Lorem Ipsum...

Matthias Ulbrich
2743974

Bachelorarbeit

Erstprüfer: Prof. Dr. Michael Meier
Zweitprüferin: Jun.-Prof. Dr.-Ing. Delphine Reinhardt
Betreuerin: Dipl.-Inform. Saffija Kasem-Madani

Institut für Informatik IV
Arbeitsgruppe für IT-Sicherheit
Rheinische-Friedrich-Wilhelms-Universität Bonn

Danksagung

Besonders bedanken möchte ich mich bei meiner Betreuerin Dipl.-Inf. Saffija Kasem-Madani für die akademische Begleitung während der Bachelorarbeit und bei Prof. Dr. Michael Meier für die Gelegenheit in seiner Arbeitsgruppe die Bachelorarbeit zu schreiben.

Weiterer Dank gilt meinen Eltern, meiner Freudin und meiner Großtante für jegliche Unterstützung in meinem Studium bis zu diesem Punkt. Ich widme diese Arbeit euch.

Selbstständigkeitserklärung

Hiermit versichere ich, die vorliegende Bachelorarbeit ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Bonn, 23. April 2017

Matthias Ulbrich

Zusammenfassung

Diese Bachelorarbeit untersucht den vielfältigen Einsatz homomorpher Kryptosysteme in Forschungsarbeiten. Dabei wird erörtert, welches Kryptosystem zum Einsatz kommt und aus welchen Gründen ein Forscherteam sich für dieses entschieden hat. Es wird untersucht in welchem Anwendungskontext das Kryptosystem zum Einsatz kommt und ob die Malleabilität von homomorphen Kryptosystemen bei der Implementierung berücksichtigt wurde. Kurzgesagt ist Malleabilität ist die Fähigkeit auf Chiffren eine Funktion auszuführen, welche man im Klartext wiederfindet. Findet diese Veränderung unbemerkt statt, kann die Integrität von Rechenoperationen im Chiffreraum verletzt werden. Damit ist Malleabilität eine Eigenschaft die direkt aus der Homomorphieigenschaft resultiert.

Die evaluierten Einsatzscenarien homorpher Kryptosysteme werden im Einzelnen kurz vorgestellt und herangezogen um Kriterien zu erstellen anhand welcher eine Kategorisierung der Kryptosysteme erfolgt. Dies geschieht im Hinblick darauf Dritten bei der Entscheidung für den Einsatz homomorpher Kryptographie für eigene Arbeiten zu unterstützen.

Inhaltsverzeichnis

1 Einleitung

Ziel dieser Arbeit ist eine Erörterung der Vorteile von Anwendungen homomorpher Kryptosysteme für praktische Anwendungen. Homomorphe Kryptosysteme gliedern sich ihrerseits in semihomomorphe und vollhomomorphe Kryptosysteme. Während semihomomorphe Kryptosysteme eingeschränkt Operationen auf Chriffren ermöglichen - abhängig von dem gewählten Kryptosystem, ermöglichen vollhomomorphe Kryptosysteme die Berechnung jeder booleschen Funktion im Chiffreraum. Da letztere jedoch aufgrund hoher Laufzeiten weniger praktikabel sind greift man in der Praxis fast ausschließlich auf semihomomorphe Kryptosysteme zurück. Dies führt in der Praxis zu verschiedenen, vom jeweiligen Anwendungsfall abhängigen Einsatz eines Kryptosystems.

1.1 Beitrag dieser Bachelorarbeit

Es ist von vornherein nicht klar welche Gründe jemand gewählt hat ein Kryptosystem dem anderen vorzuziehen, insbesondere wenn sie den gleichen Operator im Chiffreraum zur Verfügung stellen. Weiter lässt sich mit einem Kryptosystem wie Paillier der Operator von Goldwasser-Micali simulieren. Diese Arbeit will die Gründe für den Einsatz verschiedener homomorphen Kryptosysteme untersuchen und Informationen sammeln um die einzelnen Kryptosysteme zu kategorisieren als Referenz für den zukünfigen Einsatz in Forschungsarbeiten.

Es wird untersucht wie mit der Malleablilität der homomorphen Kryptosysteme im einzelnen Umgegangen wird um die Möglichkeit einer Manipulation der Daten auszunutzen die einen Angreifer ermöglichen mehr Information aus dem System abzuschöpfen.

1.2 Aufbau

Zunächst werden im Kapitel 2 der Begriff eines Kryptosystems und darauf aufbauende Erweiterungen bis hin zum probabilistischen asymmetrischen Kryptosystem erläutert welches die Grundlage vieler semihomomorpher Verfahren bildet. Dann werden in Kapitel 3 verschiedene Sicherheitskriterien erläutert die in den untersuchten Anwendungsfällen entweder realisiert werden oder zum Verständnis benötigt werden. Einige dieser Sicherheiskriterien sind insbesondere von Notwendigkeit für die Klassifierung. In Kapitel 4 werden die untersuchten Studien vorgestellt, allerdings nicht in ihrem vollem Umfang, sondern in Bezug auf den Einsatz homomorpher Kryptosysteme, der Gründe für den Einsatz und dem Umgang mit Malleablilität.

