VERSITA

TATRA MOUNTAINS
Mathematical Publications
DOI: 10.2478/v10127-012-0044-6
Tatra Mt. Math. Publ. **53** (2012), 147–154

EXPERIMENTS WITH THE PLAINTEXT SPACE IN GENTRY'S SOMEWHAT HOMOMORPHIC SCHEME

Michal Mikuš

ABSTRACT. In this paper we propose an improvement of the implementation of the original Gentry-Halevi somewhat homomorphic scheme. We suggest to choose a bigger plaintext space, by changing the underlying ideal from $I = (2)$ to $I = (p)$ for some bigger prime p.

Our analysis shows that bigger plaintext space will improve the homomorphic computation of the somewhat homomorphic scheme while it only slightly increases the complexity of the key generation procedure. The encryption and decryption functions have the same complexity. We provide also some experimental computations that support the analysis.

1. Introduction

The area of homomorphic cryptosystems has been extensively studied in the recent years. The beginnings are due to [16] in 1978, followed by [2]. Other important papers include [3], [4], [5], [15], but the main reason behind the increased interest is the work of Craig Gentry from 2009 [8], [9] that showed a promising direction of research.

Further publications mostly follow the Gentry's framework, firstly somewhat homomorphic scheme (SHS) by [17], then a simplified (integer) version of cryptosystem by [7]. Both schemes were not effective enough to permit bootstrapping and thus could not be turned into a fully homomorphic scheme (FHS). The latest results [6], [11] and [12] are fully homomorphic schemes that can potentially perform unlimited number of homomorphic operations, but their complexities are still too big for practical purposes.

2010 Mathematics Subject Classification: 62K05.
Keywords: homomorphic encryption, efficient homomorphic computing, lattice based cryptography.
This work was supported by the grants APVV-0513-10 and APVV-0586-11.

Our contribution. The main idea of this paper is to adjust the Gentry-Halevi scheme [11] so that it has a bigger plaintext space and show a simple way how to use it for practical computations. The adjustment is done, however, only on the somewhat homomorphic scheme, and there is probably no way how to extend this scheme to a fully homomorphic one.

2. Gentry-Halevi somewhat homomorphic scheme

The SHS that was proposed by Gentry is based on lattices, it was inspired by works of Goldreich et al. [10], Ajtai, Dwork [1], Micciancio and Lyubashevsky [13], [14].

The basic somewhat homomorphic scheme ξ is defined by four polynomial time algorithms $Keygen()$, $Encrypt()$, $Decrypt()$ and $Eval()$. It is based on the ring of integer polynomials $R = \mathbf{Z}[x]/\big(f(x)\big)$ and two ideals $I, J \subseteq R$, that can be also viewed as lattices. The ideal I defines the plaintext space and was chosen $I = (2)$ in [11], while two different ("bad" and "good") bases of the J ideal form public and private key of the scheme.

The idea of the SHS is to encode the plaintext as a small error vector and add it to some random point of the lattice. The decryption algorithm needs to solve the closest vector problem and that is possible only with some "good" basis, that has nearly orthogonal vectors. Solving the closest vector problem for an arbitrary basis is NP-hard problem.

The paper [11] contains a very detailed algorithms for effective key generation, encryption and decryption procedures.

2.1. Adjustment to the scheme

Our proposal is to enlarge the plaintext space of the scheme. As it was stated in [11] the ideal I only has to be *relatively prime* to J. As the source codes used to generate public challenges to this cryptosystem were optimized to bit-operations (i.e., the setting $I = (2)$), we changed the cryptosystem so that it would work with arbitrary $I = (p)$ for some prime p, so that the plaintext space is $\mathbf{Z}_p$ at the cost of decreased performance.

This change can be easily implemented into the original scheme, as the operations with the plaintext space affect only a few steps in the original algorithms. The modular divisions are simply extended to p and the only non-trivial step is the fourth step in the key generation algorithm. Here, the original condition was to find an odd coefficient w_{i_0} of some secret-key polynomial $w(x)$. Further examination of the decryption algorithm leads to a more precise formulation $w_{i_0} \equiv 1 \bmod 2$, so the condition in key generation algorithm was extended to $w_{i_0} \equiv 1 \bmod p$.

