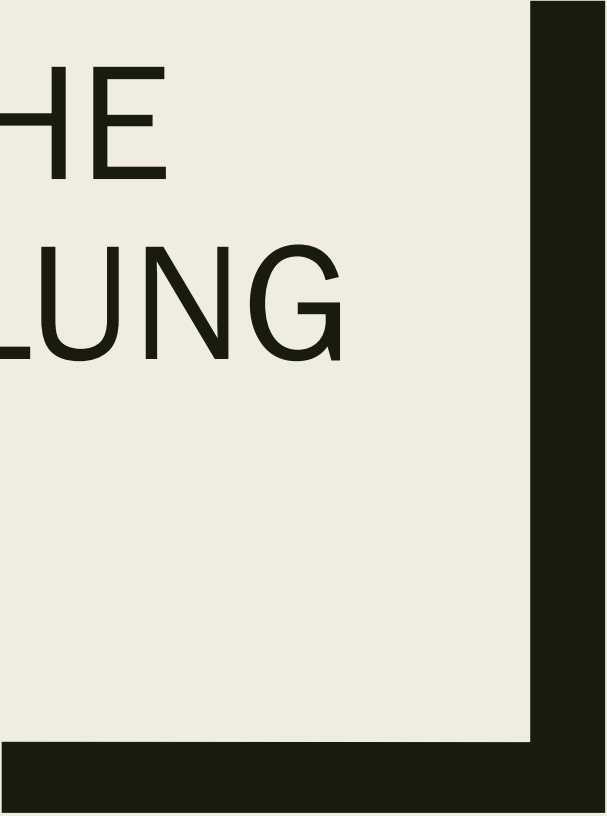




HOMOMORPHE VERSCHLÜSSELUNG

Tanja Kohler
Hannah Köppl
Tobias Mitterreiter
Raimund Petzel



Übersicht

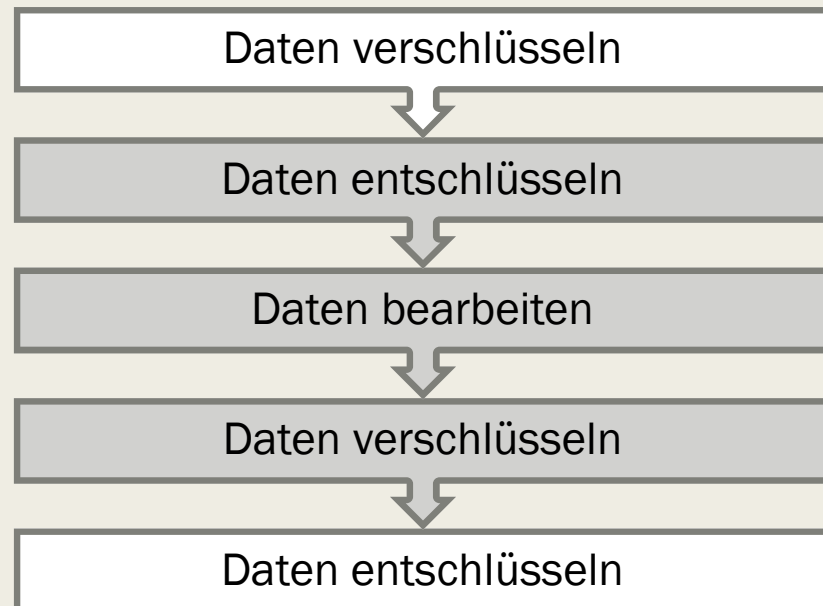
- Motivation & Allgemeines
- Mathematische Definitionen
- Teilhomomorphe Verschlüsselungen
 - *RSA*
 - *Goldwasser-Micali*
 - *Paillier*
- Vollhomomorphe Verschlüsselungen
 - *Hybrid-Homomorphe Verschlüsselung*
- Praktischer Teil



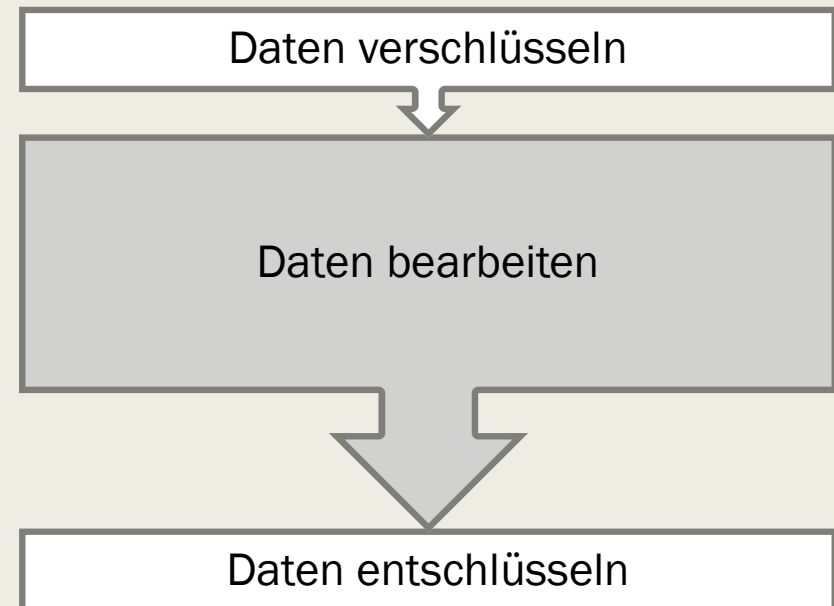
WOZU BRAUCHT MAN
HOMOMORPHE
VERSCHLÜSSELUNG
ÜBERHAUPT?

