Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Verschlüsselung

April 16, 2023

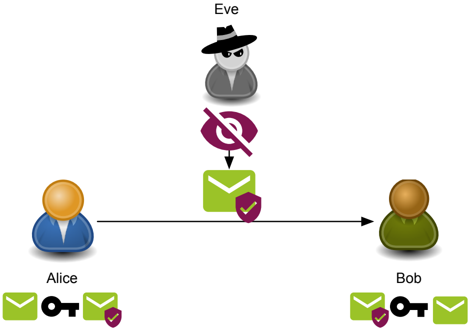
Marc Stöttinger

> We need to think about encryption not as this sort of arcane, black art. It’s a basic protection.
>
> Edward Snowden

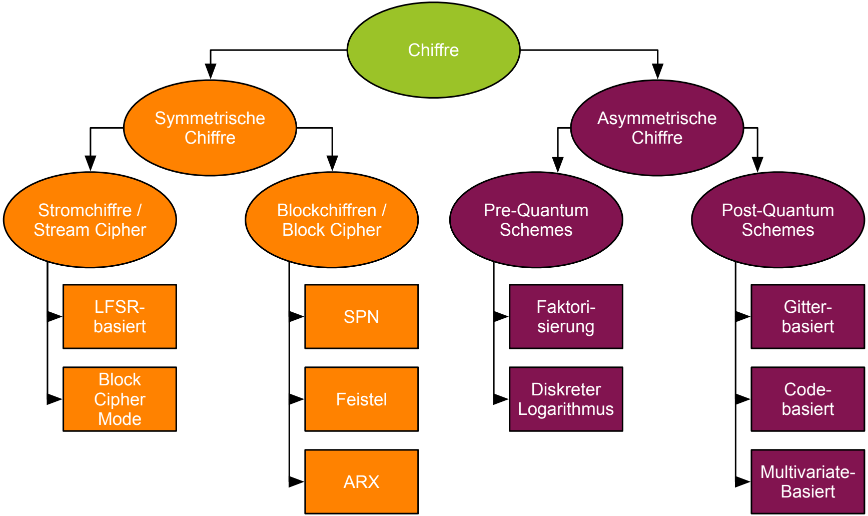
WIEDERHOLUNG: VERTRAULICHKEIT DURCH VERSCHLÜSSELUNG

→ **Bedrohung:**
Eve liest die Nachricht mit

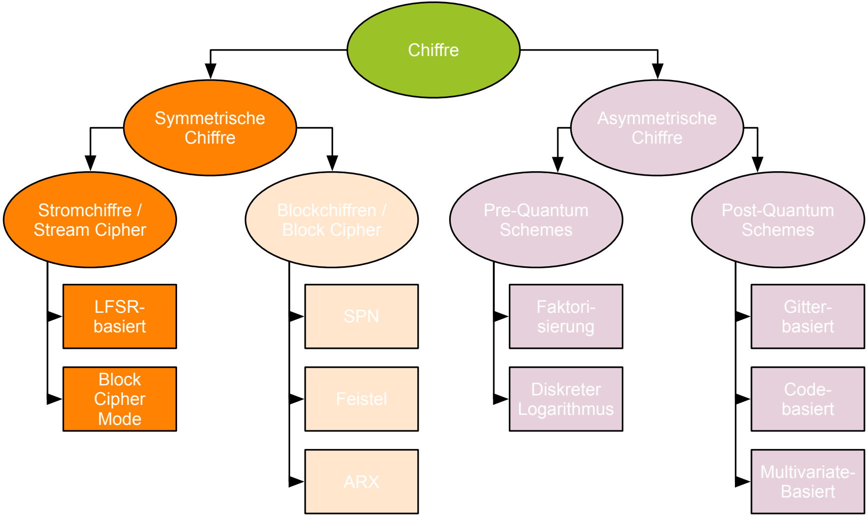
→ **Ziel:**
Personen ohne den entsprechenden Schlüssel können keine Informationen aus verschlüsselter Nachricht gewinnen



ÜBERSICHT VON VERSCHLÜSSELUNGEN



STROMCHIFFREN



PERFEKTE GEHEIMHALTUNG - ONE-TIME PAD

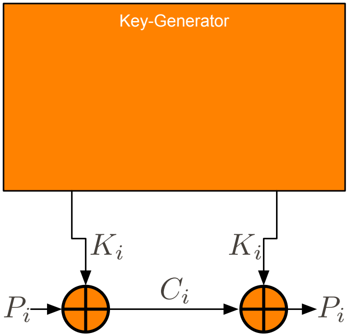
→ Substitution wobei P und K gleich lang sind

Plaintext	A	U	F	S	T	A	N	D
Schlüssel	J	A	T	U	C	O	B	I
Ciphertext	J	U	Y	M	V	O	O	L

→ Vorschrift für Binärdaten:
$C_i = P_i \oplus K_i \pmod{n}$

→ Die einzelnen Schlüsselbits K_i können durch einen Key-Generator erzeugt werden.