In the following we describe the modified algorithms of the scheme. The parameters (p, N, t) are inputs to the scheme and represent plaintext space, lattice dimension and the bitsize of the coefficients of vectors/polynomials. The real number q controls the amount of "noise" added to ciphertexts during the encryption and is usually chosen as $q = 1 - 20/N$. Variables m and c represent some plaintext and ciphertext, respectively.

For any two integers a, d by $[a]_d$ we denote modular reduction into interval $\langle -d/2, d/2)$.

$Keygen(p, N, t)$:

(1) set $f(x) = x^N + 1$,

(2) choose a random 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,

(3) compute $w(x)$ s.t. $w(x)v(x) \equiv d \bmod f(x)$, where $d = resultant\big(f(x), v(x)\big)$,

(4) output $PK = (p, N, t, d, r)$ and $SK = (w_{i_0})$, where $w_{i_0} \equiv 1 \bmod p$ is some coefficient of $w(x)$.

$Encrypt(PK, p, m, q)$:

(1) choose random $u(x) \in \mathbf{Z}[x]$ of degree $(N-1)$, where $u_i = \pm 1$ with probability $(1-q)$ and $u_i = 0$ with probability q,

(2) set $c(x) = m + p \cdot u(x)$,

(3) output $c = \big[c(r)\big]_d$.

$Decrypt(SK, p, c)$:

(1) $m = [c \cdot w_{i_0}]_d$,

(2) output $m \bmod p$.

The $Eval(PK, \circ, c_1, c_2)$ algorithm simply computes $[c_1 \circ c_2]_d$, where "$\circ$" is either addition or multiplication. We use it without modification.

2.2. Efficiency

The key generation. It can be seen from the codes above that only the last step—computation of w_{i_0}—is affected. Since we now search for coefficient of $w(x)$ that is equal to 1 modulo p, the average number of tries has grown from 2 to p. As the complexity of the procedure was $O(N)$ multiplications, under the assumption that $p = O(N)$ the asymptotic complexity will remain the same. Indeed the assumption $p = O(N)$ is necessary, because the average number of coefficients of $w(x)$ with a specific remainder modulo p is N/p, so we need $p \leq N$ in fact.

Encryption. Only the third step of the Encrypt procedure is changed. In the original scheme the multiply-by-2 operation was implemented by the right shift of bits, in the suggested modification it is transformed into a regular multiplication. The costly operation in the procedure, however, is the evaluation of the polynomial $u(r)$, hence, the overall complexity remains the same.

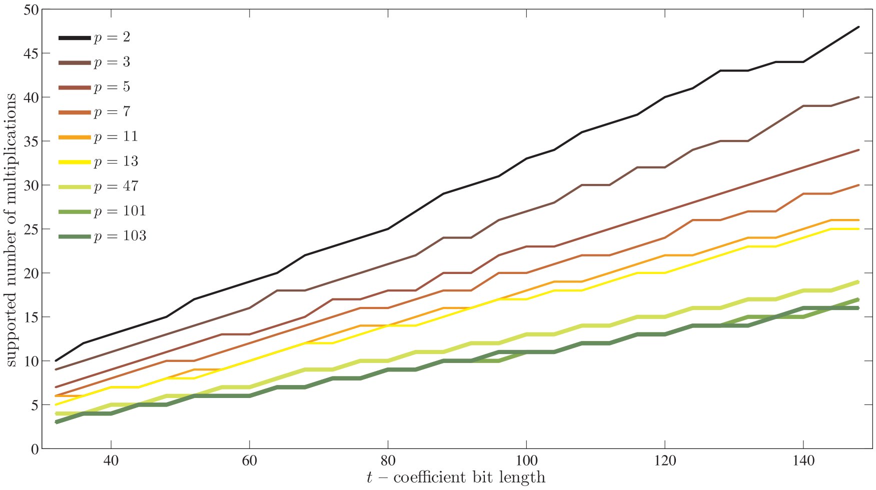

FIGURE 1. The supported number of multiplications of SHS for $N = 128$, $p = 2, 3, \ldots, 13, 47, 101, 103$ and various parameters t.

Decryption. The complexity impact is similar to encryption. The modulo-2 operation in the original scheme has been implemented with logical AND, now it becomes a regular division. Since the scheme comprises also another division and multiplication, the asymptotic complexity of decryption remains the same[1].

2.3. Experimental results

We performed several experiments that show the homomorphic properties of the scheme with some small primes $p = 2, 3, \ldots, 13$ and $47, 101, 103$. These experiments are just for illustration of the properties of the SHS, so the parameter settings were rather small. All experiments were performed in dimension $N = 128$ and $N = 256$. As both results were nearly the same, we display the results for $N = 128$ only.

The first group of experiments focused on multiplications alone. Sufficient number of PT-CT pairs were created and ciphertext monomials of increasing degree were computed. The result was decrypted and compared with the corresponding product on plaintexts. The process was interrupted on the first decryption error and the previous degree of monomial was denoted as the largest supported number of multiplications. The whole experiment was repeated 30 times and the minimum of the returned values is shown on Fig. 1.

[1] This change will have greater impact on the decryption function of the FHS. We see no straightforward way how to construct the squashed decryption function yet.

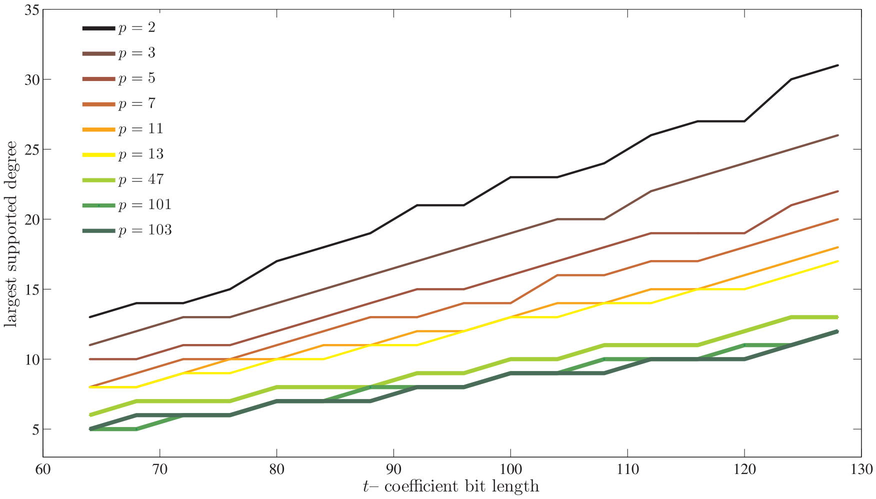

FIGURE 2. The largest supported degree of symmetric polynomials for $N = 128$, $p = 2, 3, \ldots, 13, 47, 101, 103$, $m = 80$ and various parameters t.

The second group of experiments computed the largest supported degree of symmetric polynomials of 80 variables. The symmetric polynomials of 80 ciphertexts were computed for every degree up to 80 and the results were decrypted and compared to corresponding symmetric polynomials evaluated on plaintexts. Denote d_{err} the minimal degree that produced decryption error. The largest supported degree is then $lsd = d_{err} - 1$. This experiment was repeated 10 times for every parameter setting (p, N, t) and the minimal value of lsd was returned. Results are shown on Fig. 2.

Conclusions. The obtained results indicate that the scheme behaves similarly with p higher than 2. The dependency of lsd is still linear in t, so the supported degree is only x $(x \approx 2)$ times lower with $p = 13$ than with the original setting of $p = 2$.

Specifically the results imply that if we need to compute homomorphically with numbers up to 8, it is better to set $p = 11$ and use the modified scheme than to set $p = 2$ and compute with 3-bit numbers encrypted as binary vectors. The reason behind this fact is that $\omega(m^2)$ bit-multiplications are needed for a single multiplication of two m-bit binary numbers and a polynomial of degree $2m$ needs to be evaluated during the computation.

2.4. Somewhat practical, somewhat homomorphic scheme

The results of previous section show that even the modified somewhat homomorphic scheme can correctly evaluate a reasonable number of operations.

Furthermore, the homomorphic properties are easily scalable by the parameter t. In this section we show how to construct a more practical somewhat homomorphic scheme by enlarging its plaintext space.

We use the Chinese remainder theorem (CRT) to further enlarge the plaintext space of the scheme. The idea is to set a bound b for the plaintexts and create k independent schemes $\xi_1(p_1), \ldots, \xi_k(p_k)$ such that the product of the primes $p_1, \ldots, p_k$ is larger than b. Then we "split" every message $m \in Z_b$ into the corresponding moduli m mod p_i and encrypt each separately. After decryption we use CRT to compute the original message back. It is clear that the scheme is correct if each of the underlying schemes ξ_i is correct, so the supported number of homomorphic operations is the minimum of the ξ_i's.

The detailed description of our proposed scheme:

Input parameters:

- b – plaintext bound,
- N, t, q – parameters of the SHS.

Algorithms:

- $KeyGen(\lambda)$:
  (1) choose integers k and $p_1, \ldots, p_k$ such that $\prod_{i=1}^{i=k} p_i \geq b$,
  (2) generate k somewhat homomorphic schemes $\xi_1(p_1), \ldots, \xi_k(p_k)$,
  (3) public key (resp. private key) is k-tuple of public (resp. private) keys of schemes ξ_i,
  (4) plaintext space $\mathcal{P} = \mathbb{Z}_b$ and ciphertext space $\mathcal{C} = (\mathbb{Z}_d)^k$.
- $Encrypt(pk, m, q)$:
  (1) for all $i = 1, \ldots, k$ compute first $m_i = m \bmod p_i$ and then $c_i = Encrypt_{\xi_i}(pk, m_i, q)$,
  (2) output the vector of ciphertexts $c = (c_1, \ldots, c_k)$.
- $Decrypt(sk, c)$:
  (1) for all $i = 1, \ldots, k$ compute $m_i = Decrypt_{\xi_i(sk, c_i)}$,
  (2) from m_i compute via Chinese remainder theorem unique m such that $m \equiv m_i$ mod p_i for every i.
- $Eval(pk, \circ, c_1, c_2)$: output vector c_3 is computed componentwise for every $i = 1, \ldots, k$ as $Eval_{\xi_i}(c_{1,i}, c_{2,i})$.

It should be noted that when we need to compute with numbers up to a fixed b, we obtain the best homomorphic properties if we choose successive primes from $2, 3, \ldots, p_k$, because in this case the p_k is minimal.

3. Conclusions

In this paper we followed the Gentry-Halevi implementation of the somewhat homomorphic scheme and examined the homomorphic properties of modified scheme with plaintext space $\mathbb{Z}_p$ for some prime p.

As expected, for increasing prime p the supported number of operations decreases. The graphs 1 and 2 show that the number of operations is still reasonable high, so we proposed a somewhat homomorphic scheme that can perform homomorphic operations for arbitrary large plaintext space.

It is easy to see that the proposed scheme is more efficient than existing fully homomorphic schemes, but its application scope is restricted by the limitations on the supported number of operations.

REFERENCES

[1] AJTAI, M.—DWORK, C.: *A public key cryptosystems with worst-case/average-case equivalence*, in: Proc. of the 29th Annual ACM Symp. on Theory of Comput.—STOC '97 (F. T. Leighton and P. Shor, eds.), El Paso, TX, USA, 1997, ACM, New York, NY, pp. 284–293.

[2] AHITUV, N.—LAPID, Y.—NEUMANN, S.: *Processing encrypted data*, Commun. ACM **30** (1987), 770–780.

[3] ARMKNECHT, F.—SADEGHI, A.: *A new approach for algebraically homomorphic encryption*, Cryptology ePrint Archive, Report 2008/422, 2008, `http://eprint.iacr.org/2008/422`.

[4] BAO, F.: *Cryptanalysis of a provable secure additive and multiplicative privacy homomorphism*, in: Internat. Workshop on Coding and Cryptography—WCC '03, Versailles, France, 2003, pp. 43–49.

[5] BONEH, D.—GOH, J.—NISSIM, K.: *Evaluating 2-DNF formulas on ciphertexts*, in: Theory of Cryptography, The 2nd Theory of Cryptography Conf.—TCC '05 (J. Kilian, ed.), Cambridge, MA, USA, 2005, Lecture Notes in Comput. Sci., Vol. 3378, Springer, Berlin, 2005, pp. 325–342.

[6] CHUNSHENG, G.: *New fully homomorphic encryption over the integers*, Cryptology ePrint Archive, Report 2011/118, 2011, `http://eprint.iacr.org/2011/118`.

[7] VAN DIJK, M.—GENTRY, C.—HALEVI, S.—VAIKUNTANATHAN, V.: *Fully homomorphic encryption over the integers*, in: Advances in Cryptology—EUROCRYPT '10 (H. Gilbert, ed.), Lecture Notes in Comput. Sci., Vol. 6110, Springer, Berlin, 2010, pp. 24–43.

[8] GENTRY, C.: *Fully homomorphic encryption using ideal lattices*, in: Proc. of the 41st Annual ACM Symposium on Theory of Computing—STOC '09, Bethesda, USA, 2009, ACM, New York, 2009, pp. 169–178.

[9] GENTRY, C.: *A Fully Homomorphic Encryption Scheme*. Dissertation Thesis, Standford University, Standford, 2009, `http://crypto.stanford.edu/craig/`.

[10] GOLDREICH, O.—GOLDWASSER, S.—HALEVI, S.: *Public key cryptosystems from lattice reduction problems*, in: Advances in Cryptology—CRYPTO '97, The 17th Annual Internat. Cryptology Conf. (B. Kaliski, Jr., ed.), Santa Barbara, CA, USA, 1997, Lecture Notes in Comput. Sci., Vol. 1294, Springer, Berlin, 1997, pp. 112–131.

[11] GENTRY, C.—HALEVI, S.: *Implementing Gentry's fully-homomorphic encryption scheme*, Cryptology ePrint Archive, Report 2010/520, 2010, `http://eprint.iacr.org/2010/520`.

[12] LOFTUS, C.—MAY, A.—SMART, N. P.—VERCAUTEREN, F.: *On CCA-secure fully homomorphic encryption*, Cryptology ePrint Archive, Report 2010/560, 2010, `http://eprint.iacr.org/2010/560`.

[13] LYUBASHEVSKY, V.—MICCIANCIO, D.: *On bounded distance decoding, unique shortest vectors, and the minimum distance problem*, in: Adv. in Cryptology—CRYPTO '09, The 29th Annual Internat. Cryptology Conf. (S. Halevi, ed.), Santa Barbara, CA, USA, 2009, Lecture Notes in Comput. Sci., Vol. 5677, Springer, Berlin, 2009, pp. 577–594.

[14] MICCIANCIO, D.: *Improving lattice based cryptosystems using the Hermite normal form*, in: Cryptography and Lattices, The 1st Internat. Conf.—CaLC '01 (J. H. Silverman, ed.), Providence, RI, USA, 2001, Lecture Notes in Comput. Sci., Vol. 2146, Springer, Berlin, 2001, pp. 126–145.

[15] MELCHOR, C.—GABORIT, P.—HERRANZ, J.: *Additively homomorphic encryption with t-operand multiplications*, Cryptology ePrint Archive, Report 2008/378, 2008, `http://eprint.iacr.org/2008/378`.

[16] RIVEST, R.—ADLEMAN, L.—DERTOUZOS, M.: *On data banks and privacy homomorphisms*, in: Foundations of Secure Computation (R. DeMillo et al., eds.), Academic Press, New York, 1978, pp. 169–180.

[17] SMART, N. P.—VERCAUTEREN, F.: *Fully homomorphic encryption with relatively small key and ciphertext sizes*, Cryptology ePrint Archive, Report 2009/571, 2009, `http://eprint.iacr.org/2009/571`.

Received October 18, 2011

Institute of Computer Science and Mathematics
Faculty of Electrical Engineering
and Information Technology
Slovak University of Technology
Ilkovičova 3
SK–812-19 Bratislava
SLOVAKIA
E-mail: michal.mikus@stuba.sk