Cryptanalysis of fully homomorphic encryption schemes

Ahmed EL-YAHYAOUI, Mohamed Dafir ELKETTANI

Information Security Research Team, CEDOC ST2I ENSIAS, Mohammed V University in Rabat, Rabat, Morocco

***Abstract*- MORE (Matrix Operation for Randomization or Encryption) is a free-noise fully homomorphic encryption scheme [8]. It is a symmetric scheme invented in 2012 by Kipnis and Hibshoush. The homomorphy of this cryptosystem is obtained via matrix operations. Although the authors said that the security of the algorithm is based on the hardness of factoring big integers, it suffers from new vulnerabilities.**

The Brenner et al, homomorphic encryption scheme's [11] is a simple algebraic somewhat homomorphic cryptosystem that is supposed to became fully homomorphic after using a new refreshment procedure introduced by Iti Sharma in [10], but unfortunately this refresh method fails in general.

In this paper we will provide a cryptanalysis of the MORE cryptosystem based on a single known plaintext and we will show the fail of the Iti Sharma's refreshment procedure.

***Keywords*: homomorphic encryption, MORE, cryptanalysis, refreshment procedure, cloud, attack.**

I. INTRODUCTION

Fully homomorphic encryption (FHE) is a genius solution to delegate secure computations. It consists of doing operations on encrypted data without need to decryption. Cloud computing is one of the recipients of this technological progress. In fact, cloud offers today a highly available storage and huge parallel computing resources with high performance computing capacities and low cost. However, taking advantage from cloud's performance computing is risked by information leakage and theft of privacy. Secure computation of arbitrary functions on confidential data is highly demanded to profit from cloud computing performances'. In a paper entitled "on data banks and privacy homomorphisms [1]" and dated 1978, Rivist et al have established the underlying notion of homomorphic encryption. Their conjecture remained unsolved until 2009, when Gentry proposed his breakthrough [2]. In his thesis, Gentry presented a fully homomorphic encryption scheme based on the hardness of Euclidean lattices problems. Gentry's cryptosystem [2] is considered the first plausible work which allows arbitrary computations on ciphertexts. Gentry have not designed just a single scheme but he presented a framework to construct fully homomorphic encryption schemes. His framework is consisting of two principal steps:

- Somewhat Homomorphic Encryption Scheme (SWHE): Gentry started from a scheme said SWHE or simply homomorphic which supports a limited number of homomorphic multiplications and reduces the complexity of its decryption circuit to obtain a squashed SWHE scheme.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

- Bootstrapping: This step consists of removing the noise resulted by multiplications in order to allow evaluating circuits of arbitrary depth.

A lot of works following after had inspired from Gentry's framework [3, 4, 5, 6]. These cryptosystems can be classified in the category of noise based fully homomorphic encryption schemes [7]. A second category of FHE schemes has appeared after which is free-noise based [7]. Bootstrapping technique is not required in these cryptosystems and its homomorphy is obtained, in general, via matrix operations. The security of most of these cryptosystems is undermined because it suffers from vulnerabilities.

This article is dedicated to cryptanalysis of two symmetric FHE schemes [8, 10]. The rest of the paper is organised as: In the second part we will present some notations and definitions, after we will present the MORE encryption scheme and its cryptanalysis in third and fourth parts. In part five we will give a toy example of our attack. Finally, in the next part we will provide the Iti Sharma FHE scheme [10] and its fail.

II. NOTATION AND DEFINITIONS

For a matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, $\det(A) = ad - bc$ denote the determinant of A, $tr(A) = a + d$ denote the trace of A and $Antitr(A) = c + d$ denote the anti-trace of A.

If $\det(A) \neq 0$ so A is invertible, $A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$ will denote its inverse and it verify $A.A^{-1} = I$.

A. Fully homomorphic encryption

A fully homomorphic encryption scheme can be defined as a tuple of three algorithms $\mathcal{E} = (KeyGen, Encrypt, Decrypt)$ for which the message space is a ring $(R, +, .)$ and the ciphertext space is also a ring $(R', \oplus, \otimes)$ such that for all messages $m_1, m_2 \in R$, and all outputs $(pk, sk) \leftarrow KeyGen(1^\lambda)$, we have:

$$m_1 + m_2 = Decrypt(Encrypt(m_1, pk) \oplus Encrypt(m_2, pk), sk)$$

$$m_1 . m_2 = Decrypt(Encrypt(m_1, pk) \otimes Encrypt(m_2, pk), sk)$$

If $\mathcal{E}$ is a symmetric fully homomorphic encryption scheme, we will have a single key for encryption and decryption, so the role of pk will be played by sk.

