Kryptographie für CTFs

Eine Einführung - Teil 2

KITCTF

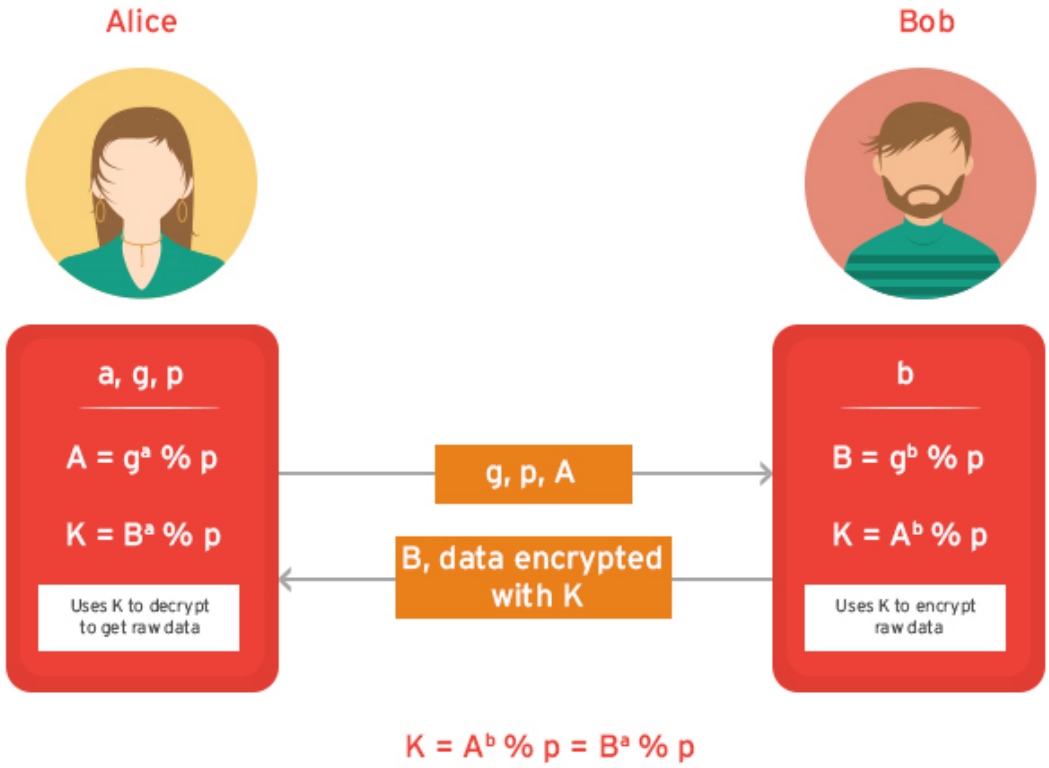
Letztes Mal: Einführung

- Klassiker
  - Caesar-Chiffre
  - Vigenère-Chiffre
- Symmetrische Verschlüsselungsverfahren (DES, AES)
  - Blockchiffren
  - Stromchiffren
- Asymmetrische Verschlüsselungsverfahren
  - RSA
  - ElGamal
  - McEliece

Heute

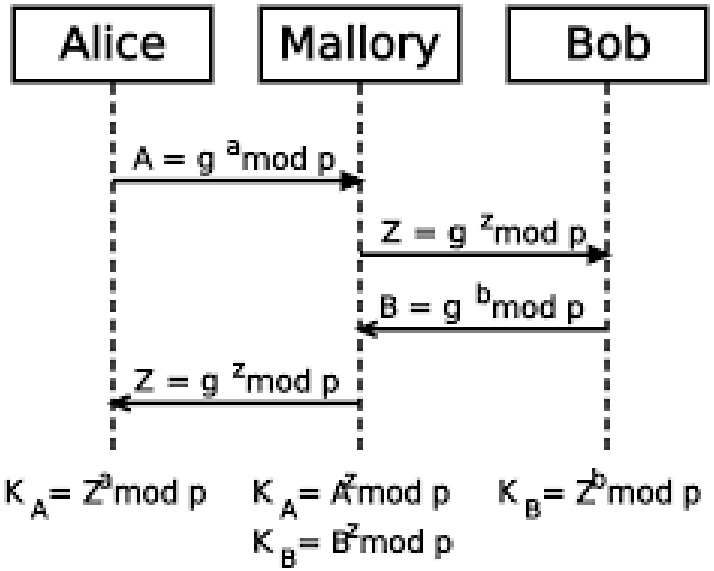
- Diskreter Logarithmus
  - Gegeben sei eine endliche zyklische Gruppe *G* mit Erzeuger α der Ordnung *n*. Finde $x, 0 \leq x \leq n-1$ mit $\alpha^x = \beta$.
- Verfahren
  - ElGamal
  - Diffie-Hellman-Schlüsselaustausch
- Attacken