Motivation

ohne homomorpher
Verschlüsselung

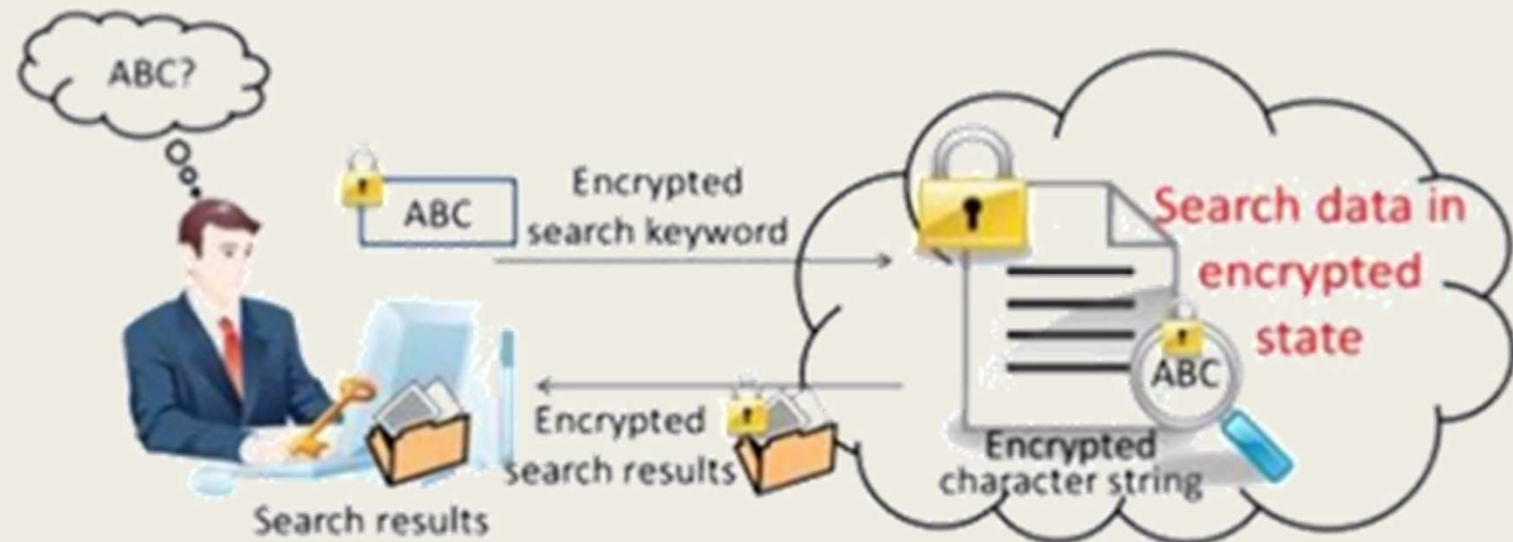


mit homomorpher
Verschlüsselung



Anwendungen

- Cloud Computing
- E-Voting
 - *Unser Praxisteil*



Definition Gruppe

- Eine Gruppe ist ein Paar (G, \circ) . G ist eine Menge und \circ eine zweistellige Verknüpfung $\circ: G \times G \rightarrow G$ und $(a, b) \mapsto a \circ b$.
- mit den folgenden Eigenschaften:
 - Assoziativität $\forall a, b, c \in G : (a \circ b) \circ c = a \circ (b \circ c)$
 - neutrales Element: $\exists e \in G \forall a \in G : a \circ e = e \circ a = a$
 - inverses Element: $\forall a \in G \exists a^{-1} \in G : a \circ a^{-1} = a^{-1} \circ a = e$
- Eine Gruppe heißt abelsch, wenn das Kommutativgesetz gilt:

$$\forall a, b \in G : a \circ b = b \circ a$$

Definition Ring

- Ein Ring $(R, +, *)$ ist eine Menge R mit zwei inneren binären Verknüpfungen $+$ und $*$.
- Dabei muss gelten:
 - $(R, +)$ ist eine abelsche Gruppe.
 - R ist bezüglich $*$ abgeschlossen
 - Bezüglich $*$ gilt die Assoziativität
 - Die Distributivgesetze gelten

