Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

A DETAILED STUDY ON HOMOMORPHIC ENCRYPTION

A.Saranyadevi M.E., [1], S.Anguraj M.E., Ph.D., [2] ,S.Senbhaga M.E.,[3]
[1]*Department of Computer Science, K.S.R.College of Engineering*
[2,3]*Assistant Professor, Department of Computer Science, K.S.R College of Engineering*

Abstract—Homomorphic encryption is the encryption strategy which tells that the functioning on the encrypted data. Homomorphic encryption can be applied in any structure by using different public key algorithms. When the data is conveyed to the public area, there are numerous encryption algorithms to tied the functions and the storage of the data. But to procedure data discovered on remote server and to conserve privacy, homomorphic encryption is useful that permits the functioning on the cipher text, which can get the same results after calculations as the operating directly on the sore data. In this paper, the most prominent focal point is on public key cryptographic algorithms based on homomorphic encryption strategy for conserving security.

Keywords—Homomorphic Encryption, Partially Homomorphic Encryption, Benaloh Cryptosystem, The Naccache-Stern Cryptosystem. Pailier Cryptosystem

I. INTRODUCTION

The evolution of homomorphic encryption gives still another clear-cut approach. Informally, a homomorphic encryption strategy permits operation directly on encrypted data. Alice can now encrypt the input x and send the ciphertext to Bob. Bob will calculate f(x) straightforwardly on the ciphertext and send back the encrypted result that only Alice can decrypt. In this way, Bob will not be able to learn anything about x as long as the security of the homomorphic encryption strategy grips. Homomorphic properties of standard public key encryption strategy, e.g. RSA and ElGamal encryption, were identified early on. However they were mainly observed as feebleness rather than an asset. Applications where data is stable typically need non-malleable encryption. However, the community has grown to believe the security of these strategies and, recently, the work of Gentry and others show that, when carefully worked, such homomorphic properties can be quite precious. Indeed, a number of recent specific applications such as data aggregation in distributed networks, electronic voting, biometrics and privacy preserving data mining have led to reignited interest in homomorphic strategy.

Numerous robust HE strategys were suggested during the past decades. One of the earliest discoveries relevant here was the Goldwasser-Micali cryptosystem whose security is based on the quadratic residuosity problem which allows homomorphic estimation of a bitwise exclusive-or (XOR). Other additive homomorphic encryption strategy that gives semantic security is Benaloh, Naccache-Stern, Paillier, Damg°ard-Jurik, Okamoto-Uchiyama and Boneh-Goh-Nissim. Some additively homomorphic encryption strategy use lattices or linear codes. These HE strategies only support homomorphic estimation of definite classes of circuits. E.g. the Goldwasser-Micali Strategy only supports bitwise exclusive-or, Paillier's Strategy only supports additions.

This paper presents the concepts, functions and properties about Pailier cryptosystem, Benaloh cryptosystem and Naccache-stern Cryptosystem. The aim of this paper is to present a survey of some partial homomorphic encryption techniques. In Section 2, we present some basic concepts

on cryptography and some types of encryption strategys. In Section 3, we discuss some of fundamental description about homomorphic encryption strategys in the literature. Section 4 presents a discourse on partial homomorphic encryption strategys which are the great robust encryption strategys for presenting a skeleton for operating over encrypted data. Finally, Section 5 concludes the chapter while contour the number of research directions and promising trends in this stirring field of operation which has a marvelous possible of discovering applications in the real-world deployments.

II. TOWARDS HOMOMORPHIC ENCRYPTION

2.1. Basics about encryption

In this section, we will remind some main concepts regarding encryption strategy. For more exact information, Encryption strategys are, first and leading, intended to conserve confidentiality. According to Kerckoffs' principle, their security must not rely on the unclear of their code, but only on the secrecy of the decryption key. We can make a distinction two kinds of encryption strategys: *symmetric* and *asymmetric* ones.

Symmetric encryption strategy

Here "symmetric" means that encryption and decryption are carried out with the same key. Hence, the sender and the receiver have to agree on the key they will use before carried out any secure communication. Therefore, it is not possible for two people who never met to use such strategy directly. This also implies to share a different key with everyone we want to communicate with. Nevertheless, symmetric strategies present the advantage of being really fast and are used as often as possible. In this category, we can differentiate block ciphers (AES) and stream ciphers (One-time Pad) presented which are even faster.

