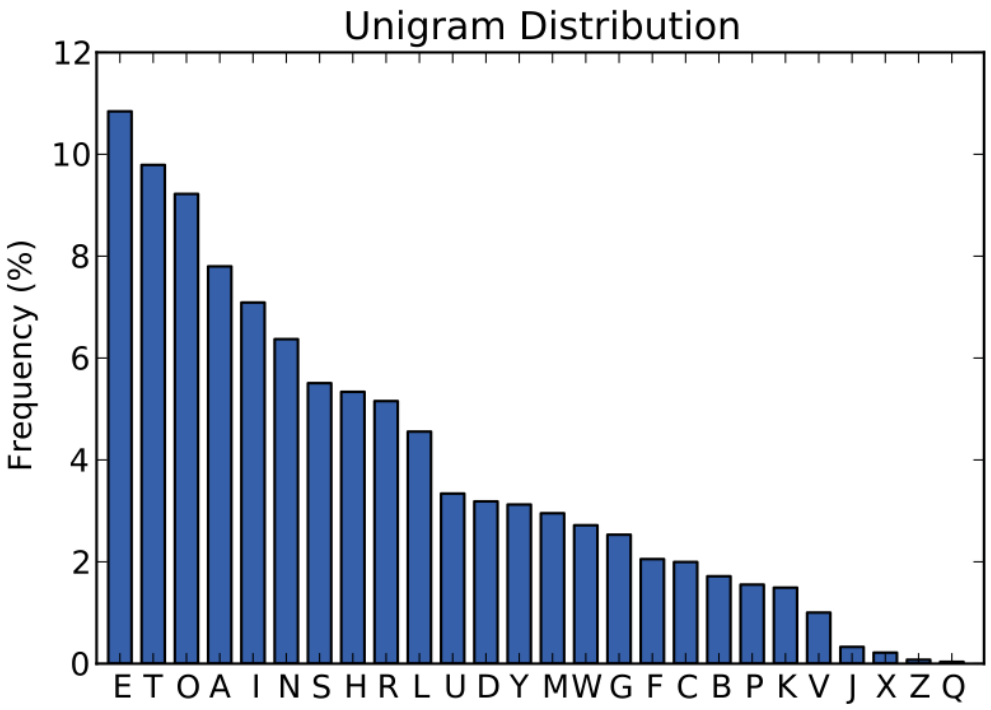
Frequenz-Analyse (in der Caesar-Verschlüsselung)



Multiplikations- und affine Verschlüsselung

Multiplikative Inverse

In $\mathbb{Q}$: $a = 2$ $\quad a^{-1} = \frac{1}{2}$ $\quad a \cdot a^{-1} = 2 \cdot \frac{1}{2} = 1$

In $\mathbb{Z}_{26}$: $a = 3$ $\quad a^{-1} = 9$ $\quad 3 \cdot b = 1$

$3 \cdot 9 = 27 \bmod 26 = 1$

Haben multiplikative Inverse in $\mathbb{Z}_{26}$

$(3,9), (11,19), (5,21), (15,7), (25,25)$

Theorie: ~~Alle Zahlen, welche 26 nicht teilen.~~

In $\mathbb{Z}_{20}$: Multiplikatives Inverses von 15?

Allgemeine Regel für $\mathbb{Z}_n$: Alle Zahlen, welche mit der (Gruppen)ordnung (n) von $\mathbb{Z}_n$ teilerfremd sind.

ausser 0 → Alle Zahlen haben ein multiplikatives Inverses, falls die Ordnung n prim ist.

Beispiel: $\mathbb{Z}_2$

10. April 2017

Finden von multiplikativen Inversen

1	0	26
0	1	15
1	-1	11
-1	2	4
3	-5	3
-4	7	1
15	-26	0

← Ordnung des Zahlenraums $\mathbb{Z}_n$

← Schlüssel

```
def findInverse(a, m):
    if gcd(a, m) != 1:
        return None

    u = [1, 0, a]
    v = [0, 1, m]

    while v[2] != 0:
        q = u[2] // v[2]
        for i in range(3):
            v[i], u[i] = u[i] - q*v[i], v[i]

    return u[0] % m
```

Ganzzahlige Division

1

Klartextbuchstabe — Schlüssel — Geheimtextbuchstabe

$3 \cdot 15 \;\%\; 26 = 19$ (Verschlüsseln)

$19 \cdot 7 \;\%\; 26 = 3$ (Entschlüsseln)

Multiplikatives Inverses des Schlüssels

Erweiterter Euklid Algorithmus

$\exists x, y: \; a \cdot x + b \cdot y = ggT(a,b)$

Euklid findet x, y und $ggT(a,b)$

Beispiel: $a = 15 \quad b = 26$

$15 \cdot x + 26 \cdot y = 1$

In $\mathbb{Z}_{26}$ ist dieser Term 0

$\Rightarrow \quad 15 \cdot x = 1 \quad \Rightarrow \quad x$ ist multiplikatives Inverses von 15

In $\mathbb{Z}_{26}$: $15^{-1} = 7$ $\quad 15^{-1} \neq \frac{1}{15} \notin \mathbb{Z}_{26}$!