2 Homomorphe Kryptosysteme

2.1 Schutzziele der Kryptografie

Die Kryptografie hat zur Aufgabe Lösungen für die Realisierung verschiedener Schutzziele bei der Speicherung, Vervielfältigung und Übertragung von Informationen umsetzen. Die Kryptografie stellt dazu verschiedene Algorithmen und Protokolle bereit. Grundlegende Schutzziele beim Übertragen von Informationen zwischen mehreren Parteien in Nachrichten sind nach [MVOV96, p.4][DKK02, p.2]:

1. **Vertraulichkeit:** Keine unauthorisierte Kenntnisnahme. Nur dazu berechtigte Personen sollen eine bestimmte Information lesen können oder Zugang zu dieser Information erhalten. Dieser Begriff ist Synonym mit Geheimhaltung. Vertraulichkeit kann physisch erreicht werden oder durch mathematische Algorithmen welche die Daten unverständlich machen.
2. **Integrität:** Keine unauthorisierte unbemerkte Datenmanipulation. Um die Integrität von Daten zu gewährleisten müssen die Möglichkeit einer Detektion von Veränderungen in den Daten realisiert werden. Dies schützt insbesondere vor dem Hinterlegen von Falschdaten in einer Nachricht oder dem Fehlen von Teilen einer Nachricht.
3. **Authentizität:** Authentizität meint die Fähigkeit einer Identifikation in Bezug auf die Information als auch auf die Kommmunikationspartner (Entitäten). Fordert man, dass die kommunizierenden Teilnehmer in der Lage sein sollen sich gegenseitig zu identifizieren spricht man von Authentizität der Entität. Bei einseitiger Kommunikation fordert man lediglich, den Urheber einer Nachricht identifizieren zu können, also die Authentizität des Datenursprungs.
4. **Nichtabstreitbarkeit:** Der Versand einer Nachricht kann von dem Sender nach dem Versand nicht mehr abgestritten werden. Zum Beispiel wenn eine Entität den Kauf in einer unabstreitbaren Nachricht zunächst authorisiert, aber später verneint, so kann ihr Konfliktfällen die ursprüngliche Zusage nachgewiesen werden.

2.2 Kryptosysteme

Ein Kryptosystem ist ein Sammlung von Algorithmen um das Schutzziel der Vertraulichkeit bei der Übertragung von Informationen 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 Zugang zu der geschützten der Information erhält.

Die zugrundelegenden Algorithmen und resultierende Eigenschaften über die Beziehung von Klartexten zu Chiffretexten führen zu verschiedenen Klassen von Kryptosystemen. Diese Krypto-

systeme 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).

In der Literatur lassen sich verschiedene Formalisierungen für ein Kryptosystem finden, und es ist nicht trivial sie zu harmonisieren. Wir benutzten die Definition von Douglas R. Stinson [Stio6, p.1] und erweitern diese Definition im Anschluss für eine bessere Differenzierung um Eigenschaften wie Determinismus oder Symmetrie.

Definition 2.2.1 (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 muss ein Kryptosystem also mindestens drei Algorithmen bereitstellen: Einen für die Erzeugung des Schlüssels, einen für die Verschlüsselung und einen für die Entschlüsselung [Cry]. Man beachte, dass der Schlüssel k als *ein* Element des Schlüsselraums selber aus *mehreren* Elementen zusammengesetzt werden kann. Diese Eigenschaft führt zu den nächsten beiden Erweiterungen.

Definition 2.2.2 (Symmetrisches Kryptosystem). *Sei $K = (\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D})$ ein Kryptosystem. Dann nennen wir K symmetrisch, wenn jede Verschlüsselungsfunktion e_k als auch die zugehörige Entschlüsselungsfunktion d_k vollständig von demselben Schlüssel $k \in \mathcal{K}$ abhängen. Vollständig bedeutet, dass diese Funktionen insbesondere nicht nur von einer Teilmenge von k abhängen, wenn der Schlüssel k aus mehreren Parametern zusammengesetzt ist. Alle Parameter von k gehen in die Verschlüsslungsfunktion und Entschlüsselungsfunktion ein. Letzteres ist in 2.2.1 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 bevor sie geheime Nachrichten austauschen können. Daher sind symmetrische Kryptosysteme auch bekannt als private-key Kryptosysteme.

Im Gegensatz dazu gibt es Kryptosysteme bei denen k aus einem privaten und einem öffentlichen Teilschlüssel zusammensetzt ist. Diese Teilmengen müssen nicht disjunkt[1] sein. Alice kann nun ihren öffentlichen Teilschlüssel bekannt geben um so dritten zu ermöglichen ihr Informationen vertraulich zukommen zu lassen. Daher spricht man auch von public-key Kryptosystemen, ein Konzept das Ursprünglich von Diffie und Hellmann in [DH76, p.648] eingeführt wurde.

Definition 2.2.3 (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 $k = (k_s,k_p)$ mit $k_s,k_p \in K$. 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.*

[1] In RSA enthalten die Mengen beider Teilschlüssel den Modulus n [RSA78, p.6].

Öffentlicher und privater Teilschlüssel stehen in einem mathematischen Zusammenhang, der jedoch für Angreifer mit begrenzter Rechenkapazität praktisch unmöglich ist zu erschließen.

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 Sicherheitsklassen welche in 3.3 vorgestellt werden.

