Studienarbeit

Fully Homomorphic Encryption

Irena Schindler

Leibniz Universität Hannover

Fakultät für Elektrotechnik und Informatik

Institut für Theoretische Informatik

Contents

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

_____

Hannover, den 15.07.2012

1 Introduction

The public key encryption cryptography started in 1978, as the first step was done by Ron Rivest, Adi Shamir and Leonard Adleman, who described RSA algorithm for public-key cryptography. RSA is a multiplicative homomorphic encryption scheme - i.e., it is possible to compute a ciphertext that is the encryption of a product of the genuine plaintexts [1], [2], [3].
For a long time there was the desire for fully homomorphic encryption scheme. The fully homomorphic encryption scheme is far more powerful, it provides the opportunity to perform any efficient computable calculation on encrypted data, without knowing the content. However it was not sure, if the fully homomorphic encryption scheme was even possible [3], [4], [5].
Since then there were published many papers about partially homomorphic cryptosystems, like ElGamal [6], Goldwasser-Micali [7], or Boneh-Goh-Nissim cryptosystem [8]. But the breakthrough was done by Crain Gentry, as he announced fully homomorphic encryption using ideal lattices on June 2009 [9].
The potential applications for fully homomorphic encryption scheme could be for example improved spam identification in encrypted emails [10]. Furthermore fully homomorphic encryption makes possible to work on a strictly confidential project in a team, without involving the staff in the content [3].
The aim of this work is to provide an insight into the construction of a fully homomorphic encryption scheme $\mathcal{E}$ presented by Crain Gentry. The fully homomorphic encryption scheme will be constructed step by step, and starts with nearly homomorphic encryption scheme. Later on the ideal lattices will be used to obtain self-embedding, bootstrappable homomorphic scheme, in order to "refresh" the ciphertext and reduce the noise parameter and afterwards to convert into fully homomorphic encryption.

2 Basic Definitions

Fully homomorphic encryption scheme will be establish stepwise. First it is required to give an overview of conventional public-key encryption scheme $\mathcal{E}$, which is based on the following algorithms:

- (sk,pk)$\leftarrow$**KeyGen**$_{\mathcal{E}}(\lambda)$
- $\psi_i \leftarrow$ **Encrypt**$_{\mathcal{E}}$(pk, π_i)
- **Decrypt**$_{\mathcal{E}}$(sk,ψ)$\rightarrow \pi$

KeyGen$_{\mathcal{E}}$: is randomized algorithm that generates both the public key pk and secrete key sk, based on security parameter λ. The public key defines a plaintext space $\mathcal{P}$ and ciphertext space $\mathcal{C}$.
Encrypt$_{\mathcal{E}}$: is also a randomize algorithm that uses the valid public key and the plaintext $\pi_i \in \mathcal{P}$ for computing associated ciphertext $\psi_i \in \mathcal{C}$.
Decrypt$_{\mathcal{E}}$: outputs the original plaintext π with the aid of secrete key sk and ciphertext ψ.

The computational complexity of those algorithms should not exceeds $\lambda^{O(1)}$.

To construct a homomorphic encryption scheme $\mathcal{E}$, is an additional efficient and possibly randomized algorithm $\psi \leftarrow$**Evaluate**$_{\mathcal{E}}$(pk,C, Ψ) required. With a circuit C from permitted set $\mathcal{C}_{\mathcal{E}}$ of circuits, and a tuple of ciphertexts $\Psi = \langle \psi_1, \ldots, \psi_t \rangle$ for the input wires of C.

Obviously, the information after the encryption should be not adulterated. The correctness is defined as follows:

Definition 1 (**Correctness of Homomorphic Encryption**)**.**
We say that a homomorphic encryption scheme $\mathcal{E}$ is correct for circuits in $\mathcal{C}_\mathcal{E}$ if, for any key-pair (sk,pk) outputs by $\boldsymbol{GenKey}_\mathcal{E}(\lambda)$*, any circuit $C \in \mathcal{C}_\mathcal{E}$, any plaintexts $\pi_1, \ldots, \pi_t$ and any ciphertexts $\Psi = \langle \psi_1, \ldots, \psi_t \rangle$ with $\psi_i \leftarrow$* $\boldsymbol{Encrypt}_\mathcal{E}(pk, \pi_i)$*, it is the case that:*
if
$\psi \leftarrow$ $\boldsymbol{Evaluate}_\mathcal{E}$ (pk, C, Ψ),
then
$\boldsymbol{Decrypt}_\mathcal{E}(sk, \psi) \rightarrow C(\pi_1, \ldots, \pi_t)$
except with negligible probability over the random coins in $\boldsymbol{Evaluate}_\mathcal{E}$*.*

The definition of the correctness enables to give an formal definition of the homomorphic and fully homomorphic encryption scheme $\mathcal{E}$.

Definition 2 (Homomorphic Encryption)**.**
E is homomorphic for circuits in $\mathcal{C}_\mathcal{E}$ if $\mathcal{E}$ is correct for $\mathcal{C}_\mathcal{E}$ and $\boldsymbol{Decrypt}_\mathcal{E}$ *can be expressed as a circuit $D_\mathcal{E}$ of size poly(λ).*

Definition 3 (Fully Homomorphic Encryption)**.**
$\mathcal{E}$ is fully homomorphic if it is homomorphic for all circuits.

Definition 4 (Leveled Fully Homomorphic Encryption)**.**
A family of schemes $\left\{\mathcal{E}^{(d)} : d \in \mathbb{Z}^+\right\}$ is leveled fully homomorphic if they all use the same decryption circuit, $\mathcal{E}^{(d)}$ is homomorphic for all circuits of depth at most d (that use some specified set of gates Γ), and the computational complexity of $\mathcal{E}^{(d)}$'s algorithms is polynomial in λ, d, and (in the case of $\boldsymbol{Evaluate}_{\mathcal{E}^{(d)}}$*) the size of C.*

3 Bootstrappable Encryption - Step 1

To realize a bootstrappable encryption, is one of the principal purposes to reach the fully homomorphic encryption scheme. Such quality enables to "refresh" the ciphertext recurrently and therefore to reduce the noise parameter. But first it is required to give an formal definition of bootstrappable encryption.