Definition Homomorphismus

- Seien (G, \circ) und (F, \diamond) Gruppen, dann heißt die Abbildung $f: G \rightarrow F$ Gruppenhomomorphismus, wenn $\forall a, b \in G$ gilt:

$$f(a \circ b) = f(a) \diamond f(b)$$

- Seien $(R, +_r, *_r)$ und $(S, +_s, *_s)$ Ringe, dann heißt die Abbildung $f: R \rightarrow S$ Ringhomomorphismus, wenn $\forall a, b \in R$ gilt:

$$f(a +_r b) = f(a) +_s f(b)$$

$$f(a *_r b) = f(a) *_s f(b)$$

Historische Entwicklung

- 1978 erster Versuch von Rivest, Adleman und Dertouzes
- 1987 entschlüsselt durch Brickell und Yacobi
- 2009 Gentry – Vollhomomorphe Verschlüsselung ist möglich

Varianten

- additiv homomorph
- multiplikativ homomorph
- voll-homomorph – additiv und multiplikativ homomorph
- hybrid-homomorph

Beispiel additiv-homomorph

Verschlüsselung $E(x) = x * k, k = 3$

Entschlüsselung $D(x) = x/k$

- $2 + 3 + 7 = 12$
- $E(2) + E(3) + E(7) = 6 + 9 + 21 = 36$
- $D(36) = 36/3 = 12$

Beispiel multiplikativ-homomorph

Verschlüsselung $E(x) = x^2$

Entschlüsselung $D(x) = \sqrt{x}$

- $2 * 3 * 7 = 42$
- $(2 * 3 * 7)^2 = 42^2 = 1764$
- $E(2) + E(3) + E(7) = 2^2 * 3^2 * 7^2 = 4 * 9 * 49 = 1764$
- $D(36) = \sqrt{1764} = 42$

TEILHOMOMORPHE VERSCHLÜSSELUNGEN

RSA
Goldwasser-Micali
Paillier

RSA

- Für den Vortrag gehen wir nicht auf die Funktionsweise von RSA ein (siehe VO)
- Das klassische RSA besitzt teilhomomorphe Eigenschaften im Bezug auf die Multiplikation von Chiffraten zur Multiplikation (*mod N*) der Plaintexte.

RSA Teilhomomorphie

- Konkret betrachten wir die Multiplikation von 2 Chiffraten c_1, c_2 , die mit einem gültigen RSA-Schlüssel (e, N) aus den Plaintexten m_1, m_2 generiert wurden, sowie die Entschlüsselung des Produkts mit dem zugehörigen Schlüssel (d, N) .

$$C_1 * C_2 = (m_1^e \pmod N) * (m_2^e \pmod N) = (m_1 m_2)^e \pmod N$$

- Wird dies nun mit dem 2. RSA-Schlüssel (d, N) entschlüsselt erhalten wir die Restklasse der Produkte der Plaintexte.

$$(m_1 m_2^e \pmod N)^d \pmod N = m_1 m_2 \pmod N$$

Padded RSA - OAEP

- Um den Determinismus von RSA, durch den ein Plaintext der mit gleichem Schlüssel verschlüsselt wird, zu umgehen, wird RSA im Regelfall „gepadding“. Hier wird die Nachricht bevor sie mit RSA verschlüsselt wird noch mit einem anderen Verfahren „vorbereitet“.
- Wir betrachten das Optimal Asymmetric Encryption Padding.

Komponenten OAEP

m	Originalnachricht in Bit (auf fixe Länge mit 0er aufgefüllt)
r	Sicherungsblock (fixe Länge, gefüllt mit Zufallszahlen)
G	Kryptographische Hashfunktion von $ r $ auf $ m $
H	Kryptographische Hashfunktion von $ m $ auf $ r $
x	Komponente von m'
y	Komponente von m'
m'	Vorbereitete Nachricht die mit RSA verschlüsselt werden kann. $(x \parallel y)$

Komponenten OAEP

m

Sender bekannt

r

G

Sender & Empfänger bekannt (Optimalfall)