A scheme is supposed to be somewhat homomorphic if it permits only a limited number of additions and multiplications.

B. Ciphertexts indistinguishability

Ciphertext indistinguishability is an important security property of modern cryptography. If a cryptosystem possesses the property of indistinguishability, it implies that adversaries will be unable to distinguish pairs of ciphertexts based on the message they encrypt.

An encryption scheme is supposed to be secure in terms of indistinguishability if no adversary $\mathcal{A}$, given a ciphertext of a cleartext randomly chosen from a two-element message space determined by the adversary, can identify the

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

cleartext choice with a probability significantly better than that of hazard (1/2). If any adversary can succeed in distinguishing the chosen ciphertext with a probability significantly greater than 1/2, then this adversary is considered to have an ‘‘advantage” in distinguishing the ciphertext, and the scheme is considered insecure in terms of indistinguishability[9].

Indistinguishability under known-plaintext attack (IND-KPA):

In this situation the attacker has access to couples of plaintext/ciphertext.

IND-KPA game:

-Setup phase: Challenger generates random key $sk \longleftarrow G(k)$.

-Challenge phase:

- Adversary choses two messages (m_0^*, m_1^*) of the same length, and sends them to challenger.
- Challenger choses a random bit $b \in \{0,1\}$. She sends $c^* \longleftarrow E_{sk}(m_b^*)$ to adversary.

-Guessing phase: Adversary outputs a bit b^*, she wins if $b^* = b$.

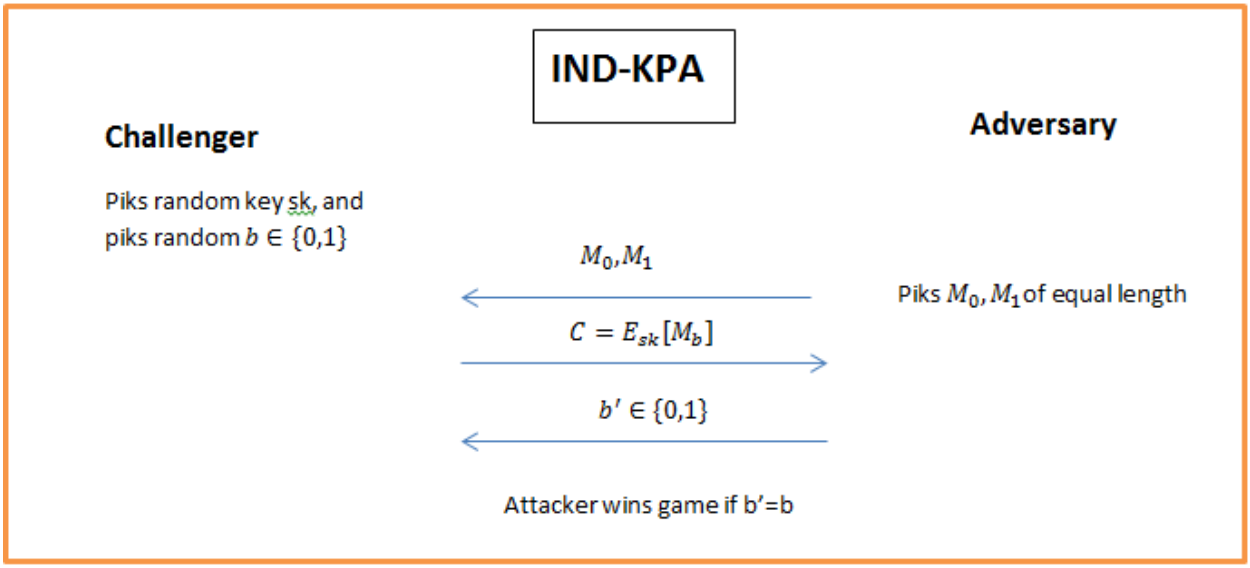

Figure 1: Known Plaintext Attack scenario

Indistinguishability under chosen-plaintext attack (IND-CPA):

In this situation the attacker can encrypt messages of his choice.

