Universität Bonn
Fakultät für Mathematik und Naturwissenschaften
Institut für Informatik IV


Bachelorarbeit

# Lorem Ipsum... 💭

Matthias Ulbrich
2743974

10. April 2017

Erster Gutachter: Dipl.-Inf. Saffija Kasem-Madani
Zweiter Gutachter: Prof. Dr. Michael Meier

# Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbstständig und nur unter Zuhilfenahme der ausgewiesenen Hilfsmittel angefertigt habe. Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach anderen verfügbaren Werken entnommen sind, habe ich mit Quellenangaben kenntlich gemacht.

Datum: ..... Unterschift: .....

# Danksagung und Widmung

Vielen Dank an...

# Abkürzungen

**DSK** Determinisitisches Symmetrisches Kryptosystem

# 1 Einleitung

Inhaltlicher Leitfaden für die gesamte Bachelorarbeit. Zu fully-, semi- und partial homomorphic encryption je eine umgesetzte Realisierung recherchieren und überlegen wie man sie klassifizieren könnte.

Für jede identifizierte und untersuchen, ob die Malleability des verwendeten Kryptosystems betrachtet wurde und wie damit umgegangen wurde. Bzw. ob integritätssichernde Maßnahmen ergriffen werden.

**Fragen beim Lesen der Paper**

1. Handelt es sich um ein fully-, semi-, oder partial homomorphic Kryptosystem?
2. Warum haben sich die Wissenschaftler für das jeweilige Kryptosystem entschieden?
3. Wurde malleability des Kryptosystems beachtet?
4. Wurden integritätssichernde Maßnahmen ergriffen? (vgl. „Integritätsprüfung homomorpher Operationen“)

# 2 Homomorphe Kryptosysteme

## 2.1 Schutzziele der Kryptografie

Die Kryptografie möchte mehrere Schutzziele für die Speicherung, Vervielfältigung und Übertragung von Informationen umsetzen und dazu verschiedene Verfahren bereitstellen. Grundlegende Schutzziele beim Übertragen von Informationen zwischen mehreren Personen mit Nachrichten sind dann [Mei16][DKK02, p.2]:

1. **Vertraulichkeit:** Keine unauthorisierte Kenntnisnahme. Nur dazu berechtigte Personen sollen eine Information lesen können oder Zugang zur einer Information bekommen.

2. **Integrität:** Keine unauthorisierte unbemerkte Datenmanipulation. Dies schützt insbesondere vor dem Hinterlegen von Falschdaten in einer Nachricht oder einer oder dem Fehlen von Teilen einer Nachricht.

3. **Authentizität:** Der Empfänger kann den Verfasser einer Nachricht verifizieren.

4. **Nichtabstreitbarkeit:** Der Sender kann dem Versand nicht mehr abstreiten der Verfasser einer Nachricht gewesen zu sein.

## 2.2 Kryptosysteme

Ein Kryptosystem ist ein Sammlung von Algorithmen um das Schutzziel der Vertraulichkeit beim Übertragen von Informationen in Nachrichten umzusetzen.

Damit ermöglicht ein Kryptosystem zwei Parteien Alice und Bob über einen ungeschützten Kanal in dem die Nachricht übertragen wird zu kommunizieren, ohne das eine dritte Partei welche mithört Zugang zu dem Inhalt der Nachricht bekommt.

Die zugrundelegenden Algorithmen und resultierende Eigenschaften über die Beziehung von Klartexten zu Chiffretexten führen zu ein verschiedenen Klassen von Kryptosystemen. Diese Kryptosysteme führen wir in diesem Abschnitt ein. In der Literatur ist es üblich bei „einfacheren“ Kryptosystem die deterministisch oder symmetrisch sind, diese Bezeichnungen wegzulassen. Zur besseren Abgrenzung werden in diesem Abschnitt Kryptosysteme immer mit ihren Eigenschaften genannt (z.B. deterministisches symmetrisches Kryptosystem).