→ K_i wird dann auch als Schlüsselstrom (Key stream) bezeichnet.

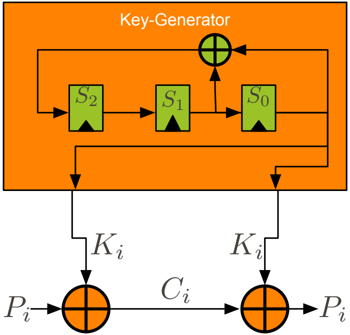


STROMCHIFFRE DESIGN

→ Eine einfache Linear Feedback Shift Register (LFSR) Schaltung wird zum Erzeugen von K_i genutzt.

clk	FF_2	FF_1	$FF_0 = s_i$
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0
8	0	1	0

→ $s_{i+3} \equiv (s_{i+1} \oplus s_i) \mod 2$

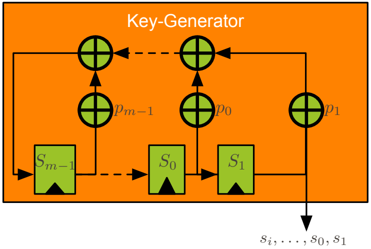


PRIMITIVE POLYNOM BASIERTE LFSRS

→ Generalisierte Form eines LFSR:
$$s_{i+m} \equiv \sum_{j=0}^{m-1} p_j \cdot s_{i+j} \mod 2;$$
$$s_i, p_i \in 0, 1; i = 0, 1, 2, \dots$$

→ Primitive Polynome, ein spezieller Typ von nicht reduzierbaren Polynomen, haben die Form
$$P(x) = x^m + p_{m-1}x^{m-1} + \cdots + p_1 x + p_0$$

→ Nur Primitive Polynome erzeugen eine maximale Sequenz von $2^m - 1$.



Maximale Sequenzlänge

Die maximale Sequenzlänge, die von einem LFSR vom Grad m erzeugt werden kann, ist $2^m - 1$.

KEY-GENERATOR FÜR STROMCHIFFREN

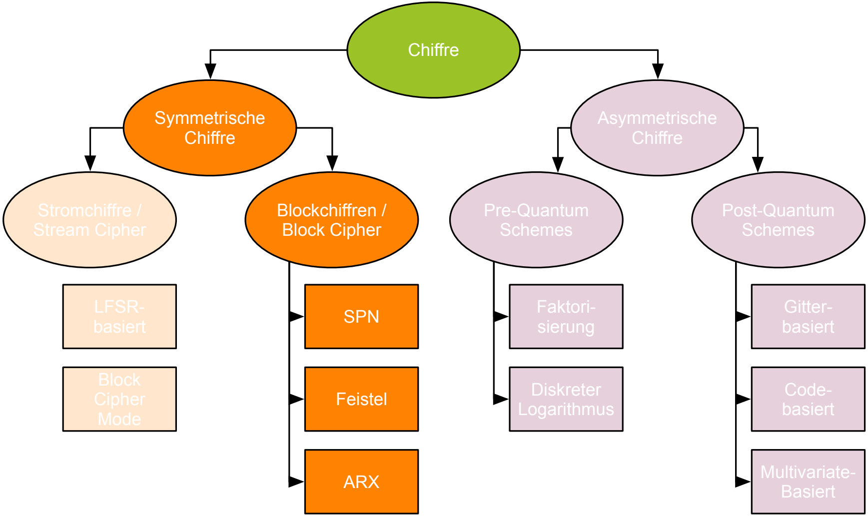
→ Die Schlüsselgeneratoren von modernen Stromchiffren haben meist einen großen internen Zustand.

→ Zur Konstruktion der zufälligen Schlüsselsstromsequenz werden meist mehrere LFSR Konstruktionen verwendet .

→ Der geheime Schlüssel wird zur Initialisierung des internen Zustands benutzt.

Chiffre	Erstellungsdatum	Schlüssellänge	Interner State	Komplexität bester Angriff
RC4	1987	8–2048 Bits	2064 Bits	2^{13} oder 2^{33}
A5/2	1989	54 Bits	64 Bits	komplett gebrochen
MICKEY	2004	80 BIts	200 Bits	$2^{32.5}$
Trivium	2004	80 Bits	288 Bits	2^{135}
Salsa20	2004	256 Bits	512 Bits	2^{251} (für 8 Runden)

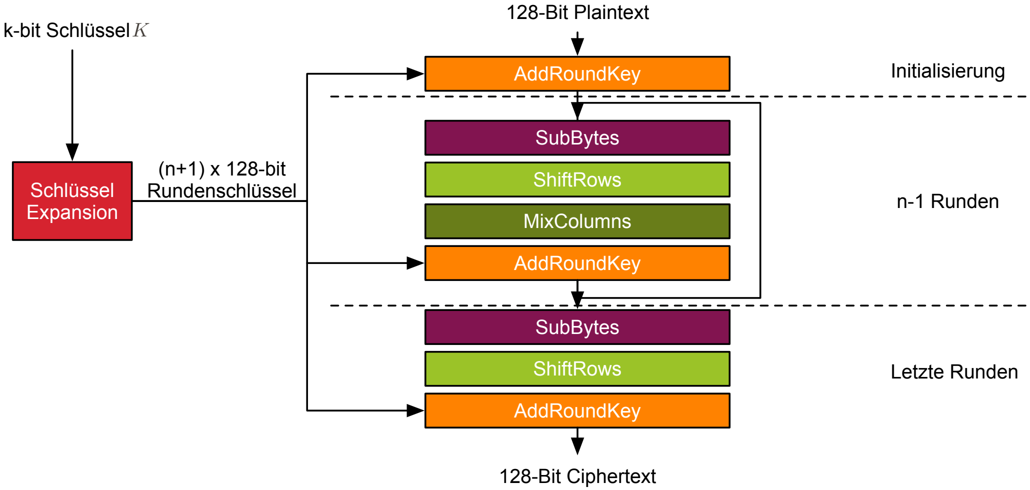
BLOCKCHIFFREN



ADVANCE ENCRYPTION STANDARD (AES)

- In 2000 wurde Rjindael zum Sieger einer Ausschreibung gekürt und als **AES** standardisiert
- AES ist eine **Blockchiffre**, die auf 128-bit Blöcken arbeitet
  - **Blockchiffre**: Der Plaintext wird in Blöcke eingeteilt und blockweise verarbeitet
  - **Stromchiffre**: Zeichen werden einzeln verarbeitet (z.B., monoalphabetische Substitution, One-Time Pad)
- Die Blöcke werden in n Runden durch ein Substitutions-Permutations-Netzwerk (SPN) verschlüsselt
- Es existieren drei AES Varianten mit Schlüssellänge K und Rundenanzahl n:
  - AES-128: $K = 128$-bit Schlüssel mit $n = 10$ Runden
  - AES-192: $K = 192$-bit Schlüssel mit $n = 12$ Runden
  - AES-256: $K = 256$-bit Schlüssel mit $n = 14$ Runden

ADVANCE ENCRYPTION STANDARD (AES)

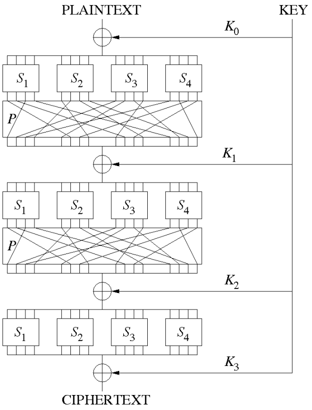


MODERNE METRIKEN FÜR KRYPTOGRAPHISCHE ALGORITHMEN

- Shannonsche Theorie
  - Wichtige **Konstruktionsprinzipien** für die kryptographische Sicherheit sind **Konfusion** und **Diffusion**.
- **Konfusion**:
  - Die Konfusion einer Blockchiffre ist dann groß, wenn die statistische Verteilung der Chiffretexte in Abhängigkeit von der Verteilung der Klartexte für den Angreifer zu groß ist (keine Ausnutzbarkeit).
  - Meistens wird die S-Box als nicht-lineares Element in der Blockchiffre für die Konfusion genutzt.
- **Diffusion**:
  - Die Diffusion einer Blockchiffre ist dann groß, wenn jedes einzelne Bit des Klartextes (und des Schlüssels) möglichst viele Bits des Chiffretextes beeinflusst (typisch etwa 50 %).
  - Permutationen oder Schiebeoperationen werden in Blockchiffren genutzt, um die Diffusion zu realisieren.

SUBSTITUTIONS-PERMUTATIONS-NETZWERK-CHIFFRE (SPN)

- → Der Plaintext $\mathcal{P}$ wird in mehrere gleiche große Blöcken aufgeteilt $P_1, P_2, \ldots, P_n \in \mathcal{P}$
- → Die Verschlüsselungsvorschrift besteht aus einer mehrfach wiederholten Rundenfunktion $f_R(\cdot)$ mit individuellem Rundenschlüssel K_i
- → Die Rundenfunktion besteht aus einer nichtlinearen Sbox und einer Permutation.
- → Für die Entschlüsselung wird die Umkehrfunktion $f_R^{-1}(\cdot)$ zu $f_R(\cdot)$ benötigt.

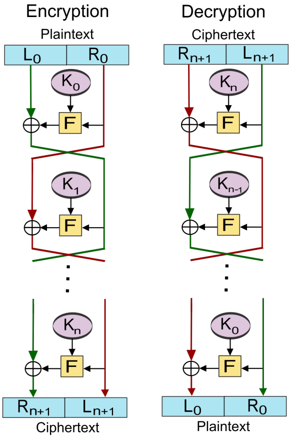


Quelle:

https://de.wikipedia.org/wiki/Substitutions-Permutations-Netzwerk

FEISTEL-CHIFFRE (LUBY-RACKOFF BLOCKCHIFFREN)

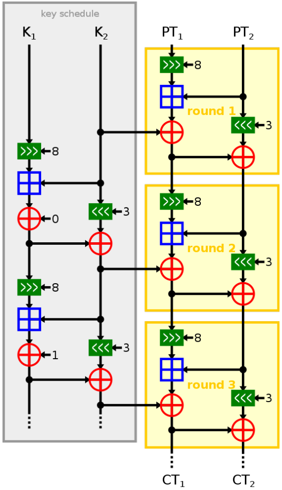
- → Basiert auch auf der Mehrfachausführung von Rundenfunktionen mit Rundenschlüsseln.
- → Plaintext wird in zwei Blöcke (L und R) aufgeteilt, die nach jeder Runde vertauscht werden.
- → Verschlüsselung und Entschlüsselung kann mit den gleichen Rundenfunktionen $f_R(\cdot)$ ausgeführt werden
- → Das Design von $f_R(\cdot)$ ist schwieriger als bei SPN-Chiffren



Quelle: https://en.wikipedia.org/wiki/Feistel_cipher

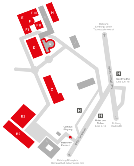
ADD–ROTATE–XOR-CHIFFRE

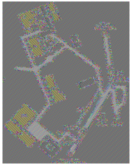
- ARX-Chiffren benutzen als Basisoperationen nur Addition, Rotation und XOR
- Dadurch sind diese sehr kompakt implementierbar und effizient für Standardprozessoren
- Nicht bester Trade-off bei der Umsetzung in Hardware
- Die Resistenz gegen kryptanalytische Angriffe noch nicht umfänglich, da es recht junge Verfahren sind

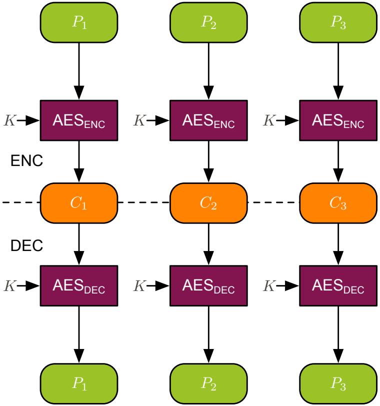


ELECTRONIC CODE BOOK (ECB)

- → AES verarbeitet die 128-bit Blöcke $P_1, P_2, P_3 \in \mathcal{P}$ des Plaintextes unabhänghig von einander.
- → Auf die gleiche Eingabe erfolgt eine gleiche Ausgabe, ähnlich wie monoalphabetischen Chiffren.
- → Spezielle Betriebsmodi sind notwendig!







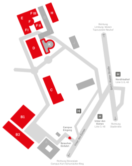
BETRIEBSMODUS VON BLOCKCHIFFREN

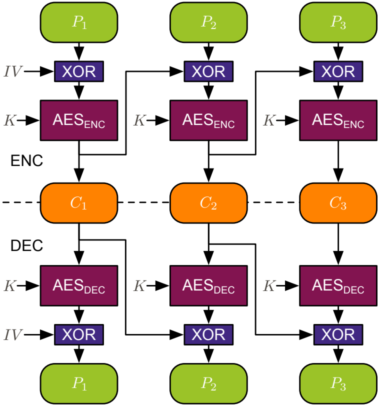
Blockchiffren können in verschiedenen Modi betrieben werden

Name	Bezeichnung	Einsatzgebiet
ECB	Electronic Code Book	Einsatz in Ausnahmefällen oder wenn nur ein Block verschlüsselt werden muss
CBC	Cipher Block Chaining	Verschlüsselung bei Datenübertragung
CFB	Cipher Feedback Mode	Verschlüsselung entspricht einer selbstsynchronisierenden Stromchiffre
OFB	Ouput Feedback Mode	Verschlüsselung mit Fehlerresistenz
CTR	Counter Mode	Verschlüsselung mit Fehlerresistenz; macht aus Blockchiffre eine Stromchiffre
XTS	Ciphertext Stealing	Festplattenverschlüsselung; Besonders gesichert gegen Angriffe auf Implementierung
GMAC/C-MAC	Galois/Cipher Message Authentication Mode	Authentifikation von Daten (Abschnitt „Message Authentication Codes“)
GCM	Galois-Counter Mode	Verschlüsselung und Authentifikation von Daten (Abschnitt „Message Authentication Codes“)

CIPHER BLOCK CHAINING (CBC)

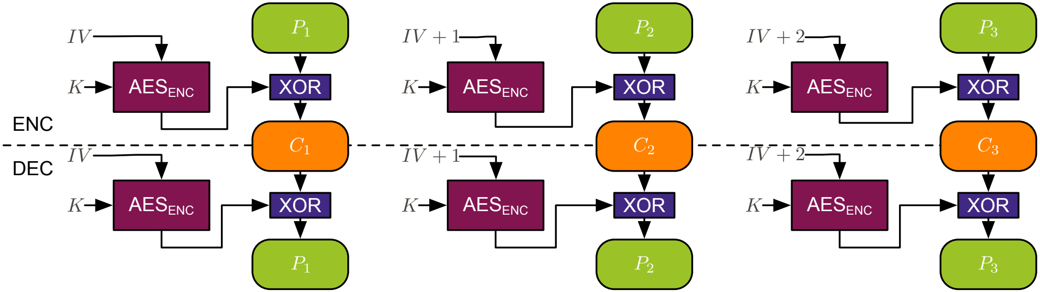
- Ciphertext des vorherigen Blocks fließt in nächsten Block mit ein (via XOR)
- Zufälliger Initialisierungsvector IV, um gleiche Plaintexte $P_1 = P_2$ zu unterschiedlichen Ciphertexten $C_1 \neq C_2$ zu verschlüsseln
- Nachteil ist, dass der Mode nicht parallelisiert ist und Übertragungsfehler propagiert werden



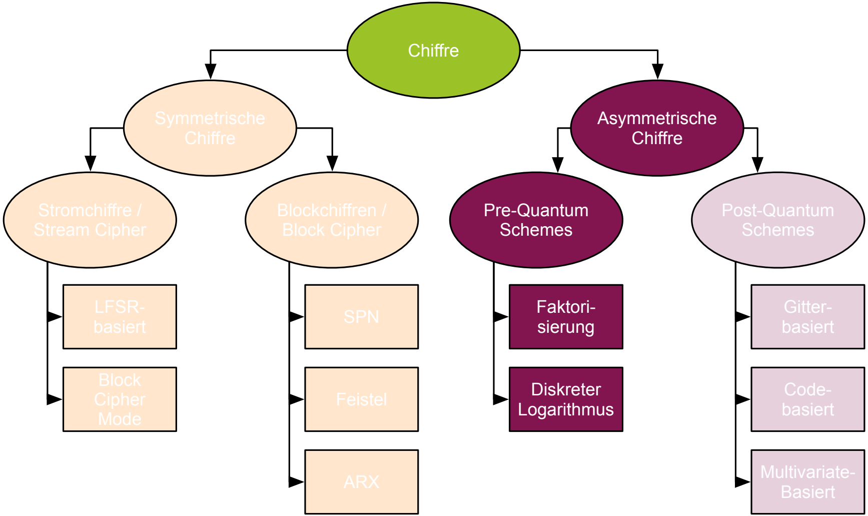


COUNTER MODE (CTR)

- → Zufälliger IV wird verschlüsselt und mit Plaintext ver-XORed
  - → Hochgradig parallelisierbar und AES kann vorberechnet werden (Stromchiffre)
  - → Übertragungsfehler wirken sich nur auf lokalen Block aus
  - → Nur die Verschlüsselungsvorschrift wird benötigt für Ver- und Entschlüsselung

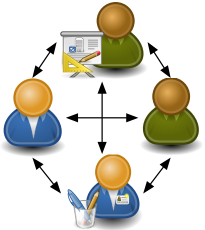


ASYMMETRISCHE VERSCHLÜSSELUNGSVERFAHREN



SYMMETRISCHE VS. ASYMMETRISCHE VERSCHLÜSSELUNGSVERFAHREN

- Bei AES benötigen beide Parteien den gleichen, geheimen Schlüssel K
  - AES fällt daher in die Kategorie der **Symmetrischen** oder **Private-Key** Verschlüsselungsverfahren
- Meist existiert aber kein geheimer, ausgetauschter Schlüssel
  - Ad-hoc Kommunikation mit unbekannten Parteien im Internet
  - Jedes Paar Parteien benötigt eigenen Schlüssel ($\frac{m(m-1)}{2}$ bei m Parteien)
- Lösung: **Asymmetrische** oder **Public-Key** Verschlüsselungsverfahren



ASYMMETRISCHE VERSCHLÜSSELUNG GRUNDPRINZIP

1. Empfänger generiert ein Schlüsselpaar K_E, K_D.
   - → K_E: **öffentlicher** Schlüssel, der von allen Parteien zum Verschlüsseln genutzt werden kann.
   - → K_D: **geheimer** Schlüssel, mit dem Ciphertexte entschlüsselt werden können.
   - → K_E und K_D stehen in einer Relation $K_E = f(K_D)$ und $K_D = f^{-1}(K_E)$.

2. Sender nutzt K_E, um Plaintext P mit $C = Enc_{K_E}(P)$ zu verschlüsseln.

3. Nur Empfänger kann C mit $P = Dec_{K_D}(C)$ zu entschlüsseln.

4. Asymmetrische Verfahren basieren auf mathematisch schweren Problemen, um sicherzustellen, dass nicht von K_E auf K_D geschlossen werden kann.

ASYMMETRISCHE VERSCHLÜSSELUNG RIVEST-SHAMIR-ADLEMAN (RSA)

- RSA wurde 1977 entwickelt von R. **R**ivest, A. **S**hamir und L. **A**dleman.
- RSA kann zur asymmetrischen Ver-/Entschlüsselung genutzt werden.
- Die Sicherheit von RSA basiert auf:
  - Dem RSA Problem (e-te Wurzel modulo N)
  - Der Schwierigkeit der Primfaktorzerlegung für große Zahlen
- RSA Ver-/Entschlüsselung mit n-bit Modulus N hat Komplexität $\mathcal{O}(n^3)$
  - Multiplikation zweier n-bit Werte hat $\mathcal{O}(n^2)$
  - Exponentation mit n-bit Exponent hat $\mathcal{O}(n^3)$
- Schlüsselgenerierung ist sehr rechenintensiv
  - Finden und Verifizieren von Primzahlen

ASYMMETRISCHE VERSCHLÜSSELUNG RIVEST-SHAMIR-ADLEMAN (RSA)

Alice		Bob
Schlüsselgenerierung		
Wähle zufällige Primzahlen p und q		
Berechne $N = p \cdot q$		
Wähle e zufällig mit $ggT(\phi(N), e) = 1$		
Berechne d als: $e \cdot d \mod \phi(N) = 1$		
Setze $K_E = (N, e)$ und $K_D = d$	$\xrightarrow{K_E = (N, e)}$	**Verschlüsselung**
		Berechne $C = P^e \mod N$
Entschlüsselung	$\xleftarrow{C}$	
Berechne $P = C^d \mod N$		

EINSCHUB ZUR EULERSCHE PHI-FUNKTION

- $\phi(m)$ gibt die Anzahl derjenigen natürlichen Zahlen $n < m$ an, die teilerfremd zu m sind; $m, n \in N, \phi m = |0 \leq n \leq m \mid ggT(n, m) = 1|$
  - Beispiel: $\mathbb{Z}_5 = 0, 1, 2, 3, 4$, $\phi(5) = 5$, da nur $ggT(0, 5) \neq 1$ ist für $\forall n \in \mathbb{Z}_5$

- Spezialfälle:
  - $p \in \mathbb{P} \Rightarrow \phi(p) = (p - 1)$
  - $k \in \mathbb{N} \Rightarrow \phi(p^k) = p^{k-1} \cdot (p - 1)$
  - $p, q \in \mathbb{P}$ und $p \neq q \Rightarrow \phi(p \cdot q) = \phi(q) \cdot \phi(p) = (p - 1) \cdot (q - 1)$

- Weitere nützliche Eigenschaften:
  - Wenn $ggT(a, n) = 1$ ist, dann gilt: $a^{\phi(n)} \mod n = 1$
  - Ist $n = p \in \mathbb{P}$, so ergibt sich der Satz von Ferma: $a^{p-1} \mod p = 1; (a \neq 0)$
  - Somit kann man das modular Inverse berechnen: $a^{-1} mod\, p = a^{p-2} \mod p; (a \neq 0)$

WESHALB IST RSA SICHER?