Definition 2.2.4 (Deterministisches Kryptosytem)**.** *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 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 Verschlüsselungsfunktion d_k zu einem identischen Chiffrat!

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

Das Konzept eines Probalistischen Kryptosystems wurde ursprünglich von Goldwasser und Micali eingeführt in [GM84]. Wir definieren in Anlehnung an [Stio6, p.345]:

Definition 2.2.5 (Probalistisches Kryptosytem)**.** *Ein Probalistisches Kryptosystem ist ein Sextupel $(\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D},\mathcal{R})$. Wie bereits in Definition 2.2.1 ist $\mathcal{P}$ ist der Klartextraum, $\mathcal{C}$ der Chiffreraum und $\mathcal{K}$ der Schlüsselraum. Neu sind:*

- *$\mathcal{R}$ ist eine endliche Menge von Randomisierern*
- *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 Randomisierer $r \in \mathcal{R}$ folgende Identität gilt: $d_k(e_k(x,r)) = x$*

Für ein festes $k \in K$ und ein beliebigen Klartext $x \in P$ definieren wir die Wahrscheinlichkeitsverteilung $p_{K,x}$ auf C, so dass $p_{K,x}(y)$ die Wahrscheinlichkeit angibt, dass y ein Chiffrat von x unter e_k ist. Nun fordern wir:

- *Gegeben $x_1,x_2 \in P, x_1 \neq x_2$ und $k \in K$. Dann sind die Wahrscheinlichkeitsverteilungen p_{K,x_1} und p_{K,x_2} nicht in Polynomialzeit unterscheidbar.*

Nicht unterscheidbar hat insbesondere zur Folge, dass wir auch nicht wissen ob die gleiche Wahrscheinlichkeitsverteilung hinter zwei Chiffraten steckt. In anderen Worten: Die wiederholte Verschlüsselung eines Klartextes führt im Allgemeinen zu verschiedenen Chiffraten.

Ein Proballistisches Kryptosystem nutzt 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 Randomizierer $\mathcal{R}$ entspricht dann der Menge möglicher Initialisierungsvektoren.

Semihomomorphes Kryptosystem

Zusammen mit 2.2.1 können wir nun ein semihomomorphe Kryptosystem definieren. Laut 2.2.12 ist ein Homomorphismus Strukturerhaltend. Für Verknüpfungen im Chiffreraum findet man eine

entsprechende Verknüpfung im Klartextraum. Diese Eigenschaft ist *die* Stärke der homomorphen Kryptographie, denn sie ermöglicht eine Verarbeitung von Daten unter Wahrung der Vertraulichkeit dieser und wird somit oft einsetzt um bekannte Protokolle privacy-preserving zu machen wie mehrere der vorgestellten Projekte in 4 zeigen. Auf der anderen Seite kann die Homomorphieeigenschaft bösartig ausgenutzt werden wie in 3.1 an einem Beispiel erläutert wird.

Definition 2.2.6 (Semihomomorphes Kryptosytem)**.** *Sei $K = (\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D})$ ein asymmetrisches Kryptosystem. Wir nennen K semihomomorph, wenn $(P,\otimes)$ und $(C,\odot)$ Gruppen bilden und die Verschlüsselungsfunktion ein Homomorphismus von Gruppen ist.*

- *Das heißt alle unter e_{k_p} erzeugten Chiffrate bilden einen Gruppe.*
- *Sei $e_{k_p}(x_1) = c_1, e_{k_p}(x_2) = c_2$. Dann gilt: $d_{k_s}(c_1 \odot c_2) = m_1 \otimes m_2$*

Diese Definition ist orientiert an [KL14, p.499].

Vollhomomorphes Kryptosystem

Lange Zeit nahm man an, dass es kein Kryptosysteme gibt, welches beliebige Rechenoperationen im Chiffreraum ermöglicht - d.h. vollhomomorph wäre. Bekannte semihomomorphe Kryptosysteme erlauben nur eingeschränkte Operationen(z.B. Addition oder XOR). Im Jahre 2009 stellte schließlich Craig Gentry [Gen09] erstmals ein Verfahren vor, wo im Chiffreraum alle Operationen eines Rings möglich sind, d.h. Addition und Multiplikation. Analog definieren wir nun [YPB14, p.47]:

Definition 2.2.7 (Vollhomomorphes Kryptosytem)**.** *Sei $K = (\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D})$ ein asymmetrisches Kryptosystem. Wir nennen K vollhomomorph, wenn $(P,\otimes,\ominus)$ und $(C,\odot,\oplus)$ Ringe bilden und die Verschlüsselungsfunktion ein Homomorphismus von Ringen ist.*

- *Das heißt alle unter e_{k_p} erzeugten Chiffrate bilden einen Ring.*
- *Sei $e_{k_p}(x_1) = c_1, e_{k_p}(x_2) = c_2$. Dann gilt: $d_{k_s}(c_1 \odot c_2) = m_1 \otimes m_2$ sowie $d_{k_s}(c_1 \oplus c_2) = m_1 \ominus m_2$*

Gentry konnte zeigen, dass ein beliebe Schaltung aus NANDs im Chiffreraum evaluiert werden kann. Dies ist ein Durchbruch, denn NANDs bilden für sich ein vollständiges Operatorensystem mit dem jede boolesche Funktion beschrieben werden kann [Hof10, p.129]. Damit ermöglicht ein vollhomomorphes Kryptosystem die Berechnung einer *beliebigen* booleschen Funktion im Chiffreraum!