Formal definieren wir [Sti06, p.1]:

**Definition 2.1** (Kryptosytem)**.** *Ein Kryptosystem ist ein Quintupel* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *welches folgenden Eigenschaften genügt:*

1. $\mathcal{P}$ *ist eine endliche Menge von Klartexten, der Klartextraum.*

2. $\mathcal{C}$ *ist eine endliche Menge von Chiffretexten, der Chiffreraum.*

3. $\mathcal{K}$ *ist eine endliche Menge möglicher Schlüssel, der Schlüsselraum.*

4. *Für alle Schlüssel $k \in \mathcal{K}$ gibt es eine Verschlüsselungsfunktion $\mathcal{E} \ni e_k : \mathcal{P} \to \mathcal{C}$ und zugehörige Entschlüsselungsfunktion $\mathcal{D} \ni d_k : \mathcal{C} \to \mathcal{D}$, so dass für alle Klartexte $x \in \mathcal{P}$ folgende Identität gilt: $d_k(e_k(x)) = x$*

Grundsätzlich gibt es also drei Algorithmen in einem Kryptosystem: Einen für die Erzeugung des Schlüssels, einen für die Verschlüsselung und einen für die Entschlüsselung [Cry].

**Definition 2.2** (Symmetrisches Kryptosystem)**.** *Sei $K = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ ein Kryptosystem. Dann nennen wir K symmetrisch, wenn sowohl die Verschlüsselungsfunktion $e_k$ als auch die Entschlüsselungsfunktion $d_k$ vollständig von demselben Schlüssel $k \in \mathcal{K}$ abhängen. Vollständig bedeutet, dass diese Funktionen insbesondere nicht nur von einen Teilschlüssel von k anhängen, wenn der Schlüssel k aus mehreren Parametern zusammengesetzt ist. Letzteres ist in 2.1 nicht vorrausgesetzt.*

Ein Nachteil von Symmetrischen Kryptosystemen liegt auf der Hand: Da sowohl Alice als auch Bob den gleichen geheimen Schlüssel benötigen, muss dieser über einen sicheren Kanal übertragen werden. Daher sind symmetrische Kryptosysteme auch bekannt als private-key Kryptosysteme.

Im Gegensatz dazu gibt es Kryptosysteme bei denen sich $K$ aus einem privaten und öffentlichen Teilschlüssel zusammensetzt von Alice den öffentlichen Teilschlüssel bekannt geben kann um dritten zu ermöglichen ihr Informationen vertraulich zukommen zu lassen. Öffentlicher und privater Teilschlüssel stehen in einem Zusammenhang, der jedoch für Angreifer mit begrenzter Rechenkapazität nicht möglich ist zu erschließen.

**Definition 2.3** (Asymmetrisches Kryptosystem)**.** *Sei $K = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ ein Kryptosystem. Dann nennen wir K asymmetrisch wenn sich der Schlüssel $k \in \mathcal{K}$ zusammensetzt aus $k = (k_s, k_p)$. Die Verschlüsselungsfunktion ist dann $\mathcal{E} \ni e_{k_p} : \mathcal{P} \to \mathcal{C}$, während die Entschlüsselungsfunktion $\mathcal{D} \ni d_{k_s} : \mathcal{P} \to \mathcal{C}$ ist. Während $e_k$ von beliebigen Parteien ausgeführt werden kann, kann $d_k$ nur vom Besitzer des privaten Teilschlüssel $k_s$ ausgeführt werden. $k_s$ muss geheim gehalten werden.*

Diese drei Definitionen genügen noch nicht um zu beschreiben, in welcher Beziehung Klartexte $x$ zu ihren Chiffraten $c$ stehen wenn sie mit dem gleichen Schlüssel $k$ in verschiedenen Ausführungen von $e_k$ erzeugt werden. Dies ist von Bedeutung für mögliche Angriffe der Kryptoanalyse welche in 3 vorgestellt werden.

**Definition 2.4** (Deterministisches Kryptosytem)**.** [1] *Sei $K = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ ein Kryptosystem. Dann nennen wir K deterministisch wenn gilt: Für einen beliebigen festen Schlüssel $k \in \mathcal{K}$ ist $e_k$ ist injektiv.*

Seien nun $c_1, c_2 \in \mathcal{C}, c_1 = c_2$ zwei Chiffrate unter $e_k$, dann folgt daraus für ihre Klartexte, dass $x_1 = x_2$. Also führt der gleiche Klartext unter Verwendung desselben Schlüssel bei verschiedenen Ausführungen von der Veschlüsselungsfunktion $d_k$ zu einem identischen Chiffrat!

