# Homomorphic Encryption

## Ist künstliche Intelligenz gefährlich?

Carine Dengler

# Table of Contents

# Intro

- confidentiality
- authentication
- integrity
- non-repudiation

# Symmetric Encryption

- $(K, P, C, KeyGen, Enc, Dec)$

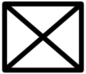
# Symmetric Encryption

- $(K, P, C, KeyGen, Enc, Dec)$
- keyspace
- $k \in K$ key
- $KeyGen$ outputs $k \in K$ s.t. length $k \geq n$

# Symmetric Encryption

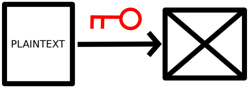

- $(K, P, C, KeyGen, Enc, Dec)$
- plaintext space
- $m \in P$ plaintext

# Symmetric Encryption



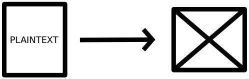
- $(K, P, {\color{blue}C}, KeyGen, Enc, Dec)$
- ciphertext space
- $c \in C$ ciphertext

# Symmetric Encryption



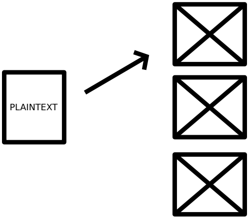

- $(K, P, C, KeyGen, Enc, Dec)$
- encryption algorithm
- $Enc : K \times P \rightarrow C, (k, m) \mapsto c$

# Symmetric Encryption



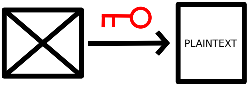

- $(K, P, C, KeyGen, Enc, Dec)$
- encryption algorithm
- $Enc : K \times P \to C, (k, m) \mapsto c$

# Symmetric Encryption




- $(K, P, C, KeyGen, Enc, Dec)$
- encryption algorithm
- $Enc : K \times P \rightarrow C, (k, m) \mapsto c$

# Symmetric Encryption




- $(K, P, C, KeyGen, Enc, Dec)$
- decryption algorithm
- $Dec : K \times C \rightarrow P, (k, c) \mapsto m$

# Symmetric Encryption

- $(K, P, C, KeyGen, Enc, Dec)$
- DES (Data Encryption Standard)
- AES (Advanced Encryption Standard)

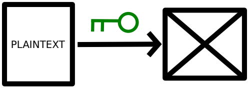
# Asymmetric Encryption

- $(K, P, C, KeyGen, Enc, Dec)$

# Asymmetric Encryption

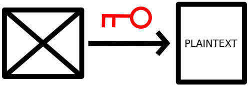
- $(K, P, C, KeyGen, Enc, Dec)$
- $(pk, sk) \in K$ s.t. length $pk \geq n$ and length $sk \geq n$

# Asymmetric Encryption




- $(K, P, C, KeyGen, Enc, Dec)$
- $Enc(pk, m) = c$

# Asymmetric Encryption

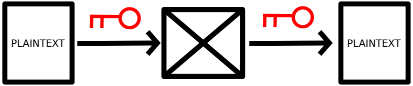



- $(K, P, C, KeyGen, Enc, Dec)$
- $Dec(sk, c) = m$

# Asymmetric Encryption

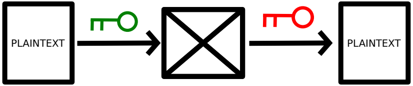
- $(K, P, C, KeyGen, Enc, Dec)$
- RSA
- ElGamal

# Correctness



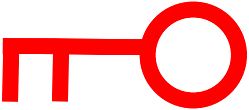

- $Dec(k, Enc(k, m)) = m,\ k \in K, m \in P$

# Correctness



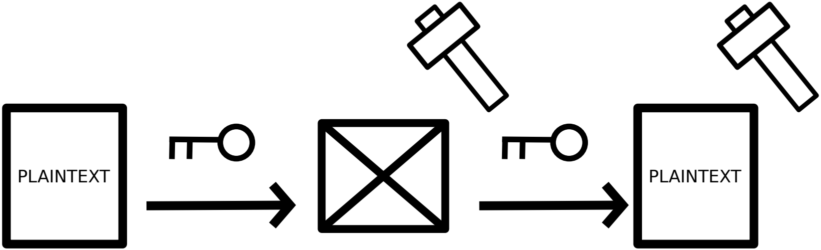

- $Dec(sk, Enc(pk, m)) = m$, $(pk, sk) \in K, m \in P$

# Kerckhoff's Principle



# Table of Contents

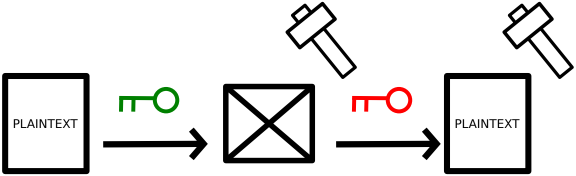
# Intro



# Intro

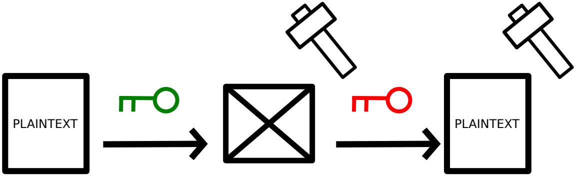
- $(K, P, C, KeyGen, Enc, Dec, Eval, \mathcal{F})$

# Intro




- $\forall f \in \mathcal{F}$ and $c_i$ with $Enc(pk, m_i) = c_i, m_i \in P, i = 1, \ldots, t$ :
  $Eval(pk, f, c_1, ..., c_t) = c_{Eval}$ s.t.
  $Dec(sk, c_{Eval}) = f(m_1, ..., m_t)$

# Intro




- $\forall f \in \mathcal{F}$ and $c_i$ with $Enc(pk, m_i) = c_i, m_i \in P, i = 1, \ldots, t$ : $Eval(pk, f, c_1, ..., c_t) = c_{Eval}$ s.t. $Dec(sk, c_{Eval}) = f(m_1, ..., m_t)$
- ElGamal

# Requirements

- manipulations
- leaked information

# Starting Point

- $f$ can be expressed as a boolean circuit
- evaluate gates for $x, y \in \{0, 1\}$
  - $\texttt{AND}(x, y) = xy$
  - $\texttt{OR}(x, y) = 1 - (1 - x)(1 - y)$
  - $\texttt{NOT}(x, y) = 1 - x$

# Starting Point

- $f$ can be expressed as a boolean circuit
- evaluate gates for $x, y \in \{0, 1\}$
  - $\texttt{AND}(x, y) = xy$
  - $\texttt{OR}(x, y) = 1 - (1 - x)(1 - y)$
  - $\texttt{NOT}(x, y) = 1 - x$

# Somewhat Homomorphic Encryption Scheme (Symmetric)

- $KeyGen(n) = p$ s.t. $p \in \mathbb{Z}$, $p$ odd

# Somewhat Homomorphic Encryption Scheme (Symmetric)

- $KeyGen(n) = p$ s.t. $p \in \mathbb{Z}$, $p$ odd

# Somewhat Homomorphic Encryption Scheme (Symmetric)

- $KeyGen(n) = p$ s.t. $p \in \mathbb{Z}$, $p$ odd
- $Enc(p, m) = m' + pq$, $m \in \{0, 1\}$ with $m'$ $n$-bit length number s.t. $m' \bmod 2 = m \bmod 2$, $q$ random number

# Somewhat Homomorphic Encryption Scheme (Symmetric)

- $KeyGen(n) = p$ s.t. $p \in \mathbb{Z}$, $p$ odd
- $Enc(p, m) = m' + pq$, $m \in \{0, 1\}$ with $m'$ $n$-bit length number s.t. $m' \bmod 2 = m \bmod 2$, $q$ random number

# Somewhat Homomorphic Encryption Scheme (Symmetric)

- $KeyGen(n) = p$ s.t. $p \in \mathbb{Z}$, $p$ odd
- $Enc(p, m) = m' + pq$, $m \in \{0, 1\}$ with $m'$ $n$-bit length number s.t. $m'$ mod $2 = m$ mod 2, $q$ random number
- $Dec(p, c) = (c \bmod p) \bmod 2$
- $c$ mod $p = m'$ noise associated to $c$
- $m' \in (-\frac{p}{2}, \frac{p}{2})$

# Somewhat Homomorphic Encryption Scheme (Symmetric)

- $KeyGen(n) = p$ s.t. $p \in \mathbb{Z}$, $p$ odd
- $Enc(p, m) = m' + pq$, $m \in \{0, 1\}$ with $m'$ $n$-bit length number s.t. $m' \bmod 2 = m \bmod 2$, $q$ random number
- $Dec(p, c) = (c \bmod p) \bmod 2$
- $c \bmod p = m'$ noise associated to $c$
- $m' \in (-\frac{p}{2}, \frac{p}{2})$

# Somewhat Homomorphic Encryption Scheme (Symmetric)

- $KeyGen(n) = p$ s.t. $p \in \mathbb{Z}$, $p$ odd
- $Enc(p, m) = m' + pq$, $m \in \{0, 1\}$ with $m'$ $n$-bit length number s.t. $m'$ mod $2 = m$ mod 2, $q$ random number
- $Dec(p, c) = (c \bmod p) \bmod 2$
- $c$ mod $p = m'$ noise associated to $c$
- $m' \in (-\frac{p}{2}, \frac{p}{2})$

# Operations

- $Add(c_1, c_2) = c_1 + c_2$, $Sub(c_1, c_2) = c_1 - c_2$, $Mult(c_1, c_2) = c_1 c_2$

# Operations

- $Add(c_1, c_2) = c_1 + c_2$, $Sub(c_1, c_2) = c_1 - c_2$, $Mult(c_1, c_2) = c_1 c_2$

# Operations

- $Add(c_1, c_2) = c_1 + c_2$, $Sub(c_1, c_2) = c_1 - c_2$, $Mult(c_1, c_2) = c_1 c_2$

# Operations

- $Add(c_1, c_2) = c_1 + c_2$, $Sub(c_1, c_2) = c_1 - c_2$, $Mult(c_1, c_2) = c_1 c_2$

# Operations

- $Add(c_1, c_2) = c_1 + c_2$, $Sub(c_1, c_2) = c_1 - c_2$, $Mult(c_1, c_2) = c_1 c_2$
- $Mult(c_1, c_2) = m'_1 m'_2 + pq'$ with $m'_i$ noise associated to $c_i$, $i = 1, 2$

# Operations

- $Add(c_1, c_2) = c_1 + c_2$, $Sub(c_1, c_2) = c_1 - c_2$, $Mult(c_1, c_2) = c_1 c_2$
- $Mult(c_1, c_2) = m'_1 m'_2 + pq'$ with $m'_i$ noise associated to $c_i$, $i = 1, 2$
- $f^+(c_1, c_2, ..., c_t) = f^+(m'_1, m'_2, ..., m'_t) + pq$ with $m'_i$ noise associated to $c_i$, $i = 1, 2, ..., t$

# Somewhat Homomorphic Encryption Scheme (Asymmetric)

- $KeyGen(n) = (pk, sk)$
- $sk = p$
- $pk$ list of encryptions of 0

# Somewhat Homomorphic Encryption Scheme (Asymmetric)

- $KeyGen(n) = (pk, sk)$
- $sk = p$
- $pk$ list of encryptions of 0

# Somewhat Homomorphic Encryption Scheme (Asymmetric)

- $KeyGen(n) = (pk, sk)$
- $sk = p$
- $pk$ list of encryptions of 0

# Somewhat Homomorphic Encryption Scheme (Asymmetric)

- $KeyGen(n) = (pk, sk)$
- $sk = p$
- $pk$ list of encryptions of 0
- $Enc(pk, m) = m + \sum_{pk_i \in I} pk_i$ with $I$ random subset

# Noise

- the noise becomes too large ($|f^{+}(m_1', ..., m_t')| > \frac{p}{2}$)

# Noise

- the noise becomes too large ($|f^+(m'_1, \dots, m'_t)| > \frac{p}{2}$)
- decryption removes noise

# Solution

- bootstrappable

# Solution

- circular-secure

# Solution

- $\mathcal{F} = \{Dec, Dec_{\texttt{Add}}, Dec_{\texttt{Sub}}, Dec_{\texttt{Mult}}\}$

# Solution

- $\mathcal{F} = \{Dec, Dec_{\texttt{Add}}, Dec_{\texttt{Sub}}, Dec_{\texttt{Mult}}\}$
- $c_1 = Enc(pk, m)$