Definition 5 (Bootstrappable Encryption)**.**
Let $\mathcal{C}_{\mathcal{E}}$ be a set of circuits with respect to which $\mathcal{E}$ is homomorphic. And let Γ be a set of gates with inputs and output in plaintext space $\mathcal{P}$ including the trivial gate (input and output are the same). We say that $\mathcal{E}$ is bootstrappable with respect to Γ if $D_{\mathcal{E}}(\Gamma) \subseteq \mathcal{C}_{\mathcal{E}}$. Where $D_{\mathcal{E}}$ is the decryption circuit of $\mathcal{E}$.

To realize the bootstrappability, the scheme $\mathcal{E}$ must be able compactly evaluate the augmented decryption circuit $D_{\mathcal{E}}$. Which is defined as follows:

Definition 6 (Augmented Decryption Circuit)**.**
Let $D_{\mathcal{E}}$ be $\mathcal{E}$'s decryption circuit, which takes a secret key and ciphertext as input, each formatted as an element of $\mathcal{P}^{l(\lambda)}$, where $\mathcal{P}$ is the plaintext space. Let Γ be a set of gates with input and output in $\mathcal{P}$, which includes the trivial gate. We call a circuit composed of multiple copies of $D_{\mathcal{E}}$ connected by a single g gate (the number of copies equals the number of inputs to g) a "g-augmented decryption circuit". We denote the set of g-augmented decryption circuits, $g \in \Gamma$, by $D_{\mathcal{E}}(\Gamma)$.

As soon as $\mathcal{E}$ can handle the augmented decryption, it can be used to construct an efficient scheme, that computes circuits of arbitrary depth. To present this construction, the following theorem is required.

Theorem 1. *Let $\mathcal{E}$ be bootstrappable with respect to a set of gates Γ. Then the family $\{\mathcal{E}^{(d)}\}$ is leveled fully homomorphic (for circuits with gates in Γ).*

This theorem states that the scheme $\mathcal{E}$ works correctly for circuits of depth $d \cdot m$, where each gate $g \in \Gamma$ is an arbitrary circuit of depth m. Now it is

possible to give an formal definition of a leveled fully homomorphic encryption scheme $\mathcal{E}^{(d)}$ =(**KeyGen**$_{\mathcal{E}^{(d)}}$, **Encrypt**$_{\mathcal{E}^{(d)}}$, **Evaluate**$_{\mathcal{E}^{(d)}}$, **Decrypt**$_{\mathcal{E}^{(d)}}$) which consists of four algorithms. Where $\mathcal{E}^{(d)}$ is bootstrappable with respect to the gates $g \in \Gamma$ and $\mathcal{E}^{(d)}$ evaluates all circuits of depth d with gates in Γ.
For simplifying the description of **Evaluate**, the secret keys are given in reverse order.

KeyGen$_{\mathcal{E}^{(d)}}(\lambda, d)$ takes as input the security parameter λ and $d \in \mathbb{Z}^+$. For $l = l(\lambda)$ as in definition 6, it sets

$$\begin{aligned} (\mathrm{sk}_i,\mathrm{pk}_i) &\xleftarrow{R} \textbf{KeyGen}_{\mathcal{E}}(\lambda) \quad \text{for } i \in [0,d] \\ \overline{\mathrm{sk}_{ij}} &\xleftarrow{R} \textbf{Encrypt}_{\mathcal{E}}(\mathrm{pk}_{i-1},\mathrm{sk}_{ij}) \quad \text{for } i \in [1,d], j \in [1,l] \end{aligned}$$

Where $\mathrm{sk}_{i1}, \ldots, \mathrm{sk}_{il}$ is the representation of sk_i as elements of $\mathcal{P}$. It outputs the secret key $\mathrm{sk}^{(d)} \leftarrow \mathrm{sk}_0$ and the public key $\mathrm{pk}^{(d)} \leftarrow \left(\langle \mathrm{pk}_i \rangle, \langle \overline{\mathrm{sk}_{ij}} \rangle\right)$. At which $\overline{\mathrm{sk}_{ij}}$ is an encryption of the j-th bit of the i-th secret key sk_i. Let $\mathcal{E}^{(\delta)}$ refer to the sub-system that uses $\mathrm{sk}^{(\delta)} \leftarrow \mathrm{sk}_0$ and $\mathrm{pk}^{(\delta)} \leftarrow \left(\langle \mathrm{pk}_i \rangle_{i\in[0,\delta]}, \langle \overline{\mathrm{sk}_{ij}} \rangle_{i\in[1,\delta]}\right)$ for $\delta \leq d$.
Encrypt$_{\mathcal{E}^{(d)}}(\mathrm{pk}^{(d)}, \pi)$ takes as input a public key $\mathrm{pk}^{(d)}$ and a plaintext $\pi \in \mathcal{P}$. It outputs a ciphertext $\psi \xleftarrow{R}$ **Encrypt**$_{\mathcal{E}}(\mathrm{pk}_d, \pi)$.
Decrypt$_{\mathcal{E}^{(d)}}(\mathrm{sk}^{(d)}, \psi)$ takes as input a secret key $\mathrm{sk}^{(d)}$ and a ciphertext ψ (which should be an encryption under pk_0). It outputs **Decrypt**$_{\mathcal{E}}(\mathrm{sk}_0, \psi)$.
Evaluate$_{\mathcal{E}^{(d)}}(\mathrm{pk}^{(\delta)}, C_\delta, \Psi_\delta)$. Takes as input a public key $\mathrm{pk}^{(\delta)}$, a circuit C_δ of depth at most δ with gates in Γ, and a tuple of input ciphertexts Ψ_δ (where each input ciphertext should be under pk_δ). With the assumption that each wire C_δ connects gates at consecutive levels; if not, add trivial gates to make it so. If $\delta = 0$, it outputs Ψ_0 and terminates. Otherwise, it does the following:

- Sets $(C^\dagger_{\delta-1}, \Psi^\dagger_{\delta-1}) \leftarrow$ **Augment**$_{\mathcal{E}^{(\delta)}}(\mathrm{pk}^{(\delta)}, \mathrm{C}_\delta, \Psi_\delta)$.
- Sets $(C_{\delta-1}, \Psi_{\delta-1}) \leftarrow$ **Reduce**$_{\mathcal{E}^{(\delta-1)}}(\mathrm{pk}^{(\delta-1)}, C^\dagger_{\delta-1}, \Psi^\dagger_{\delta-1})$.
- Runs **Evaluate**$_{\mathcal{E}^{(\delta-1)}}(\mathrm{pk}^{(\delta-1)}, C_{\delta-1}, \Psi_{\delta-1})$.

Augment$_{\mathcal{E}^{(\delta)}}(\mathrm{pk}^{(\delta)}, \mathrm{C}_\delta, \Psi_\delta)$. Takes as input a public key $\mathrm{pk}^{(\delta)}$, a circuit C_δ of depth at most δ with gates in Γ, and a tuple of input ciphertexts Ψ_δ (where each input ciphertext should be under pk_δ). It augments C_δ with $D_\mathcal{E}$; call the resulting circuit $C^\dagger_{\delta-1}$. Let $\Psi^\dagger_{\delta-1}$ be the tuple of ciphertexts formed by replacing each input ciphertext $\psi \in \Psi_\delta$ by the tuple $\left\langle \left\langle \overline{\mathrm{sk}_{\delta j}} \right\rangle, \left\langle \overline{\psi_j} \right\rangle \right\rangle$, where $\overline{\psi_j} \leftarrow$ **WeakEncrypt**$_{\mathcal{E}^{(\delta-1)}}(\mathrm{pk}^{(\delta-1)}, \psi_j)$ (where image of **WeakEncrypt** is always a subset of the image of **Encrypt**) and the ψ's from the properly formatted representation of ψ as elements of $\mathcal{P}$. It outputs $(C^\dagger_{\delta-1}, \Psi^\dagger_{\delta-1})$.
Reduce$^{(\delta)}_\mathcal{E}(\mathrm{pk}^{(\delta)}, C_\delta, \Psi_\delta)$ Takes as input a public key $\mathrm{pk}^{(\delta)}$, a tuple of input ciphertexts $\Psi^\dagger_\delta$ (where each input ciphertext should be in the image of **Encrypt**$_{\mathcal{E}^{(\delta)}}$) and a circuit $C^\dagger_\delta \in D_\mathcal{E}(\Gamma, \delta+1)$. It sets C_δ to be the sub-circuit of $C^\dagger_\delta$ consisting of the first δ levels. It sets Ψ_δ to be the induced input ciphertexts of C_δ, where the ciphertext $\psi^{(w)}_\delta$ associated to wire w at level δ is set to **Evaluate**$_\mathcal{E}(\mathrm{pk}_\delta, C^{(w)}_\delta, \Psi^{(w)}_\delta)$ where $C^{(w)}_\delta$ is the sub-circuit of $C^\dagger_\delta$ with output wire w, and $\Psi^{(w)}_\delta$ are the input ciphertexts for $C^{(w)}_\delta$. It outputs (C_δ, Ψ_δ).

The presented scheme evaluates gates $g \in \Gamma$ that are augmented by the decryption circuit only. The next step is to create a scheme that is bootstrappable with respect to the universal set of gates $g \in \Gamma$.

4 Initial Construction - Step 2

The aim of this section is to maximize the "evaluative capacity". The security of the earlier homomorphic encryption schemes is based on the "Ideal Membership Problem", like "Polly Cracker" scheme by the Fellows and Koblitz [11]. But the problem is that the homomorphic operations like multiplication, expand the ciphertext-size potentially exponentially in the depth. The new approach is now to base the security on an "Ideal Coset Problem" (ICP). The ICP will first defined in term of rings and ideals, which are used to minimize the circuit complexity.

Definition 7 (Ideal Coset Problem (ICP))**.**
Fix ring R, basis of the ideal I B_I, algorithm ***IdealGen****, and an algorithm* $\boldsymbol{Samp}_1$ *that efficiently samples R. The challenger sets $b \xleftarrow{R} \{0,1\}$ and ($\boldsymbol{B}_J^{sk}$, $\boldsymbol{B}_J^{pk}$) $\xleftarrow{R}$ **IdealGen**(R, $\boldsymbol{B}_I$). If $b = 0$, it sets $\boldsymbol{r} \xleftarrow{R} \boldsymbol{Samp}_1(R)$ and $\boldsymbol{t} \leftarrow \boldsymbol{r}$ mod $\boldsymbol{B}_J^{pk}$. If $b = 1$, it samples $\boldsymbol{t}$ uniformly from R mod $\boldsymbol{B}_J^{pk}$. The problem: guess b given ($\boldsymbol{t}$, $\boldsymbol{B}_J^{pk}$).*

So the ICP requires a decision whether $\mathbf{t}$ is uniform modulo J, or whether it was chosen according to a known "clumpier" distribution induced by $\mathbf{Samp}_1$. The initial scheme $\mathcal{E}$ uses a fixed ring R, which is adjusted related to the security parameter λ, a fixed basis $\mathbf{B}_I$ of an ideal $I \subset R$, and an algorithm ($\mathbf{B}_J^{sk}$, $\mathbf{B}_J^{pk}$) $\xleftarrow{R}$ $\mathbf{IdealGen}(R, \mathbf{B}_I)$ with the output of a secret basis $\mathbf{B}_J^{sk}$ and a public basis $\mathbf{B}_J^{pk}$, where J is a variable ideal but relatively prime to the ideal I. It is sufficient, if the secret basis $\mathbf{B}_J^{sk}$ defines a lattice $\mathcal{L}(\mathrm{B}_J^{sk})$ for a possibly fractional ideal that contains J, rather then being exactly J. The assumption for the initial scheme $\mathcal{E}$ is, if $\mathbf{t} \in R$ and $\mathbf{B}_M$ is a Basis for ideal $M \subset R$, then the value $\mathbf{t}$ mod $\mathbf{B}_M$ is unique and can be computed efficiently. The algorithm $\mathbf{Samp}(\mathbf{B}_M, \mathbf{x})$ samples from the coset $\mathbf{x}+I$.

$\mathbf{KeyGen}(R, \mathbf{B}_I)$. Takes as input a ring R and basis $\mathbf{B}_I$ of an ideal I. It

sets $(\mathbf{B}_J^{sk}, \mathbf{B}_J^{pk}) \xleftarrow{R} \mathbf{IdealGen}(R, \mathbf{B}_I)$. The plaintext space $\mathcal{P}$ is (a subset of) R mod $\mathbf{B}_I$. The public key pk includes R, $\mathbf{B}_I$, B_J^{pk}, and **Samp**. The secret key sk also includes B_J^{sk}.

Encrypt(pk, π). Takes as input the public key pk and plaintext $\pi \in \mathcal{P}$. It sets $\psi' \leftarrow \mathbf{Samp}(\mathbf{B}_I, \pi)$ and outputs $\psi \leftarrow \psi'$ mod $\mathbf{B}_J^{pk}$.

Decrypt(sk, ψ). Takes as input the secret key sk and a ciphertext ψ. It outputs $\pi \leftarrow (\psi \bmod \mathrm{B}_J^{sk}) \bmod \mathbf{B}_I$.

Evaluate(pk, C, Ψ). Takes as input the public key pk, a circuit C in some permitted set $\mathcal{C}_{\mathcal{E}}$ of circuits composed of $\mathbf{Add}_{B_I}$ and $\mathbf{Mult}_{B_I}$ gates and a set of input ciphertexts Ψ. It invokes **Add** and **Mult**, given below, in the proper sequence to compute the output ciphertext ψ. (If desired, one could use different arithmetic gates.)

Add(pk, ψ_1, ψ_2). Outputs $\psi_1 + \psi_2$ mod $\mathbf{B}_J^{pk}$.

Mult(pk, ψ_1, ψ_2). Outputs $\psi_1 \times \psi_2$ mod $\mathbf{B}_J^{pk}$.

It is possible to proof that the abstract scheme $\mathcal{E}$ works correct for a certain set of circuit, called "permitted circuits", which are defined as follows:

Definition 8 (Permitted circuits)**.**
Let $\mathcal{C}'_{\mathcal{E}}$=$\{C : \forall(x_1, \ldots, x_t) \in X_{Enc}^t, g(C)(x_1, \ldots, x_t) \in X_{Dec}\}$

Where X_{Enc} is:

Definition 9 (X_{Enc} and X_{Dec})**.**
Let X_{Enc} *be the image of* ***Samp****. Notice that all ciphertexts output by* ***Encrypt*** *are in* $X_{Enc} + J$*. Let* X_{Dec} *equal* R *mod* $\boldsymbol{B}_J^{sk}$*, the set of distinguished representatives of cosets of* J *wrt the secret basis* $\boldsymbol{B}_J^{sk}$*.*

And $g(C)$ is:

Definition 10 (Generalized Circuit)**.**
Let C be a mod- $\boldsymbol{B}_I$ circuit. We say generalized circuit $g(C)$ of C is the circuit formed by replacing C's $\boldsymbol{Add}_{B_I}$ $\boldsymbol{Mult}_{B_I}$ operations with addition '$+$' and multiplication '$\times$' in the ring R.

The aim is now to maximize the set of permitted circuits. For this purpose the mathematical knowledge is assumed. To respond to the mathematical issue goes beyond the scope of this work. Only the central term of "Lattice" is defined to understand the following argument.

Definition 11 (Lattice)**.**
An n-dimensional lattice of rank $k \leq n$ is L=$\mathcal{L}(\boldsymbol{B})$= $\{\boldsymbol{Bc} : \boldsymbol{c} \in \mathbb{Z}^k\}$, $\boldsymbol{B} \in \mathbb{R}^{n \times k}$

Where the k columns $b_1, \ldots, b_k \in \mathbb{R}^n$ of the basis are linearly independent. All lattices in this work are full rank - i.e., $k = n$. The Hermite normal form of a lattice L HNF(L) qualifies for being a public key, by the reason of HNF(L) is identical with an upper triangular basis that can be efficiently computable from any other basis of L.

Definition 12 (Ideal Lattice)**.**
Let $R = \mathbb{Z}[x]/(f(x))$ be a ring, where $f(x)$ is a monic polynomial of degree n. The ideal $(\boldsymbol{v})$ generated by $\boldsymbol{v}$ directly corresponds to the lattice generated by the column vectors

$$\left\{\boldsymbol{v}_i \leftarrow \boldsymbol{v} \times x^i \mod f(x) : i \in [0, n-1]\right\}.$$

We call this the rotation basis of the ideal lattice $(\boldsymbol{v})$.

The main reason to prefer ideal lattices instead of ideals over general rings, is that lattices provides the opportunity of a clean analysis of X_{Enc} and X_{Dec} in terms of Euclidean length. The implementation of the abstract scheme $\mathcal{E}$ using a polynomial rings $R = \mathbb{Z}[x]\backslash(f(x))$ where $f(x) \in \mathbb{Z}[x]$ is a monic polynomial of degree n, and a ideal lattice, the sets X_{Enc} and X_{Dec} become subsets of $\mathbb{Z}^n$. These two sets X_{Enc} and X_{Dec} can also be defined geometrically as follows:

Definition 13 (r_{Enc} and r_{Dec})**.**
Let r_{Enc} be the smallest value such that $X_{Enc} \subseteq \mathcal{B}(r_{Enc})$, where $\mathcal{B}(r)$ is the ball of the radius r. Let r_{Dec} be the largest value such that $X_{Dec} \supseteq \mathcal{B}(r_{Dec})$.

Now it is possible to re-define the set of permitted circuits $\mathcal{C}'_{\mathcal{E}}$ as follows:

Definition 14 (Permitted Circuits)**.**
$\mathcal{C}_{\mathcal{E}} = \{C : \forall(x_1, \ldots, x_t) \in \mathcal{B}(r_{Enc})^t, g(C)(x_1, \ldots, x_t) \in \mathcal{B}(r_{Dec})\}$.

So X_{Enc} and X_{Dec} are replaced by $\mathcal{B}(r_{Enc})$ and $\mathcal{B}(r_{Dec})$, therefore applies $\mathcal{C}_{\mathcal{E}} \subseteq \mathcal{C}'_{\mathcal{E}}$. Limiting the set of permitted Circuits $\mathcal{C}_{\mathcal{E}}$ relieve the the complexity of the decryption algorithm. For fixed r_{Enc} and r_{Dec} it is possible to bound $\mathcal{C}_{\mathcal{E}}$ further, via using the Euclidean length for generalized circuit $\|g(C)(x_1, \ldots, x_t)\|$. In terms of $\|\mathbf{u}\|$ and $\|\mathbf{v}\|$, for addition the Euclidean length is bounded by using the triangle inequality $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$, where $\mathbf{u}, \mathbf{v} \in R$. For the multiplication it is possible to proof the following, $\|\mathbf{u} \times \mathbf{v}\| \leq \gamma_{Mult}(R) \cdot \|\mathbf{u}\| \cdot \|\mathbf{v}\|$, where γ_{Mult} is an expansion factor which depends only on the ring R. The next theorem reveal the depth, what the scheme $\mathcal{E}$ can handles.

Theorem 2. *Suppose $r_E \leq 1$ and that circuit C's additive fan-in is $\gamma_{Mult}(R)$, multiplicative fan-in is 2, and depth is at most*

$$\log \log_D - \log \log \left(\gamma_{Mult}(R) \cdot r_E\right)$$

Then, $C(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t) \in \mathcal{B}(r_D)$ for all $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t \in \mathcal{B}(r_E)$.

So $\mathcal{E}$ correctly evaluates a circuit C up to a depth of $\log \log_D - \log \log \left(\gamma_{Mult}(R) \cdot r_E\right)$. In order to maximize the depth of the circuit C, $\gamma_{Mult}(R)$ and r_{Enc} should be minimized and r_{Dec} should be maximized.
In the process $\frac{r_{Dec}}{r_{Enc}}$ is bounded by the semantic security. Therefore $r_{Dec} < \lambda_1(J)$, than later it will be the shortest non zero vector in J. Furthermore r_{Dec} depends on the secret basis, than it defines the radius of the largest sphere that is circumscribed by $\mathbf{B}_J^{sk}$. Thereby is $\mathcal{P}(\mathbf{B}_J^{sk})$ is an parallelepiped that is to maximize.

Theorem 3. *Let $\boldsymbol{B}$ be a lattice basis and $\boldsymbol{B}^* = (\boldsymbol{B}^{-1})^T$. Let r be the radius of the largest sphere, centered at 0, circumscribed by $\mathcal{P}(\boldsymbol{B})$ (permitting tangential overlap). Then, $r = 1/(2 \cdot \|\boldsymbol{B}^*\|)$. In particular:*

$$r_{Dec} = 1/\left(2 \cdot \left\| \left(\left(\boldsymbol{B}_J^{sk} \right)^{-1} \right)^T \right\| \right)$$

Suppose $\|t\| < r$; then each coefficient of $\boldsymbol{B}^{-1} \cdot \boldsymbol{t}$ has magnitude at most $1/2$.

Regarding r_{Enc}, so $\lambda_1(J)/r_{Enc}$ should not exceeds 2^n. Otherwise the lattice reduction techniques allow to recover the J-vector closet to $\mathbf{t}$, which breaks the ICP. So it is agreed that r_{Enc} should be polynomial in n. The same applies for γ_{Mult}. Also γ_{Mult} should be polynomial in n, what the following theorem states:

Theorem 4. *Let $R = \mathbb{Z}[x]/f(x)$ and suppose $f(x) = x^n - h(x)$ where $h(x)$ has degree at most $n - (n-1)/k$ for $k \geq 2$. Then, $\gamma_{Mult}(R) \leq \sqrt{2n} \cdot \left(1 + 2n \cdot \left(\sqrt{(k-1)\,n}\,\|f\|\right)^k\right)$.*

In further two tweaks will be presented. The first one reduce the secret key size and therefore reduce the computational complexity of the decryption circuit, to enables the bootstrappability . It computes from $\mathbf{B}_I$ and $\mathbf{B}_J^{sk}$ a short vector $\mathbf{v}_J^{sk} \in J^{-1}$ and redefine decryption to output $\pi = \psi - \lfloor \mathbf{v}_J^{sk} \times \psi \rceil \mod \mathbf{B}_I$. And redefine the set of permitted circuits $\mathcal{C}_\mathcal{E}$ as follows:

Definition 15 (Permitted Circuits)**.**
$\mathcal{C}_\mathcal{E} = \{C : \forall (x_1, \ldots, x_t) \in \mathcal{B}(r_{Enc})^t, g(C)(x_1, \ldots, x_t) \in \mathcal{B}(r_{Dec}/n^{2.5} \cdot \|f\| \cdot \|\boldsymbol{B}_I\|)\}$.

Furthermore the first tweak makes the already shallow decryption circuit less wide. The permitted distance of ciphertexts from lattice J needs to be reduce. But it does not affect the maximum evaluation depth very much when $|f|$ and $\|\mathbf{B}_I\|$ are only polynomial in n, and r_{Dec}/r_{Enc} is superpolynomial (as it will need to be to make the scheme bootstrappable.)
Now it is possible to formulate the following lemma:

Lemma 1. *Let $\boldsymbol{B}_J^{sk}$ be an initial secret basis that decrypts correctly for parameter r_{Dec}. From $\boldsymbol{B}_J^{sk}$ and $\boldsymbol{B}_I$, we can compute in polynomial time a vector $\boldsymbol{v}_J^{sk} \in J^{-1}$ such that the rotation basis of $1/\boldsymbol{v}_J^{sk}$ circumscribes a ball of radius at least $r_{Dec}/(n^{2.5} \cdot \|f\| \cdot \|\boldsymbol{B}_I\|)$. In particular, if ψ is a valid ciphertext according Tweak 1, in the sense that it equals $\pi + i + j$ for plaintext π, $i \in I$, $j \in J$ and $\pi + i \in \mathcal{B}(r_{Dec}/n^{2.5} \cdot \|f\| \cdot \|\boldsymbol{B}_I\|)$, then $\pi = \psi - (\boldsymbol{v}_J^{sk})^{-1} \times \lfloor \boldsymbol{v}_J^{sk} \times \psi \rceil \mod \boldsymbol{B}_I$. For our particular value of $\boldsymbol{v}_J^{sk} \in J^{-1}$, it will also hold that $\pi = \psi - \lfloor \boldsymbol{v}_J^{sk} \times \psi \rceil \mod \boldsymbol{B}_I$.*

The second tweak redefines $\mathcal{C}_{\mathcal{E}}$ again by replacing $\mathcal{B}(r_{Dec})$ with $\mathcal{B}(r_{Dec}/2)$, in order to place the ciphertext vectors closer to the lattice J to reduce the restrictions on the correctness of the decryption. After the second tweak the following lemma can be formulated:

Lemma 2. *If ψ is a valid ciphertext after Tweak 2, then each coefficient of $(\boldsymbol{B}_J^{sk2})^{-1} \cdot \psi$ is within $1/4$ of an integer.*

So the new maximum evaluation depth of the scheme $\mathcal{E}$ after the second tweak is

$$\log\log(r_{Dec}/2) - \log\log(\gamma_{Mult}(R) \cdot r_{Enc}),$$

which is less then the original amount of by only a sub-constant additive factor. Overall the depth of the tweaked decryption circuit was reduced for enabling bootstrapping. However, it will be ascertained after a closer examination of decryption complexity that the bootstrapping is not possible.