IND-CPA game:

-Setup phase: Challenger generates random key $sk \longleftarrow G(k)$.

-Query phase 1:

- For $i = 1\ to\ \gamma$ do:
  - Adversary $\mathcal{A}$ sends to challenger query m_i.
  - Challenger replies with $c_i \longleftarrow E_{sk}(m_i, r_i)$ for fresh random r_i.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

-Challenger phase:

- $\mathcal{A}$ Chooses two messages (m_0^*, m_1^*) of the same length, and sends them to challenger.
- Challenger choses a random bit $b \in \{0,1\}$ and fresh random string r^* She sends $c^* \longleftarrow E_{sk}(m_b^*, r^*)$ to adversary.

-Query phase 2: As query phase 1, but for γ_2 queries.

- Guessing phase: Adversary outputs a bit b^*, she wins if $b^* = b$.

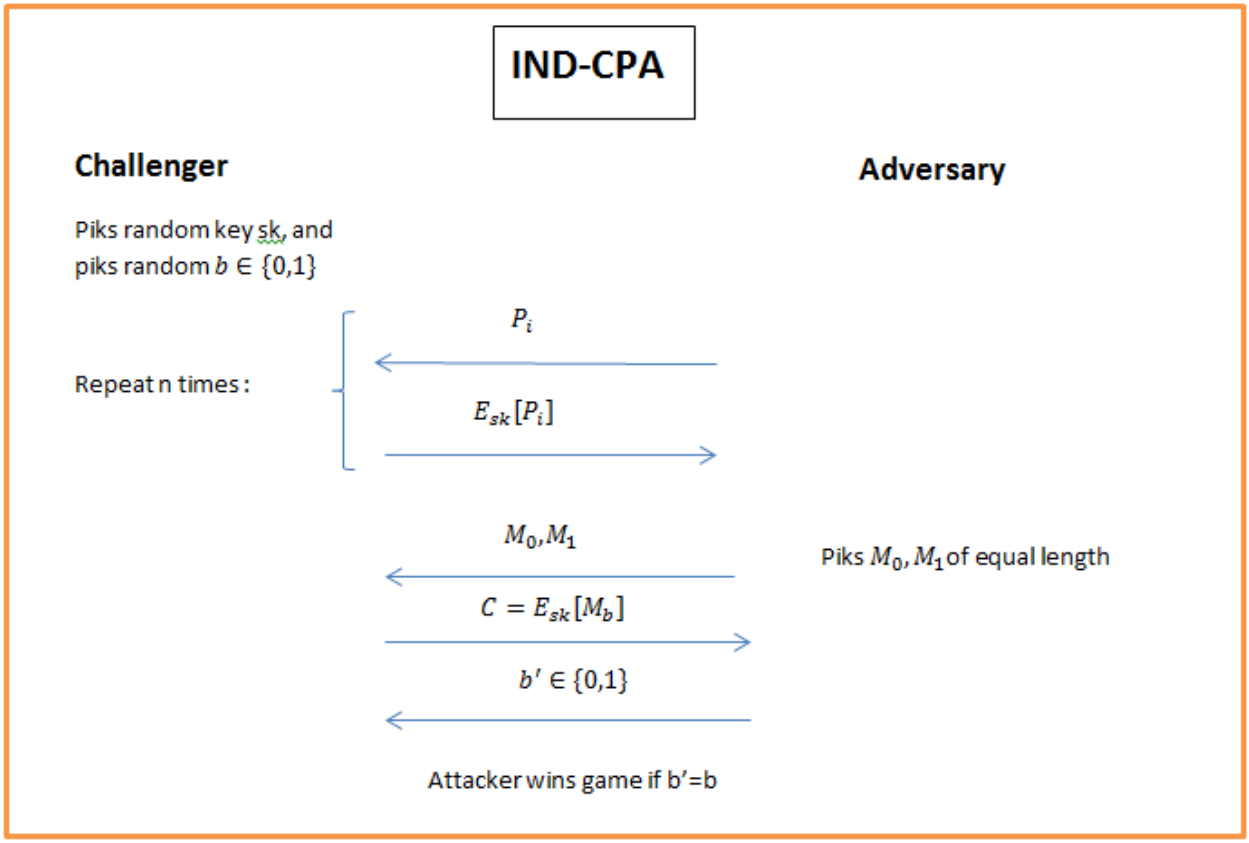

Figure 2: Chosen Plaintext Attack scenario

Indistinguishability under chosen-ciphertext attack (IND-CCA):

In this situation can decrypt arbitrary messages.