H

x

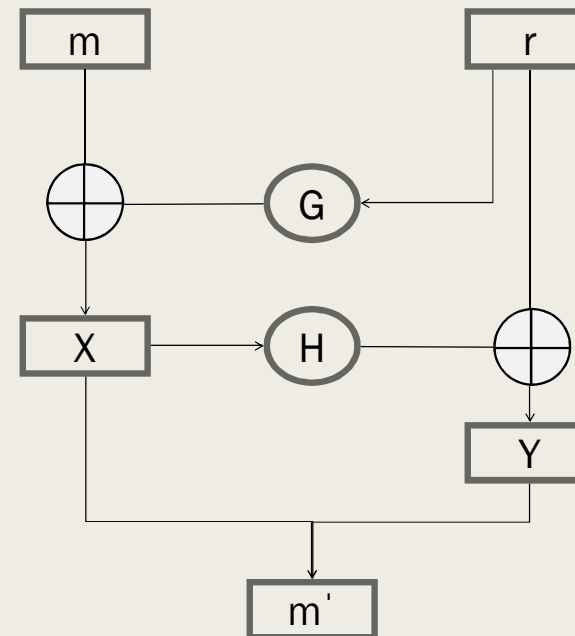
Teil der verschickten Nachricht

y

m'

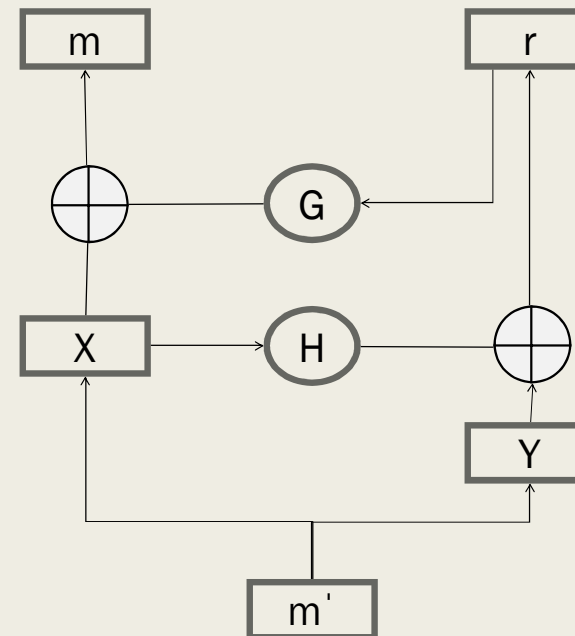
Ablauf OAEP - Verschlüsselung

1. m vorbereiten, r generieren
2. r auf $G(r)$ hashen
3. X aus $m \oplus G(r)$ berechnen
4. X auf $H(X)$ hashen
5. Y aus $r \oplus H(X)$ berechnen
6. X, Y zu m' konkatenieren



Ablauf OAEP - Entschlüsselung

1. X, Y aus m' auslesen
2. X auf $H(X)$ hashen
3. r aus $Y \oplus H(X)$ berechnen
4. r auf $G(r)$ hashen
5. m aus $X \oplus G(r)$ berechnen



RSA-OAEP / RSA

- Zwar ermöglicht RSA-OAEP (Verknüpfung der Algorithmen) die Verschlüsselung von einem Wert auf mehrere Chiffrate, jedoch besitzt es **KEINE** (teil-) homomorphen Eigenschaften.
- Zudem besitzt es noch den Nachteil, dass m' um die Länge des Sicherungsblocks r länger ist, und damit einen längeren Keyspace im RSA Verfahren benötigt.

Goldwasser-Micali

- Der Goldwasser-Micali (fortführend mit GM abgekürzt) Algorithmus ist ein kryptographisches Verfahren um einzelne Bits zu Verschlüsseln.
- Hierbei wird von einem Bit auf eine (deutlich größere) Zahl verschlüsselt.
- Der GM Algorithmus besitzt eine teilhomomorphe Eigenschaft bei der Multiplikation von Chiffraten zur Addition (*mod* 2) von Plaintexten.

GM Quadratischer Rest

- Einer der Kernthemen des GM ist die Bestimmung von ein Wert quadratischer Rest oder quadratischer Nichtrest ist. Demensprechend hier eine Wiederholung:
- Eine Zahl z ist ein quadratischer Rest (modulo eines zu z teilerfremden p) falls es eine Zahl x gibt, sodass gilt:

$$z \equiv_p x^2$$

- Ein quadratischer Nichtrest ist analog eine Zahl z wo keine Zahl x gibt, die dies erfüllt.
- Die Berechnung ist für p Prim einfach, sonst sehr komplex (Sicherheit von GM).

GM Setup/Keygen

- Analog zu RSA generieren wir zwei ausreichend große Primzahlen p, q und berechnen das Produkt N .
- Die Faktorisierung (p, q) bilden hier den privaten Schlüssel.
- Wir wählen ein x das ein quadratischer Nichtrest ($\text{mod } N$) ist.
- Hierbei können wir auf die „Falltür“ des Algorithmus zurückgreifen:
- Für p, q Prim gilt: wenn x sowohl modulo p als auch modulo q quadratischer Rest dann ist x auch modulo $p * q$ ein quadratischer Rest.
- Wird ein passendes x gefunden, bildet (x, N) den öffentlichen Schlüssel.