Asymmetric encryption strategy

In contrast to the previous family, asymmetric strategy introduce a fundamental difference between the abilities to encrypt and to decrypt. The encryption key is public, as the decryption key remains private. When Bob desires to mail an encrypted message to Alice, he uses her *public key* to encrypt the message. Alice will then use her *private key* to decrypt it. Such strategies are more functional than symmetric ones since there is no need for the sender and the receiver to be in agreement on anything before the transaction. Moreover, they often provide more features. These strategy, however, have a big problem: they are based on nontrivial mathematical computations , and much slower than the symmetric ones.

Probabilistic encryption

The most well-known cryptosystems are *deterministic*: for a rigid encryption key, a given plaintext will always be encrypted in the same ciphertext. This may lead to some cons.RSA is a good example to illustrate this point:

(1) particular plaintexts may be encrypted in a too much planned way: with RSA, messages 0 and 1 are constantly encrypted as 0 and 1, respectively;
(2) it may be easy to calculate half-done information about the plaintext: with RSA, the ciphertext c leaks one bit of information about the plaintext m, namely, the so called Jacobi symbol;

(3) when use a deterministic encryption strategy, it is easy to detect when the same message is mail twice while processed with the same key.So, in practice, we desire encryption strategy to be probabilistic.

In the case of symmetric strategy, we initiate a random vector in the encryption process (e.g., in the pseudorandom creator for stream ciphers, or in the working mode for block ciphers), commonly called *IV*. This vector may be public, and transferred as it is, without being encrypted, but *IV* must be altered every time we encrypt a message. For asymmetric ciphers, the security analysis is more mathematical, and we desire the randomized strategy to remain discoverable in the same way as the deterministic strategy. Some adequate modes have been proposed to randomize already published deterministic strategy, as the Optimal Asymmetric Encryption Padding OAEP for RSA (or any strategy based on a trap-door one-way permutation).Some new strategy, randomized by nature, have also been proposed. A simple consequence of this requirement to be probabilistic become visible in the so-called *expansion*: since for a plaintext we require the subsistence of numerous possible ciphertexts, the number of ciphertexts is greater than the number of potential plaintexts. This means the ciphertexts cannot be as small as the plaintexts, they have to be rigorously larger. The ratio between the length, in bits, of ciphertexts and plaintexts is called the *expansion*. Of course, this parameter is of practical importance. We will see that efficient probabilistic encryption strategy have been proposed with an expansion less than 2 (e.g., Paillier's strategy).

III. HOMOMORPHIC ENCRYPTION

The security requirements for data and algorithms have become very tough in the past decades. Due to the huge development of technology, an enormous amount of attacks on digital goods and technical devices are allowed. For storing and reading data protectively there exist numerous possibilities such as secure data encryption. The problem becomes more complex when asking for the possibility to calculate (publicly) with encrypted data or to alter functions in such a way that they are still executable while our privacy is provided. That is where homomorphic cryptosystems can be used.

Homomorphic encryption

Homomorphic encryption is the encryption strategy which tells that the functioning on the encrypted data. Homomorphic Encryption that permits the functioning on the cipher text, which can get the same results after calculations as the operating directly on the sore data. Homomorphic encryption is a form of encryption which allows specific types of calculations to be taken out on ciphertext and get an encrypted result which decrypted matches the result of operations carried out on the plaintext. A public-key encryption strategy E = (KeyGen, Enc, Dec) is homomorphic if for all k and all (pk, sk) output from KeyGen (k), it is able to define M, C so that: The plaintext M, and all ciphertexts output by Enc_{pk} are elements of C. For some m1, m2 ∈ M and c1, c2 ∈ C with m1 = Dec_{sk} (c1) and m2 = Dec_{sk} (c2) it holds that Dec_{sk} ($c1 * c2$) = $m1 * m2$ where the multiplicative operations * are taken out in C and M, respectively. Here the properties are additive homomorphic and multiplicative homomorphic using addition and multiplication operation respectively. And the functions are Key Generation, Encryption, Evaluation and Decryption.

Fully homomorphic encryption

This strategy is possible to give a ciphertext that encrypts f (m1,..., mt), where f is any preferred function, which of course must be proficiently calculable. No information about m1, ..., smt or f (m1, ..., mt), or any intermediary plaintext values should trickle The inputs, outputs and intermediary values are always encrypted and therefore ineffective for an opponent. A public key encryption strategy (KeyGen, Enc, Dec) is fully homomorphic if there exists an additional proficient

algorithm Eval that, for a valid public key pk, a allowed circuit C and a set of ciphertexts μ = {c1, ..., ct} where ci ← Encpk (mi), outputs c ← Evalpk (C, μ) under pk.

IV. SOME PARTIALLY HOMOMORPHIC ENCRYPTION

1. BENALOH CRYPTOSYSTEM

Benaloh's \Dense Probabilistic Encryption" explains a homomorphic encryption strategy with a significant enhancement in terms of expansion factor equated to Goldwasser-Micali. The main enhancement of the Benaloh Cryptosystem over GM is that lengthy blocks of data can be encrypted at once, whereas in GM each bit is encrypted individually. It also generalizes quadratic residuosity to r-th residuosity. For the same security parameter (the size of the RSA modulus n), the ciphertext is in both instance an integer mod n, but in Benaloh's strategy is an integer mod r for some parameter r depending on the key, whereas in Goldwasser-Micali is only a bit.

Key Generation: The public and private key are created as follows:

- Choose a block size r and two large primes p and q such that:
  - r divides (p - 1).
  - r and (p - 1)/r are reasonably prime.
  - r and q - 1 are reasonably prime.
  - n = pq.

- Select $y \varepsilon (Z_n)^* = \{ x \varepsilon Z_n : \gcd(x,n) = 1\}$ such that

$$Y^{\phi/r} \neq 1 \bmod n$$

where ϕ denotes (p-1)(q-1).
The public key is (y, r, n), and the private key is the two primes p and q.

Encryption: If m is an element in Z_r and u a random number in $(Z_n)^*$ then we calculate the randomized encryption of m using the following formula:

$$E_r(m) = \{y^m u^r \bmod n : u \varepsilon (Zn)^*\}$$

It is easily verified that:

$$E_r(m1) * E_r(m2) = E_r(m1 + m2)$$

Decryption: We first notice that for any m, u we have:

$$(y^m u^r)^{(p-1)(q-1)} = r = y^{m(p-1)(q-1)/r} u^{(p-1)(q-1)} = y^{m(p-1)(q-1)/r} \bmod n$$

$$y^{(p-1)(q-1)/r} \equiv 1 \bmod n$$

As $m<r$ and, we can conclude that if and only if $m = 0$
So if is an encryption of *m*, given the secret key *p*, *q* we can determine whether $m=0$

$$Z=y^m u^r \bmod n$$

$$(y^m u^r)^{(p-1)(q-1)/r} \equiv 1 \bmod n$$

If *r* is small, we can decrypt *z* by doing an exhaustive search, i.e. decrypting the messages *y-iz* for *i* from 1 to *r*. By precomputing values, using the Baby-step giant-step algorithm, decryption can be done in time $o\left(\sqrt{r}\right)$.

Homomorphic Properties

The Benaloh cryptosystem supports the homomorphic addition and subtraction of ciphertexts. Given two ciphertexts $C_1=y^{m1}u_1^{r}$ and $C2=y^{m2}u_2^{r}$,

$C_1 C_2 \bmod n = (y^{m1}u_1^{r})(y^{m2}u_2^{r}) \bmod n$
is a valid decryption of m_1+m_2, and

$$C_1 C_2^{-1} \bmod n = (y^{m1}u_1^{r})(y^{m2}u_2^{r})^{-1} \bmod n$$
$$= (y^{m1}u_1^{r})(y^{m2}(u_2^{-1})^{r}) \bmod n$$
$$= y^{m1-m2}(u_1u_2^{-1})^{r} \bmod n$$

is a valid encryption of m1-m2 .

Security

The security of this strategy rests on the Higher residuosity problem, specifically, given *z*,*r* and *n* where the factorization of *n* is unknown, it is calculationally impracticable to find whether *z* is an *r*th residue mod *n*, i.e. if there exists an *x* such that $Z\equiv x^{r} \bmod n$

2. THE NACCACHE-STERN CRYPTOSYSTEM

Naccache-Stern homomorphic cryptosystem is an improvement on the Benaloh cryptosystem, based on the hardness of computing higher residues modulo a composite RSA integer, to resolve the problem of r which has to be small for the decryption to be efficient. The encryption phase is analogously as Benaloh (and hence its expansion factor),and the expansion is still equal to l(n)/l(k). The decryption complexity however is of magnitude $O(l^5(n)\log(l(n)))$, which is much more efficient when r becomes large, and the authors claim it is reasonable to choose the parameters as to get an expansion equal to 4.