IND-CCA game:

-setup phase: Challenger generates random key $sk \longleftarrow G(k)$.

- Query phase 1:

- $\mathcal{A}$ has adaptive access to encryption and decryption oracles.

- Challenger phase:

- $\mathcal{A}$ Chooses two messages (m_0^*, m_1^*) of the same length, and sends them to challenger.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

- Challenger choses a random bit $b \in \{0,1\}$ and fresh random string r^* She sends $c^* \leftarrow E_{sk}(m_b^*, r^*)$ to adversary.

-Query phase 2:

- $\mathcal{A}$ has adaptive access to encryption and decryption oracles.
- Except she is not allow to query $D_{sk}(.)$ on input c^*.

- Guessing phase: $\mathcal{A}$ outputs a bit b^*, she wins if $b^* = b$.

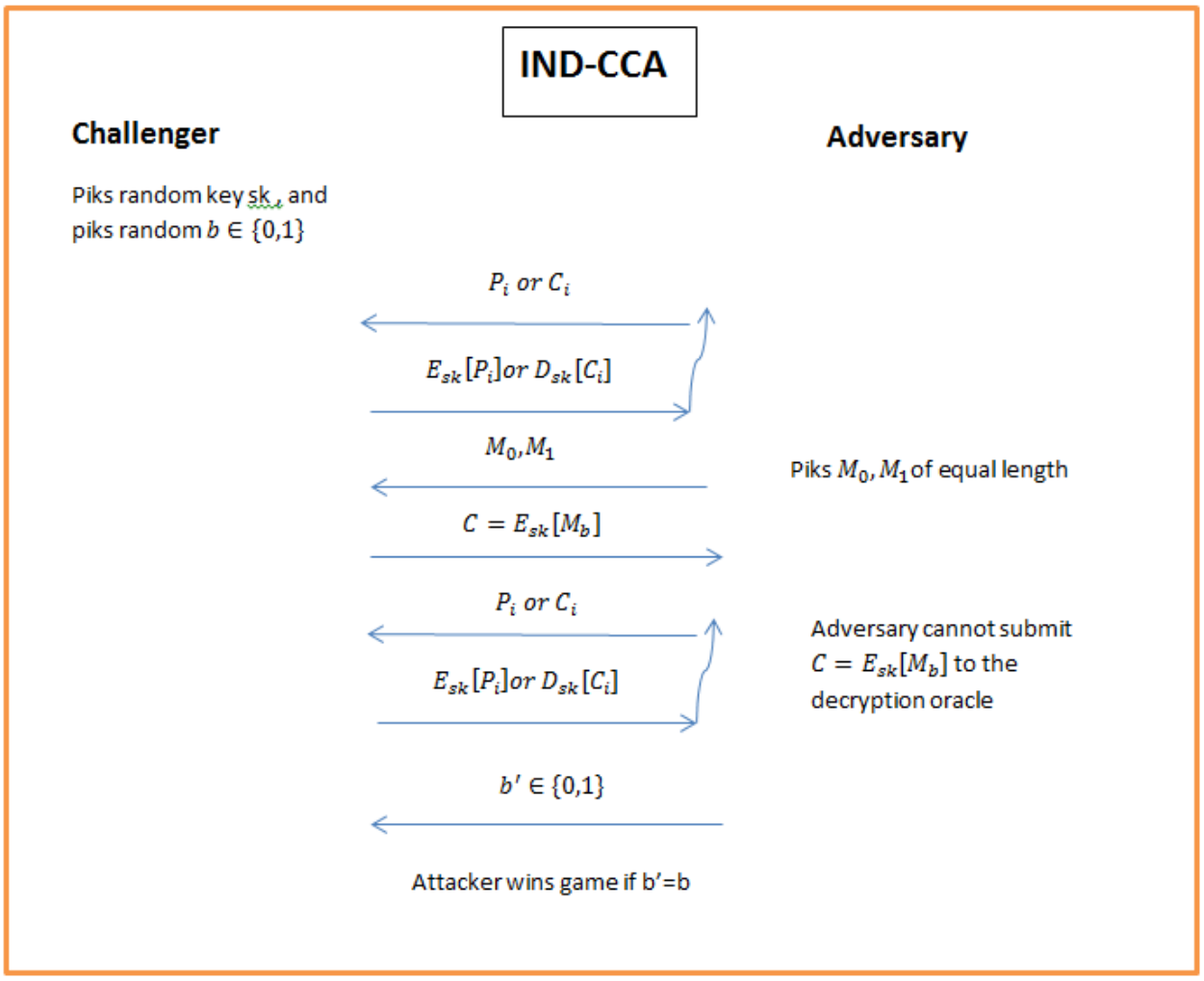

Figure 3: Chosen Ciphertext Attack scenario

III. THE MORE ENCRYPTION SCHEME

The scheme MORE [8] is constructed using the matrix ring $M_2(\mathbb{Z}_N)$ for an RSA modulo N.

The cleartext is a message $m \in \mathbb{Z}_N$.

The secret key consists of an invertible matrix $A \in GL_2(\mathbb{Z}_N)$.

To encrypt a message $m \in \mathbb{Z}_N$ we choose a random integer $r \in \mathbb{Z}_N$.

The ciphertext $E_A(m)$ is a matrix such that:

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

$E_A(m) = A^{-1}\begin{pmatrix} m & 0 \\ 0 & r \end{pmatrix} A.$

To decrypt a ciphertext $E_A(m)$ we compute:

$m = D_A\big(E_A(m)\big) = [AE_A(m)A^{-1}]_{11}.$

This cryptosystem is a fully homomorphic encryption scheme because we have:

$E_A(m_1) + E_A(m_2) = A^{-1}\begin{pmatrix} m_1 & 0 \\ 0 & r_1 \end{pmatrix} A + A^{-1}\begin{pmatrix} m_2 & 0 \\ 0 & r_2 \end{pmatrix} A = A^{-1}\begin{pmatrix} m_1 + m_2 & 0 \\ 0 & r_1 + r_2 \end{pmatrix} A = E_A(m_1 + m_2)$