During the decryption is

$$(\psi - \mathbf{B}_J^{sk1} \cdot \lfloor \mathbf{B}_J^{sk2} \cdot \psi \rceil) \mod \mathbf{B}_I$$

to compute. Where $\psi \in \mathbb{Z}^n, \mathbf{B}_J^{sk1} \in \mathbb{Z}^{n \times n}, \mathbf{B}_J^{sk2} \in \mathbb{Q}^{n \times n}$, and $\mathbf{B}_I$ is a basis of an ideal I of $R = \mathbb{Z}[x]/(f(x))$. After the second tweak the coefficients of $\mathbf{B}_J^{sk2} \cdot \psi$ are all within $1/4$ of an integer. And after the first tweak is $\mathbf{B}_J^{sk1}$ the

identity matrix and $\mathbf{B}_J^{sk2}$ is a rotation matrix. The computation of

$$(\psi - \mathbf{B}_J^{sk1} \cdot \lfloor \mathbf{B}_J^{sk2} \cdot \psi \rceil) \mod \mathbf{B}_I$$

is splitted into three steps as follows:

Step 1: Generate n vectors $x_1, \ldots, x_n$ with sum $\mathbf{B}_J^{sk2} \cdot \psi$.

Step 2: From the n vectors $x_1, \ldots, x_n$ generate integer vectors $y_1, \ldots, y_{n+1}$ with sum $\lfloor \sum \mathbf{x}_i \rceil$.

Step 3: Compute $\pi \leftarrow \psi - \mathbf{B}_J^{sk1} \cdot (\sum \mathbf{y}_i) \mod \mathbf{B}_I$.

The problem here is the second step by adding $x_1, \ldots, x_n \in [0, 1)$ binary numbers, each one with the precision of k. To compute this numbers is an constant fan-in boolen circuit of the depth $\Omega(\log n + \log k)$ required. After recursive using of the "3-for-2" trick [12], the circuit depth is reduced to $c \cdot \log_{3/2} \cdot n$. But for the last two numbers is a circuit depth of $\Omega(\log k)$ required, because the least significant bit of the addend could impair the most significant bit of the sum. The second tweak gives the promise that the sum is very close to an integer. This integer is sufficient for the further calculation and can be computed in $c \cdot \log_{3/2} \cdot n + O(\log \log n)$ depth. Nonetheless the term $c \cdot \log_{3/2}$ is to high to permit the bootstrappability, than the bootstrappable scheme can handles the depth of $O(\log n)$ only. So it is possible to formulate the following lemma:

Lemma 3. *For $i \in [1, t]$, let $a_i = (\ldots, a_{i,1}, a_{i,0}, a_{i,-1}, \ldots)$ be a real number given in binary representation $\mod \boldsymbol{B}_I$ with the promise that $\sum_i a_i \mod 1 \in [-1/4, 1/4]$. There is a mod-$\boldsymbol{B}_I$ circuit C for generating "t+1" integers $z_1, \ldots, z_{t+1}$ (also represented binary) whose sum is $\lfloor \sum_i a_i \rceil$, such that if the generalized circuit $g(C)$'s inputs are in $\mathcal{B}(r_{in})$, then its outputs are in $\mathcal{B}(r_{out})$ for:*

$$r_{out} \leq \left(\gamma_{Mult}(R) \cdot n \cdot \|\boldsymbol{B}_I\| \cdot (1 + \gamma_{Mult}(R) \cdot r_{in})^t \cdot t\right)^{polylog(t)}$$

For $\|\boldsymbol{B}_I\| \leq r_{in}, t \leq n$, and $\gamma_{Mult}(R) = n^{\Omega(1)}$, we have:

$$r_{out} \leq (\gamma_{Mult}(R) \cdot r_{in})^{t \cdot polylog(t)}.$$

So adding terms represented in general “base-I” is to expensive to permit the bootstrappability. Therefore is squashing the decryption circuit required.

5 Squashing the Decryption Circuit - Step 3

Let $\mathcal{E}^*$ be the encryption scheme described in section 4 include the first and second tweak. The circuit complexity of decryption scheme $\mathcal{E}^*$ is too high. The problem is to add n numbers. In the following the transformation of the scheme $\mathcal{E}^*$ will be presented, where **Decrypt** adds only sub-linear quantity of numbers, thus to low the the complexity of the decryption circuit. This has no influence of the set of permitted circuits. The idea is now to place a hint about the $\mathcal{E}^*$ secret key inside the transformed scheme $\mathcal{E}$ public key. Of cause, has this hint an affect of the security, as more information about the secret key is revealed. First this transformation is described abstract and will filled with details later.

The original decryption will be splitted into two phases. The first one is computationally intensive preprocessing phase, without using the secret key, where encrypter pretreat its own initial ciphertext to leave less work for the decrypter to do. The second one is computationally lightweight with using the secret key, performed by decrypter.

The transformation of the initial encryption scheme $\mathcal{E}^*$ uses two new algorithms, $\mathbf{SplitKey}_{\mathcal{E}}$ and $\mathbf{ExpandCT}_{\mathcal{E}}$, where $\mathcal{E}$ is the abstract modified scheme.

$\mathbf{KeyGen}_{\mathcal{E}}(\lambda)$. Runs $(\mathrm{pk}^*, \mathrm{sk}^*) \xleftarrow{R} \mathbf{KeyGen}_{\mathcal{E}^*}(\lambda)$ and $(\mathrm{sk}, \tau) \xleftarrow{R} \mathbf{SplitKey}_{\mathcal{E}}(\mathrm{sk}^*, \mathrm{pk}^*)$. The secret key is sk. The public key pk is (pk^*, τ). Where τ is a secret key dependent tag.

