ATLANTIS PRESS

Advances in Engineering Research, volume 164

8th International Conference on Manufacturing Science and Engineering (ICMSE 2018)

An Improved Homomorphic Technique in Construction of Fully Homomorphic Encryption Scheme from LWE

Chengbo Xu[1, a], Shuying Yang[2,b]

[1]School of Mathematical Sciences, University of Jinan, Jinan 250022, Shandong Province, P. R. China

[2]Department of Data and Computer Science, Shandong Women's University, Jinan 250300, Shandong Province, P. R. China

[a]cbqysy@163.com, [b]ysytudy2005@163.com

Keywords: Fully Homomorphic Encryption, Lattice With Error, Cloud Computing

Abstract. In this paper, we study the homomorphic techniques used in GSW fully homomorphic encryption scheme, and point out the drawback which prevents the GSW scheme from efficiency. In order to conquer the weakness of low efficiency, we propose an improved homomorphic technique. Through comparative analysis, our proposed technique is proved to not only improve the efficiency substantially, but also keep the functionality essentially required.

Introduction

With the development of cloud computing and other complex networks, fully homomorphic encryption (FHE) becomes a very attractive cryptography primitive that allows arbitrary computation on encrypted data and has numerous theoretical and practical applications[1,2]. After conceptualizing FHE in 1978 [2], for more than 30 years, it was unclear whether fully homomorphic encryption was even possible. At STOC 2009, Gentry proposed the first FHE scheme based on ideal lattices [1]. Since then, a lot of developments and improvements were witnessed [3-8].

A line of FHE schemes arose from the SHE scheme of Gentry-Sahai-Waters(GSW) [6]. This SHE scheme supports a different class of functions, including branching programs, and this was also proved sufficient to bootstrap it to FHE via Barrington's theorem. Interestingly, this approach theoretically allows obtaining FHE from a weaker version of the LWE assumption [7].

In this paper, we first review GSW fully homomorphic encryption scheme [6, 7], and then point out the drawback of low efficiency in the process of implementation and a vital property of GSW evaluation algorithm which makes the GSW scheme [6, 7] feasible in practice. In order to conquer the weakness of low efficiency, an improved homomorphic technique is proposed. Through comparative analysis, we illustrate that our proposed technique not only improves the efficiency substantially, but also keeps the functionality essentially required.

Homomorphic Techniques in GSW Homomorphic Encryption scheme

In this section, we will review the GSW homomorphic encryption scheme [6, 7], and then point out the homomorphic technique used in its construction.

The GSW homomorphic encryption scheme [6, 7] is parameterized by a dimension n, a modulus q with $l=\lceil \log_2 q \rceil$, and some error distribution c over $\mathbf{Z}$ which we assume to be subgaussian. Formally, we describe the scheme as follows:

– GSW.Gen(): choose $\bar{s} \leftarrow c^{n-1}$ and output secret key $s=(\bar{s},1)\in \mathbf{Z}^n$.

– GSW.Enc(s, $m\in \mathbf{Z}$): choose $\bar{C} \leftarrow \mathbf{Z}_q^{(n-1)nl}$ and $e \leftarrow c^m$, let $b^T = e^T - \bar{s}^T\bar{C} (\mathrm{mod}\, q)$, and output the Ciphertext:

$$C=\begin{bmatrix}\bar{C}\\ b^T\end{bmatrix}+mG$$

Copyright © 2018, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

Where G is the gadget matrix. Notice that $s^T C = e^T + m s^T G (\mathrm{mod}\, q)$.

– GSW.Dec(s, C): Let c be the penultimate column of C, and output $m = \lfloor \langle s, c \rangle \rceil_2$.

– GSW.Eval(C_1, C_2):

- Homomorphic addition: $C_1 \oplus C_2 = C_1 + C_2$.

- Homomorphic multiplication: $C_1 \mathbf{e} C_2 \leftarrow C_1 \cdot G^{-1}(C_2)$, and is right associative.