And $E_A(m_1) * E_A(m_2) = A^{-1}\begin{pmatrix} m_1 & 0 \\ 0 & r_1 \end{pmatrix} A * A^{-1}\begin{pmatrix} m_2 & 0 \\ 0 & r_2 \end{pmatrix} A = A^{-1}\begin{pmatrix} m_1 * m_2 & 0 \\ 0 & r_1 * r_2 \end{pmatrix} A = E_A(m_1 * m_2)$

IV. CRYPTANALYSIS OF THE MORE ENCRYPTION SCHEME

Suppose $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in GL_2(\mathbb{Z}_N)$. And put $\tau = \frac{bd-ac}{ad-bc}$ (Writing this, makes sense because the matrix A is invertible and $\det(A) = ad - bc$).

We have $A^{-1} = \frac{1}{\det(A)}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$ so $E_A(m) = A^{-1}\begin{pmatrix} m & 0 \\ 0 & r \end{pmatrix} A = \frac{1}{\det(A)}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}\begin{pmatrix} m & 0 \\ 0 & r \end{pmatrix}\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

That means $\det(A)\, E_A(m) = \begin{pmatrix} adm - bcr & bdm - bdr \\ -acm + acr & -bcm + adr \end{pmatrix}$

So $tr\big(\det(A)\, E_A(m)\big) = (ad - bc)(m + r) = \det(A)(m + r)$ And $Antitr\big(\det(A)\, E_A(m)\big) = (bd - ac)(m - r)$

So $tr\big(E_A(m)\big) = (m + r)$ and $Antitr\big(E_A(m)\big) = \tau(m - r)$

Supposing $\tau \wedge N = 1$ that means τ is invertible which gives (otherwise we will choose the plaintext until getting an invertible τ).

$m - r = \tau^{-1} Antitr\big(E_A(m)\big)$ Putting $\propto = \tau^{-1}$ so we will have:

$m - r = \propto Antitr\big(E_A(m)\big)$ And $m + r = tr\big(E_A(m)\big)$ which gives by adding the two equations:

$2m = tr\big(E_A(m)\big) + \propto Antitr\big(E_A(m)\big) \qquad (*)$

So we obtain a linear equation in one unknown, so knowing a message m and its ciphertext $E_A(m)$ one can find $\propto$. Finally, finding $\propto$ one can find any encrypted message based on its cryptogram (known plaintext attack).