GM Ver-/Entschlüsselung

- Wollen wir ein Bit m mit dem Public Key (x, N) verschlüsseln, generieren wir eine Zufallszahl r die teilerfremd zu N ist. Das Chiffre c wird dann berechnet mit der folgenden Formel:

$$c = r^2 * x^m \pmod{N}$$

- Bei der Entschlüsselung wird nun berechnet ob c ein quadratischer Rest ($m = 0$) oder ein quadratischer Nichtrest ($m = 1$) ist. Dementsprechend wird der Wert 0 oder 1 zurückgegeben.
- Damit liegt die Sicherheit in der Komplexität der Fakturierung oder des quadratischen Rests.

GM Teilhomomorphie

- Wir betrachten nun wieder die Multiplikation ($\text{mod } N$) zweier Chiffre c_1, c_2 , die mit dem öffentlichen Schlüssel (x, N) erzeugt wurden:

$$c_1 * c_2 = (r_1^2 * x^{m_1}(\text{mod } N)) * (r_2^2 * x^{m_2}(\text{mod } N)) = (r_1 r_2)^2 * x^{m_1 + m_2}(\text{mod } N)$$

- Sollte in diesem Beispiel $m_1 = m_2 = 1$ sein, lässt sich folgendes zeigen.

$$(r_1 r_2)^2 * x^{1+1}(\text{mod } N) = (r_1 r_2 x)^2 * x^0(\text{mod } N)$$

- Anhand dieser Veranschaulichung lässt sich nun leicht nachvollziehen, dass die Multiplikation der Chiffrierte eine Addition Modulo 2 (XOR) der Plaintexte entspricht.

Paillier

- Erfunden von und benannt nach Pascal Paillier im Jahr 1999.
- Ist ein additives homomorphes Verschlüsselungssystem.
- Anwendungen:
 - *E-Voting*
 - *Zero-Knowledge-Beweise*

Paillier KeyGen

- Wähle zwei ausreichend große Primzahlen p und q
 - *ca 1024 Bit, Zahlen mit 309 Ziffern*
- Berechne $n = pq$
- Berechne $\lambda = \text{kgV}(p - 1, q - 1)$
 - *oder vereinfacht: $\lambda = (p - 1)(q - 1)$*
- Wähle g zufällig aus $(\mathbb{Z} / n^2\mathbb{Z})^*$
 - *oder vereinfacht: $g = n + 1$*
- Berechne $\mu\lambda \equiv 1 \text{ mod } n$

PUBLIC KEY

(n, g)

PRIVATE KEY

(λ, μ)

Paillier Verschlüsselung

- Berechnung des Ciphertextes mit dem Public Key und einer Zufallszahl r für die gilt:

$$ggT(r, n) = 1 \text{ und } 0 < r < n$$

- Für eine Nachricht m ergibt sich der Ciphertext c mit der Formel:

$$c = g^m r^n \bmod n^2$$

PUBLIC KEY

(n, g)

Paillier Entschlüsselung

- Berechne die Nachricht m aus dem Ciphertext c

$$m = \frac{(c^\lambda \bmod n^2) - 1}{n} \mu \bmod n$$

PRIVATE KEY

(λ, μ)

Paillier Teilhomomorphie

- Die Verschlüsselung ist additiv homomorph.
- Eine Multiplikation von zwei verschlüsselten Werten entspricht der Addition der unverschlüsselten Werte

$$c_1 * c_2 \equiv_{n^2} (g^{m_1} r_1^n) * (g^{m_2} r_2^n) \equiv_{n^2} g^{m_1+m_2} (r_1 r_2)^n$$

- Da $ggT(r_1 r_2, n) = 1$ ist $c_1 c_2$ ein gültiger Ciphertext von $m_1 + m_2$.

$$c_1 \equiv_{n^2} g^{m_1} r_1^n$$

$$c_2 \equiv_{n^2} g^{m_2} r_2^n$$

Übersicht über Teilhomomorphe Algorithmen

Algorithmus	Homomorphie Eigenschaft
RSA	multiplikativ homomorph
Padded RSA (OAEP)	nicht homomorph
Goldwasser Micali	additiv homomorph
Paillier	additiv homomorph

VOLLHOMOMORPHE VERSCHLÜSSELUNGEN

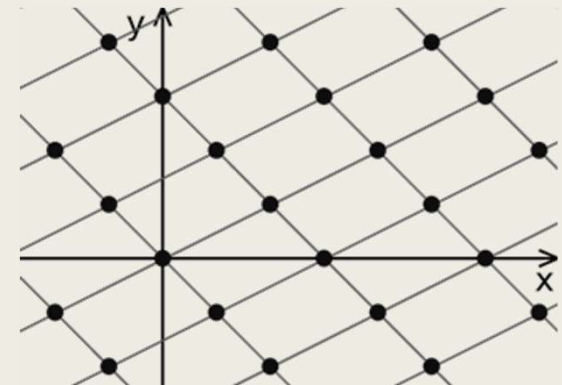
Hybrid Homomorphe Verschlüsselungen



Vollhomomorphismen

- Eine Funktion behält die Ringstruktur von $(R, +, *)$ bei.
- Also $(f(R), +, *)$ ist immer noch ein Ring, wenn f vollhomomorph ist.
- d.h. es sind beliebige Operationen auf dem Ciphertext durchführbar.
- Gentry's Algorithmus (2009) war der erste vollhomomorphe Verschlüsselungsalgorithmus, mit Verwendung von Zahlengittern.

Laufzeitkomplexität: $O(\lambda^{10})$



aktuelle Laufzeiten

Berechnungen in verschiedenen Anwendungsbereichen:

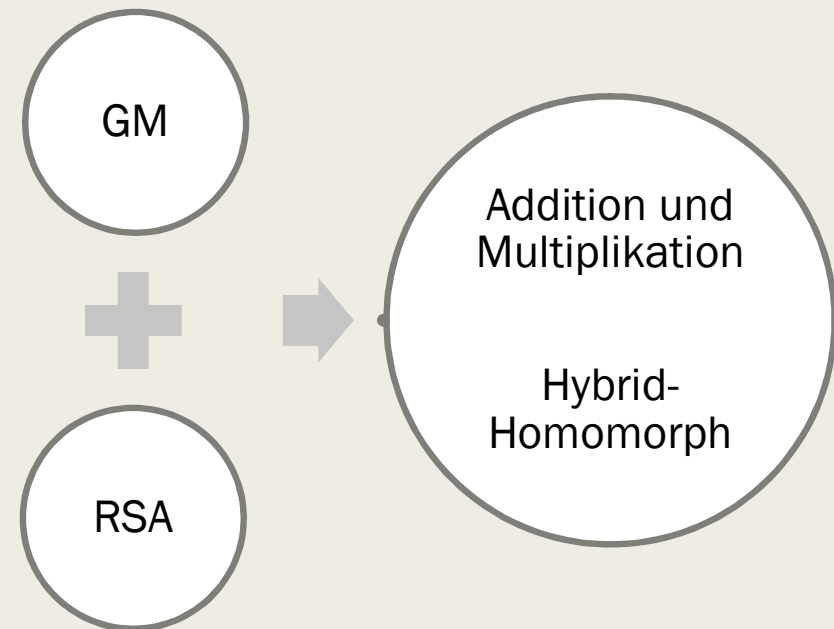
- Profilklassifikation von Energieverbrauch < 1 Sekunde
- Verschiedene medizinische Diagnosen < 2 Minuten
- Gen-basierte Diagnosen < 10 Minuten
- Lauflängenkodierung (bei 48 Kernen) ca 30 Minuten
 - Zur Bild/Videokompression.
- Komplexität basiert auf Sicherheitslevel und Optimierung.

Probleme

- Die Wahl geeigneter Parameter ist schwer
- Bisherige Implementierungen sind nicht wirklich alltagstauglich in der Praxis
- Oft verrauschen die Daten
 - *Bei Gentry muss nach 30 Operationen eine Bereinigung durchgeführt werden*
- Sicherheit
 - *Anfällig auf Chosen-Ciphertext-Attacken*

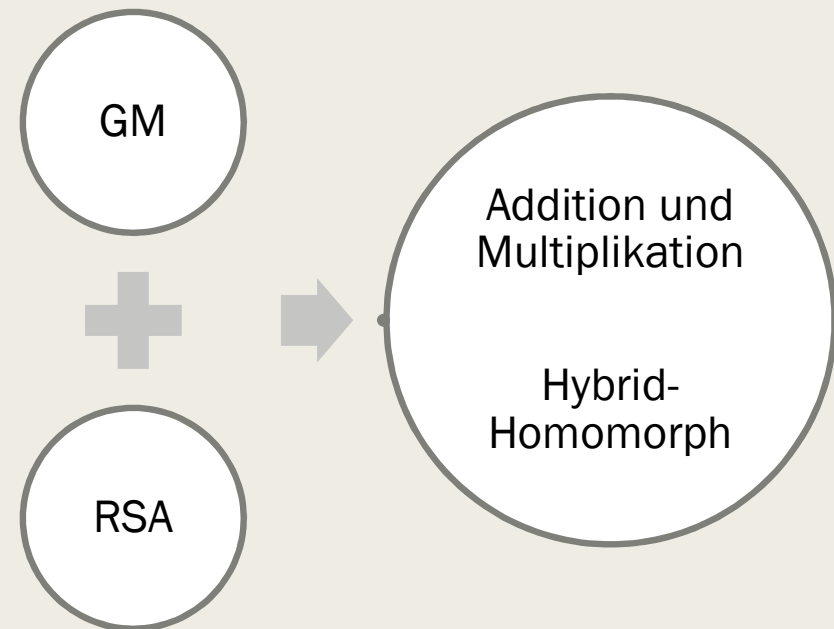
Hybrid-Homomorph

- Der Plaintext wird mittels Goldwasser-Micali und RSA verschlüsselt.
- Es ist also eine vollhomomorphe Verschlüsselung basierend auf zwei Teilhomomorphen Verschlüsselungen.
- Goldwasser-Micali regelt die Addition und RSA die Multiplikation.



Hybrid-Homomorph

1. Erstellen der Keys für G. Micali
2. Verschlüsseln mittels G. Micali
3. Erstellen der Keys für RSA
4. Verschlüsseln mittels RSA
5. Durchführen der Operationen
6. Entschlüsseln mittels RSA
7. Entschlüsseln mittels G. Micali



Vor- und Nachteile

Nachteile

- Sehr hohe Laufzeiten
- Große Rechnerkapazität benötigt
- gültige Rechtsprechung ist abhängig vom Standort der Server

Vorteile

- Gewinn an Datenschutz
- Mobilität
- Outsourcing

ANWENDUNGSGEBIETE

E-Voting, Cloud Computing



Cloud Computing

- Gründe für Verschlüsselung:
- Gewinn an Datenschutz durch Homomorphe Verschlüsselung
- unklar in welchem Land die Daten gespeichert werden
 - *Rechtssprechung des Serverstandortes*

E-Voting

■ Anforderungen:

- *Manipulationssicherheit*
- *Wahlgeheimnis soll erhalten bleiben*

■ Herausforderungen:

- *jeder Internetnutzer kann versuchen in die Wahl einzugreifen*
- *nur Wahlberechtigte dürfen abstimmen*

E-Voting Protokoll

1. Wähler erhält Zugangsdaten zusammen mit Wahlbescheid
2. Wähler schickt auf der Wahlwebsite seine Stimme ab
3. Zusätzliche Identitätsprüfung mittels e-ID

E-Voting

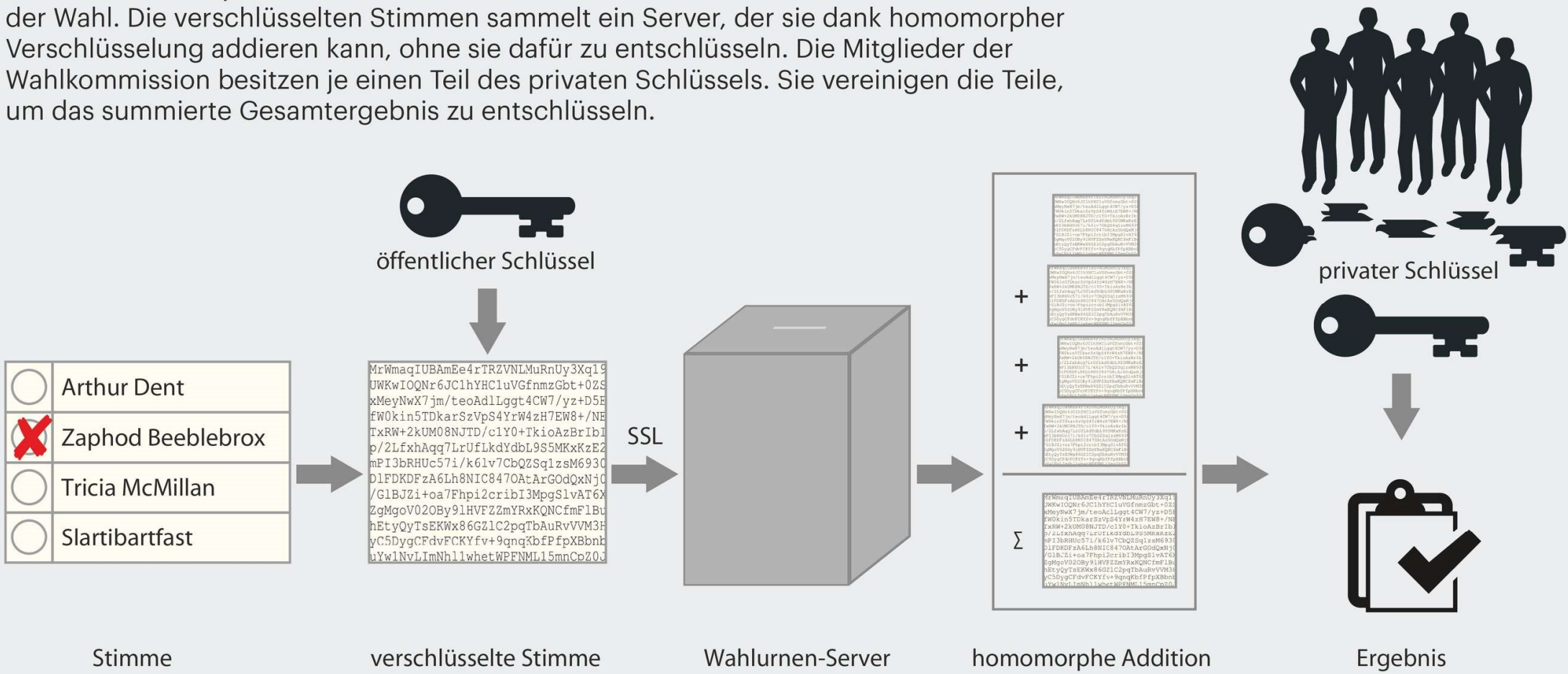
- Angreifer kann sich zwischen Wähler und digitaler Wahlurne schalten
 - *lässt sich mittels asymmetrischer Verschlüsselung lösen*
- “Böser Wahlleiter” - Wahlurnenserver kennt Zusammenhang zwischen Wähler und jeweiligem Geheimtext
 - *Erweiterung des Protokolls mittels Homomorpher Verschlüsselung*

E-Voting

- Stimmen werden im verschlüsselten Zustand addiert
 - *Geheimtext enthält das Gesamtergebnis der Wahl*
- Secret Sharing - privater Schlüssel wird in Teilen erzeugt, wobei nur je ein Teil an eine Partei geht
- zusätzliche Überprüfungsmöglichkeiten für den Wähler
 - *Wähler soll nachvollziehen können, ob die Stimme von der Software korrekt verarbeitet und ins das Ergebnis integriert wurde*

Geheime Wahl dank homomorpher Verschlüsselung

Die Wahlsoftware jedes Wählers verschlüsselt die Stimme mit dem öffentlichen Schlüssel der Wahl. Die verschlüsselten Stimmen sammelt ein Server, der sie dank homomorpher Verschlüsselung addieren kann, ohne sie dafür zu entschlüsseln. Die Mitglieder der Wahlkommission besitzen je einen Teil des privaten Schlüssels. Sie vereinigen die Teile, um das summierte Gesamtergebnis zu entschlüsseln.





PRAKTISCHER TEIL

E-Voting



E-Voting

Simulation einer Europawahl – Verschlüsselung nach Paillier

1. Stimmabgabe (Verschlüsselung)

`e_vote = crypt.encrypt(puk, vote)`

2. Auszählung (Addition der Stimmen)

`c = crypt.sum(puk, c, e_vote)`

3. Auswertung (Entschlüsselung)

`d = crypt.decrypt(prk,puk,c)`


```
tobi@Tobi:~$ python3 demo.py
```

Ergebnis:

```
ÖVP hat 21 Stimmen (plain: 21)
SPÖ hat 13 Stimmen (plain: 13)
FPÖ hat 15 Stimmen (plain: 15)
GRÜNE hat 16 Stimmen (plain: 16)
NEOS hat 15 Stimmen (plain: 15)
EUROPA hat 5 Stimmen (plain: 5)
KPÖ hat 15 Stimmen (plain: 15)
```

Die meisten Stimmen hat ÖVP

```
tobi@Tobi:~$
```

Quellen

- *Homomorphe Verschlüsselung für Cloud-Datenbanken*, H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): *Sicherheit 2018, Lecture Notes in Informatics (LNI)*, Gesellschaft für Informatik, Bonn 2018
- *Dr. Michael Brenner, Rechnen mit sieben Siegeln, Verschlüsselt rechnen mit homomorpher Verschlüsselung*, 6/2016, S. 176
- *Budurushi et al., Pretty Understandable Democracy – A Secure and Understandable Internet Voting Scheme*, *International Conference on Availability, Reliability and security 2013*, S.198
- *Fully Homomorphic Encryption, Implementation Progresses and Challenges* Caroline Fontaine, *FIC 2019, Lille*
- *New Fully Homomorphic Encryption Scheme Based on Multistage Partial Homomorphic Encryption Applied in Cloud Computing*, Z.M. Hikmat, M. I. Khalel, 11/2018, IEEE
- *Performance Analysis of Goldwasser-Micali Cryptosystem*, R. Shruthi, P. Sumana, A.K. Koundinya 7/2017
- *Probabilistic encryption*, S. Goldwasser, S. Micali, 1984

