Schnelldurchlauf durch die Themen der Krypto-Vorlesung

- Kerckhoff-Prinzip: Chiffrierverfahren öffentlich
- verschiedene Angriffs-Szenarien, z.B. Cyphertext only oder Man-in-the-Middle
- klassische Krypto: Sender und Empfänger haben von vornherein den geheimen Schlüssel
- Kryptographie – Kryptoanalyse – Steganographie – Signaturen etc.
- Historische Verfahren: Cäsar, affin, monoalphabetisch, Playfair, Enigma, Vigenère, Autokey, homophon
- Buchstabenhäufigkeiten im Deutschen. Maßzahlen:
  Koinzidenzindex $IC = \sum_{i=1}^{n} p_i^2$ bzw. Entropie $-\sum_{i=1}^{n} p_i \log_2 p_i$
  für W'Verteilung $(p_1, \dots, p_n)$.
  Schätzung des Koinzidenzindex bei gegebenem Text:
  $$\widetilde{IC} = \sum_a \frac{h(a)}{m} \cdot \frac{h(a)-1}{m-1}$$, m Textlänge
  $h(a)$ Häufigkeit von a
- Kasiski-Methode, Friedman-Methode zum Knacken der Vigenère-Chiffre
- Rückgekoppelte Schieberegister. Max. Periodenlänge: $2^n - 1$
- Stromchiffre versus Blockchiffre

- Pseudozufallsgeneratoren, z.B. $x_{i+1} = (a \cdot x_i + b) \bmod n$
  Blum-Blum-Shub: $x_{i+1} = x_i^2 \bmod n$, wobei $n = p \cdot q$
  beim Pollard-ρ-Algorithmus: $x_{i+1} = (x_i^2 + 1) \bmod n$
- Feistel-Netzwerk (DES), inverses DES
- Shannon-Entropie: gemeinsame, bedingte, Transinformation, absolute Sicherheit, One-time-pad
- Visuelle Kryptographie
- Moderne Kryptographie: Info-Austausch ausschließlich über unsicheren Kanal; hybrides System
- Modular-Arithmetik: Gruppen $(\mathbb{Z}_n, +)$; $(\mathbb{Z}_n^*, *)$
  Körper $(\mathbb{Z}_n, +, *)$ wenn n Primzahl; Inverse; ggT
- Euler-Funktion $\varphi(n) = |\mathbb{Z}_n^*|$. Berechnung von $\varphi(n)$, wenn Faktorisierung von n bekannt.
- multiplikative Inverse mittels Extended Euklid / Bezout:
  $(a,b) \mapsto (d, x, y)$; $d = ggT(a,b)$, $d = ax + by$
- Komplexität von Algorithmen, worst-case vs. average-case, polynomial vs. exponentiell, probabilistische Algorithmen, O-Notation
- Sicherheitsniveau von k Bit, falls bester Knack-Algor. Laufzeit 2^k hat. $k=80$ ist ausreichend.
- Satz von Euler / Fermat: $a^{\varphi(n)} \equiv 1 \pmod n$ bzw.
  $a^{n-1} \equiv 1 \pmod n$, n prim. $a \in \mathbb{Z}_n^*$, $\varphi(n) = |\mathbb{Z}_n^*|$.
- Primzahlsatz: $\pi(n) \sim \frac{n}{\ln n}$