Ein vollhomomorphes Kryptosystem hat besteht aus vier Algorithmen. Zusätzlich zum Schlüsselgenerierung, Verschlüsselung und Entschlüsselung gibt es jetzt einen Algorithmus für die Evaluierung eines Schaltkreises im Chiffreraum. Sei k_p der öffentliche Schlüssel unter dem die Chiffrate $c_1,\ldots,c_l$ erzeugt worden. Dann ermöglicht der Evaluierungsalgorithmus das berechnen eines beliebigen Schaltkreises S zu dem Ergebnis c:

$$\textbf{Evaluate}_{k_p}(S, c_1,\ldots,c_l) = c$$

Trotz dieser großen Errungenschaft, finden bis heute vollhomomorphe Kryptosysteme wenig Einsatz in der Praxis, da die Verknüpfungen im Chiffreraum zu viel Rechenkapazität kosten.

Statt drei jetzt vier algorithmen: KeyGen, Encrypt, Decrpyt, und Evaluate (homorphic evaluation) [**?**]

expansion rate Ratio between lenght of ciphertext and the length of cleartext. [NS98, p.63]

The expansion rate of probalistic cryptosystems is usually very large, i.e. one kilobit is needed to encrypt on a few bits. [NS98, p.60]

Stattdessen weicht man auf Semihomomorphe Kryptosysteme auf, wenn sie den Anforderungen genügen oder realisiert eine Kombination aus zwei Kryptosystemen von denen eins additiv und das andere multiplikativ ist. Ein Beispiel für so eine Realisierung werden wir in 4.2 sehen.

Abschließende Anmerkungen:
Es kommen fast ausschließlich probalistische homomorphe Kryptosysteme zum Einsatz, obwohl es ebenso deterministische homomorphe Kryptosysteme gibt. Jedoch wurde bereits von Beoneh und Lipton in 1996 gezeigt, das *jedes* deterministische homomorphe Kryptosystem in subexponentieller Zeit gebrochen werden kann [**?**].

Stufenfixes Homomorphes Kryptosystem

2.2.1 Algebraische Strukturen

Wir werden später Homomorphe Kryptosystem im Detail einführen. Um ihre Anwendung zu verstehen, ist es jedoch nötig folgende Algebraische Strukturen nach [Fis11] einzuführen um ein Verständnis dafür zu schaffen wie Homomorphe Kryptosysteme verwendet werden könnnen.

Definition 2.2.8 (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 2.2.9 (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 2.2.10 (Körper). *Sei K ein kommutativer Ring mit Eins wie in 2.2.9, 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 2.2.11 (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*

[2] Evtl. ausführlicher formulieren

- $\otimes$ *ist assoziativ, hat ein neutrales Element bzgl. V*
- *Elemente aus K und V sind distributiv*

Dann nennen wir V einen K-Vektorraum.

Definition 2.2.12 (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:*

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

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! [p.380][KL14]

Ciphertext expansion

3.4 provable security

uses reduction

3.5 ???

Außer dem One-Time Pad wurde kein anderes Kryptosystem als unconditioneal secure bewiesen. Daher betrachtet man bei der Sicherheit immer die Rechenkapazitäten des Angreifers

3.6 Attack Model

chosen-plaintext IND-CPA nonapdaptive chosen chiphertext IND-CCA1 adaptive chosen ciphertext IND-CCA2 (implies nonmalleability) IND stands for indistinguishability In asyymetrischen Verfahren kann jeder beliebige plaintexte verschüssel. Daher gegenüber asyymetrische Verfahren der Angreifer immer Fähigkeit eines chosen-plaintext Angriffs

semantic security [34] in homo for non-experts polynomial security (= indestinguishability) [36] semantic security and polynomial security are equivalent! [34]

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 byteweise 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 $x = (x_1,\dots,x_n)$ und $y = (y_1,\dots,y_n)$ wobei alle Einträge Klartexte sind. Das mit pub Paillierverschlüsselte Skalarprodukt ist dann:

$$Enc_{pub}(\langle x,y\rangle) = \prod_i Enc_{pub}(y_i)^{x_i} \bmod N^2 \tag{4.1}$$

noch auszuführen

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 μ 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 Efficient and Secure Comparison for On-Line Auctions

Ivan Damgard et al. stellen in [DGK07] ein neues additives Kryptosystem DGK vor um schnelle vergleiche einer öffentlich bekannten Zahl x und einer bitweise (vgl. 4.8) verschlüsselten Zahl m durchzuführen die auf einem Server und einem Hilfsserver verteilt ist. Bei der online 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 erweiterbar, so das beide Eingabeparameter privat sind. Ein auf ihrem Verfahren basierender Vergleich von zwei privater Zahlen wird in 4.7 vorgestellt.

Kategorisierungskriterien: additiv-homomorph, kleiner Klartextraum

Malleability: Das vorgestellen Kryptosystems wurde nicht auf Malleability untersucht obwohl bösartige Anfragen möglich 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.

4.6 Fingerprinting Protocol for Images Based on Additive Homomorphic Property

[KT05] Das vorgestellte Protocol wurde unter dem Hintergrund eingeführt, da bisher bekannte Lösungen entweder eine zu langsame Verschlüsselungsrate hatten oder das Einsetzen des Wasserzeichens zu aufwendig ist. Es wird ein homomorphes public-key Kryptosystem eingesetzt um asymmetrische Fingerprints zu erzeugen. Dies ist notwendig, da bei symmetrischen Fingerprinting der Verkäufer ein benutzerspezifisches Wasserzeichen erzeugen kann und somit auch einen rechtmäßigen Käufer eine Raubkopie unterstellen könnte.

Obwohl das Protocol mit Pailler umgesetzt werden könnte, hat man sich für das Kryptosystem von Okamoto-Uchiyama [OU98] entschieden, da weniger Rechenoperationen durchgeführt werden müssen [p.2132].

Die Asymmetrie und Homomorphie vom Okamoto-Uchiyama Kryptosystem wird wie folgt beim Fingerprinting ausgenutzt:

1. Der Käufer erzeugt einen Fingerabruck, verschlüsselt ihn mit seinem öffentlichen Schlüssel, und sendet ihn an den Verkäufer. Mit einem Zero-Knowledge-Proof wird dem Verkäufer nachgewiesen, dass das Chiffrat tatsächlich einen nutzerspezifischen Fingerabdruck enthält.
2. Nun verschlüsselt der Verkäufer sei digitales Bild unter dem gleichen öffentlichen Schlüssel wie der Käufer und bettet den Fingerabruck durch homomorphe Verknüpfung ein.
3. Der Käufer entschlüsselt das Bild mit dem Wasserzeichen ohne jedoch in der Lage zu sein das Wasserzeichen zu entfernen, da ihm nicht bekannt ist in welchen Positionen oder Frequenzbereichen (bei Anwendung des Wasserzeichens im Frequenzraum unter diskreter Kosinustransformation) das Wasserzeichen das ursprüngliche Bild verändert hat.

Weiter gewährleistet die semantische Sicherheit von Okamoto-Uchiyama anonymes Kaufen [p.2134], d.h. der Verkäufer ist nicht in der Lage die Identität des Käufers aufzudecken.

Kategorisierungskriterien: additiv-homomorph, semantisch-sicher

Malleability: Die Autoren sind sich bewusst der Mlleability des verwendeten Kryptosystems, ohne jedoch mögliche Angriffe zu untersuchen. Der Hauptfokus des Sicherheitsbegriffs liegt in der Unmöglichkeit des Käufers, das Wasserzeichen zu entfernen und für den Verkäufer, nicht in der Lage zu sein die Identität des Käufers zu aufzudecken.

4.7 Privacy Preserving Face Recognition

Das Team um Zekeriya Erkin et al. [EFG+09] stellt ein originelles privacy-preserving Gesichtserkennungssystem vor bei dem sowohl die Eingabebilder als auch das Ergebnis ihrer Analyse vom Server verdeckt bleiben. Der Analyse zugrunde liegt ein Eigenfacesalgorithmus, welcher auf verschlüsselten Bildern arbeitet. Eingesetzt werden die Kryptosysteme Paillier und DKG [DGK07] welches auch in 4.5 zum Einsatz kommt. Sie wählen die Basis g des öffentlichen Schlüssel als $g = n + 1$, welches die Verschlüsslung beschleunigt [p.237][DJ01]. DKG wird lediglich auch Effizienzgründen anstelle von Paillier eingesetzt. Der Klartextraum im implementierten DKG ist kleiner. Da die Exponenten kleiner sind, ist die Verschlüsselung effizienter.

Da die Verfahren nur mit Integers arbeiten, werden Featurevektoren des Gesichtserkennungssytem diskretisiert in dem auf den nächsten Integer gerundet wird.

Rechenoperationen im Chiffreraum im privaten Gesichtserkennungssystem: Wann immer die eckigen Klammern auftauchen, ist ein Element verschlüsselt unter dem öffentlichen Schlüssel von Alice, welche ein Gesicht analysieren möchte. Sie übergibt dieses Gesicht verschlüsselt Bob, der dank homomorpher Kryptographie in der Lage ist die Analyse durchzuführen ohne das Gesicht direkt zu sehen.

- *Projektion* des verschlüsselten Eingabebilds Γ auf die Basis von Eigenfacevektoren $u_1, \ldots, u_K$. Hier zu wird mit der gleichen Technik wie in 4.3 ein Skalarprodukt durch Potenzieren berechnet. Das Ergebnis ist ein verschlüsselter Featurevektor des Eingabebildes $[\![\overline{\Omega}]\!]$.

- *Abstand D* von Featurevektoren $\{\Omega_1,\ldots,\Omega_M\}$ der Datenbank des Servers zum Featurevektor des Eingabebildes $[\![\overline{\Omega}]\!]$. Da man nur an der relativen Ordnung der Abstände interessiert ist, genügt der Vergleich der quadierten Abstände:

$$\begin{aligned} D(\Omega,\overline{\Omega}) &= \|\Omega-\overline{\Omega}\|^2 = (\omega_1-\overline{\omega}_1)^2+\ldots+(\omega_K-\overline{\omega}_K)^2 \\ &= \underbrace{\sum_{i=1}^{K}\omega_i^2}_{S_1} + \underbrace{\sum_{i=1}^{K}(-2\omega_i\overline{\omega}_i)}_{S_2} + \underbrace{\sum_{i=1}^{K}\overline{\omega}_i^2}_{S_3} \end{aligned}$$

  Da Bob den Server betreibt, kennt er die Komponenten ω_i der Featurevektoren in der Datenbank und kann S_1 direkt berechnen. Da er die Komponenten $\overline{\omega}_i$ des Featurevektor vom Eingabebild nur verschlüsselt vorliegen hat, muss er S_2 analog wie beim Skalarprodukt in 4.3 durch potenzieren berechnen. Letzendlich kann Bob S_3 nur in Kooperation mit Alice berechnen, da bei beide Faktoren des Produkts ihm unbekannt sind. Dazu maskiert er die Komponenten des Featurevektors mit gleichverteilten Zufallswerten r_i:

$$[\![x_i]\!] = [\![\overline{\omega}_i + r_i]\!]$$

  Diese maskierten Komponenten sendet er an Alice, welche diese mit ihrem privaten Schlüssel entschlüsselt und quadiert, das Ergebnis $S_3^{'} = \sum_{j=1}^{K} x_i^2$ wieder verschlüsselt und dann an Bob zurücksendet. Bob erhält dann S_3 durch:

$$[\![S_3]\!] = [\![S_3^{'}]\!] \cdot \prod_{j=1}^{K}\left([\![\overline{\omega}_i]\!]^{(-2r_i)} \cdot [\![-r_i^2]\!]\right)$$

  Was korrekt ist, da die i-te Komponente sich ergibt aus:

$$[\![x_i^2]\!] \cdot [\![\overline{\omega}_i]\!]^{(-2r_i)} \cdot [\![-r_i^2]\!] = [\![(\overline{\omega}_i + r_i)^2 - 2r_i\overline{\omega}_i - r_i^2]\!] = [\![\overline{\omega}_i^2]\!]$$

- *Vergleichen* zweier privater Zahlen im DGK Kryptosystem. Gegeben seien zwei bitweise verschlüsselte Zahlen $[\![d]\!]$ und $[\![r]\!]$. Alice generiert das Schlüsselpaar unter DGK und sendet den öffentlichen Schlüssel, sowie die ihre verschlüsselten Bits $[\![d_{l-1}]\!],\ldots,[\![d_0]\!]$ zu Bob. Bob berechnet dann:

$$[\![c_i]\!] = [\![d_i - r_i - 1 + 3\sum_{j=i+1}^{l-1} d_j \oplus r_j]\!] = [\![d_i]\!] \cdot [\![-r_i]\!] \cdot [\![1]\!] \cdot \left(\sum_{j=i+1}^{l-1} [\![d_j \oplus r_j]\!]\right)^3$$

  Diese Formel hat die Eigenschaft: *alle* c_i sind ungleich Null genau dann, wenn d die größere der beiden Zahlen ist. Umgekehrt ist mindestens ein c_i null, wenn $d_i \leq r_i$. Nämlich das höchstwertigste Bit wo sich die beiden Zahlen unterscheiden. Die 1 wird addiert, damit ein c_i gerade Null annimmt, wenn d kleinergleich ist. Der Faktor 3 garantiert, dass c_i ungleich Null ist sobald sich die Zahlen in einem Bit unterscheiden.

Man sieht, dass alleine durch die Möglichkeit Linearkombinationen von Chiffren zu bilden ein additiv-homomorphen Kryptosystems komplexer Operatoren ausführen kann!

Kategorisierungskriterien: additiv-homomorph, Simulierte Operatoren: Projektion und Abstand (wobei teilweise Parameter unverschlüsselt vorlagen) und Vergleich (beide Zahlen verschlüsselt)

Malleabiliy: Anstelle von malleability wird in dieser Veröffentlichung lediglich von „re-randomization“ gesprochen, was sich bezieht auf eine Neuverschlüsselung einer Zahl z.B. durch Addition von Null. Alice vertraut Bob der korrekten Durchführung des Eigenfacesalgorithmus und Bob akzeptiert, dass Alice grobe Eigenschaften über seiner Gesichtserkennungssystem lernen kann. Es wird von einem honest-but-curious Modell ausgegangen. Eine Malleability wurde nicht untersucht, jedoch ist klar, dass Bob nie Einsicht in private biometrische Daten von Alice erhält. Denn Bob operiert auf Chiffren die unter Alices öffentlichen Schlüssel verschlüsselt sind. Bob kann eine Malleability nicht ausnutzen, weil er keine entschlüsselten Informationen bei Alice anfragen kann.

Alice ist zwar in der Lage beliebig veränderte Chiffrate Bob zurückzugeben, z.B. die x_i bei der Abstandsberechnung. Jedoch führt dies lediglich zu einem verfälschten Ergebnis des Eigenfacesalgorithmus welchen Bob ausführt. Damit kann Alice Malleability nicht für eigene Vorteile ausnutzen.

4.8 Private predictive analysis on encrypted medical data

In [BLN14] stellen Joppe W. Bos et. al stellen ein Verfahren vor zur privaten Analyse von in die Cloud ausgelagerten medizinischer Daten. Sie implementieren eine logische Regression auf homomorph verschlüsselten Daten um die Wahrscheinlichkeit einer Herzkreislauferkrankung zu prädizieren. Zum Einsatz kommt ein Stufenfixes (engl. leveled) homomorphes Kryptosystem (2.2). In dem Präditkionsmodell mit logistischer Regression ist die Prädiktionsformel

$$P(x) = \frac{e^x}{e^x - 1}$$

wobei x eine Linearkombination von gewichteten Regressionskoeffizienten repräsentiert. Somit ist x offenbar homomorph berechenbar. Die Prädiktionsformel wird homomorph berechenbar indem man sie durch eine Taylorreihe annähert.

In Anlehnung an [BLLN13] [NLV11] stammen daher die zu verknüpfenden Variablen aus einem Polynomring $R = \mathbb{Z}/(X^n + 1)$. Der Klartextraum und Chiffreraum besteht aus Polynomen $\sum_{i=0}^{n-1} a_i X^i, a_i \in \mathbb{Z}$. Operationen im Chiffreraum entsprechen der Addition und Multiplikation von Polynomen mod $X^n + 1$. Mit diesem Stufenfixen homomorphen Kryptosystem ist es dann möglich alle Operationen zur Berechnung der Prädiktionsformel durchzuführen. Trotzdem ist noch folgendes Problem vorhanden: Die Eingabe welche eine Linearkombination von prädizierenden Regressionskoeffizienten ist, sowie die letztere selbst sind in der Anwendung typischerweise rationale Zahlen.

Um das Kryptosystem verwenden zu können, müssen alle Eingaben als Polynome dargestellt werden, z.B. durch die 2-adische Darstellung: $\sum_{i=0}^{n-1} a_i 2^i, a_i \in \{0,1\}$. Wir können mit rationalen Zahlen rechnen wenn wir eine fixe Fließkommagenauigkeit vorab festgelegen (d.h. Grad des Polynoms). Dann werden rationale Zahlen zu Integern rück-/transformiert durch Division/Multiplikation mit dem zugehörigen Fakter k^{10}.

Kategorisierungskriterien: leveled-homomorphic-encryption (multiplikativ, additiv), simuliert p-adische Darstellung womit rationale Zahlen dargestellt werden können

Malleabiliy: Eine malleability der Kryptosystems wurde nicht untersucht. Der Cloudprovider übernimmt die Rolle outgesourcter Rechenkapazität die Eingaben deterministisch verarbeitet. Im

Kontext des Anwendungsfalls sind sowohl Kunde als auch Cloudprovider an komplett ehrlicher, aber vertraulicher Verarbeitung interessiert.

4.9 Sichere Berechnung von Funktionen 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

4.10 Bitweise Verschlüsselung vs Integerverschlüsselung

Mit bitweiser Verschlüsslung lassen sich Grundoperatoren XOR und AND erzeugen. Und damit nach Abschnitt 2.2.1 jede boolesche Funktion bestimmen. Nachteil ist, dass ein Chiffretext weniger Information erhält. Bitweise Verschlüsslung erzeugt einen separaten Chiffretext für jedes Bit und erzeugt damit einen großen Overhead, so dass nach Möglichkeit auf Integerverschlüsselung ausgewichen wird. [?]

4.11 Zusammenfassung...

Homomorphe Kryptosysteme können verwendet werden für Sichere Wahlsysteme (paillier), Kollisionsresistente hashfunktionen und Private Informationsrückgewinnungssysteme (Kushilevitz and Ostrovsky single server PIR protocol) [MPS12, p.228] (Lieber andere Quellen!)

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	

5 Verwandte Arbeiten

In [MPS12] werden verschiedene partiell- und vollhomorphe Kryptosysteme vorgestellt als generelle Lösung für Vertraulichkeitsprobleme im Cloud Computing. Sie erwarten, dass homomorphe Kryptosysteme in der Zukunft schneller werden und somit ihre Relevanz für den Praxiseinsatz steigt. In ihrer Veröffentlichung werden die Ver- und Entschlüsselungsalgorithmen der verschiedener Kryptosysteme kurz vorgestellt und in Abschnitt 3 erste Beispiele für einen Einsatz in Cloud Computing, Sicheren Wahlsystemen und Privaten Informationsrückgewinnungssystemen erwähnt.

Der Einsatz von homomorpher Kryptographie geschieht nur unter dem Hintergrund der Sicherung von Vertraulichkeit. Weder eine Malleability der Kryptosysteme wird erörtert noch die Möglichkeit von Angreifern Schwächen in den vorgestellten Beispielen auszunutzen.

Der Artikel [FG07] dient als Übersicht und Einführung in Homomorphe Kryptographie für fachfremde Wissenschaftler. Die Autoren betonen ausdrücklich die Malleablität von homomorphen Kryptosystemen und stellen klar, dass diese geforderte Eigenschaft das Erreichen von IND-CCA2 Sicherheit unmöglich macht. Sie gehen auch darauf ein, dass die Forderung nach semantischer Sicherheit zur Folge hat, dass eingesetzte Kryptosysteme proballistisch sein müssen. In einer anschließenden Vorstellung einzelner Kryptosysteme bleiben die Autoren jedoch der Vorstellung eines Anwendungskontexts schuldig.