V. A TOY EXAMPLE

In this section we will provide a practical example to illustrate the attack.

As a first step we suppose that we have a plaintext and its correspondent ciphertext. These two elements will allow us to resolve the equation $(*)$ and find $\propto$.

After this step we will suppose that we have any ciphertext and we will find its correspondent plaintext.

All operations will be done modulo N Such that:

p = 123456789012345678949 And $q = 203860925102478233479$

$\Rightarrow$ N = p. q = 25168015218238260179270970125792397333571

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

The secret key is:

$$A = \begin{pmatrix} 75847318622406900105579689116433473883 55 & 18065526363928829631581590675417499804 87 \\ 12723295490223195205914401583135697300360 & 58055582753977699132135960279425577974 25 \end{pmatrix}$$

We have $\det(A) = 10220422593855438666285562884767752307994$

$$\det(A)^{-1} = 307707810054253720804216039598203188741 6 \neq 0$$

$$\text{So } A^{-1} = \begin{pmatrix} 14949553640134651155553421071085697695540 & 51358580071887597291068635621813599434 91 \\ 13507645766406454776954334200685057904712 & 23372093982187685937786230888460382909 3 \end{pmatrix}$$

The known plaintext is $m = 458741546254500241569634475589620$ and the known ciphertext will be:

$$E_A(m) = AMA^{-1} = \begin{pmatrix} 22072769804179064877305514857357796908590 & 24485016013233905873633568075633294319909 \\ 18874160419472626946586522868706641172477 & 30952458728107260774660333932820749597 11 \end{pmatrix}$$

$$\text{Such that } M = \begin{pmatrix} 458741546254500241569634475589620 & 0 \\ 0 & 99845210003365552129989451 10 \end{pmatrix}$$

As consequences we get:

$$\mathrm{tr}\left(E_A(m)\right) = 25168015676989790954771548250639871868301$$

$$\text{And Antitr}\left(E_A(m)\right) = 18191161214468272640949120818547538158815$$

The equation $(*)$ will became:

$$2 * 458741546254500241569634475589620 = 25168015676989790954771548250639871868301$$

$$+ 18191161214468272640949120818547538158815 *\propto$$

$$\Rightarrow 18191161214468272640949120818547538158815 *\propto = 45873156173349990501442147664 4510$$

$$\Rightarrow \propto = 18424826429299116161931774386381554307 86$$

Step 2: $m' = 25413665487989441205689922011452$ is a cleartext,

$r' = 78455456969746699871213697746568$ is a random which permits to build the matrix of plaintext M'

And C' is the ciphertext of m' such that:

$$C' = AM'A^{-1} =$$

$$\begin{pmatrix} 63491597612143057844289835083133860744 29 & 15974040780571889813417667953081617162 150 \\ 23544966323535110323156840518816977996 7 & 18818855560893076852578127694382631017 162 \end{pmatrix}$$

After calculating $\propto$, the attacker can find any plaintext only by knowing its ciphertext.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

In this step $(*) \Rightarrow$

$$2m' = 10386912245773614107690361975802 0 + 162094904438072409166492363582697869421 17 * 184248264292991161619317743863815543078 6 = 508273309759788824113798440229 04$$

$$\Rightarrow \quad m' = 254136654879894412056899220114 52$$

Finally we obtained the first plaintext only by using the equation $(*)$.

VI. FAIL OF THE ITI SHARMA'S REFRESHMENT PROCEDURE

In a paper entitled "A Symmetric FHE Scheme Based on Linear Algebra" [10], Iti Sharma proposed a new fully homomorphic encryption scheme based on the Brenner et al [11] somewhat homomorphic encryption scheme. The author introduced a simple procedure to refresh ciphertexts based on modulus reduction. Unfortunately this refreshment method fails after some few homomorphic operations and does not work in general. In this part we will introduce this cryptosystem and show its fail.

A. The Iti Sharma's FHE scheme.

Let λ be the security parameter.

Key generation: Generate the secret key p, a prime number of length λ bits. Select q as refresh key such that it is an even multiple of p, that is q = k p, where k is an even number.

Encryption: Encryption involves following steps:

1. Choose m' such that m≡m' mod 2
2. Choose a random number r of length λ^2 bits.
3. Output c = m' + pr.

Decryption: To decrypt we compute the message as m = c mod p mod 2.

Homomorphic operations are obtained via algebraic addition and multiplication of ciphertexts. Direct addition of two ciphertexts is homomorphic to XOR of two plaintext bits and integer multiplication of two ciphertexts is homomorphic to AND of two plaintext bits.

Refresh procedure: The refreshment method is done via refresh key as: c' = c mod q.

B. Cryptanalysis

The idea of the encryption algorithm is enciphering bits (0 or 1) by hiding the parity of a random integer. This is done in two steps: the first step consists of choosing an integer m' of the same parity as the plaintext bit (m' contains the useful information for us). The second step consists of randomising this parity by adding to m' an integer pr of hazardous parity.

The idea of the proposed refresh procedure is the fact that modeling an integer by an even modulo does not change its parity. This procedure is gainful in reducing the size of a given ciphertext but it is unable to preserve the parity of the useful information in the ciphertext. In other words, the first part of the ciphertext (m') can grow after operations

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

until exceeding p which can change the parity of the useful information, so the decryption will fail in this situation. Bellow we give a simple example for clarification:

Key generation: Choose $p = 23$ and $q = 2 * 19 * 23 = 874$.

Encryption: encrypt $m_1 = 1\ and\ \ m_2 = 0$ to $\ c_1 = 5 + 4 * 23 = 97\ and\ \ c_2 = 8 + 6 * 23 = 146$.

Multiplication: $\ c_1 * c_2 = 97 * 146 = 14162 \equiv 174\ mod\ 874$.

Decryption: $(\ c_1 * c_2\ mod\ 23) mod 2 = (174\ mod\ 23) mod 2 = 1 \neq (m_1 * m_2 = 0)$.

VI. CONCLUSION

In this paper we presented a cryptanalysis of a free-noise fully homomorphic encryption scheme. Our attack is based on a single known plaintext. The couple (plaintext, ciphertext) allows us to resolve an equation and find the $\propto$ parameter. By knowing this parameter an adversary can learn any message after without the need of secret key. As a consequence, the cryptosystem MORE is not IND-KPA secure. We cryptanalyzed also a symmetric FHE scheme and showed that its refreshment method can't work in general.

REFERENCES

[1] R. Rivest, L. Adleman, and M. Dertouzos. "On data banks and privacy homomorphisms", Foundations of Secure Computation, pp 169-180, 1978.
[2] Gentry, C. A fully homomorphic encryption scheme. PhD thesis, Stanford University, 2009. http://crypto.stanford.edu/craig.
[3] 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/.
[4] van Dijk, M., Gentry, C., Halevi, S., and Vaikuntanathan, V. Fully homomorphic encryption over the integers. Cryptology ePrint Archive, Report 2009/616, 2009. http://eprint.iacr.org/.
[5] G. Chunsheng. " Fully Homomorphic Encryption Based on Approximate Matrix GCD ". Aavailable at eprint.iacr.org/2011/645.
[6] Z. Brakerski, V. Vaikantanathan. " Efficient Fully Homomorphic Encryption from (Standard) LWE". Available at http://eprint.iacr.org/2011/344
[7] Ahmed EL-YAHYAOUI, Mohamed Dafir ELKETTANI "Fully Homomorphic Encryption: State of Art and Comparison", Vol. 14 No. 4 APRIL 2016 International Journal of Computer Science and Information Security (pp. 159-167).
[8] A. Kipnis and E. Hibshoosh, « Efficient Methods for Practical Fully Homomorphic Symmetric-key Encrypton, Randomization and Verification », Cryptology ePrint Archive, Report 2012/637.
[9] https://en.wikipedia.org/wiki/Ciphertext_indistinguishability
[10] Iti Sharma, "A SYMMETRIC FHE SCHEME BASED ON LINEAR ALGEBRA". Vol. 5 No. 05 May 2014, International Journal of Computer Science & Engineering Technology (IJCSET) ISSN : 2229-3345 available in: http://www.ijcset.com/docs/IJCSET14-05-05-079.pdf
[11] M. Brenner, J. Wiebelitz, G. von Voigt and M. Smith, "Secret Program Execution in the Cloud Applying Homomorphic Encryption" in 5th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2011), 31 May -3 June 2011, Daejeon, Korea, pp. 114-119.