# Solution

- $\mathcal{F} = \{Dec, Dec_{\texttt{Add}}, Dec_{\texttt{Sub}}, Dec_{\texttt{Mult}}\}$
- $c_1 = Enc(pk, m)$
- $\bar{sk} = Enc(pk, sk_i)$

# Solution

- $\mathcal{F} = \{Dec, Dec_{\texttt{Add}}, Dec_{\texttt{Sub}}, Dec_{\texttt{Mult}}\}$
- $c_1 = Enc(pk, m)$
- $\bar{sk} = Enc(pk, sk_i)$
- $Recrypt(pk, Dec, \bar{sk}, c_1)$
  - $\bar{c_1} = Enc(pk, c_{1i})$
  - $c = Eval(pk, Dec, \bar{sk}, \bar{c_1})$

# Operations

- $c_1 = Enc(pk, m_1), c_2 = Enc(pk, m_2)$

# Operations

- $c_1 = Enc(pk, m_1), c_2 = Enc(pk, m_2)$
- $\bar{c}_1 = Enc(pk, c_{1i}), \bar{c}_2 = Enc(pk, c_{2i})$

# Operations

- $c_1 = Enc(pk, m_1), c_2 = Enc(pk, m_2)$
- $\bar{c_1} = Enc(pk, c_{1i}), \bar{c_2} = Enc(pk, c_{2i})$
- $c = Eval(pk, Dec_{\texttt{Add}}, \bar{sk}, \bar{c_1}, \bar{c_2})$

# Operations

- $c_1 = Enc(pk, m_1), c_2 = Enc(pk, m_2)$
- $\bar{c}_1 = Enc(pk, c_{1i}), \bar{c}_2 = Enc(pk, c_{2i})$
- $c = Eval(pk, Dec_{\texttt{Add}}, \bar{sk}, \bar{c}_1, \bar{c}_2)$
- $c = Enc(pk, (m_1 + m_2) \bmod 2)$

# Fully Homomorphic Encryption Scheme

- express $f$ as circuit

# Fully Homomorphic Encryption Scheme

- express $f$ as circuit
- arrange its gates into levels

# Fully Homomorphic Encryption Scheme

- express $f$ as circuit
- arrange its gates into levels
- evaluate the levels sequentially

# But . . .

- the noise of $Dec$ is $\gg \frac{p}{2}$
- $Dec(p, c) = (c \bmod p) \bmod 2 = (c - \lfloor \frac{c}{p} \rceil) \bmod 2 \Leftrightarrow \mathtt{LSB}(c)\ \mathtt{XOR}\ \mathtt{LSB}(\lfloor \frac{c}{p} \rceil)$
- $\lfloor \frac{c}{p} \rceil$

# But ...

- the noise of $Dec$ is $\gg \frac{p}{2}$
- $Dec(p, c) = (c \bmod p) \bmod 2 = (c - \lfloor \frac{c}{p} \rceil) \bmod 2 \Leftrightarrow \texttt{LSB}(c) \texttt{ XOR } \texttt{LSB}(\lfloor \frac{c}{p} \rceil)$
- $\lfloor \frac{c}{p} \rceil$

# But ...

- the noise of $Dec$ is $\gg \frac{p}{2}$
- $Dec(p, c) = (c \bmod p) \bmod 2 = (c - \lfloor \frac{c}{p} \rceil) \bmod 2 \Leftrightarrow \texttt{LSB}(c)\ \texttt{XOR}\ \texttt{LSB}(\lfloor \frac{c}{p} \rceil)$
- $\lfloor \frac{c}{p} \rceil$

# But ...

- the noise of $Dec$ is $\gg \frac{p}{2}$
- $Dec(p, c) = (c \bmod p) \bmod 2 = (c - \lfloor \frac{c}{p} \rceil) \bmod 2 \Leftrightarrow$ $\texttt{LSB}(c)\ \texttt{XOR}\ \texttt{LSB}(\lfloor \frac{c}{p} \rceil)$
- $\lfloor \frac{c}{p} \rceil$

# But . . .

- the noise of $Dec$ is $\gg \frac{p}{2}$
- $Dec(p, c) = (c \bmod p) \bmod 2 = (c - \lfloor \frac{c}{p} \rceil) \bmod 2 \Leftrightarrow \mathtt{LSB}(c)\ \mathtt{XOR}\ \mathtt{LSB}(\lfloor \frac{c}{p} \rceil)$
- $\lfloor \frac{c}{p} \rceil$

# Hint

- make the decryption function simpler
- include a hint about the secret integer $p$

# Hint

- *KeyGen* with parameters $\alpha, \beta$
- $(pk, sk)$ with $sk = p$
- $Y = \{y_1, ..., y_\beta\}, y_i \in \mathbb{Q}$ s.t. $\exists J \subset \{1, ..., \beta\}$ with $\sum_{j \in J} y_j = \frac{1}{p}$ mod 2 and $\mid J \mid = \alpha$
- $x \in \{0, 1\}^\beta$ with Hamming weight $\alpha$

# Hint

- *KeyGen* with parameters $\alpha, \beta$
- $(pk, sk)$ with $sk = p$
- $Y = \{y_1, ..., y_\beta\}, y_i \in \mathbb{Q}$ s.t. $\exists J \subset \{1, ..., \beta\}$ with $\sum_{j \in J} y_j = \frac{1}{p}$ mod 2 and $\mid J \mid = \alpha$
- $x \in \{0, 1\}^\beta$ with Hamming weight $\alpha$

# Hint

- *KeyGen* with parameters $\alpha, \beta$
- $(pk, sk)$ with $sk = p$
- $Y = \{y_1, ..., y_\beta\}, y_i \in \mathbb{Q}$ s.t. $\exists J \subset \{1, ..., \beta\}$ with $\sum_{j \in J} y_j = \frac{1}{p} \bmod 2$ and $\mid J \mid = \alpha$
- $x \in \{0,1\}^\beta$ with Hamming weight $\alpha$

# Hint

- *KeyGen* with parameters $\alpha, \beta$
- $(pk, sk)$ with $sk = p$
- $Y = \{y_1, ..., y_\beta\}, y_i \in \mathbb{Q}$ s.t. $\exists J \subset \{1, ..., \beta\}$ with $\sum_{j \in J} y_j = \frac{1}{p} \bmod 2$ and $\mid J \mid = \alpha$
- $x \in \{0, 1\}^\beta$ with Hamming weight $\alpha$

# Hint

- *KeyGen* with parameters $\alpha, \beta$
- $(pk, sk)$ with $sk = p$
- $Y = \{y_1, ..., y_\beta\}, y_i \in \mathbb{Q}$ s.t. $\exists J \subset \{1, ..., \beta\}$ with $\sum_{j \in J} y_j = \frac{1}{p}$ mod 2 and $\mid J \mid = \alpha$
- $x \in \{0, 1\}^\beta$ with Hamming weight $\alpha$

# Hint

- $Encrypt(pk, m) = m$
- $\bar{z}$ s.t. $z_i = cy_i \bmod 2, i \in \{1, ..., \beta\}$

# Hint

- $Dec(x, z, c) = \mathrm{LSB}(c) XOR\ \mathrm{LSB}(\lfloor \sum x_i z_i \rceil)$
- $\sum x_i z_i = \sum c x_i y_i = \frac{c}{p} \bmod 2$

# Hint

- $Dec(x, z, c) = \mathrm{LSB}(c) XOR\ \mathrm{LSB}(\lfloor \sum x_i z_i \rceil)$
- $\sum x_i z_i = \sum c x_i y_i = \frac{c}{p} \bmod 2$

# Hint

- $Dec(x, z, c) = \mathrm{LSB}(c) XOR \ \mathrm{LSB}(\lfloor \sum x_i z_i \rceil)$
- $\sum x_i z_i = \sum c x_i y_i = \frac{c}{p} \bmod 2$

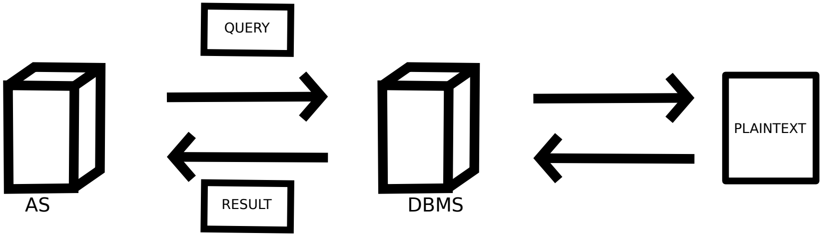
# Hint

- replace multiplication by summation
- if $\alpha$ is small enough, the noise is small enough

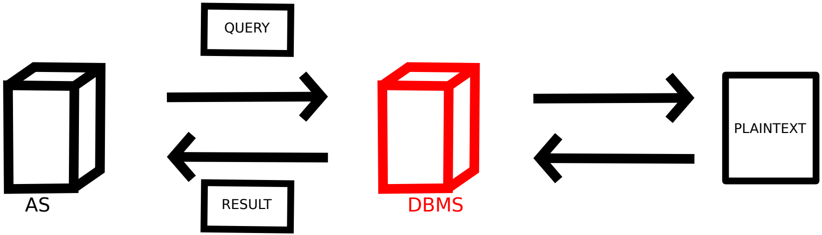
# Table of Contents

Intro

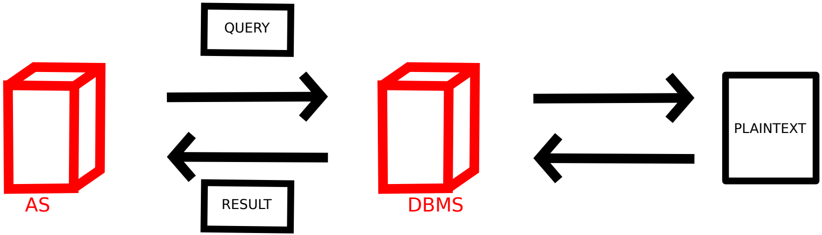
# Database-backed Application



# Threats



# Threats

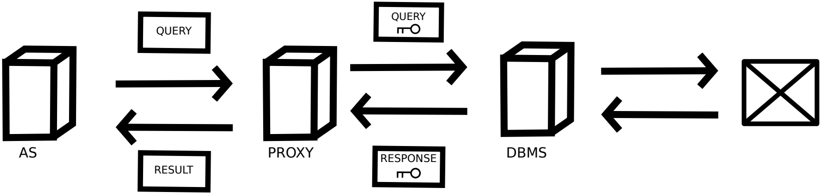


# Key Ideas

- SQL-aware encryption strategy
- adjustable query-based encryption
- chaining encryption keys to user passwords

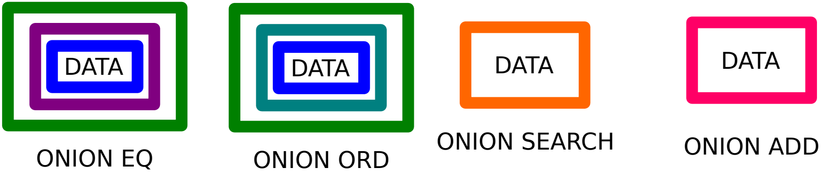
Intro

# Architecture



CryptDB

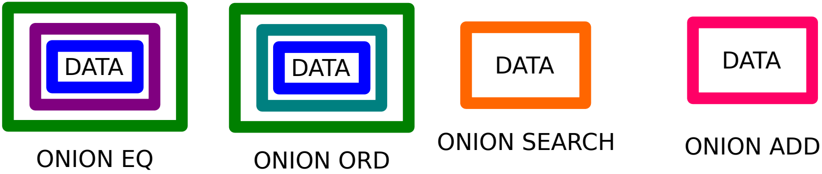
# SQL-aware Encryption



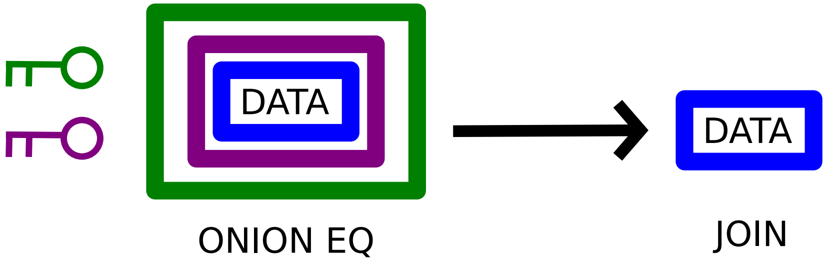

- Random (RND)
- Homomorphic encryption (HOM)
- Word search (SEARCH)
- Deterministic (DET)
- Join (JOIN and OPE-JOIN)
- Order-preserving encryption (OPE)

CryptDB

# Adjustable Query-based Encryption

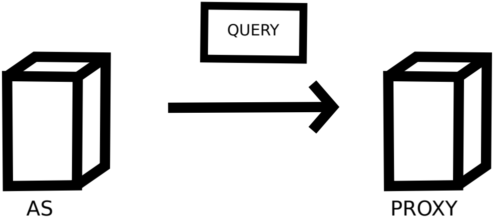


# Adjustable Query-based Encryption



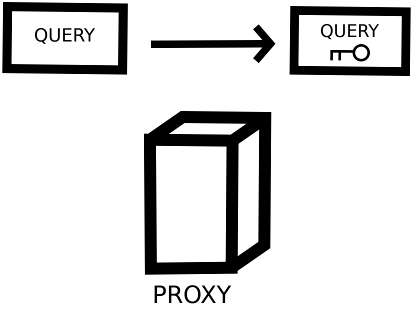
CryptDB

# Query Execution



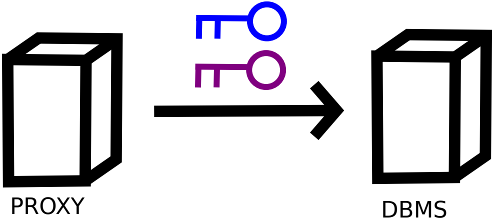
CryptDB

# Query Execution



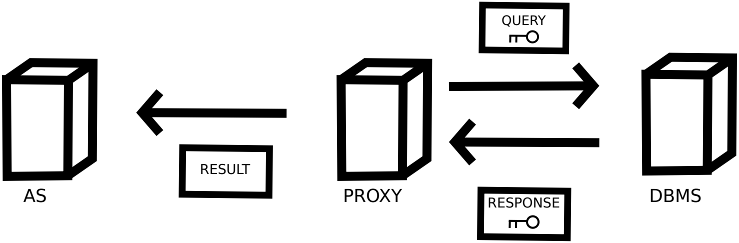
CryptDB

# Query Execution



CryptDB

# Query Execution

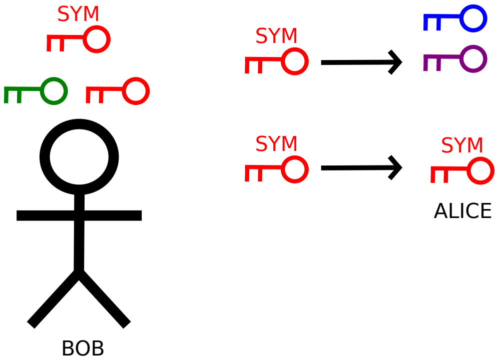


# Access Policy

- principal
- annotations
- delegation rules

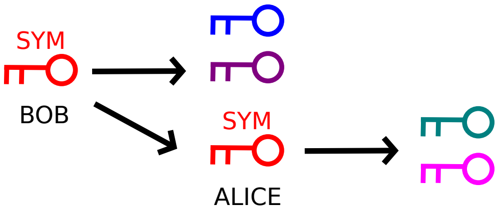
CryptDB

# Key Chaining



CryptDB

# Key Chaining



# Security

- sensitive data
- revealed information
- compromise

CryptDB

# Further Reading I


- Buchmann, J.
  *Einführung in die Kryptographie*
- Katz, J., Lindell, Y.
  *Introduction to Modern Cryptgraphy*
- Beutelspacher A., Schwenk, J., Wolfenstetter, K.-D.
  *Moderne Verfahren der Kryptographie*
- Petrlic, R., Sorge C.
  *Datenschutz. Einführung in technischen Datenschutz, Datenschutzrecht und angewandte Kryptographie*


- Illustrations in section Crypto 101 based on descriptions in the works listed above.

CryptDB

# Further Reading II

Gentry, C.
*Computing Arbitrary Functions of Encrypted Data*
Illustrations in section Fully Homomorphic Encryption based on descriptions in this work.

Popa, R. A., Redfield, C. M. S., Zeldovich, N., Balakrishnan, H.
*CryptDB: Protecting Confidentiality with Encrypted Query Processing*
Illustrations in section CryptDB based on descriptions and illustrations in this work.