[1] Diese Definition ist eine Formalisierung von saloppen Beschreibungen in der Literatur.

Jetzt können wir in Abgrenzung zu dieser Definition das Proballistische Kryptosystem einführen. Ein Proballistisches Kryptosystem erzeugt für gleiche Klartexte bei demselben Schlüssel mit jeder Ausführung der Verschlüsselungsfunktion ein anderes Chiffrat.

Das Konzept eines Proballistischen Kryptosystems wurde ursprünglich von Goldwasser und Micali eingeführt in [GM84]. Wir definieren in Anlehnung an [DKK02, p.345]:

**Definition 2.5** (Proballistisches Kryptosytem)**.** *Ein Proballistisches Kryptosystem ist ein Sextupel* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{R})$*. Wie schon in 2.1 ist* $\mathcal{P}$ *ist der Klartextraum,* $\mathcal{C}$ *der Chiffreraum und* $\mathcal{K}$ *der Schlüsselraum. Neu sind:*

- $\mathcal{R}$ *ist eine endliche Menge von Zufallszahlen*
- *Für alle Schlüssel* $k \in \mathcal{K}$ *gibt es eine Verschlüsselungsfunktion* $\mathcal{E} \ni e_k : \mathcal{P} \times \mathcal{R} \to \mathcal{C}$ *und zugehörige Entschlüsselungsfunktion* $\mathcal{D} \ni d_k : \mathcal{C} \times \mathcal{R} \to \mathcal{D}$*, so dass für alle Klartexte* $x \in \mathcal{P}$ *und alle Zufallszahlen* $r \in \mathcal{R}$ *folgende Identität gilt:* $d_k(e_k(x, r)) = x$
- *Für eine Zufallszahl* $r \in \mathcal{R}$ *und verschiedene Klartexte* $x_1, x_2 \in \mathcal{P}, x_1 \neq x_2$ *gilt:* $e_k(x_1, r) \neq e_k(x_2, r)$*.*

Ein Proballistisches Kryptosystem benutzt also Zufall in der Verschlüsselungsfunktion, so dass der gleiche Klartext verschieden verschlüsselt wird. Mit Proballistischen Kryptosystemen werden meistens Asymmetrische Verschlüsselungsverfahren gemeint, es ist jedoch auch möglich mit Symmetrischen Verschlüsselungsverfahren diese Eigenschaft zu erreichen, z.B. bei Verwendung von Blockchiffren im Cipher Block Chaining Mode. Die Menge der Zufallszahlen $\mathcal{R}$ entspricht dann der Menge möglicher Initialisierungsvektoren [Mei16].

### 2.2.1 Homomorphes Kryptosystem

## 2.3 Mathematische Grundlagen

Wir werden später Homomorphe Kryptosystem im Detail einführen. Um ihre Anwendung zu verstehen, ist es jedoch nötig folgende Algebraische Strukturen nach [Rä15] einzuführen um ein Verständnis dafür zu schaffen wie Homomorphe Kryptosysteme verwendet werden könnnen.

**Definition 2.6** (Gruppe)**.** *Eine Gruppe ist ein Tupel* $(G, +)$ *bestehend aus der Menge* $G$ *und einer Verknüpfung* $+$ *auf* $G$ *mit folgenden Eigenschaften:*

- $+$ *ist assoziativ*
- *Es existiert bzgl.* $+$ *ein neutrales Element* $e$ *in* $G$*.*
- *Jedes* $g$ *in* $G$ *ist invertierbar.*

*Ist die Verknüpfung einer Gruppe zusätzlich kommutativ, so nennt man sie abelsch.*

**Definition 2.7** (Ring)**.** *Ein Ring ist ein Tripel $(R, +, \cdot)$ bestehend aus der Menge $R$ und zwei Verknüpfungen $+$ und $\cdot$ auf $R$ mit folgenden Eigenschaften:*

- *$R$ ist bzgl. $+$ eine abelsche Gruppe.*
- *$\cdot$ ist assoziativ*
- *Es gelten die Distributivgesetze: $\forall a, b, c \in R : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ und $(a + b) \cdot c = (a \cdot c) + (b \cdot c)$*

Hat $R$ bzgl. $\cdot$ ein neutrales Element, so nennen wir $R$ einen „Ring mit Rins“. Ist $R$ bzgl. $\cdot$ kommutativ, so nennen wir $R$ einen „kommutativen Ring“.

