Hardware Assisted Homomorphic Encryption in a Real Time VOIP Conference Application

Ahmed Kamal Hisham Dahshan Ashraf Diaa

Department of Communications
Military Technical College
Cairo, Egypt

***Abstract*—In this paper, a new scheme to perform a secure Voice over IP (VoIP) teleconferencing between a number of VOIP Clients is proposed. The Teleconference is done between clients regarderless of their "based on" working environment (OS (Windows, Linux, Android,..), HW (Mobile, Desktop, IP Phone, ..), ...etc). It is assumed that the SIP Server lies in an untrusted area or administered by untrusted persons. Therefore, it is highly required to prevent those persons from observing all the communications between participants and obtain unencrypted data when they have access to teleconferencing servers. An end-to-end Homomorphic Encryption (HE) is applied with the assistance of a special external Hardware appliance based on an Altera Cyclone IV 4CE115 FPGA to support the somewhat homomorphic encryption scheme. Further, this external appliance is used to boost the performance of the Homomorphic operations. Two different approaches for mixing the VOIP data streams required for the teleconference are proposed. In the first approach, the Mixing function for the encrypted VOIP data streams is performed in the SIP Server and the hardware appliance will do only the HE operations. In the second approach both the mixing function for the VOIP data streams and the HE operations are performed in the external appliance.**

***Index Terms*—Some What Homomorphic encryption, Hardware implementation, VOIP Conference.**

I. Introduction

Nowadays cloud computing is a booming technology that is evolving very fast. Much of individuals and businesses data is stored and computed on by third parties such as Google, Microsoft, Apple, Amazon, Facebook, Dropbox and many others. Classically, cryptography provided solutions to protect data in motion from point A to point B. But these are not always sufficient to protect data at rest and particularly data in use [16]. VOIP calls can easily be traced and tracked or listened and the attacker can edit also the communication or some confidential conversation may be leaked or misused while using the internet as it is a public network. These issues are covered regarding to normal VOIP calls by using VPNs and end to end encryption. The main remaining threat is when using the Teleconferencing feature, as even after applying end to end encryption it only terminates at the SIP Server. The Server has to get the audio in a clear form in order to apply the mixing function on the coming VOIP Data streams before forwarding it to the designated clients. This leads to the ability of the SIP Server administrators to listen to all the conferences that is held by the Server. That is considered a critical threat especially if the SIP Server is hosted at any Cloud Service provider. The above issues could be covered either by Secure Multi-Party Computation (SMC) [5] or by applying Homomorphic encryption on mixed VOIP Data streams. It would be encrypted by a secret key which is only shared by the participating clients and so prevent the SIP server administrators from having access to the clear data streams. Homomorphic Encryption has previously been considered inefficient for practical usage. A lot of research have been addressed for improving schemes efficiency and implementations to solve the runtime challenge of the HE operations beside its huge memory consumption [6], [7], [8], [9], [10], [11]. The paper is organized as follows. Section II gives a Background of the SIP Technology and Conferencing Feature. Section III discusses Homomorphic Encryption and exploring two different kinds of it (Paillier CryptoSystem and Elementary Modular Arithmatic). Section IV discusses the Related work on the Same Context. In Section V, a describtion of the threat model and goals of a Secure Teleconference Application, In Section VI, a detailed explanation of the proposed solution and exploration of the two different approaches for achieving the required security for the mentioned application is presented. The analysis of the proposed schemes is introduced in Section VII. Finally, Section VIII concludes the paper.

II. Background

A. TeleConferencing using SIP

The Session Initiation Protocol (SIP) can support many different conferencing architectures including the centralized conferencing server model. It defines how to establish, maintain and terminate Internet sessions including multimedia conferences. In centralized mixing, a server receives media streams from all the clients in a conference. It mixes or filters these based on pre-defined policy and distributes the streams to the clients. Different types of media streams need to be handled differently. For example audio streams are typically summed. While video streams are selected [3]. The main functions of a conference server is the mixing and redistribution of media streams. Typically, Internet audio streams are added (mixed). While video streams and other media are simply replicated. For audio: the server needs to ensure that a participant does not receive a copy of his own media in

the mixed stream. RTP allows a sender to indicate which sources have been combined in a single media packet. In

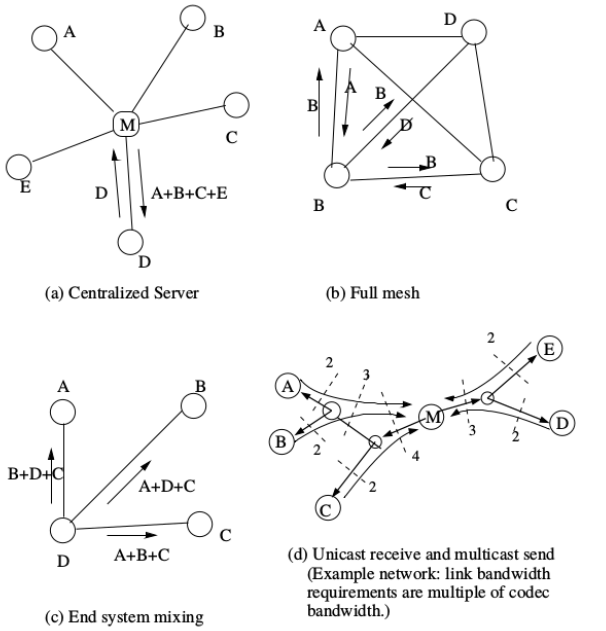

Fig. 1. Types of TeleConferencing distribution model

the centralized model. A server receives media streams from all participants. it mixes them if needed and redistributes the appropriate media stream back to the participants. The server needs to create a customized stream for each of the currently active M senders. it assumes that they can all support the same media format. The server needs to decode/decrypt audio streams before mixing, as mixing can only be performed on plain audio. Decoding the M coming audio streams and then encoding and combining/adding $m - 1$ audio streams and sending it to the designated client [3].

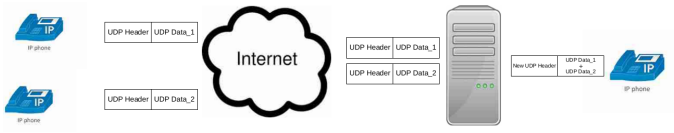

Fig. 2. Visual Example of a Centralized Server

1) Audio Mixing: Figure 3 shows how an audio mixing module can be implemented. Client A support G.711, B DVI ADPCM and C both GSM and G.711. Clients list the codecs they support in their INVITE requests. The server selects an intersection of the algorithms supported by the Client as well as by the server. This selection is returned in the signaling success response to the Client. These algorithms are listed in order of preference in the SDP of the INVITE or its response. The mixing algorithm can be defined as a decode-mix-encode sequence. When an audio packet arrives at the mixing module, it is decoded into 16-bit linear samples and enqueued in the per-Client audio queue. The jitter in packet arrivals is absorbed by a play-out delay algorithm. Every few milliseconds, a timer triggers a routine that mixes a range of the buffer streams into combined packets by simple addition of the sample values. Then, for each of the Clients, the linear sample values from the per-Client queue (e.g., A) is subtracted from the mixed data X and the resulting data $X - A$ is encoded using the preferred audio algorithm. The encoded data is packetized and sent to the Client. If there are m Clients, then both mixing and redistribution will take m additions and m subtractions [3].

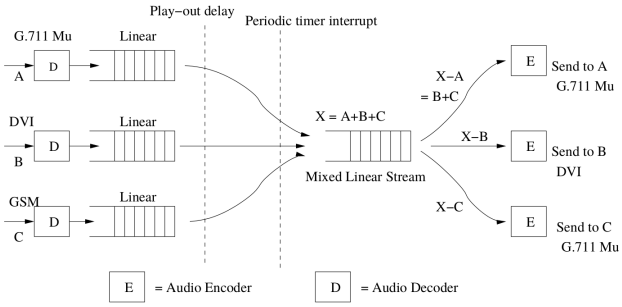

Fig. 3. Audio mixing

III. Homomorphic Encryption

As cloud computing has grown increasingly nowadays, the privacy and confidentiallity of data on the cloud becomes an issue. Encryption is an effective way to enforce data Confidentiallity. However most of the existing encryption schemes require data to be decrypted for processing, where data then becomes vulnerable for exposure [4]. Full Homomorphic encryption is considered the optimum solution for such an issue. It is an encryption scheme which allows for computations on encrypted data and obtains an encrypted result which when decrypted gets the same result of computations on the plain original data.

$$EncryptionScheme = \{Key, *, ENC, DEC\} \quad (1)$$

$$c_i = ENC(k, m_i) \quad (2)$$

$$DEC_k(f, ENC_k(x)) = f(x) \quad (3)$$

- m_i/c_i is the plain/cipher text.
- $*$ is the Operation (Summation/Multiplication).
- Key is the secret key.
- ENC/DEC is the Encryption/Decryption process.
- $f(x)$ is the function to be applied.

Full Homomorphic encryption makes it possible to process data in an open computing environment such as the Cloud while enforcing data privacy. Data can be outsourced for any third party for applying computation without any fear of original information leakage. Despite the urgency of applying this feature as the world now is increasingly adopts Cloud computing and cloud services, full homomorphic encryption suffers from the high computational cost. For example a simple computation on two plain texts needs a large memory storage ”1 Mb of data results in more than 10 Gb of encrypted

data" besides the huge processing power consumed. However Homomorphic addition is still manageable as it takes under 1 ms in some cases. However, multiplication takes over 5 seconds per multiplication [17]. So, it's still not relevant till now on today's practical applications. But using somewhat Homomorphic encryption is still somehow practical and could be applied on many different applications where only additive property is required [2]. There are many different algorithms which can be used in somewhat homomorphic encryption and mainly they are divided into two main categories. One that depends on the Discrete Logarithmic Problem such as the Paillier Cryptosystem [14]. And the other depends on approximate integer greatest common divisors (approximate GCD) problem, such as Elementary Modular Arithmetic [15]. In our context the later would be used and the reason would be illustrated later in section 6. In the following section an overview would be given for both algorithems (DLP / GCD).

A. Paillier Cryptosystem

The Paillier cryptosystem is introduced by Pascal Paillier in 1999, it is a probabilistic asymmetric algorithm for public key cryptography. The problem of computing $n-th$ residue classes is believed to be computationally difficult (DLP). The scheme is an additive homomorphic cryptosystem; this means that given only the public-key and the encryption of m_1 and m_2, one can compute the encryption of $m_1 + m_2$ [14].

B. DGHV Encryption Scheme

The Elementary Modular Arithmetic was introduced by a group of IBM researchers (M. van Dijk, C. Gentry, S. Halevi and V.Vaikuntanathan) [15] in 2010. It is a very simple algorithem, it is used here in the proposed application as an example for representing the concept and it is considered secure if the secret value p chosen properly.

1) Algorithem Preparation [15]:

- Choose p to be a large secret prime number.
- Choose q and r to be two random numbers where q and $r \in Z$.

2) Encryption:

- $c_i = m_i + (p * q_i) + (255 * r_i)$ (4)

3) Decryption:

- $m_i = (c_i \ mod \ p) \ mod \ 255$ (5)

4) Homomorphic properties [15]:

- The Sum of two ciphertexts will decrypt to the sum of their corresponding plaintexts.
- $c_1 + c_2 = (m_1 + m_2) + (p * (q_1 + q_2)) + (255 * (r_1 + r_2))$ (6)
- $m_1 + m_2 = ((c_1 + c_2) \ mod \ p) \ mod \ 255$ (7)

IV. Related Work

A great work has been done by Kurt Rohloff, David Bruce Cousins, Daniel Sumorok [2]. Where an end to end Secure Teleconference by homomorphic encryption is established. Mumble client is installed at the clients side (Mobile phones (IOS, Android), PC) with its source code modified to perform an additive encoding after sampling in order not to be distorted when mixed during the encryption process at the SIP Server. Besides the Homomorphic encryption module is also added to the client's source code. Murmur server is installed with its source code also modified to apply the mixing function on the incoming encrypted audio data streams. The function used called $EvalMix$ which is responsible for safely adding the corresponding encrypted Samples. It then forwards it to the designated clients.

V. Threat Model and Goals

The following considerations are considered in our approach as follows [2]:

A. Encryption Work Factor

for the lattice based scheme, considering 80-bits as a security factor is currently sufficient [12].

B. Server Compromise

the SIP Server administrators or any intruders shouldn't be able to extract any data or listen to any conversations done through the SIP Server. This is because all operations in the mentioned scheme should be done on the encrypted data.

C. Latency

latency of less than 150 ms is considered acceptable in practice [13].

D. Sound Quality

the Quality of sound shouldn't differ as of the normal conference, so the user shouldn't recognise any degradation in the sound quality.

E. Scalability

the teleconferencing capability should scale to support more than two participants without any degradation in sound quality or latency.

F. Bandwidth Usage

As this system would mainly operate over the Internet, the existing Internet speed and availability are considered high enough. An increase in the packet size between the external appliances would be accepted as long as it will not affect the sound quality or latency.

G. Wide Geographic Area

the system should work all over the world as long as there is an internet connectivity. Beside the above mentioned goals mentioned in [2], there are two extra goals that should also be fulfilled:

H. Solution Scalability

the proposed solution would be applicable to most of the SIP applications rather than to be attached to a certain kind or version of an application.

I. Computations Offload

Offloading Computations to external resources other than the conferencing server will increase the performance especially for the HE Operations as it needs huge computational power and memory.

VI. Proposed Solution

Two different approaches are proposed below for implementing an Encrypted end to end Teleconference using an external appliance. The appliance would be used to perform HE operations only in the first approach. In the second approach the appliance would be used for the mixing operation as well.

A. First Approach

A SIP Server would be installed with the conference feature. A VOIP client with the codec to be adjusted to PCMA (G711A). Intoducing an external HE device just next to the clients connected through the ethernet port from the plain side (trusted port) and to the Edge Router (Internet Gateway) (untrusted port) as shown in figure 4. The reason of using the Elementry Modular Arithmetic is its additive property which is performed homomorphically by addition of the corresponding cipher texts opposite to Paillier cryptosystem which depends on the discrete logarithmic problem as the additive property is performed by multiplication of the corresponding cipher texts. It is assumed here that is the maximum number of active speakers in a conference is four and this number could be generalized and it will affect linearly the size of the output IP packet from the external HE Device.

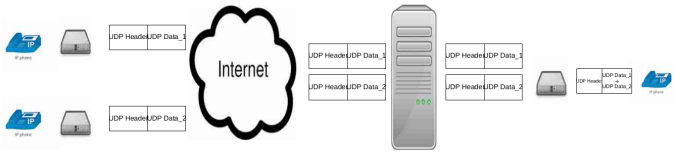

Fig. 4. Visual Example of a Centralized Server after introducing the External HE Device

- The function of the device is to do the Homomorphic encryption to the incoming audio stream as the incoming IP packet would be modified as follows:
- The IP packet header will remain the same for the outgoing packet similar to the incoming one.
- A secret Prime number P is chosen. For simplicity P of length 2^{15} is chosen. The payload length will then increase by multible of $3 * n$ as shown in figure 5.
- Each byte would be encrypted using Homomorphic encryption with Elementary Modular Arithmetic algorithem.
- $(B_i)_j$ is the byte number i in client number j where $0 \leq B \leq 255$.
- Choose q and r to be two random numbers where q and $r \in Z$ and $r \leq 255/2n$ and $3 \leq q \leq 255$.

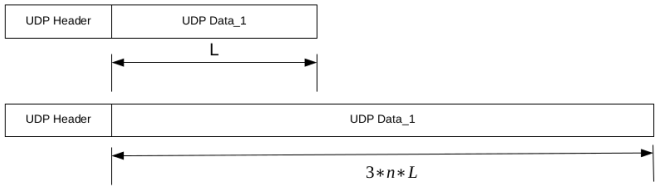

Fig. 5. IP Packet before and after modification by the External HE device, n is number of conference participants.

- $(c_i)_j$ to be the ciphertext of byte number i in client number j.
  $C_(i)_j = B'_j + (P * q_j) + (255 * r_j)$ (8)
- The output value after encryption should be distributed to $3 * n$ bytes to make sure that each byte value must be less than $255/n$. This is done to prevent distortion after applying the mixing operation in the conference Server side as shown in figure 6.

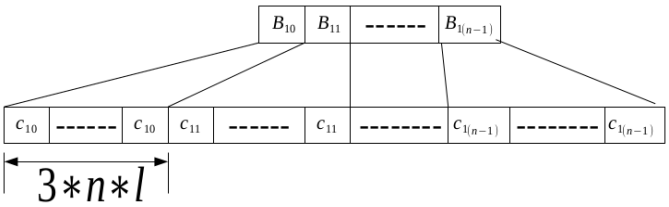

Fig. 6. The output value after encryption should be distributed to $3 * n$ bytes

- The Conference SIP server applies the mixing operation to the incoming IP packets which is a simple sample adder and then forward the packet to the designated clients as shown in figure 7.
  - Choose $(C_{output})_j$ to be the output ciphertext.
  - $C_{output_j} = \sum_{i=1}^{n} c_{ij}$ (9)
    $C_{output_j} = (B'_1 + (p * q_1) + (255 * r_1)) + (B'_2 + (p * q_2) + (255 * r_2)) + ... + (B'_n + (p * q_n) + (255 * r_n))$ (10)
    $C_{output_j} = \sum_{i=1}^{n} B_i + P * \sum_{i=1}^{n} q_i + 255 * \sum_{i=1}^{n} r_i$ (11)
    where C_{output_j} is the output cipher which will be distributed to $3 * n$ bytes.
  - Each IP packet outgoing from the Conference SIP server should pass first to the remote external HE device before reaching the reciepient VOIP client. Each n bytes in the payload coming from different clients would be decrypted to get the plain mixed byte.
    $\sum_{i=1}^{n} B_i = (C_{output} \ mod \ p) \ mod \ 255$ (12)
  - This means that the clients participants would not be affected by the IP packet length increase as this only applies at the IP network traffic in between the external devices.

B. Second Approach

A SIP Server without the conference feature. A VOIP client with the codec PCMA (G711A) to be adjusted. The external

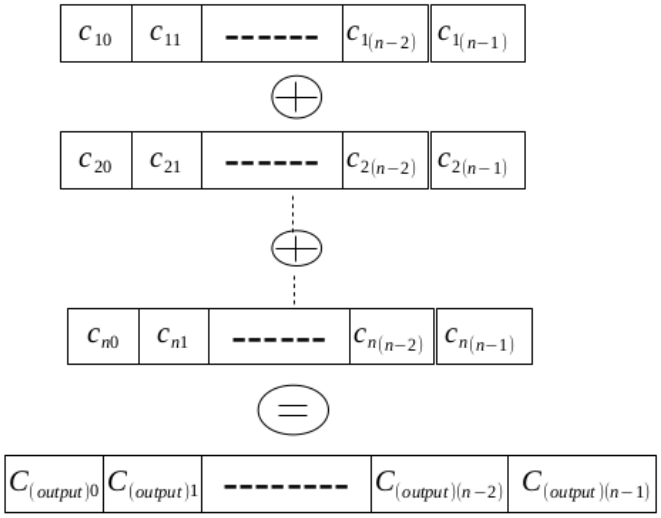

Fig. 7. Mixing function applied in the Conference Server

HE device to be introduced just next to the clients the same exactly as in the first approach figure 4. The HE device would perform also the mixing operations beside the HE operations. The main advantage in this approach is that there is no significant increase in the IP Packet length compared to the previous approach. However there is a drawback that the Clients are recieving a copy of the mixed streams from the main organizer including their own data streams.

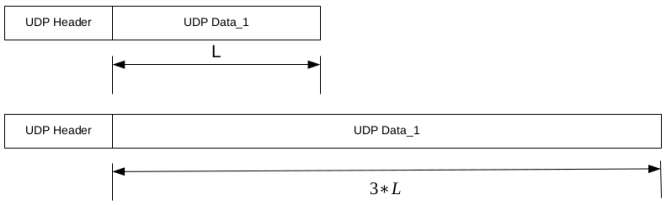

Fig. 8. The IP Packet payload would be only tripled

- Each byte would be encrypted using Homomorphic encryption using Elementary Modular Arithmetic algorithem.
- $(B_i)_j$ is the byte number i in client number j where $0 \leq B \leq 255$.
- A secret Prime number P is chosen of length 2^{15}.
- Choose q and r to be two random numbers where q and $r \in Z$ and $r \leq 255/n$ and $3 \leq q \leq 255$.
- $(c_i)_j$ to be the cipher text of byte number i in client number j.
  $(c_i)_j = B'_j + (p * q_j) + (255 * r_j)$ (13)
- The output value after encryption should be distributed to three bytes as shown in figure 9.

Below is the operations that would be done at the HE Device next to the organizing client:

- $C_{output_j} = \sum_{i=1}^{n} c_{ij}$ (14)
  $C_{output_j} = (B'_1 + (p * q_1) + (255 * r_1)) + (B'_2 + (p * q_2) + (255 * r_2)) + ... + (B'_n + (p * q_n) + (255 * r_n))$ (15)
  $C_{output_j} = \sum_{i=1}^{n} B'_i + P * \sum_{i=1}^{n} q_i + 255 * \sum_{i=1}^{n} r_i$ (16)
  where C_{output_j} is the summation of all the corresponding input cipher values.
  After applying the mixing function as shown above, the output value would be decrypted as follows:
  $\sum_{i=1}^{n} m_i = (C_{output} \ mod \ p) \ mod \ 255$ (17)
- This means that the clients participants would not be affected by the IP packet length increase as this only applies at the IP network traffic in between the external devices.

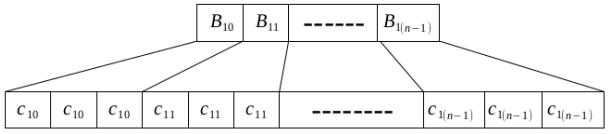

Fig. 9. The encrypted should be distributed to three bytes

VII. Security Analysis

The proposed scheme achieves the correctness and secrecy requirements necessary for HE as follows:

A. Correctness

Analysis of both approaches would be stated below:

1) First Approach: **Theorem 1.** Given B_1,...,B_t, The sum of the plain bytes is equal to the decryption of the sum of the encrypted bytes.

$$\sum_{i=0}^{t} B_i = ((\sum_{i=0}^{t} C_i) \ mod \ P) \ mod \ 255 \tag{18}$$

$$r_i \leq (255/2n), \quad \sum_{i=0}^{n} B_i \leq P \tag{19}$$

therefore

$$\sum_{i=0}^{n} r_i \leq 255 \tag{20}$$

$$\sum_{i=0}^{t} C_i = \sum_{i=0}^{t} B_i + P * \sum_{i=0}^{t} q_i + (255 * \sum_{i=0}^{t} r_i) \tag{21}$$

$\sum_{i=0}^{t} C_i$ is of size 2^{24} $(i.e\ 3\ bytes)$ which means that the payload would be tripled as shown in figure 8.

The mixing function at the SIP Server is then applied for each byte for the incoming data streams which is a simple addition $\sum_{j=0}^{n} C_{ij}$ where i is the byte order number and j is the client number.

In the decryption side at the HE appliance, decryption is applied and the original value of corresponding bytes summation is achieved.

$$\sum_{i=0}^{n} B_i = (\sum_{i=0}^{n} C_i \ mod \ P) \ mod \ 255 \tag{22}$$

2) Second Approach: For this approach the mixing function is applied in the External device beside the HE operations so C_{Total} which has a maximum size of 2^{24} is distributed only over three bytes. So, the size of the output packet after applying the Homomorphic encryption is almost three times the size of the original packet. As we don't have in this approach the same restriction applied in the first one where the mixing function is applied in the conference server.

$$\sum_{i=0}^{t} B_i = ((\sum_{i=0}^{t} C_i)\ mod\ P)\ mod\ 255 \tag{23}$$

$$r_i \leq (255/2n), \qquad \sum_{i=0}^{n} B_i \leq P \tag{24}$$

therefore

$$\sum_{i=0}^{n} r_i \leq 255 \tag{25}$$

$$\sum_{i=0}^{t} C_i = \sum_{i=0}^{t} B_i + P * \sum_{i=0}^{t} q_i + (255 * \sum_{i=0}^{t} r_i) \tag{26}$$

$\sum_{i=0}^{t} C_i$ is of size 2^{24} $(i.e\ 3\ bytes)$ and each byte is distributed to n bytes to make sure that each byte has a value $\leq 255/n$.

The mixing function opposite to the first approach will be applied in the external HE device. Each byte in the incoming data streams would be added with its corresponding one $\sum_{j=0}^{n} C_{ij}$ where i is the byte order number and j is the client number.

In the decryption side at the HE appliance, decryption is applied and the original value of corresponding bytes summation is achieved.

$$\sum_{i=0}^{n} B_i = (\sum_{i=0}^{n} C_i\ mod\ P)\ mod\ 255 \tag{27}$$

B. Security

The algorithem applied here in the two approachs depends on Elementary Modular Arithematic (EMA) which is first proposed by Gentry [15]. The level of security achieved depends on the length of the secret prime P and the hardness of the approximate integer greatest common divisors (approximate GCD) problem. In the proposed schemes each byte in the client's outgoing IP packet would be encrypted with the EMA algorithem. If any adversary or the SIP Server administrators monitored the traffic. They will get nothing useful as they will face the GCD problem to get the value of B.

VIII. Conclusion

The main contribution here is to apply the Homomorphic Encryption in existing working environments. It won't work in certain environments where a mandatory change in the core of the application is required (modifying source code). It is applicable in certain applications which uses only the additive property in the stored data. such as (Electronic Voting, VOIP TeleConferences, some Medical applications, ...etc). An increase in the IP packet size specially in the first approach after applying HE operations would be recognized. A higher bandwith for the transmission media is required to avoid any latencies. Applying this approach with a good quality codec may decrease the packet size significantly. However, the codec should maintain the additive property as the PCMA to avoid distortion of the voice content after applying encryption and mixing operations. Introducing an external device to do the HE operations (huge amount of memory and processing power usage) will ease the burden on the application Server/Client.

References

[1] Muhammad Zulkifl Hasan, Muhammad Zunnurain Hussain, "Collective Study On Security Threats In VOIP Networks," International Journal of Scientific and Technology Research, vol. 6, Issue 01, January 2017.

[2] Kurt Rohloff, David Bruce, Daniel Sumorok, "Scalable, Practical VoIP Teleconferencing with End-to-End Homomorphic Encryption", IEEE Transactions on Information Forensics and Security, Vol 12, Issue 5, May 2017.

[3] Kundan Singh, Gautam Nair and Henning Schulzrinne, "Centralized Conferencing using SIP,".

[4] Ryan Hayward, Chia-Chu Chiang, "Parallelizing Fully Homomorphic Encryption," International Symposium on Computer, Consumer and Control, 2014, IEEE.

[5] J. Launchbury, D. Archer, T. DuBuisson, and E. Mertens, Application-scale secure multiparty computation, in Programming Languages and Systems, ser. Lecture Notes in Computer Science, Z. Shao, Ed. Springer Berlin Heidelberg, 2014, vol. 8410, pp. 826.

[6] Z. Brakerski, C. Gentry, and V.Vaikuntanathan, (Leveled)fully homomorphic encryption without bootstrapping,ACM Transactions on Computation Theory (TOCT), vol. 6, no. 3, p. 13, 2014.

[7] C. Gentry, A. Sahai, and B. Waters, Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute based, in Advances in CryptologyCRYPTO 2013. Springer 2013, pp. 7592.

[8] M. Naehrig, K. Lauter, and V. Vaikuntanathan, Can homomorphic encryption be practical? in Proceedings of the 3rd ACM workshop on Cloud computing security workshop, ser. CCSW 11. ACM, 2011, pp. 113124.

[9] [14] Y. Dorz, Y. Hu, and B. Sunar, Homomorphic AES evaluation using the modified LTV scheme, Designs, Codes and Cryptography, vol. 80, no. 2, pp. 333358, 2016.

[10] C. Gentry and S. Halevi, HElib, https://github.com/shaih/HElib, 2014.

[11] L. Ducas and D. Micciancio, FHEW:Bootstrapping homomorphic encryption in less than a second, in EUROCRYPT 2015. Springer, 2015, pp. 617640.

[12] T. Lepoint and M. Naehrig, A Comparison of the Homomorphic Encryption Schemes FV and YASHE. Cham: Springer International Publishing, 2014, pp. 318335.

[13] S. Na and S. Yoo, Allowable propagation delay for VoIP calls of acceptable quality, in Advanced Internet Services and Applications. Springer, 2002, pp. 4755.

[14] GEMPLUS Cryptography Department and ENST Computer Science Department, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes,".

[15] Marten van Dijk, Craig Gentry, Shai Halevi and Vinod Vaikuntanathan, "Fully Homomorphic Encryption over the Integers,", EUROCRYPT 2010.

[16] Boaz Barak, "Fully homomorphic encryption : Introduction and bootstrapping,".

[17] Connor Rset, Van Warren and Chia-Chu Chiang, "Enhanced Database Security Using Homomorphic Encryption,", International Conference on Information Science and Applications, 2017.