- Öffentlich ist: $K_E = (N, e), C$
- Geheim sind: $K_D = d, p, q, P$

- Berechnung des Plaintextes $C = P^e \mod N$
  - Invertierung: $P = \sqrt[e]{C} \mod N \rightarrow$ Problem der e-ten Wurzel $\mod N$.

- Alternative: Berechnung des privaten Schlüssels $K_D = d$
  - Bedingung $e \cdot d \mod \phi(N) = 1$
  - Berechne $\phi(N) = (p-1) \cdot (q-1) \Rightarrow$ Problem der Primfaktorzerlegung

RSA – SICHERHEIT

- RSA ist als asymmetrisches Verfahren bereits im Chosen-Plaintext Modell
  - Angreifer kann beliebige Plaintexte mit öffentlichem Schlüssel K_E verschlüsseln
- Kurze Plaintexte können via Brute-Force gebrochen werden
  - Telefonnr. (≈32 bit): Verschlüsseln aller Nummern mit K_E und Vergleich mit Ciphertext
- Exponent e für Verschlüsselung wird kurz gewählt, um Berechnung zu beschleunigen
  - $e \in \{3, 65537\}$
- Textbuch RSA benötigt weitere Paddingverfahren, um Brute-Force Angriffe auszuschließen
  - **RSA-OAEP Padding**: Nachricht wird um Zufallszahl und Prüfsumme erweitert

IMPLEMENTIERUNG ASYMMETRISCHE VERSCHLÜSSELUNG

- → Asymmetrische Verfahren nur sehr schwer sicher zu implementieren [B99]
  - → Primzahlen in RSA dürfen weltweit nicht doppelt vorkommen [ND+12]
  - → Bestimmte Primzahlen müssen vermieden werden [C96]
  - → Bestimmte Werte für d und e müssen vermieden werden
  - → Fehler im Paddingverfahren können zur Kompromittierung des Schlüssels führen [B98]

- → Etliche Tricks können asymmetrische Verfahren beschleunigen
  - → Chinesischer Restsatz
  - → Wahl einer Basis aus einer Restklassengruppe mit kleinerer Ordnung

- → Implementieren Sie asymmetrische Verfahren **nicht selbst**, sondern nutzen Sie bestehende Bibliotheken!

HYBRIDE VERSCHLÜSSELUNG (1/2)

Aspekt	Symmetrische Verschlüsselung	Asymmetrische Verschlüsselung
Vorteile	Sehr schnell (∼Gigabyte/Sekunde)	Es muss kein geheimer Schlüssel ausgetauscht sein
Nachteile	Geheimer Schlüssel muss ausgetauscht sein	Langsam (∼Hunderte Kilobyte/Sekunde)

→ Hybride Verschlüsselung kombiniert die Vorteile beider Verfahren:

1. Asymmetrische Verfahren, um einen symmetrischen Schlüssel auszuhandeln
2. Symmetrische Verfahren, um die Daten zu übertragen

HYBRIDE VERSCHLÜSSELUNG (2/2)

Alice		Bob
	$\xleftarrow{\quad K_E \quad}$	Schlüsselpaar(K_E, K_D)
Sym. Schlüssel K wählen	$\xrightarrow{\quad C_K = Enc_{C_E}(K) \quad}$	
		$K = Dec_{K_D}(C_K)$
	$\xleftarrow{\quad C = Enc_K(P) \quad}$	
	$\xleftrightarrow{\quad C_K = Enc_K(\dots) \quad}$	

DIFFIE-HELLMAN VERFAHREN (DH)

- → Asymmetrisches Verfahren zur Schlüsselvereinbarung, entwickelt in 1976
- → Basiert auf dem diskreten Logarithmusproblem in primen Restklassenringen
- → Voraussetzung: Alice und Bob kennen öffentliche Primzahl p und Basis g
  - → Mögliche Primzahlen und Basen sind in Standards definiert DHP
- → DH kann nicht für Verschlüsselung genutzt werden, sondern nur für Schlüsselvereinbarung
  - → DH benötigt weiteres Verschlüsselungsverfahren (z.B. symmetrisches Verfahren)

EINSCHUB ZYKLISCHE GRUPPEN

- Eine Gruppe $(G, \circ)$ hat eine endliche Anzahl von Elementen. Die Anzahl der Elemente gibt die Ordnung (Kardinalität) der Gruppe G mit $|G|$ an.
  - Beispiele:
    - $\mathbb{Z}_9 = \{0, 1, 2, 3, 4, 5, 6, 7, 8\} \Rightarrow |\mathbb{Z}_9| = 9$
    - $\mathbb{Z}_9^* = \{1, 2, 4, 5, 7, 8\} \Rightarrow |\mathbb{Z}_9^*| = \phi(9) = 6$

- Die Ordnung $ord(a)$ eines Elements $a \in < G, \circ >$ ist die kleinste positive ganze Zahl k mit $a^k = a \circ a \circ a \ldots \circ a = 1$.

- Eine Gruppe G ist zyklisch, wenn die Gruppe G ein Element α mit $ord(\alpha) = |G|$ enthält. α ist ein Generator oder primitives Element von G.
  - Beispiel: Das Element $\alpha^i = a = 2$ ist ein Generator für $\mathbb{Z}_{11}^*$

$$
\begin{array}{llll}
a^1 \equiv 2 \mod 11 & a^2 \equiv 4 \mod 11 & a^3 \equiv 8 \mod 11 & a^4 \equiv 5 \mod 11, \\
a^5 \equiv 10 \mod 11 & a^6 \equiv 9 \mod 11 & a^7 \equiv 7 \mod 11 & a^8 \equiv 3 \mod 11, \\
a^9 \equiv 6 \mod 11 & a^{10} \equiv 1 \mod 11 & &
\end{array}
$$

DIFFIE-HELLMAN PROTOKOLL

Alice		Bob
Wählt a		Wählt b
Berechne $A \equiv g^a \mod p$		Berechne $B \equiv g^b \mod p$
	$\xrightarrow{A}$	
	$\xleftarrow{B}$	
Berechne $K \equiv B^a \mod p \equiv g^{b \cdot a} \mod p$		Berechne $K \equiv A^b \mod p \equiv g^{a \cdot b} \mod p$
	K kann für sym. Verschlüsselung genutzt werden $\longleftrightarrow$	

WIESO IST DH SICHER?

- Eve möchte den Schlüssel K berechnen
  - Öffentlich: $g, p, A \equiv g^a \mod p$, $B \equiv g^b \mod p$
  - Geheim: a, b und $K = p^{a \cdot b}$
- Um a (oder b) zu finden, muss Eve den diskreten Logarithmus berechnen:
  - $a \log_g A \mod p$ oder
  - $b \log_g B \mod p$
- Aber: Bester bekannter Algorithmus zur Berechnung des diskreten Logarithmus hat Komplexität $\mathcal{O}\left(2^{\frac{n}{2}}\right)$ für n-bit p (vereinfacht!).

DH SCHLÜSSEL BEHALTEN ODER LÖSCHEN?

- Originales DH Protokoll: a und b werden für jeden Austausch neu generiert
  - **Vorteil**: Falls a oder b einer Sitzung veröffentlich werden, ist nur die aktuelle Sitzung korrumpiert (sog. **Forward Secrecy**) $\Rightarrow$ Standard in vielen Protokollen
  - **Nachteil**: Mallory kann Schlüsselaustausch abfangen, da Alice und Bob sich nicht anhand von A und V authentifizieren können (sog. **Man-in-the-Angriff**)

Alice	Mallory	Bob
$\xrightarrow{A \equiv g^a \mod p}$		$\xleftarrow{B \equiv g^b \mod p}$
$\xleftarrow{F \equiv g^f \mod p}$		$\xrightarrow{F \equiv g^f \mod p}$
$\xleftrightarrow{K \equiv g^{a \cdot f} \mod p}$		$\xleftrightarrow{K \equiv g^{b \cdot f} \mod p}$

ZUSAMMENFASSUNG

→ Verschlüsselungsalgorithmus AES

→ Verwendungszweck von Betriebsmodi für Blockchiffren

→ Passende Betriebsmodi für einen einfachen Anwendungsfall auswählen

→ Unterschied zwischen öffentlichem und privatem Schlüssel

→ Verschlüsselungsalgorithmus RSA

→ Vor- und Nachteile von symmetrischen- und asymmetrischen Verfahren

→ Aufbau und Vorteile von hybriden Verschlüsselungsverfahren

→ DH Verfahren sowie Vor- und Nachteile des Behaltens der öffentlichen Schlüssel