For a function $f = x_1 + x_2 + \mathbf{L} + x_n$, and n fresh ciphertexts $C_1, C_2, \mathbf{L}, C_n$, the GSW evaluation algorithm will compute as the following:

$$\begin{aligned}
&\text{GSW.Eval}(C_1, C_2, \mathbf{L}, C_n, f) \\
&= C_1 \oplus C_2 \oplus \mathbf{L} \oplus C_n = C_1 + C_2 + \mathbf{L} + C_n \\
&= \begin{bmatrix} \bar{C}_1 + \bar{C}_2 + \mathbf{L} + \bar{C}_n \\ (e_1^T + e_2^T + \mathbf{L} + e_n^T) - \bar{s}^T (\bar{C}_1 + \bar{C}_2 + \mathbf{L} + \bar{C}_n) \end{bmatrix} + f(m_1, m_2, \mathbf{L}, m_n) G
\end{aligned}$$

After evaluation, the error term of result ciphertext is $e_f = e_1 + e_2 + \mathbf{L} + e_n$ which is a linear combination of those original error terms $e_1, e_2, \mathbf{L}, e_n$.

For a function $g = x_1 x_2 \mathbf{L}\, x_n$, and n fresh ciphertexts $C_1, C_2, \mathbf{L}, C_n$, the GSW evaluation algorithm will compute as the following:

$$\text{GSW.Eval}(C_1, C_2, \mathbf{L}, C_n, g) = C_1 \mathbf{e} C_2 \mathbf{e} \mathbf{L} \mathbf{e} C_n = C_1 G^{-1}(C_2 G^{-1}(\mathbf{L}\, C_{n-2} G^{-1}(C_{n-1} G^{-1}(C_n))))$$

Since there are many nests in the above structure, we compute it step by step inside and out.

$$\begin{aligned}
C_{n-1} G^{-1}(C_n) &= \left(\begin{bmatrix} \bar{C}_{n-1} \\ b_{n-1}^T \end{bmatrix} + m_{n-1} G \right) G^{-1}(C_n) \\
&= \begin{bmatrix} \bar{C}_{n-1} G^{-1}(C_n) + m_{n-1} \bar{C}_n \\ (e_{n-1}^T G^{-1}(C_n) + m_{n-1} e_n^T) - \bar{s}^T (\bar{C}_{n-1} G^{-1}(C_n) + m_{n-1} \bar{C}_n) \end{bmatrix} + (m_{n-1} m_n) G
\end{aligned}$$

$$\begin{aligned}
C_{n-2} G^{-1}(C_{n-1} G^{-1}(C_n)) &= \left(\begin{bmatrix} \bar{C}_{n-2} \\ b_{n-2}^T \end{bmatrix} + m_{n-2} G \right) G^{-1}(C_{n-1} G^{-1}(C_n)) \\
&= \begin{bmatrix} \bar{C}_{n-2} G^{-1}(C_{n-1} G^{-1}(C_n)) + m_{n-2} \bar{C}_{n-1} G^{-1}(C_n) + m_{n-2} m_{n-1} \bar{C}_n \\ b_{n-2}^T G^{-1}(C_{n-1} G^{-1}(C_n)) + m_{n-2} b_{n-1}^T G^{-1}(C_n) + m_{n-2} m_{n-1} b_n^T \end{bmatrix} + (m_{n-2} m_{n-1} m_n) G
\end{aligned}$$

and so on in a similar fasion, we will compute the evaluation result GSW.Eval($C_1, C_2, \mathbf{L}, C_n, g$) whose form is very complex, so we omit here. However, an important point worth mentioned here is that after evaluation, the error term of result ciphertext is

$$e_f = e_1 G^{-1}(C_2 G^{-1}(\mathbf{L}\, C_{n-1} G^{-1}(C_n))) + m_1 e_2 G^{-1}(C_3 G^{-1}(\mathbf{L}\, C_{n-1} G^{-1}(C_n))) + \mathbf{L} + m_1 m_2 \mathbf{L}\, m_{n-1} e_n$$

which is also a linear combination of those original error terms $e_1, e_2, \mathbf{L}, e_n$.

Our Proposed Homomorphic Techniques

The GSW homomorphic technique is a classical tool in constructions of fully homomorphic encryption scheme [6, 7]. However, in the process of evaluating multiplication gate, the efficiency is a bottleneck since the procedure cannot be implemented in a parallel way. In order to conquer this drawback, we proposed a new homomorphic method as follows:

For the addition gate $f = x_1 + x_2 + \mathbf{L} + x_n$, and n fresh ciphertexts $C_1, C_2, \mathbf{L}, C_n$, the process of evaluation is same to the GSW scheme.

