On implementation of the Gentry-Halevi somewhat homomorphic scheme

MICHAL MIKUŠ
Slovak University of Technology in Bratislava
Faculty of Electrical Engineering and Information Technology
Department of Applied Informatics and Computing Technology
SLOVAKIA
michal.mikus@stuba.sk

Abstract: We have implemented the somewhat homomorphic scheme from [7]. We examined this scheme in the same way as mentioned in [7] and extend the results for a wider set of parameters. We focused on the dependencies between the largest supported degree and various parameters of the cryptosystem. We show that the growth of the supported degree doesn’t grow linearly with the parameter $t$ and we give an explanation for this fact.

Key–Words: homomorphic encryption, cloud-computing, somewhat homomorphic scheme

1 Introduction

A homomorphic cryptosystem is a cryptosystem that for arbitrary plaintexts $p_1, p_2$ allows a computation of $p_1 \odot p_2$ only from corresponding ciphertexts $c_1 = \text{Enc}(p_1)$ and $c_2 = \text{Enc}(p_2)$ without the necessity of revealing $p_1$ or $p_2$. The operation $\odot$ is usually addition or multiplication and there are well-known examples of such additive or multiplicative homomorphic cryptosystems. A fully homomorphic cryptosystem is one that allows computation of both addition and multiplication.

The existence of a fully homomorphic cryptosystem had been an open problem for more than 30 years [11]. Various approaches were proposed to find a solution to this problem, e.g. [2], [9], however, the problem was first affirmatively solved by C.Gentry in [5], [6] and [7]. Since then many variations of the Gentry’s scheme were created, e.g. [4], [12] and very recently [3].

The main result of Gentry is that a fully homomorphic scheme can be constructed from a ”somewhat homomorphic” scheme, that can perform only a limited number of aditions/multiplications, under the assumption that the number of supported additions and multiplications is high enough. This property of the somewhat homomorphic scheme is called bootstrapability, exact definitions can be found in [6]. In this paper we examine only the somewhat homomorphic scheme (SHS), because its properties influence the effectiveness of the whole fully homomorphic system.

Many questions arise during the careful examination of the SHS in [7]. The most interesting one is how the largest supported degree is influenced by the parameters $N, t$ and $m$, or how many multiplications are supported. We will give some answers in the section 3.

2 Notations and definitions

We use the same notation as in [7], namely for integers $a, d$ the reduction of $a$ modulo $d$ into the interval $[-d/2, d/2)$ is denoted by $[a]_d$. The standard reduction modulo $d$ or polynomial $f(x)$ is denoted by $a \mod d$ or $a(x) \mod f(x)$.

2.1 The Gentry-Halevi cryptosystem

The somewhat homomorphic scheme is defined by five procedures $\text{Keygen}$, $\text{Encrypt}$, $\text{Decrypt}$, $\text{Add}$, $\text{Mul}$. Here we provide only a brief descriptions of them. For further information on the underlying algorithms and detailed implementation we refer the reader to the original article [7]. The plaintext space is $P = \{0, 1\}$, the ciphertexts are numbers of size approx. $2^t$. 

ISBN: 978-1-61804-000-8
\[\text{Keygen}(N, t)\]
- set \( f(x) = x^N + 1 \)
- choose a polynomial \( v(x) \) of degree \((N - 1)\), with a \( t \)-bit coefficients, s.t. \( v(x) \) and \( f(x) \) have a single root \( r \) in common
- compute \( w(x) \) s.t. \( w(x)v(x) \equiv d \mod f(x) \), where \( d = \text{resultant}(f(x), v(x)) \)
- output \( PK = (N, t, d, r) \) and \( SK = (w_{i_0}) \), where \( w_{i_0} \) is some odd coefficient of \( w(x) \)

\[\text{Encrypt}(PK, b, q)\]
- choose random \( u(x) \in \mathbb{Z}[x] \) of degree \((N - 1)\)
  where \( u_i = \pm 1 \) with probability \((1 - q)\)
- set \( c(x) = m + 2 \cdot u(x) \)
- output \( c = [c(r)]_d \)

\[\text{Decrypt}(SK, c)\]
- output \( m = [c \cdot w_{i_0}]_d \)

\[\text{Add}(PK, c_1, c_2)\]
- output \( c = [c_1 + c_2]_d \)

\[\text{Mul}(PK, c_1, c_2)\]
- output \( c = [c_1 \cdot c_2]_d \)

### 3 Experiments and Results

In this section we describe two groups of experiments. Accordingly to [7] we fixed the dimension \( N = 128 \) and the probability \( q = 1 - 20/N \) in both groups, so that approximately 20 coefficients of the \( c(x) \) are non-zero and the obtained results are independent from the chosen dimension \( N \).

#### 3.1 Supported number of multiplications

In the first group of experiments we examined the dependency of the supported number of multiplications on the parameter \( t \) – the bit-length of coefficients of \( v(x) \).

In each experiment we generated an instance of the cryptosystem (with \( N = 128 \) and various settings of \( t \)) and a sufficient number of PT-CT pairs. To analyze the number of multiplications that can be performed correctly we were multiplying an increasing number of ciphertexts and comparing the decrypted result to the corresponding product of plaintexts. The process stopped on the first number \( n_a \) that produced error and the supported number of multiplications was then \( n_a - 1 \).

As the encryption is a randomized process, the experiment was repeated 30 times to get statistically more significant results. The results are displayed on fig. 1 and only confirm the hypothesis that the homomorphic operation \( \text{Mul} \) multiplies the error introduced by encryption. The decryption of the scheme is correct if the error does not exceed the boundary \( 2^t \) and therefore the supported number of multiplications was expected to grow linearly in \( t \).

The results on the Figure1 clearly show the linear dependence, but also the statistical deviation, because on \( t = 196 \) the degree decreased to a value 64. This is due to a "lucky" choice of ciphertexts with the parameter \( t = 192 \), where we were able to correctly compute product of 65 ciphertexts.

#### 3.2 The largest supported degree

The next group of experiments focused on evaluation of arbitrary polynomials. Namely, we fixed a number of variables of a polynomial and then evaluated every elementary symmetric polynomial up to some degree. The largest supported degree for the parameter setting \((N, t)\) denotes the number \( \text{ltd} \), for which every tested elementary symmetric polynomial on \( m \) variables was evaluated correctly on the ciphertexts. To get more relevant results, this experiment was also repeated 30 times for every parameter setting.

These experiments were executed for dimension \( N = 128 \) and number of variables \( m = 64, 80 \) and 96, results are displayed on Figure2. In contrary to the previous experiments, the expectations are not so straightforward. The
elementary symmetric polynomials contain \( \binom{m}{\deg} \) monomials of degree \( \deg \). After the evaluation of the monomials, we expect each of them to have an error of size approximately \( c^{\deg} \) for some (unknown) constant \( c \). After the summation the error for the \( \text{lsd} \)-degree polynomial should be less than \( 2^t \).

As the \( \text{lsd} \) of a polynomial is bounded by the number of variables \( m \), the lines for \( \text{lsd} \) must remain constant after reaching the value \( m \). The results for \( m = 64 \) also show a slight change of the slope at \( \text{lsd} \approx 32 \) – which is exactly the point where the \( \binom{m}{\deg} \) starts to decrease and the same can be observed for \( m = 80 \). We conclude that the \( \text{lsd} \)-value is not linear in \( t \) (which would occur if the multiplication error would prevail over the addition), but depends also on the number of additions – in this case the ratio of \( m : \deg \).
3.3 Open questions

The other interesting parameter is the probability $q$ used to generate the “noise” polynomial. It is clear that this parameter influences the initial error (higher $q$ yields lower error), but also the security of the scheme.

Acknowledgements: The research was supported by the VEGA Grant (grant No. 1/0244/09).

References: