# Efficient Public Key Generation for Homomorphic Encryption over the Integers

Y. Govinda Ramaiah and G. Vijaya Kumari

Department of Computer Science and Engineering
JNTUH College of Engineering, Hyderabad, India
{ygovinda,vijayakumari.gunta}@acm.org

**Abstract.** The 'Holy Grail' of cryptography called Fully Homomorphic Encryption (FHE), which allows encrypted data processing and delegation of computational tasks to the remote untrusted server, has become a hot research topic in light of the privacy concerns related to cloud computing. Several FHE schemes were found after the first construction of such scheme by Craig Gentry in 2009. One of the several reasons making these theoretically feasible schemes unpractical is their high computational costs. In this paper, a simplest possible key generation method is proposed for the somewhat homomorphic scheme of Van Dijk et al., which leads to an efficient integer based FHE scheme. Also, the security and practicality of the proposed scheme is thoroughly analyzed with respect to the new key generation method suggested.

**Keywords:** Homomorphic Encryption, Key Generation, Security, Practicality.

## 1 Introduction

The problem of devising a *Fully Homomorphic Encryption* (FHE) scheme or a *privacy homomorphism* [1], which supports "processing the data while it is encrypted", has been studied for over decades. The research on the topic has gained momentum after Craig Gentry's first construction of such a scheme based on algebraic lattice theory in the year 2009 [2][3]. This breakthrough work has become an attractive solution, especially for the security and privacy problems of cloud computing [7] and the related applications, but, only theoretically promising.

The initial construction of Gentry's FHE [2] [3] consists of a strict 3-step blueprint which include, *Constructing a Somewhat Homomorphic Encryption* (SHE) *scheme*, *Squashing the decryption function of the* SHE, and finally *Obtaining the* FHE (Bootstrapping) [6]. FHE schemes that follow the Gentry's blueprint [4] [5] were found to be inefficient enough for practical implementation [9] because of the huge difference between the computational complexities of processing the ciphertexts and processing the plaintexts. The major contribution to this high complexity is by large *message expansion* (e.g., in the scheme of [5] every bit is expanded to a ciphertext of $\tilde{O}(n^5)$), and the ciphertext refreshing procedure during the bootstrapping. In [8] the first implementation of integer based FHE scheme of [5] is described. Their major

V.V. Das and J. Stephen (Eds.): CNC 2012, LNICST 108, pp. 262–268, 2012.

contribution is in reducing the public key size of that scheme from $\tilde{O}(n^{10})$ to $\tilde{O}(n^7)$. More expository survey of the recent advances in the homomorphic cryptography is given in [10].

In this work, an efficient variant of the underlying SHE scheme of [5] is presented using a comparatively smaller public key of size $\tilde{O}(n^3)$. It is shown that, the semantic security of the proposed scheme is preserved under the two-element Partial Approximate Greatest Common Divisor (PAGCD) problem. Also, the proposed variant is proved *compact* with low ciphertext expansion of $n^3$. It is estimated that, with the improvements made the Homomorphic Encryption usage and thus encrypted data processing becomes imminent for suitable applications that fall within the multiplicative capacity of the proposed scheme. Due to space constraints, the proofs for all the Theorems and Lemmas are given in the Appendix of the full version of this paper.

## 2 The SHE over the Integers and the Proposed Optimization

The Somewhat Homomorphic Encryption over the integers [5], which is denoted as HE in this paper, consists of four algorithms *KeyGen*, *Encrypt*, *Decrypt* and *Evaluate*. The size (bit length) of various integers used in the scheme is denoted by the parameters $e$, $t$, $r$, $g$, $d$, which represent the size of the secret key, number of elements in the public key, size of the noise in the public key integers, the size of each integer in the public key, size of the noise used for encryption, respectively and are polynomial in the security parameter $n$. The parameter setting suggested in view of the homomorphism and security is, $e = \tilde{O}(n^2)$, $r = n$, $d = 2n$, $g = \tilde{O}(n^5)$, and $t = g + n$. This makes the public key size as $\tilde{O}(n^{10})$, because, the public key consists of $t = \tilde{O}(n^5)$ integers each of size $g = \tilde{O}(n^5)$.

*KeyGen*($n$): Choose a random $e$-bit odd integer from the right open interval $[2^{e-1}, 2^e)$ as the secret key P. For i = 0,1,…..,$t$ , Choose a random integer $Q_i$ from $[0, 2^g/P)$, another integer $R_i$ from the open interval $(-2^r, 2^r)$, and compute $X_i = PQ_i + R_i$ until the conditions $X_0 > X_1,\ldots..., X_t$ , $X_0 \bmod 2 = 1$, and $(X_0 \bmod P) \bmod 2 = 0$ are satisfied. Output the public key PK = $(X_0, X_1,\ldots..., X_t)$ and the secret key SK = P.

*Encrypt*(PK, M ∈ {0, 1} ): Choose an integer B from $(-2^d, 2^d)$ as noise for encryption. Choose a subset $\mathbf{J} \subseteq \{1,\ldots..t\}$. Compute the sum $S = \sum_{i \in \mathbf{J}} X_i$ . Output the ciphertext as $C = [M + 2(B + S)] \bmod X_0$.

*Decrypt* (SK, C) : Compute M = ( C mod P ) mod 2.

*Evaluate*(PK,CKT, $(C_1,\ldots..,C_k)$ ): Let CKT be the binary circuit to be evaluated representing a boolean function $f$, with XOR gates and AND gates (i.e., CKT consists of mod-2 addition and multiplication gates). Replace the XOR gates and AND gates of CKT with addition and multiplication gates that operate over integers. Let GCKT be the resulting generalized circuit and $f_g$ be the corresponding multivariate polynomial. Apply GCKT over $(C_1,\ldots..,C_k)$, and output the resulting ciphertext $C_g = f_g(C_1,\ldots..,C_k)$ .

For a ciphertext in the scheme we have, $C = [M + 2(B + S)] \bmod X_0 = M + 2B_m + PQ_m$, for some integers $B_m$ and $Q_m$. The term, say $N = M + 2B_m << P$, is the noise (the distance to the multiple $PQ_m$), which makes C an approximate or near multiple of P. Multiplication or addition of such near multiples results in another near multiple. Therefore in decryption, C mod P results in N, and N mod 2 gives the plaintext bit M. For every multiplication during the *Evaluate*, the size of the resulting noise equals the sum of the sizes of the multiplicand noises, which crosses the size of P after certain number of multiplications, resulting in incorrect decryption. This makes the scheme an SHE and to make it an FHE, the transformation based on Gentry's blueprint [2][3] is used by [5]. Optimizations proposed in this paper targets only the underlying SHE.

The method being proposed is denoted as $HE^{SP}$, in which the public key consists of only two big integers $X_0$ and $X_1$. $X_0$ is an exact multiple of the odd secret integer P and $X_1$ is an approximate multiple, i.e., multiple of P containing some additive error R. To encrypt a plaintext bit M, the erroneous integer $X_1$ of the public key is multiplied with a random even integer N, the result is added to the plaintext bit and the final sum is reduced modulo the error-free integer $X_0$ in the public key. For homomorphic evaluation of a function, the addition and multiplication operations in the corresponding arithmetic circuit are performed over ciphertexts, modulo the error-free integer $X_0$ in the public key. The security of $HE^{SP}$ is based on the two-element Partial Approximate Greatest Common Divisor (PAGCD) problem (Definition 2). The parameter setting for the variant scheme being proposed is reviewed as follows.

For the given security parameter $n$, the size of the secret key integer P, denoted by $e$, is taken as $\geq d$. $\Theta(n \lg^2 n)$ to support homomorphism for evaluation of sufficiently deeper circuits. Size of the noise in the public key integer $X_1$, denoted by $r$, is taken as $\omega(\lg n)$ to foil the brute-force attack against the noise. The number of bits in each of the public key integers is denoted by $g$. More precisely, $g$ is the size of the factor Q in the multiples of P, in the public key. Since the public key consists of only two elements, the attacks related to two-element PAGCD problem only are considered. With respect to this, it is claimed that, it is sufficient to take $g > e$ against the condition used in [5] as $g > e^2$, to thwart lattice based attacks on the AGCD problem with some arbitrary $t$ number of elements. Therefore, $g$ is taken as $\omega(\, e. \lg n\,)$. The parameter $d$ denotes the size of the even noise factor N used during the encryption. With these, the theoretical parameter setting for $HE^{SP}$ can be chosen as, $e = \tilde{O}(n^2)$, $r = n$, $d = 2n$, and $g = \tilde{O}(n^3)$. This setting results in a scheme with overall complexity of $\tilde{O}(n^3)$. With this, the construction of the proposed variant is obtained as follows.

$KeyGen^{SP}(n)$: Secret key is a random $e$-bit odd integer P chosen from $[2^{e-1}, 2^e)$. Choose a random $r$-bit integer R from the interval $(-2^r, 2^r)$. For i = 0, 1, Choose a random $g$-bit integer $Q_i$ from $[0, 2^g / P)$. Compute $X_0 = P\, Q_0$, $X_1 = PQ_1 + R$. Output the secret key, SK = P and the public key, $PK = (X_0, X_1)$.

*Note*. For the reasons described in Section 4, the integers $X_0$, $X_1$ should be co-prime. Also, we take $X_0 > X_1$.

*Encrypt*$^{SP}$ (PK, M ∈ {0,1} ): For a plaintext bit M∈ {0,1}, Choose a random even integer N from the interval $[2^{d-1}, 2^d)$. The ciphertext C = [M + N. $X_1$] mod $X_0$

*Evaluate*$^{SP}$ ( PK, CKT, ($C_1$,…..,$C_k$) ), and *Decrypt*$^{SP}$ (SK, C) algorithms are same as that of the original HE.

The appealing feature of the scheme HE$^{SP}$ is the relatively smaller public key with only two integers of size $\tilde{O}(n^3)$ each. Encryption method is also comparatively simple because, the product (N. $X_1$) corresponds to the operations of choosing a random subset from the big set of public key elements, adding the elements in that subset, multiplying the sum with 2 and adding to an even noise as done in HE.

Similar to the case of HE, the limits imposed on the sizes of the noise makes the scheme somewhat homomorphic. It is quite easy to see that *the scheme* HE$^{SP}$ *is a variant of the scheme* HE *for the chosen parameter setting*. This is because, the ciphertext in HE is M + 2B + PQ. The ratio of size of P and the size of noise (M+2B) is $\tilde{O}(n^2)$ / $\tilde{O}(n)$ = $\tilde{O}(n)$. Consider a fresh ciphertext in HE$^{SP}$. We have, C = [M + N. $X_1$] mod $X_0$ = M + RN + P ($NQ_1$ – K $Q_0$) for some integer K ≥ 0. This can be written as M + 2$B_s$ + P$Q_s$ since RN is even, and due to which we have $B_s$ = RN/2, $Q_s$ = ($NQ_1$ – K $Q_0$).The ratio between the size of P and the size of noise (M + 2$B_s$) is $\tilde{O}(n^2)$ / $\tilde{O}(n)$ = $\tilde{O}(n)$, which is same as that of HE. Hence, both the schemes are identical with only difference in the methods of key generation and encryption. For *Evaluate*$^{SP}$, corresponding to the generalized circuit GCKT we have the following notion of permitted circuit.

**Definition 1.** (Permitted circuit). An arithmetic circuit with addition and multiplication gates is called a permitted circuit for the scheme HE$^{SP}$ if, for any set of integer inputs each $< 2^d$ in absolute value, the maximum absolute value output by the circuit is $< 2^{e-2}$. We denote the set of permitted circuits as **PCKT**.

**Lemma 1.** *For the scheme* HE$^{SP}$, *the ciphertexts resulting from Encrypt*$^{SP}$ *as well as Evaluate*$^{SP}$ *applied to a permitted circuit, decrypts correctly*. □

**Theorem 1.** *The encryption scheme* HE$^{SP}$ *is correct, compact and is algebraically homomorphic for the given plaintext* M ∈ {0,1}, *and for any circuit* CKT ∈ **PCKT**. □

## 3 Security of the Proposed Variant

Since HE$^{SP}$ is a variant of HE, we can follow the same strategy as that of [5] and [8] to base the security of our proposition on the hard problem of solving a version of GACD called Partial Approximate Greatest Common Divisor (PAGCD). In [8] this problem is called as error-free approximate-GCD.

**Definition 2.** (Two-element Partial Approximate Greatest Common Divisor) The two-element (*r*, *e*, *g* )-PAGCD problem is: For a random *e*-bit odd positive integer P, given $X_0$ = P$Q_0$ and $X_1$= P$Q_1$ + R, where $Q_i$ ( i=0,1), R are chosen from the intervals $[0, 2^g / P)$, and $(-2^r, 2^r)$ respectively, output P.

The recent work of Chen and Nguyen [12] shown that solving PAGCD is relatively easier than solving GAGCD. However, as mentioned by them their attack's implementation parameters are suboptimal for medium and large challenges put forth by Coron et el [8]. Hence, if the security parameter $n$ is appropriately chosen, the PAGCD problem will be intractable ensuring the semantic security of the scheme. We have the following theorem, similar to [5] to base the security of our scheme on the two-element PAGCD problem.

**Theorem 2.** *Let* $d$, $e$, $g$, $r$ *be the parameters of the scheme* $HE^{SP}$, *which are polynomial in the security parameter* $n$. *An adversary* A *with an advantage* $\varepsilon$ *against* $HE^{SP}$ *can be converted in to an algorithm* B *for solving the two-element* ($r$, $e$, $g$ )-PAGCD *problem with success probability at least* $\varepsilon$ /2. *The running time of* B *is polynomial in the running time of* A, $n$ *and* 1/ $\varepsilon$. □

## 4 Known Attacks

In $HE^{SP}$, for a given security parameter $n$ the lowest possible size of the problem instance to solve the PAGCD problem is the public key ($X_0$, $X_1$) because, the noise in $X_1$ is less when compared to noise in ciphertexts for a particular instance of the scheme. Therefore, the attacks against the two-element PAGCD problem, i.e., against the public key only are described, claiming that the high noise ciphertexts (approximate multiples of P) successfully defend all these attacks.

**Factoring the Exact Multiple.** For the chosen parameter values, the size of the exact multiple of P i.e., $X_0$ is big enough so that, even the best known integer factoring algorithms such as the General Number Field Sieve [13] will not be able to factor $X_0$. Even if the factor P is targeted which is smaller than the size of total $Q_0$, algorithms such as Lenstra's elliptic curve factoring [14] takes about exp ($O(\sqrt{e}$ ))time to find P. But, it is to be noted that, P will not be recovered directly as it is not prime and may be further decomposed in to smaller primes.

**Brute-Force Attack on the Noise.** Given the public key integers $X_0$ and $X_1$, the simple brute-force attack can be; choosing an R form the interval ($-2^r$, $2^r$), subtracting it from $X_1$, and computing GCD($X_0$, $X_1$- R) every time, which may be the required secret integer P. In a worst case, this process may need to be repeated for all the integers R in the interval. The complexity of this attack will be $2^r$. $\tilde{O}(g)$ for $g$ bit integers.

Another integer more vulnerable to brute-force attack in $HE^{SP}$ is the noise factor N used during the encryption. In fact, this integer clearly defines the overall security of the scheme because, guessing this number simply breaks the scheme, rather than guessing the secret integer P. The attack in the case of this integer will be, choosing all the possible even integers N from the interval mentioned, and encrypting 0 with each such N and public key. Then, for a plaintext bit encrypted using some N, the

difference between the corresponding ciphertext and a ciphertext that encrypted 0 using the same N will be only in the least significant bit. The complexity of this attack will be exponential in the size of N that is $2^r$ and choosing $r = \omega(\lg n)$ foils this attack.

**Continued Fractions and Lattice Based Attacks.** Howgrave Graham [11] described two methods to solve the two-element PAGCD problem. In simple terms the continued fraction based approach (Algorithm 11, [11] ) recovers P if the condition $R < P/Q$ is satisfied. Similarly, his lattice based algorithm (Algorithm 12, [11]) recovers P if the condition $R < P^2 / (PQ)^{\varepsilon}$ is satisfied for some real number ε. Also, as analyzed in [5] for the case of a two-element PAGCD problem, it is possible to recover P when $r/g$ is smaller than $(e / g)^2$ . Since the parameter setting of $HE^{SP}$ does not satisfy these constraints, the concerned methods fail to recover the value of P.

**The General Common Divisors Attack.** Consider the Theorem 31.2 and its corollaries discussed in [15]. $GCD(X_0, X_1)$, can be the smallest positive element in the set $\{AX_0 + BX_1 : A,B \in \mathbb{Z}\}$. This is possible because, A, B can be any integers including negative numbers. Now, if a common divisor exists for both $X_0$, $X_1$, it will divide all the possible linear combinations of $X_0$, $X_1$. Modular reduction of a ciphertext with such common divisor results in the plaintext, because a ciphertext contains a linear combination of $X_0$, $X_1$. Therefore, taking the pair of integers $X_0$, $X_1$ as co-prime foils this attack.

## 5 Improvement in Bit Complexity

As discussed earlier, the public key of the HE contains $\tilde{O}(n^5)$ elements each of which is $\tilde{O}(n^5)$ bits long. This will take $\tilde{O}(n^{10})$ computations for complete key generation. Also, in that scheme the bit length of a fresh ciphertext that encrypts a single bit is $\tilde{O}(n^5)$, leading to an expansion ratio of $n^5$ .

The public key in the scheme $HE^{SP}$ consists of only two elements of $\tilde{O}(n^3)$ bits long. This makes the complexity of key generation as $\tilde{O}(n^3)$. This is a considerable improvement over the somewhat homomorphic schemes of [5] and [8]. Also, the encryption of an $\tilde{O}(n)$ bit plaintext, which involves a multiplication of $\tilde{O}(n^3)$. $\tilde{O}(n)$ and a modular reduction of this with $\tilde{O}(n^3)$ bit $X_0$ takes $\tilde{O}(n^3)$ steps. Similarly, the bit complexity of decryption is roughly $\tilde{O}(n^3)$. Therefore, the overall complexity of the proposed variant $HE^{SP}$ is $\tilde{O}(n^3)$. Similarly, a single plaintext bit is embedded in a ciphertext of $\tilde{O}(n^3)$ bits making the expansion ratio also comparatively less which is $n^3$. With these drastic improvements in bit complexity and ciphertext expansion, this conceptually simple somewhat homomorphic scheme will be suitable for many practical applications that involve simple functions for homomorphic evaluation (The degree of the polynomial approximation of such functions should be within the homomorphic evaluation capacity of the scheme).

## 6 Conclusions

In this paper, an efficient and hopefully a practical variant of the existing Somewhat Homomorphic Encryption over the integers is proposed. The improvement in efficiency, from $\tilde{O}(n^{10})$ to $\tilde{O}(n^3)$, is obtained by reducing the size of the public key, which contains only two integers. The semantic security of the scheme is thoroughly analyzed by reducing the same to the hard problem of solving the two-element Partial Approximate Greatest Common Divisor, describing all the known attacks. With the improvement in bit complexity, it is expected that the Homomorphic Encryption usage and thus encrypted data processing becomes imminent practically.

## References

1. Rivest, R., Adleman, L., Dertouzos, M.: On data banks and privacy homomorphisms. In: Foundations of Secure Computation, pp. 169–180 (1978)
2. Gentry, C.: A Fully homomorphic encryption scheme. Ph.D. thesis, Stanford Univ. (2009)
3. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC 2009, pp. 169–178. ACM (2009)
4. Smart, N.P., Vercauteren, F.: Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 420–443. Springer, Heidelberg (2010)
5. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully Homomorphic Encryption over the Integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010)
6. Gentry, C.: Computing arbitrary functions of encrypted data. Communications of the ACM 53(3), 97–105 (2010)
7. GovindaRamaiah, Y., VijayaKumari, G.: State-of-the-art and Critique of Cloud Computing. In: NCNGCIS 2011, pp. 50–60. IMS, Noida (2011)
8. Coron, J.-S., Mandal, A., Naccache, D., Tibouchi, M.: Fully Homomorphic Encryption over the Integers with Shorter Public Keys. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 487–504. Springer, Heidelberg (2011)
9. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: Fully Homomorphic Encryption without Bootstrapping. Electronic Colloquium on Computational Complexity (ECCC) 18, 111 (2011)
10. Vaikuntanathan, V.: Computing Blindfolded: New Developments in Fully Homomorphic Encryption, http://www.cs.toronto.edu/~vinodv/FHE-focs-survey.pdf
11. Howgrave-Graham, N.: Approximate Integer Common Divisors. In: Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, pp. 51–66. Springer, Heidelberg (2001)
12. Chen, Y., Nguyen, P.Q.: Faster algorithms for approximate common divisors: Breaking fully homomorphic encryption challenges over the integers. Cryptology ePrint Archive, Report 2011/436, http://eprint.iacr.org/2011/436
13. Briggs, M.: An Introduction to the General Number Field Sieve. Master's Thesis, Virginia Tech (April 1998), http://scholar.lib.vt.edu/theses/available/etd-32298-93111/
14. Lenstra, H.: Factoring Integers with Elliptic Curves. Annals of Mathematics 126, 649–673 (1987)
15. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press (2002)