichen Hash erzeugen

Salt

- Ein Extrawort, dass dem Passwort vor dem Hashing hinzugefügt wird.
- Dabei ist es nicht wichtig, dass der Salt geheim ist, nur dass er bei **jedem Benutzer unterschiedlich** ist.
- Durch die Zuführung eines Saltes, kann man den Hash nicht mehr in Rainbowtables nachsehen.
- Und gleiche Passwörter anhand der Gleichheit des Hashes erkennen.
- Beispiel: Bei Kerberos ist der default salt **username@DOMAINNAME**
- krb5key = Salt + Hashfunktion(Salt+Pwd)

Encryption

- Der eigentliche Fokus der Kryptografie
- Eine **umkehrbare** Funktion, bei der es eine Funktion **e**(ncryption) und eine Funktion **d**(ecryption) gibt.
- Zwei grundlegende Operationen um starke Verschlüsselung zu erreichen sind **Konfusion** und **Diffusion**
- **Konfusion**: Verschleierung von Zusammenhang von Schlüssel und Chiffre
- **Diffusion**: Verschleierung von statistischen Eigenschaften des Klartextes

Symmetrische Verschlüsselung

- In der symmetrischen Verschlüsselung kann mit **dem selben Schlüssel ver- und entschlüsselt** werden
- Beide Teilnehmer müssen im Besitz dieses Schlüssels sein
- Die Verteilung des Schlüssels ist ein Hauptproblem beim symmetrischen Verfahren
- Darum verwendet man auch **hybride** Verfahren, bei denen der Schlüsselaustausch asymmetrisch vollstreckt wird
- Man teilt die symmetrischen Verfahren in **Blockchiffren** und **Stromchiffren** auf

Stream vs Block Cipher

- Bei der Stromchiffre wird **jeder Klartextbit einzeln** verschlüsselt
  - Es gibt keinen (automatischen) Integritätsschutz
  - Beispiel: **RC4**
- Bei der Blockchiffre wird immer ein ganzer **Block Klartextbits gleichzeitg** verschlüsselt
  - Die statistischen Eigenschaften bleiben im Chiffrat erhalten
  - Beispiel: **DES**, **AES**
  - meistens werden Blockchiffren genutzt

Stream Cipher

Stromchiffren

- Jedes Bit `x_i` wird verschlüsselt, indem ein geheimes Bit `s_i` des Schlüsselstroms per XOR mit diesem verknüpft wird.
- **XOR** ist der logische Operator der balanciert ist, und eignet sich daher zum verschlüsseln
- Die Sicherheit der Chiffre ist **vollständig** in Abhängigkeit von dem Schlüsselstrom
- Die Bit `s_i` vom Schlüsselstrom, sind **nicht der Schlüssel**
- Ein Pseudorandomgenerator wird mit dem Schlüssel als **seed** gefüttert um diesen Schlüsselstrom zu generieren
- Heutzutage werden **häufiger Blockchiffren** verwendet

Block Cipher

Blockchiffren

- Eine Blockchiffre veschlüsselt einen Block **gleichzeitig** mit dem gleichen Schlüssel
- Innerhalb des Blocks beeinflusst jedes Bit die Verschlüsselung jedes anderen Bits in dem Block.
- Heutzutage ist die **Blockbreite** 128 Bit (AES) Standart, früher 64 Bit (DES)
- Block Cipher haben verschieden **Betriebsmodi** und meist noch einen Integritätscheck
- Beispiel für einen Schlüsselnamen: **DES-CBC-CRC**
- [DES-Verschlüsselung]-[CBC-Betriebsmodus]-[CRC-Integritycheck]

DES

Data Encryption Standart

- Verschlüsselt 64 Bit Blöcke mit 56 Bit Schlüsseln
- Da der Schlüssel nur 56 Bit hat, kann DES heute leicht bruteforced werden und gilt als unsicher
- Aus diesem **Hauptschlüssel** werden **16 Rundenschlüssel** generiert
- Vor der Verschlüsselung werden die Daten in zwei Hälften zerteilt, die eine wird in die **Funktion f** eingegeben, und dann mit der anderen per XOR verknüpft
- Dann werden die Seiten getauscht und das Ganze mit dem nächsten Rundenschlüssel wiederholt

AES

Advanced Encryption Standart

- Heutzutage häufigste genutzte symmetrische Chiffre
- Das was die Werbung als **military grade encryption** nennt
- In jeder Iteration wird im Gegensatz zu DES der gesamte und nicht nur der halbe Block verschlüsselt
- AES besteht aus **drei Schichten**, die in jeder Iteration (außer der letzten) angewendet werden
- **Key-Addition-Schicht** , **Byte-Substitution-Schicht**, **Diffusionsschicht**