**Definition 2.8** (Körper)**.** *Sei $K$ ein kommutativer Ring mit Eins wie in 2.7, so heißt $K$ Körper, wenn die neutralen Elemente bzgl. der Verknüpfungen $+$ und $\cdot$ verschieden sind und alle Elemente bzgl. $\cdot$ invertierbar sind.*

Um den Homomorphismus Einführen zu können benötigen wir eine noch mächtigere algebraische Struktur.

**Definition 2.9** (K-Vektorraum)**.** *Sei $(K, +, \cdot)$ ein Körper und $V$ eine Menge. Zusätzlich existieren zwei Verknüpfungen $\oplus : V \times V \to V$ und $\oplus : K \times V \to V$, so dass gilt:*[2]

- *$\oplus$ ist assoziativ, kommutativ, hat ein neutrales Element bzgl. $V$ und für alle Elemente $v \in V$ Inverse in $V$*
- *$\otimes$ ist assoziativ, hat ein neutrales Element bzgl. $V$*
- *Elemente aus $K$ und $V$ sind distributiv*

*Dann nennen wir $V$ einen $K$-Vektorraum.*

**Definition 2.10** (Homomorphismus)**.** *Seien $V, W$ zwei $K$-Vektorräume und $f : V \to W$ eine Abbildung. Dann nennen wir $f$ einen Homomorphismus (oder linear) von $V$ nach $W$ wenn gilt:*

- $\forall x, y \in V : f(x) + f(y) = f(x + y)$
- $\forall x \in V, \lambda \in K : f(\lambda \cdot K) = \lambda \cdot f(x)$

**Theorem 2.1** (Jede boolsche Funktion ist mit NANDs konstruierbar)**.** ...

## 2.4 Semihomomorphes Kryptosystem

Ein Semihomomorphes Kryptosystem erlaubt die Ausführung von Rechenoperationen im Chiffreraum ohne vorherige Verschlüsselung (!). (z.B. ElGamal für Multiplikation, Pailler für Addition). ElGamal stellt einen Ring mit Eins zur Verfügung[3], wobei der Chiffreraum dem Körper entspricht.

[2] Evtl. ausführlicher formulieren

[3] Laut, https://security.stackexchange.com/a/98190. Es gilt hierfür noch mehr Recherche zu betreiben, z.B. zu nichtkommutativer Kryptographie

### 2.4.1 Pailler Kryptosystem

## 2.5 Voll Homomorphes Kryptosystem

Allows arbitrary computation on ciphertexts. (example: Gentry). It provides a ring structure (R,+,*) which allows to construct NAND gates. Since it is possible to create any boolean computation with NAND gates, a fully homomorphic cryptosystem allows to compute any arbitrary computation.

# 3 Sicherheitskriterien

## 3.1 Malleability

Malleability beschreibt die Möglichkeit, dass ein Angreifer einen Chiffretext $c$ von Klartext $k$ geziehlt verformen kann um einen daraus abgeleiteten Chiffretext $f(c) = c'$ zu erzeugen welcher in einer ihm bekannten Beziehung $f$ zu $c$ steht. Existiert nun zwischen den Klartexten $k$ und $k'$ eine Beziehung die der Angreifer umkehren kann, kann er zu $k'$ den ursprünglichen Klartext $k$ bestimmten. [Sma03, p. 292]

**Eigenschaften**: Ein malleable Kryptosystem ist angreifbar mit chosen ciphertext Angriffen. (CCA2)

**Kommentar**: Das zwischen den Chiffretexten und den Klartexten eine ähnliche Beziehung steht die der Angreifer kennt ist eine Eigenschaft die genau Isomorphismen ermöglichen! Daher sind homomorphe Kryptosysteme per Design anfällig für malleability.

## 3.2 Privacy-Preserving

## 3.3 Sicherheitsklassen

### 3.3.1 Ununterscheidbarkeit von Geheimtexten (Ciphertext Indistinguishability)

### 3.3.2 Semantische Sicherheit

Ein deterministisches Kryptosytem wie in **??** kann nie semantisch sicher sein!

# 4 Klassifikation Homomorpher Kryptosysteme

## 4.1 Aufteilungen

## 4.2 Autocrypt

[TSCS13] Problemstellung: Server sind ständig durch Angriffe bedroht die bis hin zu ihrer kompletten Übernahme geraten können. Um Datendiebstahl und Vertraulichkeitsverletzungen vorzubeugen ist es ratsam nur mit verschlüsselten Datenbeständen zu arbeiten. Ein IT-System oder Programm so anzupassen, dass

es auf verschlüsselten Daten korrekt arbeitet wollen die Wissenschaftler automatisieren indem sie die Arbeit der Programmtransformation mit einem Compiler abwickeln: Autocrypt.

Der Server läuft als virtuelle Maschine und Inhalte werden außerhalb der unvertrauten VM auf einem keyserver verschlüsselt. Autocrypt bestimmt automatisch benötigte Verschlüsselungsdatentypen für die Variablen und konvertiert zwischen diesen im Programmablauf her durch einfügen von hypercalls. Die Verschlüsselungsdatentypen werden gewählt nach der Verknüpfung die sie zu Verfügung stellen. Wenn also im Ursprungscode Additionen von zwei Variablen durchgeführt werden, dann werden zunächst Paillerverschlüsselungsdatentypen erzeugt. Wird das Ergebnis allerdings später multipliziert, dann muss der Datentyp konvertiert werden zu einem Elgamalverschlüsselungsdatentyp.

Bei der Entwicklung von Autocrypt sollen alle Rechenoperationen privacy-preserving sein. Als Transformationstool ist eine Integrität der Daten auf denen gerechnet wird daher nicht berücksichtigt worden.

**Kategorisierungskriterien**: Pailler wurde wegen seiner Homomorphie und Additionsverknüpfung verwendet (→ additiv-homomorph). Analoges Argument für Elgamal. Zwischen diesen beiden Verfahren wird hin und her konvertiert, da dies schneller ist als *ein* vollhomomorphes Verfahren (→ Klasse schneller Verfahren). Weiter ist Pailler flexibel einsetzbar für die Addition von Zahlen byteweiese oder bitweise. Letzteres ermöglicht die Konstrution eines homomorphen XOR Operators. (→ homomorph XOR)

**Malleability**: Die Autotoren haben als Zielsetzung die unerlaubte Kenntnisnahme von Daten auf dem Server zu unterbinden. Eine Überprüfung der Integrität von Rechenoperationen der von Autocrypt konvertierten Programmbestandteile ist daher kein Fokus der Arbeit [p. 4].

## 4.3 Machine Learning Classification over encrypted data

[BPTG15] Problemstellung: Es soll ein privacy-preserving Maschinenlernenverfahren erstellt werden, bei dem sowohl die zu klassifizierenden Daten als auch die Klassifiziererdesigns vertraulich bleiben. Es wird eine Bibliothek konstruiert, aus der Modular beliebige privacy-preserving Klassifizierer erstellt werden können.

In einem ersten Ansatz wurde überlegt privacy-preserving mit Secure Multiparty Computation umzusetzen, welches sich jedoch als zu langsam herrausgestellt hat. Aus dem gleichen Grund wird auch auf den Einsatz von vollhomomorpher Verschlüsselung verzichtet. Es ist schneller mit für Klassifizierungsverfahren spezialisierten Protokollen zu arbeiten.

Es wird wie in [TSCS13] XOR mit Pailler simuliert. Zusätzlich wird ein privates Skalarprodukt auf Basis von Pailler berechnet. Gegeben seien die Vektoren $x = (x_1, \ldots, x_n)$ und $y = (y_1, \ldots, y_n)$ wobei alle Einträge Klartexte sind. Das mit pub Paillierverschlüsselte Skalarprodukt ist dann:

$$Enc_{pub}(\langle x, y \rangle) = \prod_i Enc_{pub}(y_i)^{x_i} \bmod N^2 \tag{1}$$

Eine weitere Tatsache die im Paillerkryptosystem ausgenutzt wird ist der Klartextraum ungefähr $2^{1024}$ bit ist. Anstelle von lediglich Integern können Floatzahlen mit Pailler verschlüsselt werden, wenn man die IEEE 754 floating point Darstellung verwendet welche große Exponenten benötigt.

In dieser Arbeit wurde auch eine leveled vollhomomorphe Verschlüsselung (HElib) verwendet, jedoch der Umfang und die Gründe dafür bleiben ohne nähere Erläuterung [p. 4].

**Kategorisierungskriterien**: Es wurden die Kryptosysteme von Paillier und Goldwasser–Micali verwendet. Beide Aufgrund ihrer schnelleren Performance und der mathematischen Verknüpfung sie anbieten. (→ additiv-homomorph) (→ xor-homomorph). Analog zur unverschlüsselten Konstruktion von Gleitkommazahlen aus ganzen Zahlen kann mit Pailler ein Operator für die homomorphe Addition von Gleitkommazahlen konstruiert werden. (→ floatingpoint-additiv-homomorph)

**Malleability**: Es werden die homomorphen Kryptosysteme ledglich zum Rechnen im Chiffreraum verwendet. Ein Angriff der Malleability ausnutzt wird nicht betrachtet. Dies ist nachvollziehbar, da hier ein deterministischer Algorithmus abgearbeitet wird.

## 4.4 Privacy Preserving Matrix Factorization

[NIW+13] Bei der Generierung von userspezifischen Empfehlungen anhand vorheriger Wahlen eines Users ist Matrizenfaktorisierung ein weit verbreitetes Verfahren. Um dieses privacy-preserving zu machen soll ein System designt werden, welches Empfehlung geben kann ohne die Userbewertungen zu lernen.

Bei dem Design wird aus Performancegründen hash-ElGamal verwendet um verschlüsselte Wertungen zu maskieren für die Einheit, welche im Besitz des privaten Schlüssels ist. In dem Design bekommt das Recommendersystem (RecSys) vom User ein mit dem öffentlichen Schlüssel von Cryptoserviceprovider (CSP) verschlüsseltes Rating $c$. Damit der CSP dies Rating nicht aufdecken kann, addiert RecSys einen zufälligen Wert $\mu$ auf das Rating. CSP erhält $c' = c + \mu$.

**Kategorisierungskriterien**: Es wurde hash-ElGamal verwendet wegen seiner schnelleren Performance gegenüber Paillier und seiner Additivität (→ additiv-homomorph)

**Malleability**: Bei dem Design wird von einem honest-but-curious Angreifer ausgegangen. Also könnte RecSys aus Neugierde $\mu = 0$ addieren und so CSP ermöglichen alle Userratings zu lernen. Im HBC-Modell dürfen RecSys und CSP jedoch nicht vom Protokoll abweichen und könnten daher nicht kooperativ diese Information abschöpfen, denn CSP weiß nicht, dass RecSys eine Nulladdition durchführt welches die Maskierung aufhebt.

## 4.5 Fingerprinting Protocol for Images Based on Additive Homomorphic Property

[KT05]

## 4.6 Privacy-Preserving Face Recognition

[EFG+09]

## 4.7 Efficient and Secure Comparison for On-Line Auctions

Ivan Damgard et al. stellen in [DGK07] ein neues additives Kryptosystem vor um schnelle vergleiche einer öffentlich bekannten Zahl $x$ und einer geheimen Zahl $m$ durchzuführen die auf einem Server und einem Hilfsserver verteilt ist. Bei der on-line Versteigerung steht $x$ für das momentane Höchstgebot, während $m$ für das private mögliche Höchstgebot steht. Sie haben dieses Kryptosystem für ihren Anwendungsfall designt, da sie einen möglichst kleinen Klartextraum haben wollen. Die Verwendung eines kleinen Klartextraums hat zum Vorteil, dass mit kleineren Exponenten gerechnet wodurch ihr Verfahren an Effizienz im Vergleich gegenüber anderen Ansätzen gewonnen hat.

In ihrem Vergleich $m \leq x$ ist letztere Zahl öffentlich, jedoch ist das Verfahren erweiter bar, das beide Eingabeparameter geheim sind.

**Kategorisierungskriterien:** additiv-homomorph, kleiner Klartextraum

**Malleability:** Das vorgestellen Kryptosystems wurde nicht auf Malleability untersucht obwohl bösartige Anfragen mögliche währen, da von einem honest-but-curious Angreifermodell ausgegangen wird. Ein Teilnehmer kann somit falsche Höchstgebote abgeben, jedoch gewinnt man dadurch keinen Vorteil. Entweder wird man früher aus dem Gebotsverfahren geschmissen oder er kann nach dem Gebotsverfahren feststellen ob der Betreiber die Vergleiche inkorrekt durchgeführt hat.

## 4.8 Secure Computation mit SMC oder HE

In den Studien [DGK07, p.420] und [SSW09, p.2] wurden zwei Techniken identifiziert um Funktionen sicher zu berechnen: Secure Multi-Party Computation und Homomorphe Verschlüsselung. Dabei kommen beide Studien unabhängig voneinander zu den gleichen Schlüssen über die Vorzüge der jeweiligen Verfahren:

- SMC Vorteil: viel Kommunikation
- SMC Nachteil: geringe Runden- und Berechenkomplexität
- HE Vorteil: geringe Kommunikation
- HE Nachteil: hohe Runden- und Berechenkomplexität

# 5 Zusammenfassung

# 6 Ausblick

| Kryptosystem | hom. Operator | sim. Operator | modi |
|---|---|---|---|
| Pailler | $+$ | XOR, $\langle \cdot, \cdot \rangle$ | bitwise, bytewise, float |
| ElGamal | $\cdot$ | - | - |
| hash-ElGamal | XOR | - | - |
| Goldwasser-Micaeli | XOR | | |
| BGV (HELib) | vollhom. | any | |

## Was man noch ergänzen könnte...

- Zum deterministischen Kryptosystem **??** kurz das Beispiel der Cäsarverschlüsselung erwähnen. Und weitere Recherche unter dem Begriff „deterministische Verschlüsselung“.

# Literatur


[BPTG15] Raphael Bost, Raluca Ada Popa, Stephen Tu, and Shafi Goldwasser. Machine learning classification over encrypted data. In *NDSS*, 2015.

[Cry] Cryptosystem - wikipedia. `https://en.wikipedia.org/wiki/Cryptosystem`. (Accessed on 03/28/2017).

[DGK07] Ivan Damgård, Martin Geisler, and Mikkel Krøigaard. Efficient and secure comparison for on-line auctions. In *Australasian Conference on Information Security and Privacy*, pages 416–430. Springer, 2007.

[DKK02] Hans Delfs, Helmut Knebl, and Helmut Knebl. *Introduction to cryptography*, volume 2. Springer, 2002.

[EFG+09] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. Privacy-preserving face recognition. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 235–253. Springer, 2009.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984.

[KT05] Minoru Kuribayashi and Hatsukazu Tanaka. Fingerprinting protocol for images based on additive homomorphic property. *IEEE Transactions on Image Processing*, 14(12):2129–2139, 2005.

[Mei16] Prof. Dr. Michael Meier. *Vorlesung IT-Sicherheit*, chapter 2. Grundlegendes. WS2015/16.

[NIW+13] Valeria Nikolaenko, Stratis Ioannidis, Udi Weinsberg, Marc Joye, Nina Taft, and Dan Boneh. Privacy-preserving matrix factorization. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 801–812. ACM, 2013.

[Rä15] Prof. Dr. Thoralf Räsch. *Vorlesung Lineare Algebra*. SS2015.

[Sma03] Nigel Paul Smart. *Cryptography: An Introduction*, volume 5. McGraw-Hill New York, 2003.

[SSW09] Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg. Efficient privacy-preserving face recognition. In *International Conference on Information Security and Cryptology*, pages 229–244. Springer, 2009.

[Sti06] Douglas R Stinson. *Cryptography: theory and practice*. CRC press, 2006.

[TSCS13] Shruti Tople, Shweta Shinde, Zhaofeng Chen, and Prateek Saxena. Autocrypt: enabling homomorphic computation on servers to protect sensitive web content. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1297–1310. ACM, 2013.