For the multiplication gate $g = x_1 x_2 \mathbf{L}\, x_n$, and n fresh ciphertexts $C_1, C_2, \mathbf{L}, C_n$, the procedure of evaluation is

GSW.Eval($C_1, C_2, \mathbf{L}, C_n, g$)

$$= ((((C_1 G^{-1}(C_2))G^{-1}(C_3)) \mathbf{L}\, G^{-1}(C_{n-1}))G^{-1}(C_n))$$
$$= ((((\left(\begin{bmatrix} \bar{C}_1 \\ b_1^T \end{bmatrix} + m_1 G\right)G^{-1}(C_2))G^{-1}(C_3)) \mathbf{L}\, G^{-1}(C_{n-1}))G^{-1}(C_n))$$
$$= ((((\begin{bmatrix} \bar{C}_1 G^{-1}(C_2) \\ b_1^T G^{-1}(C_2) \end{bmatrix} + m_1 G(G^{-1}(C_2)))G^{-1}(C_3)) \mathbf{L}\, G^{-1}(C_{n-1}))G^{-1}(C_n))$$
$$= ((((\begin{bmatrix} \bar{C}_1 G^{-1}(C_2) \\ e^T G^{-1}(C_2) - \bar{s}^T \bar{C}_1 G^{-1}(C_2) \end{bmatrix} + m_1\left(\begin{bmatrix} \bar{C}_2 \\ b_2^T \end{bmatrix} + m_2 G\right))G^{-1}(C_3)) \mathbf{L}\, G^{-1}(C_{n-1}))G^{-1}(C_n))$$
$$= ((((\begin{bmatrix} \bar{C}_1 G^{-1}(C_2) + m_1 \bar{C}_2 \\ (e_1^T G^{-1}(C_2) + m_1 e_2^T) - \bar{s}^T (\bar{C}_1 G^{-1}(C_2) + m_1 \bar{C}_2) \end{bmatrix} + (m_1 m_2)G)G^{-1}(C_3)) \mathbf{L}\, G^{-1}(C_{n-1}))G^{-1}(C_n))$$
$$= \mathbf{L}$$
$$= \begin{bmatrix} C_{eval} \\ e_{eval}^T - \bar{s}^T(C_{eval}) \end{bmatrix} + (m_1 m_2 \mathbf{L}\, m_n)G$$

Where

$$C_{eval} = \bar{C}_1 G^{-1}(C_2) \mathbf{L}\, G^{-1}(C_n) + m_1 \bar{C}_2 G^{-1}(C_3) \mathbf{L}\, G^{-1}(C_n) + \mathbf{L} + m_1 \mathbf{L}\, m_{n-1}\bar{C}_n,$$
$$e_{eval}^T = e_1^T G^{-1}(C_2) \mathbf{L}\, G^{-1}(C_n) + m_1 e_2^T G^{-1}(C_3) \mathbf{L}\, G^{-1}(C_n) + \mathbf{L} + m_1 \mathbf{L}\, m_{n-1} e_n^T$$

After the process of evaluation, the form of resulting error e_{hom}^T is also linear in those original error terms $e_1, e_2, \mathbf{L}, e_n$. This illustrate our proposed homomorphic technique keeps the vital property mentioned above. Thus, our method can be implemented to replace GSW homomorphic evaluation in the construction of GSW homomorphic encryption scheme.

Comparison of These Two Homomorphic Techniques

In this section, we compare our proposed homomorphic techniques with that used in GSW encryption scheme [6, 7] in two aspects of efficiency and functionality.

Comparison in Efficiency. GSW scheme is a represent of those third generation fully homomorphic encryption scheme since it is formalized succinctly. However, one of main obstacles affecting its wide application in practice is its efficiency. By the construction of GSW, one has to do the following computation:

$$C_{eval} = \bar{C}_1 G^{-1}(C_2 G^{-1}(\mathbf{L}\, C_{n-1} G^{-1}(C_n))) + m_1 \bar{C}_2 G^{-1}(C_3 G^{-1}(\mathbf{L}\, C_{n-1} G^{-1}(C_n))) + \mathbf{L} + m_1 m_2 \mathbf{L}\, m_{n-1}\bar{C}_n$$