Prerequisites: The main difference with Benaloh is that the setup is slightly different. Let r be a B-smooth squarefree number. I.e., $r = \pi_i p_i$, with $p_i < B$, and all p_i different. Also, g does not need to be a generator, but it's order does need to be a large order multiplicative of r.

Let σ be a squarefree odd B-smooth integer, where B is small integer and let n = pq be an RSA modulus such that σ divides ϕ(n) and is prime to ϕ(n)/σ. Generation of the modulus appears rather straightforward: select a family p_i of k small odd distinct primes, with k even. Set $u = \prod_{i=1}^{k/2} p_i$, $v = \prod_{k/2+1}^{k} p_i$ and $\sigma = uv = \prod_{i=1}^{k} p_i$. choose two large primes a and b such that both p = 2au + 1 and q = 2bv + 1 are prime and let n = pq. To select g, one can choose it at random and verify whether or it has order (n) = 4.The public key is the numbers σ, n, g and the private key is the pair p, q. When k = 1 this is effectively the Benaloh cryptosystem.

Encryption: Similar to Benaloh:

$$E_r(m) = g^{m} u^{r} \bmod n$$

Decryption: The decryption is based on the Chinese remainder theorem, subsequent an idea which goes back to the Pohlig-Hellman paper.

For each pi that is a factor of r, $c^{\phi(n)/pi}$ is compared with the pre-computed values $g^{i\phi(n)/pi}$. If they are equal, then, $m \equiv i \bmod p_i$. Because all $p_i$are smaller than B, this is efficient. Via Chinese remainder theorem, m mod $\pi_i p_i$ is computed.

Homomorphic Properties

The Naccache-Stern cryptosystem allows for the homomorphic addition and subtraction of ciphertexts, as well as multiplication of a ciphertext by a constant.

Examples for the probabilistic variant are provided, but work equivalently for the basic cryptosystem.

Given two ciphertexts $c_1 = g^{m1}x_1^{\sigma} \bmod n$ and $c_2 = g^{m2}x_2^{\sigma} \bmod n$ as valid encryptions of m_1 and m respectively, then

$$c_1c_2 \bmod n = g^{m1}x_1^{\sigma}g^{m2}x_2^{\sigma} \bmod n$$

$$= g^{m1+m2}(x_1x_2)^{\sigma} \bmod n$$

is a valid encryption of $m_1 + m_2$ and

$c_1^{k} \bmod n = g^{km1}(x_1^{k})^{\sigma} \bmod n$

is a valid encryption of km_1. Subtraction is possible by taking $c_1c_2^{-1}$.

Security

The semantic security of the Naccache–Stern cryptosystem rests on an expansion of the quadratic residuosity problem known as the higher residuosity problem.

3. PAILLIER CRYPTOSYSTEM

The Paillier cryptosystem is the famous cryptosystem that we wanted to cover in the first place. As one of the most well-known homomorphic encryption strategys, it is an upgrading of the Okamoto-Uchiyama cryptosystem that decreases the expansion from 3 to 2. This cryptosystem is based on the Composite Residuosity (CR) assumption, meaning that the problem of computing n-th residue classes is computationally difficult.

Paillier came back to n = pq, with gcd (n, ϕ(n)) = 1, but with the group G = Z_{n2}^{*}, and a proper choice of H led to k = l(n). The encryption cost is practically low. Decryption desires one exponentiation modulo n^2 to the power λ (n), and a multiplication modulo n, and it can b managed efficiently through the Chinese Remainder Theorem.

Applying it to Paillier's original strategy, there are several stronger variants proposed. Cramer and Shoup proposed a general move toward to gain security against adaptive chosen-ciphertext attacks for definite cryptosystems with some particular algebraic properties. Bresson et al. proposed a slightly different version that may be more accurate for some applications.

The main observation for this cryptosystem is the following equation:

$(1 + x)^{m} = 1 + mx \bmod x^2$.

Mathematical Basis: The security of this system relies on the problem of deciding n-th residuosity.

The Cryptosystem

Prerequisites

- Choose two large primes p, q randomly and independently of each other.

- Compute n = pq and ¸ = lcm (p − 1, q − 1).
- Choose a random g ∈ Z^*_{n2}, ensure n divides the order of g, by checking the existence of the subsequent modular multiplicative inverse μ = (L(g^{λ}¸ mod n^2))$^{-1}$mod n.
- The function L is defined as L (u) = u−1/n .
- The public key is (n, g), the private key.