- Modulare Exponentiation „square + multiply": $O(m^3)$.
- Carmichael-Zahlen: $a^{n-1} \equiv 1 \pmod{n}$; für alle $a \in \mathbb{Z}_n^*$, aber n keine Primzahl.
- Miller-Rabin-Primzahltest: $n-1 = 2^t \cdot u$, $t \geq 1$, u ungerade. $a^{n-1} \equiv 1 \pmod{n}$ sowie, falls $a^{2k} \equiv 1 \pmod{n}$ so muss $a^k \equiv 1$ oder $a^k \equiv -1 \pmod{n}$ sein. Fehlerw'keit $\leq \frac{1}{4}$
- Diffie-Hellman-System liefert gemeinsamen Schlüssel; anschließend klassische Nachrichtenverschlüsselung (hybrid)
- Shamir's No Key System (mit Pohlig-Hellman-Verschlüsselung)
- Zyklische Gruppen. Erzeugendes Element a: $\langle a \rangle = G$. $\mathbb{Z}_n^*$ ist zyklisch $\Leftrightarrow n = 1, 2, 4, p^k, 2p^k$ (p ungerade Primzahl), Gauß.
- Primitivwurzel-Kriterium: $a^{\varphi(n)/q} \not\equiv 1 \pmod{n}$ für alle q die Teiler von $\varphi(n)$ sind. ($\varphi(n) = n-1$ falls n Primzahl).
- Sichere Primzahlen und Sophie-Germain-Primzahlen
  $n = 2 \cdot q + 1$
- Diskreter Logarithmus $n, a, y \mapsto x$ sodass $a^x \equiv y \pmod{n}$
- Asymptotisch: $\varphi(n) \geq \frac{n}{6 \cdot \ln(\ln(n))}$
- U Untergruppe von $G \Rightarrow |U|$ ist Teiler von $|G|$. (Lagrange)
- Wenn G zyklisch, so gibt es $\varphi(|G|)$ Erzeuger.
- $a^x \equiv a^y \pmod{n} \Leftrightarrow x \equiv y \pmod{\varphi(n)}$; a Erzeuger
- Modulo Primzahl n immer genau 2 oder 0 Quadratwurzeln
  Quadratwurzeln von 1 sind ± 1. $|QR_n| = |QNR_n| = \frac{n-1}{2}$

- Quadratwurzel-Kriterium: (Euler-Kriterium)
  a ist quadr. Rest mod n, n Primzahl, falls $a^{(n-1)/2} \equiv 1 \pmod n$
- Bei Primzahlen: $a^{(n-1)/2} \equiv \left(\frac{a}{n}\right)$ (Legendre-Symbol)
- Jacobi-Symbol über Primfaktorisierung von n definiert, aber effizient berechenbar mittels alternierend Modul-Berechnung + Quadrat. Reziprozitätsgesetz
- Solovay-Strassen-Primzahltest: $a^{(n-1)/2} \stackrel{?}{\equiv} \left(\frac{a}{n}\right)$; a zufällig
- relativer Vergleich algorithmischer Probleme mittels $\leq_{eff}$ (Orakel-Reduzierbarkeit). Def. von P und NP, NP-vollständig.
- Verschiedene Äquivalenzen bzgl. $\equiv_{eff}$ bzw. $\equiv_{probeff}$:
  $\varphi(n)$ berechnen $\equiv_{eff}$ Faktorisieren; Diffie-Hellman $\equiv_{eff}$ ElGamal
- Einwegfunktion: nicht-umkehrbar, evtl. mit „Falltür".
  $\exists$ starke Einwegfkt $\Rightarrow P \neq NP$. (doch umkehrbar bei Kenntnis eines Schlüssels)
- Faktorisieren, Diskreter Logarithmus: nur expon. Algorithmen
- Miller-Rabin-Test überprüft ob „Profil" $(a^{n-1}, a^{(n-1)/2}, \ldots, a^u) \bmod n$ „regulär" ist.
- Public Key Systeme: abstrakt mittels $D(\cdot)$ und $E(\cdot)$.
- RSA: $n = p \cdot q$; $\varphi(n) = (p-1)(q-1)$; $e \in \mathbb{Z}_{\varphi(n)}$; $d = e^{-1} \pmod{\varphi(n)}$
  $c = m^e \bmod n$; $m = c^d \bmod n$
- Public Key nach Rabin: $n = p \cdot q$; $m \mapsto m^2 \bmod n$
- Quadratwurzelziehen mod Primzahl n effizient möglich, vor allem, wenn $n \equiv 3 \pmod 4$; „$\sqrt{a}$" $= a^{(n+1)/4} \bmod n$.
  Fall $n \equiv 1 \pmod 4$ etwas schwieriger.
- Public Key nach ElGamal: Asymmetrische Version von Diffie-Hellman & anschließende klassische Verschlüsselung mittels modularer Multiplikation; Randomisierung

- Chinesischer Restsatz: $n_1, \dots, n_k$ teilerfremd. $a_1 < n_1, \dots, a_k < n_k$. Dann $\exists x \in \mathbb{Z}_{n_1 \cdots n_k}$:
  $x \equiv a_1 \pmod{n_1}, \dots, x \equiv a_k \pmod{n_k}$.
  Notation: $x \multimap\!\bullet\ (a_1, \dots, a_k)$
- Falls $x \multimap\!\bullet\ (a_1, \dots, a_k)$, $y \multimap\!\bullet\ (b_1, \dots, b_k)$, so gilt:

  $x \text{ op } y \multimap\!\bullet\ (a_1 \text{ op } b_1, \dots, a_k \text{ op } b_k)$

- Signatur von Dokument m: Eigenen, geheimen Schlüssel auf m anwenden: $\tilde{m} = D(m, k')$. Verifikation: $E(\tilde{m}, k) \stackrel{?}{=} m$
  D, E müssen kommutieren. (Das tun sie bei RSA)
- kürzere Unterschrift durch Hashing: man unterschreibt $h(m)$.
- Anforderungen an Hashfunktionen: Einwegfunktion, Kollisionsresistenz.
- Geburtstagsproblem: Nach ca. $1{,}2 \cdot \sqrt{m}$ vielen Zufallszahlen (bzw. Hashwerten) im Zahlenbereich bis m ist es bereits „wahrscheinlich", dass eine Kollision auftritt.
  (Bei Hashing bedeutet dies $h(x) = h(x')$, $x \neq x'$.)
- El Gamal-Version für das Signieren.
- Blinde Unterschrift bei RSA: m „Verblenden" zu
  $\hat{m} = m \cdot z^e$, $\hat{m}$ unterschreiben ergibt $(\hat{m})^d = m^d \cdot z^{ed} = m^d \cdot z$
  Verblendung entfernen: dies mit z^{-1} multiplizieren.
- Algorithmen fürs Faktorisieren:
  - Naiv, Probeteiler ausprobieren: $O(2^{m/2})$
  - Pollard-ρ: Pseudozufallszahlen ergeben nach $O(2^{m/4})$ vielen sehr Wahrscheinlich „Kollision": $z_i \equiv z_j \pmod{p}$
    Dann ergibt $ggT(z_i - z_j, n) = p$. Auffinden solcher z_i, z_j mit Cycle Detection Trick: z_k, z_{2k} testen.

- Pollard $(p-1)$: Gut, wenn für einen Primfaktor p von n gilt, dass $p-1$ ausschließlich kleine Primfaktoren besitzt: $p-1$ teilt $B!$
  Dann gilt: $ggT(a^{B!}-1, n) = p$. Worst Case: $O(2^{m/2})$
- Fermat-Faktorisierung: Finde Darstellung von n als Differenz zweier Quadratzahlen: $n = x^2 - y^2$
- Dixon: Finde viele x mit $x^2 \bmod n = p_1^{e_1} \cdots p_k^{e_k}$ wobei $\{p_1, \ldots, p_k\}$ Faktorbasis. Finde durch linear. Gleichungssystem über $\{0,1\}$ eine Auswahl von x'en, deren Produkt gerade Exponenten bei den p_i's ergibt.

■ Algorithmen für den Diskreten Logarithmus

- Naiv: teste alle x: $O(2^m)$
- Baby-Step-Giant-Step: Zerlege x in $x_1 \cdot \sqrt{n} + x_2$.
  Berechne Liste mit $(x_1, a^{x_1\sqrt{n}})$. Vergleiche mit $y \cdot a^{-x_2}$ - Werten. Laufzeit + Speicherplatz: $O(2^{m/2})$
- Pollard-ρ: Pseudo-zufällige Werte $a^s y^t$ generieren bis $a^s y^t \underset{(\text{mod } n)}{\equiv} a^{s'} \cdot y^{t'} \iff s + xt \equiv s' + xt' \pmod{n-1}$
  Definiere geeigneten Pseudozufallsgenerator; Cycle Detect. Trick.
  Laufzeit $O(2^{m/2})$
- Index Calculus: Faktorbasis $\{p_1, \ldots, p_k\}$. Finde viele Gleichungen $a^e \equiv p_1^{\alpha_1} \cdots p_k^{\alpha_k} \pmod{n} \iff$
  $$e \equiv \alpha_1 \cdot d\log p_1 + \cdots + \alpha_k \, d\log p_k \pmod{n-1}$$
  Bei genügend Gleichungen, löse nach den $d\log p_i$ auf.

Finde weitere Gleichung:

$$y \cdot a^{e} = p_1^{\beta_1} \cdots p_k^{\beta_k} \pmod{n} \iff$$

$$\mathrm{dlog}\, y + e = \beta_1\, \mathrm{dlog}\, p_1 + \dots + \beta_k \cdot \mathrm{dlog}\, p_k \pmod{n-1}$$

Löse nach $\mathrm{dlog}\, y$ auf.

- Zero Knowledge: mit Graphenisomorphie:

(commitment-challenge-response.)

$$\begin{tikzcd} G_0 \arrow[rr, mapsto, "\pi"] \arrow[dr, "\pi\circ\sigma"'] & & G_1 \arrow[dl, "\sigma"] \\ & H & \end{tikzcd}$$

P		V
$H := \sigma(G_1)$	$\xrightarrow{H}$	$b \in \{0,1\}$
	$\xleftarrow{b}$	
$\tau = \sigma$ oder $\tau = \pi\circ\sigma$	$\xrightarrow{\tau}$	$\tau(G_b) \overset{?}{=} H$

- Zero Knowledge-Definition: Die Kommunikation zwischen P und V (als Zufallsvariable) kann auch von Simulator geliefert werden – ohne Kenntnis des Geheimnisses.

- Zero Knowledge, allgemein mit <u>homomorpher Einwegfunktion</u> f

d.h: $f: (A, \circ) \to (B, \bullet)$ wobei $f(x \circ y) = f(x) \bullet f(y)$.

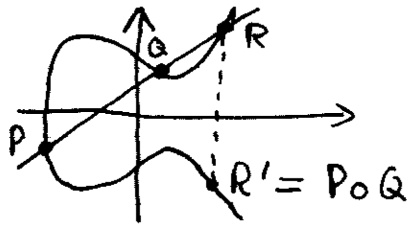
P		V
y zufällig		
$\tilde{y} = f(y)$	$\xrightarrow{\tilde{y}}$	$b \in \{0,1\}$
$z = x^{b} \circ y$	$\xleftarrow{b}$	
	$\xrightarrow{z}$	$f(z) \overset{?}{=} \tilde{x}^{b} \bullet \tilde{y}$

Umsetzung dieser allgemeinen Idee, z.B. mit modularer Exponentiation oder mit modularer Quadratfunktion (wobei $n = p \cdot q$) (Fiat-Shamir-Protokoll)

- Elliptische Kurve: $E_{a,b} = \{ (x,y) \in K^2 \mid y^2 = x^3 + ax + b \}$ wobei $4a^3 + 27b^2 \neq 0$.

$\{0\} \cup E_{a,b}$ ist Gruppe mit geeigneter Operation $\circ$.

- Veranschaulichung:



- Elliptische Kurven können bei allen Verfahren eingesetzt werden, die auf Diskret. Logarithmus beruhen.

- Aufteilen von Geheimnissen: einfachster Fall:

Hauptschlüssel $x \longmapsto \left\{ \begin{array}{l} x_1 = z_1 \\ x_2 = z_2 \\ \vdots \\ x_{k-1} = z_{k-1} \\ x_k = z_1 \oplus \dots \oplus z_{k-1} \oplus x \end{array} \right.$ $z_1, \dots, z_{k-1}$: Zufallsbitfolgen

- (k,n)-Schwellenwertsystem mit Polynom mit k Koeffizienten; $k-1$ Koeffizienten $\in \mathbb{Z}_p$ werden zufällig gewählt; der k-te Koeffizient ist x.

- Folgende Fakten/Methoden muss man für die Prüfung <u>nicht</u> auswendig wissen – dafür gibt's Bücher bzw. Skripten:
  - Kriterium, wann ein linearer Kongruenzgenerator maximale Periode erreicht.
  - die effiziente Berechnung des Jacobi-Symbols
  - Wie man die Quadratwurzel modulo einer Primzahl n berechnet, wenn n die Form hat: $n \equiv 1 \pmod 4$.
  - die Details der El Gamal-Signatur.
  - wie die $\circ$-Operation auf den Koordinaten der Punkte einer elliptischen Kurve definiert ist.