Diffie-Hellman-Schlüsselaustausch



Diffie Hellman Key Exchange

- Attacken:
  - Man-in-the-Middle-Angriff
  - Seitenkanalangriffe
    - diskrete Exponentialfunktion, square-and-multiply Algorithmus
      - Zeitangriff
      - Stromangriff
  - DLog Problem lösen



Dlog berechnen

- Verfahren:
  - Pohlig-Hellman
  - Babystep-Giantstep-Algorithmus
  - Pollard Rho
  - Pollard Lambda

Pohlig-Hellman

- Pohlig-Hellman
  - Idee: Reduziere DLog Problem auf einfachere Teilprobleme
  - Primfaktorzerlegung der Gruppenordnung muss bekannt sein

- Vorgehen:
  - Reduktion des Problems der Gruppe G in zyklische Gruppen G_{p^k} deren Ordnung p^k ist, wobei p^k ein Teiler von n ist
  - Reduktion von Gruppen mit Primzahlpotenzordnung in Gruppen mit Primordnung
  - Zusammensetzen des Ergebnisses mittels des Chinesischen Restsatzes

Babystep-Giantstep-Algorithmus

- Falls $\alpha^x = \beta$ und $x = i * m + j$, m $\approx \sqrt{n}$, dann gilt
  $\alpha^{im+j} = \beta$ und somit auch $\alpha^j = \beta(\alpha^{-im})$

- Berechne Tabelle der „baby steps“(j, α^j)
- Berechne sukzessive „giant steps“$(\mathrm{i}, \beta(\alpha^{-im}))$
- Prüfe auf Gleichheit

- -> Erhöhter Speicheraufwand
- -> Für große Gruppen immer noch nicht einsetzbar

Pollard‘s Rho

- Teile Gruppenelemente in 3 etwa gleich große Mengen auf
- Definiere Sequenz $x_0 = 1, \quad x_{i+1} = f(x_i) = \begin{cases} x_i\beta, \text{falls } x_i \in S_1 \\ x_i^2, \text{falls } x_i \in S_2 \\ x_i\alpha, \text{falls } x_i \in S_3 \end{cases}$
- Dadurch werden Sequenzen $\{a_i\}$ und $\{b_i\}$ definiert mit $x_{i+1} = \alpha^{a_i}\beta^{b_i}$
- Falls $\alpha^{a_k}\beta^{b_k} = \alpha^{a_m}\beta^{b_m}$:
$$\beta = \alpha^{(a_m - a_k)/(b_k - b_m)}$$

Dlog Verfahren

Bedingung	Verfahren	Komplexität
	BSGS	$O(\sqrt{n})$
Faktorisierung $n = \prod_{i=1}^{k} q_i^{e_i}$ bekannt	Pohlig-Hellman	$O(\sqrt{q_i})$
	Pollard‘s Rho	$O(\sqrt{n})$
$x \in \{a, \dots, b\} \in Z_n$	Pollard‘s Lambda	$O\left(\sqrt{b-a}\right)$

Und mehr!

Nützliches/Aufgaben

- SageMath http://www.sagemath.org/
  free open-source mathematics software system

- Picoctf.com (writeup https://hgarrereyn.gitbooks.io/th3g3ntl3man-ctf-writeups/2017/picoCTF_2017/problems/cryptography/ECC2/ECC2.html)
- cryptopals.com (set 5, set 8)