Encryption

- Let m be a plaintext to be encrypted where m ∈ Zn
- Choose random r where r ∈ Z^*_n.
- Compute ciphertext as: c = $g^{m.}r^n$mod n^2.

Decryption

- Ciphertext c ∈ Z^*_{n2}

Decryption is essentially one exponentiation modulo n2. The analysis of the main practical aspects of computations required by the cryptosystem and different implementation strategies for improved performance, as well as how the Chinese Remainder Theorem can be used to powerfully reduce the decryption workload of the cryptosystem.

Homomorphic properties : The strategy is an additive homomorphic cryptosystem; this means that, given only the public-key and the encryption of m1 and m2, one can calculate the encryption of m_1+m_2. The homomorphic properties of Paillier's strategy.

D (E_{r1} (m_1) Er_2 (m_2) mod n^2) = m_1+ $m_2$mod n

D(E_r (m_1) g^{m2}) = m_1+ $m_2$mod n

D(E_r (m_1)^{m2}) = m_1 · $m_2$mod n

The cryptosystem explained above provides semantic security against chosen-plaintext attacks (IND-CPA). The capability to successfully differentiate the challenge ciphertext essentially amounts to the capability to decide composite residuosity, let the decisional composite residuosity assumption (DCRA) is supposed to be inflexible. For aforementioned homomorphic properties however, the strategy is malleable, and therefore does not enjoy the highest echelon of semantic security that secure against adaptive chosen-ciphertext attacks (IND-CCA2).

V. CONCLUSION

We presented in this paper homomorphic encryption strategys discussing their parameters, performances and security issues. As we saw, these strategy are not well suited for every use, and their characteristics must be taken into account. Nowadays, such strategy are studied in wide application contexts, but the research is still challenging in the cryptographic community to design more powerful secure strategy. Performing computations using homomorphic encryption strategy nowadays takes quite a long time, but as approaches evolve things will quickly change. Researchers believe in the possibility of advancing in homomorphic encryption area and bringing new related technologies to the wide market. It can be used whenever the need of doing calculations on pieces of

un-owned information appears. We therefore conclude that focusing on these topics would be a good approach for further research.

REFERENCES

[1] Josh Benaloh. Dense probabilistic encryption. pages 120–128.
[2] R. Rivest, L. Adleman, and M. Dertouzos, "On data banks and privacy homomorphisms," in Foundations of Secure Computa- tion, pp. 169–177, Academic Press, 1978.
[3] S. C. Pohlig and M. E. Hellman. An improved algorithm for computing logarithms over gf (p) and its cryptographic significance. IEEE Transactions on Information Theory, IT-24-1:106–110, 1978.
[4] Goldwasser, S. & Micali, S. (1984). Probabilistic Encryption. Journal of Computer and System Sciences, Vol 28, Issue 2, pp. 270-299, April 1984.
[5] David Naccache and Jacques Stern. A new public key cryptosystem based on higher residues. In ACM Conference on Computer and Communications Security, pages59–66, 1998.
[6] Ivan Damg°ard and Mads Jurik. A generalization, a simplification and some applications of paillier's probabilistic public-key system. In proceedings of PKC 01, Lecture Notes in Computer Science, pages 119–136. Springer-Verlag, 2001.
[7] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai,Salil Vadhan, and Ke Yang. On the (Im) possibility of Obfuscating Programs. In Advances in Cryptology - CRYPTO 2001, volume 2139 of Lecture Notes in Computer Science, pages 1–18. Springer-Verlag, 2001.
[8] Emmanuel Bresson, Dario Catalano, and David Pointcheval. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In Advances in Cryptology - ASIACRYPT 2003, Lecture Notes in Computer Science.Springer-Verlag.
[9] Caroline Fontaine and Fabien Galand. A survey of homomorphic encryption fornon-specialists. 2007.
[10] Castagnos, G. (2007). An Efficient Probabilistic Public-Key Cryptosystem over Quadratic Fields Quotients. Finite Fields and Their Applications, Vol 13, No 3, pp. 563-576, July 2007.
[11] Brecht Wyseur and Mina Deng (K.U.Leuven) .Encrypted Code Final Report, August 2009.
[12] Gu Chun-sheng (2012)Attack on Fully Homomorphic Encryption over the Integers- International Journal of Information & Network Security (IJINS) Vol.1, No.4, October 2012, pp. 275~281