when evaluating the multiplication gate $g = x_1 x_2 \mathbf{L}\, x_n$. In above equation, we have to compute the matrix $G^{-1}(C_2 G^{-1}(\mathbf{L}\, C_{n-1} G^{-1}(C_n)))$ step by step inside and out which is the key problem to result in low efficiency. In order to conquer the weakness of sequential computation, a natural approach is applying parallel computation to improve efficiency. In our proposed homomorphic technique, we implement the idea. By the construction of our proposed homomorphic technique, when evaluating the multiplication gate $g = x_1 x_2 \mathbf{L}\, x_n$, What we have to do is only to compute the following:

$$C_{eval} = \bar{C}_1 G^{-1}(C_2) \mathbf{L}\, G^{-1}(C_n) + m_1 \bar{C}_2 G^{-1}(C_3) \mathbf{L}\, G^{-1}(C_n) + \mathbf{L} + m_1 \mathbf{L}\, m_{n-1}\bar{C}_n$$

This formula looks like the previous one, but it is able to be computed in a parallel way since no nests exist in the structure. Concretely, we can compute the $n-1$ matrices $G^{-1}(C_2)$, $\mathbf{L}$, $G^{-1}(C_n)$ at the same time, while in the previous way, we can only compute the matrix $G^{-1}(C_2 G^{-1}(\mathbf{L}\, C_{n-1} G^{-1}(C_n)))$ step by step inside and out. Thus, our proposed homomorphic technique is more efficient than the GSW method.

Comparison in Functionality. One of vital properties of GSW homomorphic technique which make GSW encryption scheme feasible is that the error growth during process of the evaluation algorithm

is a linear combination of those original error terms $e_1, e_2, \mathbf{L}, e_n$. In our proposed homomorphic technique, this vital property is maintained since the error growth during process of our proposed evaluation algorithm is

$$e_{eval}^T = e_1^T G^{-1}(C_2)\,\mathbf{L}\,G^{-1}(C_n) + m_1 e_2^T G^{-1}(C_3)\,\mathbf{L}\,G^{-1}(C_n) + \mathbf{L} + m_1\,\mathbf{L}\,m_{n-1} e_n^T$$

which is also linear in the corresponding original error terms. Moreover, the coefficients in the error linear combination of our proposed evaluation process can be computed in a parallel way, while the corresponding coefficients of GSW evaluation algorithm can only computed sequentially.

Conclusions

In this paper, we first reviewed GSW fully homomorphic encryption scheme, and then pointed out the drawback of low efficiency in the process of implementation and a vital property of GSW evaluation algorithm which makes the GSW scheme feasible in practice. In order to conquer the weakness of low efficiency, we proposed an improved homomorphic technique. Through comparative analysis, our proposed technique is proved to not only improve the efficiency substantially, but also keep the functionality essentially required.

Acknowledgements

This work was partially supported by the Doctoral Fund of University of Jinan (Granted No. XBS1455), and the project of Shandong Natural Science Foundation (Granted No. ZR2013FM009).

References

[1] C. Gentry et al. : Fully homomorphic encryption using ideal lattices. In STOC, volume 9, pages 169–178, 2009.

[2] R. L. Rivest, L. Adleman, and M. L. Dertouzos: On data banks and privacy homomorphisms. Foundations of Secure Computation, pages 169–179, 1978.

[3] S. Halevi and V. Shoup: Faster Homomorphic Linear Transformations in HElib. IACR Cryptology ePrint Archive, 2018(244), 2018.

[4] N. P. Smart and F. Vercauteren: Fully homomorphic SIMD operations. Des. Codes Cryptography, 71(1):57–81, 2014.

[5] S. Halevi and V. Shoup: Bootstrapping for HElib. In Elisabeth Oswald and Marc Fischlin, editors, EUROCRYPT 2015, Part I, volume 9056 of LNCS, pages 641–670, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany.

[6] C. Gentry, A. Sahai, and B. Waters: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, CRYPTO 2013, Part I, volume 8042 of LNCS, pages 75–92, Santa Barbara, CA, USA, August 18– 22, 2013. Springer, Heidelberg, Germany.

[7] J. Alperin-Sheri□ and C. Peikert: Faster bootstrapping with polynomial error. In Juan A. Garay and Rosario Gennaro, editors, CRYPTO 2014, Part I, volume 8616 of LNCS, pages 297–314, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany.

[8] L. Ducas and D. Micciancio: FHEW: Bootstrapping homomorphic encryption in less than a second. In Elisabeth Oswald and Marc Fischlin, editors, EUROCRYPT 2015, Part I, volume 9056 of LNCS, pages 617–640, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany.