



# Funktionsweise der Honey Encryption

Konstantin Kobs  
Tom Petersen

20. Januar 2015



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

- 
1. Einleitung
  2. Honey Encryption - Ein Beispiel
  3. Verfahren der Honey Encryption
    - Distribution Transforming Encoder
    - Verschlüsselung
  4. Einschränkungen
  5. Fazit

# Brute-Force-Angriff auf klassische Verfahren

---

$K_1 \rightarrow \text{yxV\#U}$

$K_2 \rightarrow \text{Katze}$

$K_3 \rightarrow \text{-CPK9}$

$\dots \rightarrow \dots$

## Verwendete Passwörter



<https://xato.net/wp-content/xup/passwordscLOUD.png>

# Honey Encryption - Idee

---

$K_1 \rightarrow \text{Hund}$

$K_2 \rightarrow \text{Katze}$

$K_3 \rightarrow \text{Maus}$

$\dots \rightarrow \dots$

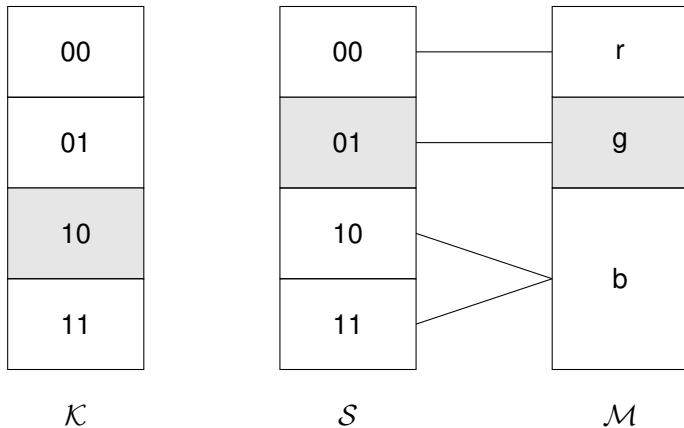
# Honey Encryption

---

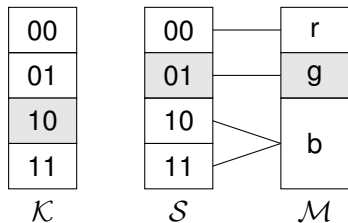
*Honey Encryption wurde entwickelt, um Ciphertexte zu generieren, die bei Entschlüsselung mit einem falschen Schlüssel zu einem plausibel wirkenden, aber unechten Klartext führen.*

- A. Juels, T. Ristenpart

# Honey Encryption - Ein Beispiel

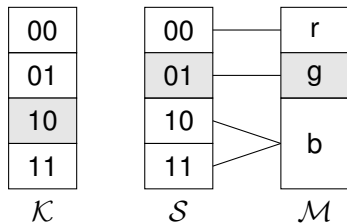


# Honey Encryption - Ein Beispiel





# Honey Encryption - Ein Beispiel



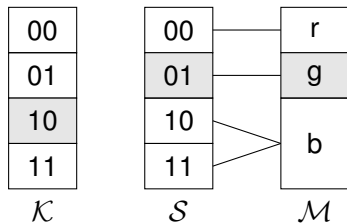
## Verschlüsselung

01  $\leftarrow$  Nachricht  $M$  (*grün*)

$\oplus$ 10  $\leftarrow$  Schlüssel  $K$

11  $\leftarrow$  Ciphertext  $C$

# Honey Encryption - Ein Beispiel



## Entschlüsselung

11  $\leftarrow$  Ciphertext  $C$

$\oplus \underline{10} \leftarrow$  Schlüssel  $K$

01  $\leftarrow$  Nachricht  $M$  (*grün*)

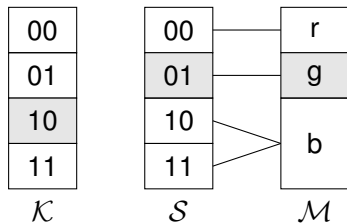
## Verschlüsselung

01  $\leftarrow$  Nachricht  $M$  (*grün*)

$\oplus \underline{10} \leftarrow$  Schlüssel  $K$

11  $\leftarrow$  Ciphertext  $C$

# Honey Encryption - Ein Beispiel



## Verschlüsselung

$01 \leftarrow \text{Nachricht } M \text{ (grün)}$   
 $\oplus \underline{10} \leftarrow \text{Schlüssel } K$   
 $11 \leftarrow \text{Ciphertext } C$

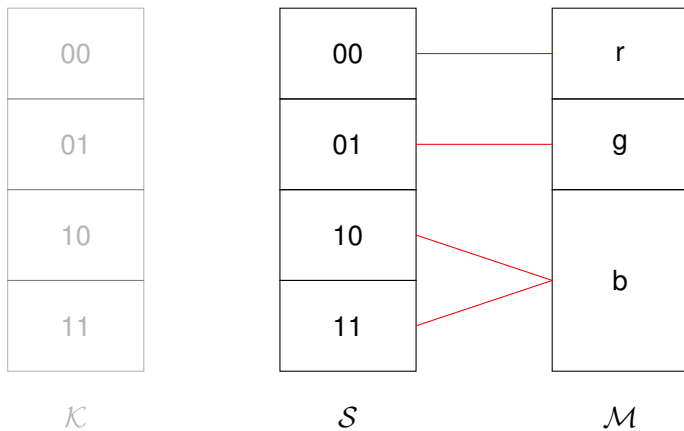
## Entschlüsselung

$11 \leftarrow \text{Ciphertext } C$   
 $\oplus \underline{10} \leftarrow \text{Schlüssel } K$   
 $01 \leftarrow \text{Nachricht } M \text{ (grün)}$

## Brute-Force-Angriff

$11 \leftarrow \text{Ciphertext } C$   
 $\oplus \underline{11} \leftarrow \text{Schlüssel } K'$   
 $00 \leftarrow \text{Nachricht } M' \text{ (rot)}$

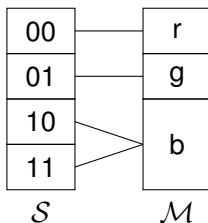
# DTE



# DTE

$$DTE = (encode, decode)$$

- *encode* meist randomisiert
- *decode* deterministisch



# DTE

---

Mögliche DTE-Formen:

- Tabelle/Datenstruktur zum Nachschauen

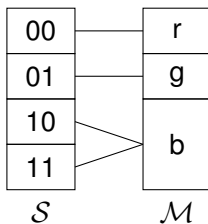
Seed	Nachricht
00	rot
01	grün
10, 11	blau

- Funktion zur Berechnung

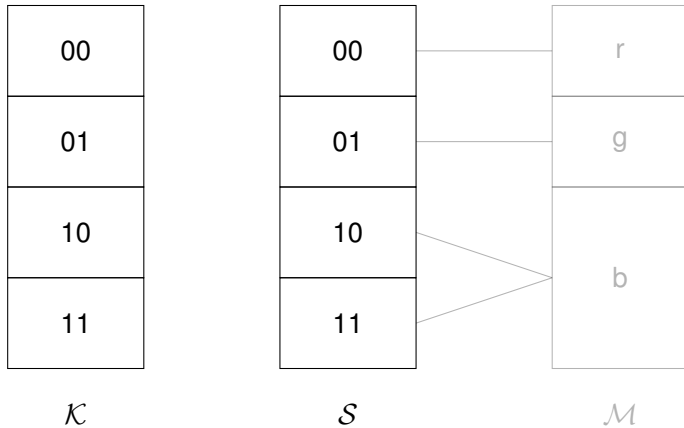
# DTE

Bekannt sein muss:

- Menge/Struktur der Nachrichten
  - endlich speicherbar/berechenbar
  - unendlich berechenbar
- Verteilung der Nachrichten
  - Nachricht wahrscheinlicher  $\Rightarrow$  mehr Seeds



# Verschlüsselung





# Hashbasierte Verschlüsselung

---

## Verschlüsselung

$\text{HEnc}_{\text{Hash}}(M, K)$

$$S \stackrel{\langle r \rangle}{\equiv} \text{DTE}_{\text{encode}}(M)$$

$$R \stackrel{\langle r \rangle}{\equiv} \{0, 1\}^k$$

$$H = \text{HF}(K, R)$$

$$C = H \oplus S$$

Return  $(C, R)$

# Hashbasierte Verschlüsselung

## Verschlüsselung

$\text{HEnc}_{\text{Hash}}(M, K)$

$S \stackrel{\langle r \rangle}{\equiv} \text{DTE}_{\text{encode}}(M)$

$R \stackrel{\langle r \rangle}{\equiv} \{0, 1\}^k$

$H = \text{HF}(K, R)$

$C = H \oplus S$

Return  $(C, R)$

## Entschlüsselung

$\text{HDec}_{\text{Hash}}((C, R), K)$

$H = \text{HF}(K, R)$

$S = H \oplus C$

$M = \text{DTE}_{\text{decode}}(S)$

Return  $M$

## Einschränkungen

---

- Freitext (noch) nicht möglich
  - *Message Space* unendlich groß
  - Verteilung nicht bekannt

## Einschränkungen

---

- Freitext (noch) nicht möglich
  - *Message Space* unendlich groß
  - Verteilung nicht bekannt
- *Typo-Safety*
  - Tippfehler führt zu falschen Daten
  - große Stärke  $\Rightarrow$  große Schwäche

# Einschränkungen

---

- Freitext (noch) nicht möglich
  - *Message Space* unendlich groß
  - Verteilung nicht bekannt
- *Typo-Safety*
  - Tippfehler führt zu falschen Daten
  - große Stärke  $\Rightarrow$  große Schwäche
- Vorab bekannte Informationen
  - Angreifer hat Zusatzinformationen  $\Rightarrow$  Verifizierung des Ergebnisses
  - Sicherheit der Verschlüsselung

## Fazit

---

- Sehr sicher
- Nicht universal anwendbar
- Forschungsgebiete:
  - Natural Language Processing
  - Stochastik
  - User Experience
- Nächstes Ziel: Passwort-Manager