Betriebsmodi

- Meistens werden mehr als nur ein Block von 64/128 Bit verschlüsseln sondern **mehrere Blöcke**
- Wie diese Blöcke verknüpft werden entscheidet der **Betriebsmodus**
- Beispielsweise **ECB** Electronic-Codebook-Modus ist vollständig deterministisch. Das Problem sieht man bei den Pinguinen.
- **CBC** Cipher-Block-Chaining-Modus verkettet die verschlüsselten Blöcke und umgeht dieses Phänomen so
- CBC ist dafür aber deutlich fehleranfälliger
- Es gibt eine ganze Reihe von Betriebsmodi mit verschiedenen Vor- und Nachteilen

Verschlüsselung mit CBC Betriebsmodus

$$y_1 = e_k(x_i \oplus IV)$$
$$y_i = e_k(x_i \oplus y_{i-1}), i \geq 2$$

- Die Blöcke werden im Cipher-Block-Chaining-Modus folgendermaßen verknüpft:
- In der **ersten Runde** wird mit einer **Nonce** verknüpft, einer Pseudo-Zufallszahl und dann die **Verschlüsselunsfunktion e** mit **Rundenschlüssel k** ausgeführt
- In jeder darauffolgenden Runde wird der zuvor verschlüsselte Block mit dem neuen Klartextblock per XOR verknüpft und **dann** mit e verschlüsselt

Asymmetrische Verschlüsselung

Public-Key-Kryptografie

- Hier gibt es **zwei verschiedene** Schlüssel, einen **verschlüsselnden** und einen **entschlüsselnden**
- Diese Schlüssel werden auch **public key** und **private key** genannt
- Der public key **muss nicht geheim gehalten** werden
- Diese Verschlüsselungsalgorithmen basieren darauf, dass es Prinzipien gibt, die für Computer leicht zu errechnen sind, deren Inverse allerdings sehr schwer zu berrechnen ist.
- Man nennt dies auch **Einwegsfunktionen** die im Grunde **injektive** Abbildungen sind
- Wird oft nur für **Signaturen** und zum **Schlüsselaustausch** verwendet

Asymmetrische Verschlüsselung

Diffie-Hellmann-Schlüsselübergabe

- Der symmetrische Schlüssel für die Kommunikation wird über einen **unsicheren Kanal gesichert übertragen**
- Basiert auf dem **Diskreten-Logarithmus-Problem**
- Vorgegeben sind eine Primzahl **p** und eine natürliche Zahl **x**
- Es ist sehr einfach

$$x^a \mod p$$

- zu errechnen. Jedoch ist es sehr schwierig aus aus dem Ergebnis **a** zu errechnen. Dies wird sich hier zu Nutze gemacht
- **TODO GRAFIK**

Asymmetrische Verschlüsselung

RSA

- Das meistgebrauchte asymmetrische Verfahren (Beispiel: ssh-keys)
- Die genutzte **Einwegfunktion** ist die Multiplikation von Primazahlen
- Es ist **einfach zwei große Primzahlen zu multiplizieren**, jedoch **sehr schwer aus dem Produkt wieder die beiden Primzahlen zu errechnen**
- **TODO Grafik**

RSA

- Es werden zwei Primzahlen **p** und **q** gewählt und das Produkt **pq** errechnet
- Außerdem bestimmt man die Zahl

$$m = (p-1)(q-1)$$

- **pq** ist öffentlich verfügbar, während **p**, **q** und **m** geheim bleiben, denn **m** ist aus **pq** nur schwer zu bestimmen
- Nun legen wir einen öffentlichen Schlüssel **e** fest, der **teilerfremd** zu **m** sein muss. Der private Schlüssel **d** ist die Lösung von

$$ed \equiv 1 \mod m$$

- Will man nun die **geheime Botschaft x** verschlüsseln rechnet man

$$y = x^e \mod pq$$

- Wobei **y** die verschlüsselte Nachricht ist. Zum **entschlüsseln von y** errechnet man

$$x = y^d \mod pq$$

Asymmetrisch vs Symmetrische Verschlüsselung

- Hauptproblem der symmetrischen Verschlüsselung ist das **Schlüsselaustauschproblem**
- Nur mit asymmetrischen Keys kann **signiert** werden
- Asymmetrische Keys müssen **sehr lang** sein um ähnlich sicher wie symmetrische Keys zu sein
- Symmetrische Verschlüsselung ist sehr **schnell und effizient**

Signaturen

- Soll **Zurechenbarkeit** und **Nichtabstreitbarkeit** erreichen, ähnlich eine gewöhnlichen Signatur
- Kann nur durch asymmetrische Verschlüsselung zustande kommen
- Dabei geht es **nicht um Verschlüsselung** der Nachricht. Nachrichten können signiert sein, ohne verschlüsselt zu sein
- **RSA-Signatur** wird häufig verwendet und basiert auf **RSA-Verschlüsselung**

Gute Quellen

- Einführung in DES: https://www.youtube.com/watch?v=H7bvLU-2JUI
- Buch Kryptografie verständlich
- Kapitel IT-Sicherheit aus "Betriebssysteme" von Tanenbaum

Danke für eure Aufmerksamkeit