$\mathbf{Encrypt}_{\mathcal{E}}(\mathrm{pk}, \pi)$. Runs $\psi^* \leftarrow \mathbf{Encrypt}_{\mathcal{E}^*}(\mathrm{pk}^*, \pi)$. It then sets ψ to include $\psi*$ and the output of $\mathbf{ExpandCT}_{\mathcal{E}}(\mathrm{pk}, \psi^*)$. ($\mathbf{ExpandCT}_{\mathcal{E}}$ makes heavy use of τ.)

$\mathbf{Decrypt}_{\mathcal{E}}(\mathrm{sk}, \psi)$. Uses sk and expanded ciphertext to decrypt more efficiently. $\mathbf{Decrypt}_{\mathcal{E}}(\mathrm{sk}, \psi)$ should work whenever $\mathbf{Decrypt}_{\mathcal{E}^*}(\mathrm{sk}^*, \psi^*)$ works.

$\mathbf{Add}_{\mathcal{E}}(\text{pk}, \psi_1, \psi_2)$. Extracts (ψ_1^*, ψ_1^*) from (ψ_1, ψ_2), computes $\psi^* \leftarrow \mathbf{Add}_{\mathcal{E}^*}(\text{pk}^*, \psi_1^*, \psi_2^*)$ and sets ψ to include ψ^* and the output of $\mathbf{ExpandCT}_{\mathcal{E}}(\text{pk}, \psi^*)$.

$\mathbf{Mult}_{\mathcal{E}}(\text{pk}, \psi_1, \psi_2)$ is analogous to $\mathbf{Add}_{\mathcal{E}}(\text{pk}, \psi_1, \psi_2)$.

The security of the transformation depends on the "**SplitKey** Distinguished Problem", which is defined as follows:

Definition 16 (**SplitKey** Distinguished Problem)**.**
The challenger sets $(sk^*, pk^*) \xleftarrow{R} \boldsymbol{KeyGen}_{\mathcal{E}^*}$ *and* $b \xleftarrow{R} \{0,1\}$. *If* $b = 0$, *it sets* $(sk, \tau) \xleftarrow{R} \boldsymbol{SplitKey}(sk^*, pk^*)$. *If* $b = 1$, *it sets* $(sk, \tau) \xleftarrow{R} \boldsymbol{SplitKey}(\perp, pk^*)$, *where* $\perp$ *is a special symbol. The problem: guess* b *given*(τ, sk^*, pk^*).

Now it is possible to formulate the following theorem about the security of the transformation:

Theorem 5. *Suppose that there is an algorithm* A *that breaks the semantic security of* $\mathcal{E}$ *above with advantage* ϵ. *Then, there exist algorithms* $\mathcal{B}_0$ *and* $\mathcal{B}_1$, *running in about the same time as* A, *such that either* $\mathcal{B}_0$*'s advantage against the* ***SplitKey*** *Distinguishing Problem or* $\mathcal{B}_1$*'s advantage against the semantic security of* $\mathcal{E}^*$ *is at least* $\epsilon/3$.

In the following, the transformation is described in detail. As a reminder:

$$\pi = \psi - \lfloor \mathbf{v}_J^{sk^*} \times \psi \rceil \mod \mathbf{B}_I$$

is the decryption output of the scheme $\mathcal{E}^*$ after the first tweak, where $\mathbf{v}_J^{sk^*}$ is the secret key vector, an element of a fractional ideal J^{-1}. Now it is required to give a hint about this secret key vector $\mathbf{v}_J^{sk^*}$. This hint will be a set of vectors that has a secret sparse subset of vectors whose sum is $\mathbf{v}_J^{sk^*}$. Concretely it will holds:

$$\sum_{i \in S} t_i = \mathbf{v}_J^{sk^*} \mod I.$$

Where $t_1, \ldots, t_{\gamma_{setsize}(n)}) \in J^{-1}$ a set of vectors τ, where $\gamma_{setsize}(n)$ is polynomial in n. And $S \subset \{1, \ldots, \gamma_{setsize}(n)\}$ is a subset of indices. The new secret

key sk is a 0/1-matrix encoding the subset S. The new **SplitKey** distinguishing problem is now: given $\mathbf{v}_J^{sk^*}$ and τ decide whether there is actually a sparse subset whose sum is $\mathbf{v}_J^{sk^*} \mod I$, or whether there is a sparse subset whose sum is $\mathbf{0} \mod I$.

In the **ExpandCT** operation, the encrypter edit the initial ciphertext ψ^*, by computing all of the products $c_i \leftarrow t_i \times \psi^* \mod \mathbf{B}_I$ and including them to ψ. The decrypter extracts $\mathbf{c}_i$ with $i \in S$ and uses the decryption equation:

$$\pi = \psi - \lfloor \sum_{i \in S} \mathbf{c}_i \rceil \mod \mathbf{B}_I.$$

The central point is that summing up $\gamma_{setsize}(n)$ values requires much less depth then $\log(n)$.

Now it is the time to give a formal version of the transformation. Which makes the scheme $\mathcal{E}$ bootstrappable, as it will shown afterwards. Let (sk*, pk*) be an $\mathcal{E}^*$ key pair. Let $\gamma_{setsize}(n)$ and $\gamma_{subsetsize}(n)$ be functions, where the former is $\omega(n)$ and $poly(n)$ and the latter is $\omega(1)$ and $o(n)$. Here are the concrete instantiations of $\mathbf{SplitKey}_{\mathcal{E}}$, $\mathbf{ExpandCT}_{\mathcal{E}}$, and $\mathbf{Decrypt}_{\mathcal{E}}$ used to construct $\mathcal{E}$.

$\mathbf{SplitKey}_{\mathcal{E}}(\text{sk}^\dagger, \text{pk}^*)$. Takes as input $\text{sk}^\dagger$, which may be either sk^* or $\perp$. If the former, it extracts the vector $\mathbf{v}_J^{sk^*}$ from sk^*; if the latter, it sets $\mathbf{v}_J^{sk^*} \leftarrow 0$. It outputs (sk, τ), where:

- τ is a set of $\gamma_{setsize}(n)$ vectors $\mathbf{t}_1, \ldots, t_{\gamma_{setsize}(n)}$that are uniformly random in $J^{-1} \mod \mathbf{B}_I$, except there exists a subset $S \subseteq \{1, \ldots \gamma_{setsize}(n)\}$ of cardinality $\gamma_{subsetsize}(n)$ such that:

$$\sum_{i \in S} \mathbf{t}_i \in \mathbf{v}_J^{sk^*} + I$$

  .

- sk is a matrix $\gamma_{subsetsize}(n) \times \gamma_{setsize}(n)$ matrix M of 0's and 1's, where $M_{ij} = 1$ iff j is the ith member of S.

ExpandCT$_{\mathcal{E}}$(pk, ψ^*). Outputs $\mathbf{c}_i \leftarrow \mathbf{t}_i \times \psi^* \mod \mathbf{B}_I$ for $i \in [1, \gamma_{setsize}(n)]$.

Decrypt$_{\mathcal{E}}$(sk, ψ). Takes as input the secret key sk and a ciphertext ψ. It performs the following steps:

Step 0: Set the vectors $\mathbf{w}_{ij} \leftarrow M_{ij} \cdot \mathbf{c}_j$

Step 1: Set the vectors $\mathbf{x}_i \leftarrow \sum_{j=1}^{\gamma_{setsize}(n)} \mathbf{w}_{ij}$

Step 2: From $\mathbf{x}_1, \ldots, \mathbf{x}_{\gamma_{subsetsize}(n)}$, generate integer vectors $\mathbf{y}_1, \ldots, \mathbf{y}_{\gamma_{subsetsize}(n)+1}$ with sum $\lfloor \sum \mathbf{x}_i \rceil$.

Step 3: Compute $\pi \leftarrow \psi^* - \sum \mathbf{y}_i \mod \mathbf{B}_I$.

The scheme $\mathcal{E}$ is formulated, now it is required to proof the bootstrappability of this scheme. This is the aim of the next section.

6 Bootstrapping for the Decryption Circuit Achieved

In the following the certain steps of $\mathbf{Decrypt}_{\mathcal{E}}$ will be analyzed and afterwards a theorem about the bootstrappability of the scheme $\mathcal{E}$ will be formulated.

The **Step 0**, to set the vectors $\mathbf{w}_{ij} \leftarrow M_{ij} \cdot \mathbf{c}_j$, requires a constant depth.

The **Step 1**, to set the vectors $\mathbf{x}_i \leftarrow \sum_{j=1}^{\gamma_{setsize}(n)} \mathbf{w}_{ij}$, requires also a constant depth, because the set $\{\mathbf{w}_{ij} : j \in [1, \ldots, \gamma_{setsize}(n)]\}$ has only one nonzero vector, therefore no expensive carry operations are required.

Step 2 and **Step 3** are already discussed in section 4. Finally the following theorem can be formulated.

Theorem 6. *The scheme $\mathcal{E}$ is bootstrappable when*

$$\gamma_{subsetsize}(n) \cdot \log^{c_1} \gamma_{subsetsize}(n) \leq \left(\frac{\log(r_{Dec}/m)}{2^{c_2} \cdot \log(\gamma_{Mult}(R) \cdot r_{Enc})} \right)$$

where $\log^{c_1} \gamma_{subsetsize}(n)$ is the polylog term arising in Lemma 3, m arises from the redefinition of $\mathcal{C}_{\mathcal{E}}$ in the Tweaks ($m = 2$ when just Tweak 2 is used), and c_2 is a constant representing the depth needed in a circuit having $\boldsymbol{Add}_{\boldsymbol{B}_I}$ gates with $\gamma_{Mult}(R) = n^{\Omega(1)}$ fan-in and $\boldsymbol{Mult}_{\boldsymbol{B}_I}$ gates with constant fan-in to sequentially perform $\boldsymbol{Decrypt}_{\mathcal{E}}$ Steps 0, 1, and 3, and a NAND gate.

7 Conclusion

This work is a compendium of the thesis from Craig Gentry " A Fully Homomorphic Encryption Scheme". In contrast to his antecessors he constructed a fully homomorphic encryption scheme which can handle a arbitrary depth of circuit. His fundamental idea was to use ideal lattices to reduce the circuit complexity. Afterwards the scheme was modified, by squashing the decryption circuit, to make the encryption scheme bootstrappable. But even after the squashing the decryption circuit, a constant factor of the depth make a huge difference in the performance and security of the scheme. Craig Gentry focus his work on the semantic security. To construct a CCA1-secure ("lunchtime attacks") fully homomorphic encryption scheme, just like to make the fully homomorphic encryption scheme practical, is still an open problem.

References

[1] http://en.wikipedia.org/wiki/RSA_(algorithm)

[2] Crag Gentry. Fully homomorphic encryption using ideal lattices. STOC 2009. pages 169-178

[3] Craig Gentry. A Fully Homomorphic Encryption Scheme (Ph.D. thesis)

[4] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. CryptographyâĂŞPKC 2010. pages 420-443

[5] Craig Gentry. Computing arbitrary functions of encrypted data. http://crypto.stanford.edu/craig/easy-fhe.pdf

[6] Taher Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transaction on information theory, Vol. IT-31, No. 4, July 1985 .http://caislab.kaist.ac.kr/lecture/2010/spring/cs548/basic/B02.pdf

[7] http://en.wikipedia.org/wiki/Goldwasser%E2%80%93Micali_cryptosystem

[8] Dan Boneh, Eu-Jin Goh, Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. TCC 2005. pages 325-341. http://crypto.stanford.edu/~dabo/papers/2dnf.pdf

[9] http://en.wikipedia.org/wiki/Fully_homomorphic_encryption#Fully_homomorphic_encryption

[10] http://www-03.ibm.com/press/us/en/pressrelease/27840.wss#contact

[11] M. Fellows and N. Koblitz. Combinatorial cryptosystems galore! In

Contemporary Mathematics, volume 168 of Finite Fields: Theory, Applications, and Algorithms, FQ2, pages 51-61, 1993

[12] R. Karp. A Survey of parallel Algorithms for Shared Memory Maschines.