Startwerte:

$26 \cdot 1 + 15 \cdot 0 = 26$ ← Zeile u
$26 \cdot 0 + 15 \cdot 1 = 15$ ← Zeile v
} beim Start

$26 \cdot 1 + 15 \cdot (-1) = 11$
$26 \cdot (-1) + 15 \cdot 2 = 4$
} 4 hat 2 Mal Platz in 11

$26 \cdot 3 + 15 \cdot (-5) = 3$
$26 \cdot (-4) + 15 \cdot 7 = 1$
$26 \cdot 15 + 15 \cdot (-26) = 0$

Ziel: Affine Verschlüsselung

Multiplikation + Caesar

~~encrypt(m, key1, key2)~~ encrypt(m, key)

key = Schlüssel für Caesar
key1 = Schlüssel für Multiplikation

Wie können wir die beiden Schlüssel in einem Schlüssel vereinen?

key = 60
- key1: $60 // 29 \Leftrightarrow \left\lfloor \frac{60}{29} \right\rfloor$
- key2: $60 \bmod 29 = 2$

$60 \equiv 2 \bmod 29$ $\quad \mathbb{Z}_{29}$

$key = key1 \cdot 29 + key2$

- Verschlüsselung und Entschlüsselung
  - Für ein beliebiges Alphabet
    alph = „ABC...“
  - Prüft, ob die Schlüssel sinnvoll sind.
    - key2 % 29 ≠ 0
    - key1 ≠ 1 key1 ≠ 0
    - ggT(key1, 29) = 1

- Zusatzaufgaben:
  - Schreibe einen Schlüsselgenerator, welcher sinnvolle Schlüssel zufällig generiert.
    → Möglichst allg. für alle Alphabetlängen

---

8. Mai 2017

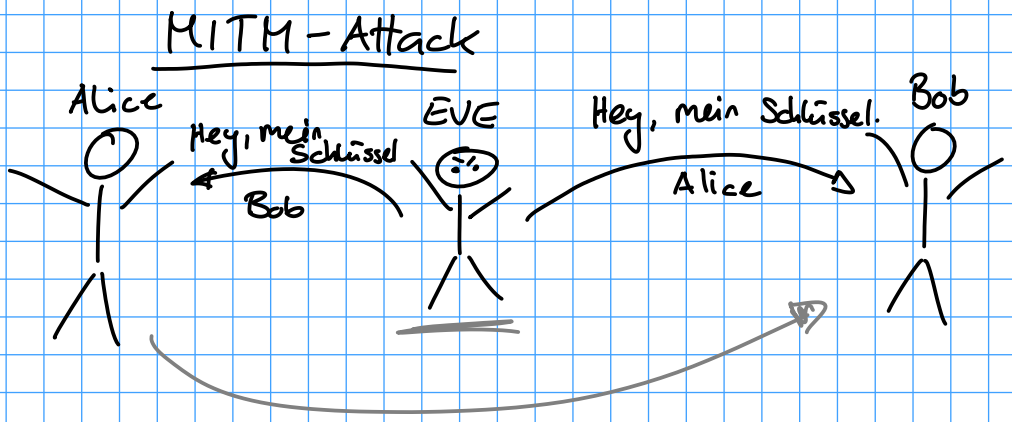
+ Definition einer Verschlüsselung
  -> Was ist zulässig
  -> Regeln für sicheren Algorithmus
  -> Begriffe (Schlüssel, Geheimtext, Klartext....)
+ Berechnen des multiplikativen Inversen in Zn
  -> Beurteilen, ob es existiert.
+ Analyse eines verschlüsselten Textes (Multiplikation, Caesar)
  -> Frequenzanalyse
+ RSA

(Prüfung als Jupyter-Notebook)

RSA

-> Warum brauchen wir nicht nur Verschlüsselung sondern auch Authentifizierung?

-> Wie funktioniert RSA



RSA

- Schlüssel generieren:
  - Zwei zufällige, grosse Primzahlen p, q
  - $n = p \cdot q$ (Schlüssellänge)
  - e eine Zahl mit $ggT(e, (p-1)\cdot(q-1)) = 1$
  - $d = e^{-1} \bmod (p-1)\cdot(q-1)$

  Public Key / öffentl. Schl.): n, e

  Private Key / priv. Schl.: n, d

- Verschlüsselung

  m: Nachricht

  $c = m^e \bmod n$ $\quad$ $m = c^d \bmod n$

  $m = (m^e)^d = m^{e \cdot d} = m^1$

$p = 41$ $q = 97$ $n = p \cdot q = 3977$

$(p-1)\cdot(q-1) = 40 \cdot 96 = 3840$

$e = 217$

$d = 3433$ Multiplikatives Inverses in $\mathbb{Z}_{3840}$

Alice: 217, 3977 $m = 42$

$c = 42^{217} \bmod 3977 = 3732$

Bob: 3433, 3977

$3732^{3433} \bmod 3977 = 42$

http://inventwithpython.com/hacking