Beide erwähnten Quellen erwähnen, die Unpraktiabilität von vollhomomorpher Kryptographie für den praktischen Einsatz wegen hoher Anforderung an vorhandene Rechenkapazität.

6 Zum Abschluss

Cloud Computing mit HE ist eine Lösung zur Wahrung von Vertraulichkeit, aber präsentiere (welche) Lösungen zeigen auf, dass dem Endnutzer noch nicht gezeigt ist, dass diese Daten auch legal und richtig verarbeitet wurden.

Jede wissenschaftliche Arbeit ist selbstverständlich den subjektiven Kriterien des Autors unterworfen, die seinen persönlichen Stil einen Sachverhalt zu präsentieren und zu charakterisieren. Dieser Stil muss nicht unbedingt mit dem des Lesers übereinstimmen. Daher sind sicher auch vereinzelte Aspekte, die in diesem Leitfaden besprochen werden, durch unsere subjektive Vorstellung motiviert, wie eine gute Ausarbeitung zu erstellen ist.

Die meisten der in diesem Leitfaden angesprochenen Punkte spiegeln jedoch Erfahrungswerte wider, die sich bei der Präsentation unserer eigenen wissenschaftlichen Texte bewährt haben. Daher lohnt es sich, diesen Leitfaden bei der Erstellung Ihrer wissenschaftlichen Arbeit im Hinterkopf zu behalten, um die gröbsten technischen Fehler im vorhinein zu vermeiden.

7 Literaturverzeichnis

[BLLN13] Bos, Joppe W. ; Lauter, Kristin E. ; Loftus, Jake ; Naehrig, Michael: Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme. In: *IMA Int. Conf.* Springer, 2013, S. 45–64

[BLN14] Bos, Joppe W. ; Lauter, Kristin ; Naehrig, Michael: Private predictive analysis on encrypted medical data. In: *Journal of biomedical informatics* 50 (2014), S. 234–243

[BPTG15] Bost, Raphael ; Popa, Raluca A. ; Tu, Stephen ; Goldwasser, Shafi: Machine Learning Classification over Encrypted Data. In: *NDSS*, 2015

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

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

[DH76] Diffie, Whitfield ; Hellman, Martin: New directions in cryptography. In: *IEEE transactions on Information Theory* 22 (1976), Nr. 6, S. 644–654

[DJ01] Damgård, Ivan ; Jurik, Mads: A generalisation, a simpli. cation and some applications of paillier's probabilistic public-key system. In: *International Workshop on Public Key Cryptography* Springer, 2001, S. 119–136

[DKK02] Delfs, Hans ; Knebl, Helmut ; Knebl, Helmut: *Introduction to cryptography*. Bd. 2. Springer, 2002

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

[FG07] Fontaine, Caroline ; Galand, Fabien: A survey of homomorphic encryption for nonspecialists. In: *EURASIP Journal on Information Security* 2007 (2007), Nr. 1, S. 1–10

[Fis11] Fischer, Gerd: Lernbuch Lineare Algebra und Analytische Geometrie. In: *Vieweg+ Teubner, Wiesbaden* (2011)

[Gen09] Gentry, Craig: *A fully homomorphic encryption scheme*, Stanford University, Diss., 2009

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

[Hof10] Hoffmann, Dirk W.: *Grundlagen der technischen Informatik*. Carl Hanser Verlag GmbH Co KG, 2010

[KL14] Katz, Jonathan ; Lindell, Yehuda: *Introduction to modern cryptography*. CRC press, 2014

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

[MPS12] Maimut, Diana S. ; Patrascu, Alecsandru ; Simion, Emil: Homomorphic encryption schemes and applications for a secure digital world. In: *Journal of Mobile, Embedded and Distributed Systems* 4 (2012), Nr. 4, S. 224–232

[MVOV96] Menezes, Alfred J. ; Van Oorschot, Paul C. ; Vanstone, Scott A.: *Handbook of applied cryptography*. CRC press, 1996

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

[NLV11] Naehrig, Michael ; Lauter, Kristin ; Vaikuntanathan, Vinod: Can homomorphic encryption be practical? In: *Proceedings of the 3rd ACM workshop on Cloud computing security workshop* ACM, 2011, S. 113–124

[NS98] Naccache, David ; Stern, Jacques: A new public key cryptosystem based on higher residues. In: *Proceedings of the 5th ACM conference on Computer and communications security* ACM, 1998, S. 59–66

[OU98] Okamoto, Tatsuaki ; Uchiyama, Shigenori: A new public-key cryptosystem as secure as factoring. In: *Advances in Cryptology—EUROCRYPT'98* (1998), S. 308–318

[RSA78] Rivest, Ronald L. ; Shamir, Adi ; Adleman, Leonard: A method for obtaining digital signatures and public-key cryptosystems. In: *Communications of the ACM* 21 (1978), Nr. 2, S. 120–126

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

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

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

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

[YPB14] Yi, Xun ; Paulet, Russell ; Bertino, Elisa: *Homomorphic encryption and applications*. Bd. 3. Springer, 2014