ISSN 2047-3338

A Trusted Third Party Based Lawful Interception Model for VoIP

Muhammad Sarwar Jahan Morshed
Department of Computer, Electrical & Space Engineering, Luleå University of Technology, Luleå, Sweden
muhammad.morshed@ltu.se

***Abstract*— Lawful Interception is a very basic method used by the law enforcement agency (LEA) all over the world for identifying criminal act and actors by monitoring telecommunication network. It plays an important role for the homeland security. As soon as the VoIP is introduced, people switches from traditional telephone system to VoIP due to higher speeds, low cost, support for mobility, and easy nature for encryption. But there are some issues such as misuse of intercepted data by the members of LEA, privacy concern of the individuals, cost, legal liabilities, etc. On the other hand, VoIP has become a popular means of communication tool for the criminals and terrorists – as VoIP provides secure conversation by encrypting the messages transferred between two communication parties. Besides other difficulties for interception cause due to the architecture of the VoIP and use of smart devices – thus it is hard to intercept both the signaling and call contents [1]. This paper presents a Trusted Third Party (TTP) or Key Escrow Agent based Lawful Interception (LI) model that would provide lawful interception over VoIP while ensuring the individual privacy.**

***Index Terms*— VoIP, Lawful Interception, Privacy, Security, Key Escrow Agent and Trusted Third Party**

I. INTRODUCTION

IN the twenty-first century, people are using different kinds of technology for communications. Exploiting the advantages of these communication technologies, different terrorist organizations and local criminals hide their internal communication and operate their activities avoiding authority's monitoring. In order to prevent them and their unlawful activities, Lawful Interception (LI) in their communication is used by the Law Enforcement Agency (LEA) all over the world. In traditional telephony system LEA can monitor the network system easily, as these networks are operated through a centrally controlled network hub and all of the communication passes through the telephone exchanges that are controlled by different telecommunication operators. In contrast, call signaling and the communication traffic between the communication parties in the VoIP system may travel through different routes - which makes the interception over VoIP difficult. Besides, endpoints of the VoIP communication consists of intelligence and have features for adding encryption/decryption with the message. Moreover, there is option of using mutually agreed but non-standard CODEC for the data traffic. These attributes of the VoIP architecture makes harder for interception in this communication method. Moreover, lawful interception becomes a disputed issue between the LEA of many countries and their citizens, since many consider lawful interception is a breach of basic human rights since it might compromise individual's privacy. On the other hand, lawful interception is one of the most useful tool for protecting national security as well as to prevent criminal activities. Besides, traditional interception in the communication network may open security hole that could make whole communication network vulnerable and cyber attacker may use this opportunity to fulfill their ill-intention. So that a secure mechanism for lawful interception over VoIP is required which will consider all shortcomings in the existing mechanisms and will be acceptable to both LEA and citizens.

In order to introduce a secure mechanism for lawful interception, we have carried out an investigation and details of this investigation has been presented in [2]. This paper reflects the findings of the work (in [2]) about the proposed secure LI model.

II. RELATED WORKS

Both VoIP and Lawful Interception are two important things which are now widely used. As it is known to all that VoIP is used by many for its low cost, while terrorists are exploiting this technology for their secure communication. VoIP has a dilemma that it uses encryption mechanism that encrypts the conversation between the communicating parties to preserve their privacy. But criminals and terrorists are misusing this encryption mechanism to avoid identification by the LEA. On the other hand, lawful interception is a mechanism that has been used by the LEA for years in order to identify the terrorists' gangs and their activities. Therefore there should be some mechanism for lawful interception over VoIP that could be used by the LEA in order to investigate the suspicious activities while preserving individual's privacy. To date there are few published research papers regarding the lawful interception over VoIP that consider individual's privacy while intercepting individual's communication? In the following some relevant researches have been briefly presented.

A. Non-repudiation of Voice-over-IP

C. Hett, et al. have presented a mechanism that would ensure the non-repudiation property of a network communication using VoIP [5]. Their proposed security mechanism provides the security of RTP packets over the communication channel by splitting the VoIP stream into intervals of flexible length. This division of VoIP-stream would be based o time, number of packets or other attributes. In this mechanism, both communicating parties (e.g. caller & callee) accumulate received packets in buffers. Packets of all intervals are digitally signed along with the meta-data, and thus create the interval signature. This method also uses hash chains to ensure cohesion. A hash of the last interval with its signature is embedded in with the metadata of the current interval. In this way an unbreakable chain is formed with a continuous stream of signatures. Additionally both channels are inter-waved with the use of chaining property.

B. Log Data Protocol by Rafael Accorsi

Rafael Accorsi proposed different Log Data protocols as digital evidence [6]. His investigation reveals that use of log data has been increased as the digital evidence for legal prosecution in judicial row. He also claims that the existing logging protocols are not sufficient in this respect. His investigation found that different protocols store in different formats for future audit. He applied Cohen's trustworthiness as the security requirements that will make sure the authentication of log data. Rafael Accorsi has proposed a log data protocol that has two phases [6]. In the first phase, a log message routs via communication channel between a sending device and the collecting device. Requirements for this phase are origin authentication, message confidentiality, message integrity, message uniqueness, and reliability delivery. The later phase is actually the storage phase which requires entry accountability, entry integrity, entry confidentiality. All logging protocols must follow the requirements when that would be implemented in reality. If the requirements would be followed, logging protocols would be able to provide digital evidence.

C. Logging protocol by V. Stathopoulos, et al.

Stathopoulos et al. present a logging protocol for open communication channel and the objective of the protocol is to prevent insider attack [7]. Assessing individual's trustworthiness is one of the difficult tasks. A member of the LEA can intentionally modify the recorded data of the audit trial or log data without any possibility of detection. This protocol provides a mechanism like "syslog-sign" with the addition of a "regulatory authority" which confirms that the logging is performed following the protocol. Collector, in this protocol, provides signed block periodically to the regulatory authority in order to use in future. If there would be a need to verify any data in future, stored signed block by the regulatory authority could be used to compare with the signed block stored at the collector. The authors argue that forgery is not possible as a copy of the signed block is stored by the regulatory authority.

D. Clipper Chip

The Clipper Chip was actually cryptographic device that used key escrow mechanism. This cryptographic device introduced by the National Security Agency (NSA) of the USA [3], [8]. This mechanism is the first kind of security measure that protects individual's private communication while provide the option to the LEA for lawful interception. The algorithm called Skipjack was used for cryptographic operation and key-exchange algorithm was Diffie-Hellman algorithm. A unique secret key and unique identifier were embedded with each Clipper Chip. During the session initiation, a session initiation key was generated dynamically and was used encrypt the conversation data. Device's secret key encrypted was encrypted and shared with other end device before starting the session. The encryption key was stored by one or more trusted third party for future use by the LEA. Later Clipper Chip was abandoned by the NSA, since Matt Blaze claimed in his publication that this device could be used to violate individual privacy by the members of the LEA. A 128 bit field known as Law Enforcement Access Field (LEAF) was transfer by the chip. LEAF contained the required information to recover the session's encryption key. A 16 bit hash was also sent with the LEAF to prevent the forgery of the LEAF field. Blaze identified that a brute force attack could generate a new LEAF field that would have valid hash. Therefore NSA decided to stop its uses.

III. DESCRIPTION OF THE PROPOSED ESCROW AGENT BASED LI MODEL

Main modules of the proposed LI model are- User Agent, Escrow Agent, LEA module, and validation module.

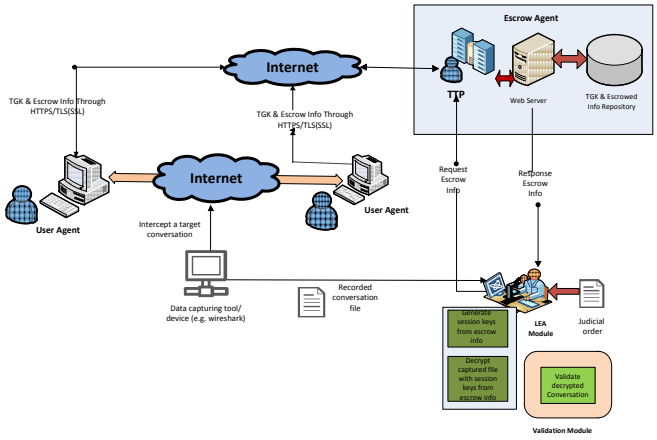

Figure 1. Security risks at 3 stages of new approach

A. User Agent Module

User Agent means the SIP user agent through which users communicate each other from the end devices. User Agents create a block of packets of the media stream, compute a hash over this block, then signs this hash using the user's public key and sends these signed hashes as packets. The User Agent escrows the key and the other information about the session after end of a successful session. In the simulation, an open source SIP User Agent called "minisip" has been used, which has been developed at KTH. This User Agent has been extended by Md. Sakhawat Hossen **Error! Reference source not found.** for creating hash on the block and to escrow to the

Escrow Information. He added a module to "*minisip"* for hashing the block, the popular hash algorithm HMAC-SHA1 has been used. For signing the hash the RSA algorithm has been used.

B. Escrow Agent Module

Escrow Agent Module is the key component of the proposed LI model as it acts as the trusted third party to which User Agents escrow TEK Generation Key (TGK) along with other escrow information after completing each session. It is notable that TGK and other fields are generated by the user agent as they use SRTP and MIKEY protocol. To read more about the protocols, follow our main investigation paper [2].

C. Required Fields for Escrow Agent Module

Escrow Agent contains a database which is used to store escrowed information of a particular conversation. In the investigation it is found that six fields of a session over VoIP are required to escrow to the trusted third party in order to retrieve the corresponding session in future. That fields are – User ID, TGK, RAND value, CSBId, signed hash of the last block and the time of escrowing [4], [10]. Details of these fields are available in [2]. First five fields will be escrowed by the SIP User Agent and the sixth value which is actually locally generated time stamp by the user agent. The User ID is the user's SIP URI or another UI of the UA. TGK is the TEK generation key of flexible length. RAND is a 128-bit or more pseudo-random bit string sent by the session initiator during initial exchange. CSBId is the Cpryto Session Bundle ID which is 32-bit unsigned integer [10].

During the simulation of the proposed model, an open source user agent called *"minisip*" was used [3]. The Crypto Session Id (CSId) is an 8-bit unsigned integer which is initialized when a cryptographic session is established between two user agents. In case of *minisip*, CSId is initialized after accomplishing MIKEY-SRTP mapping. Value of Policy number, ROC and SSRC are used during MIKEY-SRTP mapping, where policy number is fixed since *minisip* only uses SRTP. ROC is computed from SSRC with initial value zero [3]. ROC is computed locally by the receiving user agent. When the sequence numbers go over 2^{16}, the sequence number is reset to zero and ROC is incremented by one. Algorithm for calculating ROC is as follows:

1. Initialize ROC to zero (0).
2. Get sequence number of a captured SRTP packet (sequentially from the first SRTP packet until the end of the session).
3. If sequence number!=65535 then return ROC.
4. Else If (sequence number==65535)
5. Set Current_ROC=ROC.
6. Set ROC=ROC+1.
7. Return Current_ROC.

D. Escrow Agent Database

Escrow agent consists of a database with four different tables for storing and future use of escrow information of each session.

- *User Authentication Table:* This table is used for storing the User Authentication related information. It stores user's credential and other relevant information. In the simulation, SIP URI is used as the User ID. When users are provided the VoIP services, this authentication table can be updated automatically using VoIP operator's provided information. When any user agent wants to escrow information, Escrow Agent will authenticate the user agent based on the stored information in this authentication table.

- *Escrowed Information Table:* This table is used to store escrowed information and escrowed information are Escrowed Id, User Id, TGK, CSBId, RAND, CSId, Signed Hash, and a local time stamp of 64 bit similar to the format of the time stamp used for the Network Time Protocol (NTP). Escrowed information will be identified by the User Id and time stamp. We have assumed that Escrow Agent is synchronized with an NTP time server and thus all time stamps will indicate a common global meaning.

Following Table (as shown in Fig. 2) contains the sample information stored Escrowed information table.

Table 1: User Info

USER ID	RAND VALUE	CSB ID	TGK	SIGNED HASH	TIME
sip:saki@130.237.209.238	QjLmyL5EPrO7F1EiU3sriQ==	-1626928177	S1PhTWfl2jiFpvejCLCm l5L95Z1Aa3BiWcaqGiDJ 2zIdmIPfyNqj6T17qzpg 07dI9fZV6dK7GZFFwUL 1sdG9PPpalvL7hU0hM6q bi4IV93/qwOm6D 2g3mQiPjS69joGJcSDN 85KzV5gTsPaU9kO7PFFA KtN8zvpQa6cUBN787Mnl MZevm6e9YgVqNDrZ4A1Z KA/b4rGT40Nlnq9f4l1R 9rP81P926GKX0dWUeJ54 Pv3v0IUT9fUFFSr8tqoy	oSN/4DI2Q/SXI0G2/QAt Nx qgwdRtswxtZMGdz44qm6 yYSQDOuGPUv4gK33gSlx qSDmny6HhN62nxyb6ZNL rr5FT4tLFV1BwMAu80Om BPysvfAik3OewhUM6R5l xYRhiWHM50EdC7dbegHZ UT37oMtghKLmudRvYop IAPJRvJM=	2010-01-14 17:25:01

save cancel

Figure 2: Escrowed Information (adapted from [2])

LEA Authentication Table - This table stores the LEA's user credential. Here user means the relevant members of LEA those are responsible for conducting lawful interception. Fields in this table will vary depending on the agreement of authentication between Escrow Agent and LEA. In our simulation, three fields such as LEA Id, Agency Address, and password are used. In the implementation, we used a positive integer for LEA Id and other two fields were variable length text strings.

LEA Request Table - LEA must provide sufficient information such as identity, court order, and some other optional information when LEA asks for the escrowed information of a target. For Escrow Agent, it is important to store requested information and time as well as to provide information.

User Agent Identification: This risk is an insider attack. An employee may misuse his access to corporate database using mobile device; for example one can erase classified corporate information remotely using his mobile device to fulfill his financial desire.

IV. LEA MODULE

LEA module is a part of the proposed Lawful Interception model. It is used by the law enforcement agency in order to capture and recover the encrypted recorded conversation between the target parties. Law enforcement agency asks for the escrow information of a targeted conversation to the Escrow Agent with proper judicial order using LEA module of the proposed LI model. After getting the escrow information, LEA module regenerates the session keys from the TGK with other security associates parameters. To regenerate the session keys, LEA module requires the following parameters:

- TGK, RAND, CSBId, and from the escrowed information provided by the Escrow Agent.
- SSRC and Sequence number will be got from the packet header.
- ROC from the SSRC.
- The policy number is fixed as SRTP is still the only protocol used currently.

Additionally, the sequence number of the first SRTP packet must also be sent. Otherwise it would not be possible to identify the forged block. In our simulation, LEA module sent request for escrowed information to the Escrow Agent using a web-based module. After verifying the authentication and other relevant information, Escrow Agent sends back the escrowed information. LEA module uses this escrowed information to regenerate the session keys by which recorded conversation of the target parties will be decrypted for legal prosecution.

A. Steps of the Proposed LEA Module

We have implemented the LEA module using C++. Some existing *minisip* and Open SSL modules have been reused. The LEA module performs the following steps:

- To capture the conversation between two parties.
- The captured packets must be filtered to include only the SRTP packet and RTCP packet of the intercept target.
- To read the captured session and re-packet them into SRTP format.
- To read escrowed information provided by the Escrow Agent.
- The LEA module reads the captured SRTP packets for the Sequence number and SSRC of each packet.
- The LEA module calculates ROC using the Sequence Number and SSRC value. For ROC calculation, we assume that LEA captured the full conversation between two communicating peers for any electronic surveillance.
- LEA module computes a cryptographic context based upon the escrowed information, ROC, Policy number, and CSId. This cryptographic context is generated according to the MIKEY-SRTP protocol. Since SRTP is only implemented protocol, the policy number is fixed and its value is 0. CSId value for initiator of the session is 1 as a sender and 2 as a receiver and vice -versa for the responder. Therefore if the target is the initiator, then the CSId value will be 1; while if the target is responder then the CSId value will be 2.
- Next the LEA module generates the Master Key, Master Salt Key, and Authentication Key.
- Then using these keys and the Cryptographic context, the session keys are generated to decrypt each SRTP payload.
- Using the derived session keys, the LEA module decrypts the encrypted SRTP payload of each of the captured SRTP packets (similarly the SRTCP packets can be decrypted).

B. Possible Trade-offs of the LEA Module

Efficiency and reliability of the LEA module depends on the time required to analyze the captured data, the security during the LEA operations, the network overhead, and the desired transparency of the LEA module. Following trade-offs have been found during analysis of the LEA module:

1) The time required to decode the recorded SRTP packets

We have partitioned the total required time required by LEA to decrypt the recorded session as capturing of SRTP packets have already been captured, delay to get the court order, the time to get the escrowed information, the time to generate the session keys, and the time to actually decode the SRTP packets. Required time for getting the court order could be varies depending upon the procedure in different countries.

After authenticating with the Escrow Agent, LEA module sends court order and other relevant to Escrow Agent. After authenticating LEA module, Escrow Agent will return the escrowed information. After receiving a request, the required time for generating a response depends on the EA module's efficiency and the time required for the LEA module to transmit the information required for the request and the time for the EA to transmit the reply, along with the delay due to the communication channel. If LEA module and Escrow Agent use TLS protected connection - above delay, processing time for mutual authentication as well as transmitting time for authentication information and challenge responses between these entities.

$$t_{res} = t_{TLS} + t_{EA} + \Sigma t_{transmission\_delay} + N_{round\_trips} * RTT$$

Required time for establishing the TLS connection (t_{TLS}) is always constant as the number of required round trips N_{round_trips} to establish a TLS connection is fixed. There are two options to keep the connection open (or not) between the Escrow Agent and LEA module:

- It is notable that Escrow Agent and LEA will work offline. Instead of keeping the TLS connections always open, a new TLS connection can be established when requires. pause
- In contrast, if it requires communicating between Escrow Agent and LEA module frequently, the secure pause and resume session by TLS **Error! Reference source not found.** can be used to re-connect the TLS connection after initial connection.

Response time by the EA (t_{EA}) will depend on implementation mechanism of the whole system. For instance, if the EA modules store the escrowed information in a distributed system, t_{EA} must add the time to reconstruct the information from its distributed elements.

Preprocessing time for transferring packets to/from also includes Transmission delay ($t_{transmission\_delay}$) **Error! Reference source not found.**. In a packet switching based network, a store-and-forward mechanism is used to store the packets in the buffer and then checks for errors before forwarding the packet. Additionally, propagation delay for sending the packets over the link should also be added with the processing time. Transmission delay includes the time to transmit all of the response to the LEA. It is notable that batch requests & responses should also be considered.

Required time to generate session keys from TGK and to decrypt recorded SRTP packets can be divided into two: Time required to derive session keys (t_{kd}) and the time required to decode SRTP packets using the session keys(t_{decode}) = $t_{per\_packet\_decode} * N_{packets}$.

The sum of all these delays is the total time required to decode the captured SRTP packets:

$$t = t_{TLS} + t_{EA} + \Sigma t_{transmission\_delay} + N * RTT + t_{kd} + t_{decode}$$

Therefore, efficiency of the LEA module is inversely proportional to this total delay.

V. Security Of The LEA Module

The LEA module should have necessary security measures that could protect the previously escrowed information received from the Escrow Agent module. The derived session keys and the decoded contents of the captured SRTP file should also be protected by the LEA module. Since SRTP is encrypted, it might seem that there is no need to protect captured SRTP (and SRTCP) traffic. But the time stamps, source and destination IP address and port addresses are not encrypted – this information could be used to learn when and where the intercept was taking place, whose traffic was being intercepted, etc. Besides, if the results are to be presented as evidence in a legal proceedings there may be a need to use secure logging techniques (e.g., Secure Log protocol presented by Rafael Accorsi in sub-section B of Related Work) to preserve the "chain of evidence". Further a complete security analysis of the security of the LEA module is required and considered as future work.

VI. Network Overhead

If the communication between LEA module and Escrow Agent is not often for escrowed information, the network overhead of this communication is not expected to be a significant problem. In contrast, if there is a need for real-time (or near real-time) interception, then there are two problems (1) the key escrow architecture that has been used in the minisip client – prevents real-time interception – since the key information is only escorted after the session has been terminated and (2) there is no requirement that the UA escrow this information immediately after the call terminates. Hence network overhead in the communication between the LEA module and the EA module should not affect any solution. Note that this situation would change if there was a requirement for the UA to escrow the material as soon as it was available to the UA; however, further consideration of this lies outside the scope of this thesis project.

VII. Transparency Of The LEA Module

Transparency of the LEA module depends on its design and implementation. The LEA module should be designed and implemented so as to avoid (or discourage) misuse of the system and to avoid forgery or modification of the captured and decoded data.

A. Validation Module

Validation module in the proposed TTP based LI model introduce for detecting the forgery of the recorded session. Following is the procedure of the validation module:

1. Read an RTCP packet and stores the signed hash from each RTCP in a vector A.
2. Read the escrowed information.
3. Generate the authentication key using the escrowed information.
4. Check the sequence number of the first SRTP packet.
5. Read SRTP packets and create blocks of *n* packets.
6. Create a hash of these SRTP packets in the block with the authentication key (generated in step 2).
7. Save these hashes in vector B.
8. Check the last signed hash of A with the escrowed last signed hash.
9. Load public key certificate for the target and create a certificate object.
10. Call the VerifySign function within the Ossl Certificate class and pass the hashes (one by one) from vector B to compare with the hashes contain in A (obtained from the signed hashes of the RTCP packets).
11. If the return value is -1, then the certificate file or require a certificate is invalid or missing, if the return value is less than zero (0) we cannot verify the signature, if return value is zero (0), then the data is forged, and if the return value is one (1), then block is valid, i.e., it has not been forged or modified.

VIII. Making Forgery With Attacker Module

In the research work, we have modified the recorded session in different ways to identify whether the proposed module can identify or not. Following sections shows how a recorded session can be forged.

Algorithm for UDP Checksum Calculation: For any successful off-line forgery of a packet, the checksum must be recalculated and inserted into a packet, since without correct

checksum value forgery of packets can easily detected. The checksum of a UDP packet is calculated using octets of the pseudo header, UDP header, and data **Error! Reference source not found.**, **Error! Reference source not found.**. The pseudo header consists of the IP Source Address (4 bytes), IP Destination Address (4 bytes), Protocol (2 bytes), and UDP length (2 bytes). Additionally, the checksum value in the UDP header must be set to zero before computing the checksum value. The checksum computation is as follows:

1. Read pseudo header of the packet.
2. Read UDP header of the packet.
3. Set checksum byte of the UDP header to zero (0).
4. Read the data octets of the packet.
5. Check the length of the data. If the data octet is odd add a zero padding byte at the end of the packet.
6. Initialize a variable sum =0.
7. Make 16 bit words with two adjacent 8 bit octets of the data.
8. Sum all 16 bit words to create the new checksum.
9. Add the contents of the pseudo header to the sum.
10. Keep only the last 16 bits of the 32 bit calculated sum and add the carries with the sum.
11. Take one's complement of the sum and assign the sum to store the sum in the checksum field.
12. Return checksum.

Operation Procedure of the Attacker Module: If a wireless hand held device user's store data in the cloud and later want to remove from the cloud, how they could be assured that the data would be permanently deleted from the cloud.

1) Possible ways of modifying a recorded call

A recorded conversation could be modified in a number of ways. These possible modifications have been enumerated to prove that the proposed validation method is able to detect the forgery with the recorded conversation. Since this method detects forgeries on the block of the packets instead of single packets. Thus this proposed forgery detection method does not ensure that it could detect all modification, if a single packet is modified, inserted, or deleted. Reason behind this limitation is that signed hash is computed on a block of packets. **Error! Reference source not found.** enumerates some of the possible modifications of a recorded conversation between two parties.

As a reminder of how to compute the number of possible combination of *n* things taken *r* at a time:

$$^{n}C_{r} = n! / (n - r)!\ r!$$

Using this method we have calculated the number of forgery that is possible with the recorded session. Modification of a session could be done by either adding a single packet, add/or a block of packet, and/or multiple blocks. The first 3 columns of **Error! Reference source not found.** indicate set S_a {a packet, a block, blocks}.

Table 1: Possible combinations of forgery with the recorded session (adapted from [1])

Combination of Forged Options	Number of Combination	Tested	Detectable	Remarks
{a packet, a block, blocks}	7	All	Yes	Which packet is forged is not detectable in the proposed escrow mechanism.
{insert, replace, delete}	7	All	Yes	Without sequence no. of the first SRTP packet, any insertion in the front will show all session as forged.
{front, anywhere in middle, end}	7	All	Yes	Forgery can be made in these three location
{payload, whole packet}	2	All	Yes	Forgery can be made with either payload only or full packet
{with seq. no., without seq. no.}	2	All	Yes	Forgery can be made with either seq. no. or without seq. no.
{with SRTP auth, without SRTP auth}	2	All	Yes	Forgery can be made with either SRTP authentication or without SRTP authentication

Using the formula above we can calculate the number of possible combinations of these three sizes of modifications as the sum of the number of ways of choosing 1 column from 3 ($^{3}C_{1} = 3$), plus the number of ways of choosing 2 column from 3 ($^{3}C_{2} = 3$), plus the number of ways of choosing 3 column from 3 ($^{3}C_{3} = 1$), for a total number of combinations of set S_a is (3+3+1) = 7.

Next 3 columns of the **Error! Reference source not found.**1 is being referred to as set S_b {insert, replace, delete}. The number of ways of choosing 1 column from 3 is $^{3}C_{1} = 3$

The number of ways of choosing 2 column from 3 is $^{3}C_{2} = 3$

The number of ways of choosing 3 column from 3 is $^{3}C_{3} = 1$

Total combination of set S_b is (3+3+1)=7

And next 3 columns of the table set S_c {in front, in the middle, at the end}

The number of ways of choosing 1 column from 3 is $^{3}C_{1} = 3$

The number of ways of choosing 2 column from 3 is $^{3}C_{2} = 3$

The number of ways of choosing 3 column from 3 is $^{3}C_{3} = 1$

Total combination of set S_c is (3+3+1)=7

Next 2 columns of the table set S_d {payload, whole content}

The number of ways of choosing 1 column from 2 is $^{2}C_{1} = 2$

Total combination of set S_d is 2

Next 2 columns of the table set S_e {with sequence number, without sequence number}

The number of ways of choosing 1 column from 2 is ${}^2C_1 = 2$

Total combination of set S_e is 2

Next 2 columns of the table set S_f {with SRTP authentication, without SRTP authentication}

The number of ways of choosing 1 column from 2 is ${}^2C_1 = 2$

Total combination of set S_f is 2

Total combination of the table with the above condition is $(S_a \times S_b \times S_c \times S_d \times S_e \times S_f = (7x7x7x2x2x2) = 2744$.

IX. Evaluation & Results

In order to prove the effectiveness of the proposed LI model, we have simulated the system. It is mentioned earlier that an open source SIP User Agent "minisip" was used. For capturing the conversation, we have used Wireshark [13]. After capturing a full session, we filtered out the SRTP and RTCP packets of the target user and saved these packets on the local machine as a libpcap file. This recorded libcap file was used for evaluating the proposed model.

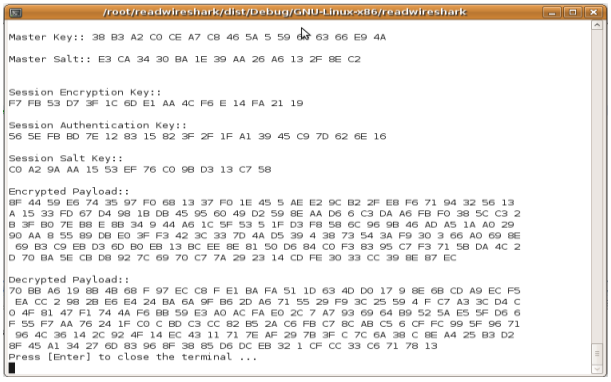

Figure 3: Session Key Generation and Payload Decryption (adapted from [2])

Simulation shows that LEA module can generate session keys from the escrowed information provided by the Escrow Agent (TTP) as shown in Figure 3.

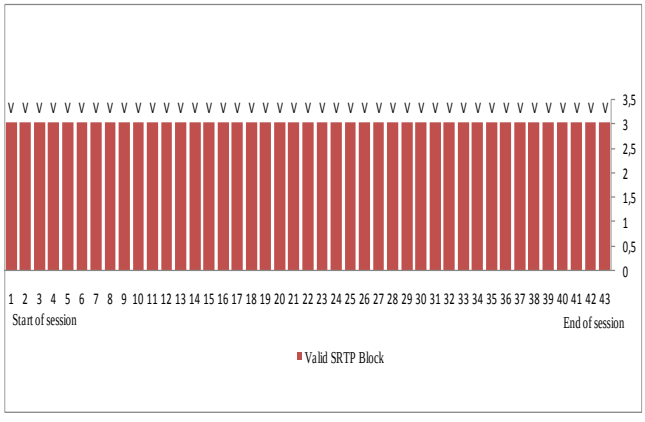

Figure 4: A Valid SRTP Session (adapted from [1])

Using this session keys, Validation module in the proposed LI model can also detect any modification as shown in Figure 4.

A. *Modification of a recorded session*

In order to forge the recorded session, an attacker module was implemented that could modified in following ways:

1. Delete a packet, a block of packets, or blocks of packets.
2. Insert a fake block that does not have the correct SRTP authentication.
3. Insert a fake block that has the correct SRTP authentication (computed using the secret obtained from the escrow agent).
4. Insert a fake block with the same sequence number as a valid block, but insert this before the block with this sequence number without the correct SRTP authentication.
5. Insert a fake block with the same sequence number as a valid block, but insert this before the block with this sequence number with the correct SRTP authentication.
6. Insert a fake block at the end of the session (after the last valid block in the real recorded session) without the correct SRTP authentication.
7. Insert a fake block at the end of the session (after the last valid block in the real recorded session) with the correct SRTP authentication.
8. Insert a fake block before the start of the session (i.e., before the first valid block in the real recorded session) with the correct SRTP authentication.
9. Insert a fake block before the start of the session (i.e., before the first valid block in the real recorded session) without the correct SRTP authentication.
10. Repeat each of the above, but rather than doing the step with a single SRTP block - use a set of 128 SRTP blocks (Where 128 was be block size used in the original captured session.).

Figure 5 shows that forgery is made by replacing the content of the SRTP packets and Figure 6 shows that forgery is made by inserting some packets without changing the sequence number. It is notable that for an actual forgery, new time stamps have to be calculated to avoid the huge difference in time between packet number 128 and packet number 129 in Figure 6.

B. *Forgery detection*

If a forgery is made by any of possible ways as shown in Table 1, investigation shows that the Validation Module can detects all of them. Figure 7 shows the visualized form detected by the Validation Module.

C. *Evaluation Result*

During experiment ~136 micro second (as shown in Figure 8) in average was required to generate session keys from the TGK and other escrowed. Certainly this short time will have less affects on an off-line session analysis.

Using statistical analysis on the required time to derive session keys from the TGK, statistical information shown in Table 2 has been found.

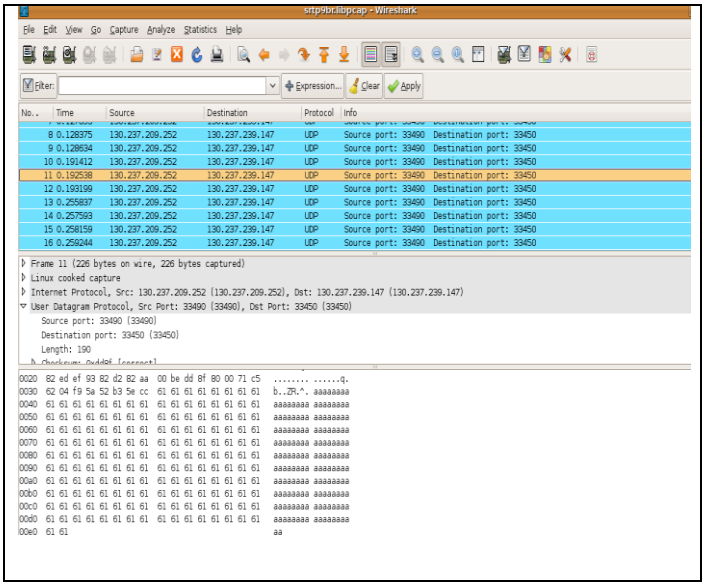

Figure 5: A Packet Forged by Replacing the Content (adapted from [2])

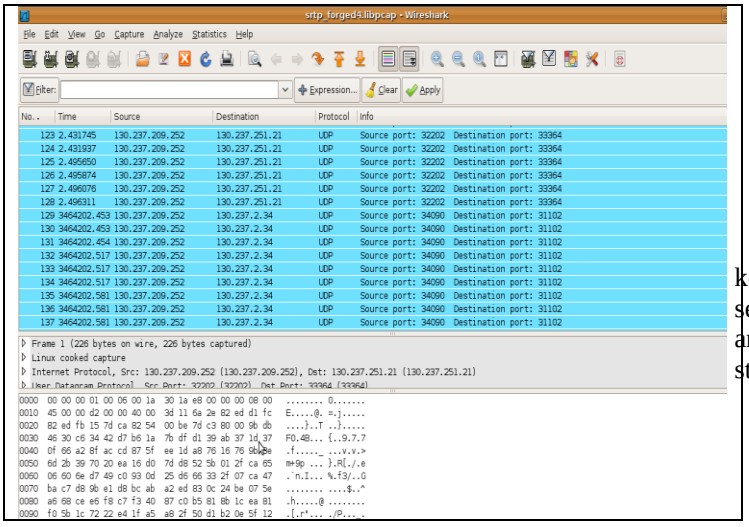

Figure 6: Replaced Block by whole Content (adapted from [2])

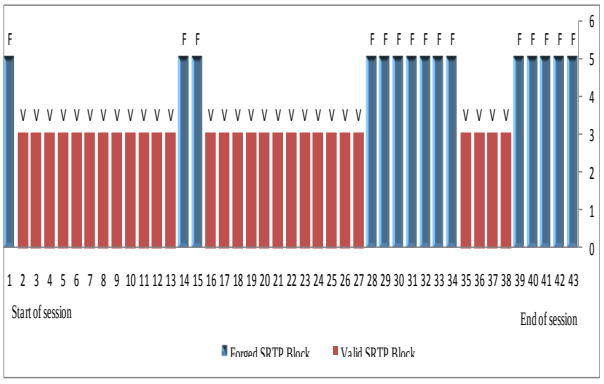

Figure 7: A Forged Session (adapted from [2])

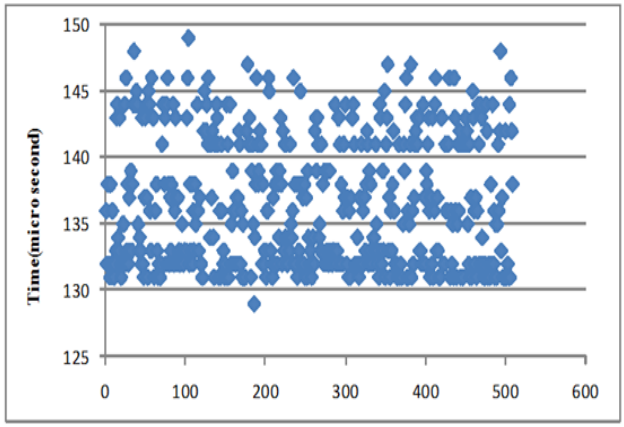

Figure 8: Time required generating Session Keys from TGK and other escrowed information (adapted from [2])

Table2: Data analysis on required time for deriving Session keys from TGK and other escrowed information

Mean	136.5
Median	136
Mode	132
Standard Deviation	4.8
Sample Variance	23
Range	19
Minimum	129

Table 2 shows that the lowest time for generating session keys is 129 micro seconds, and maximum is 149 micro seconds. Figure 9 shows the frequency of the SRTP packets and the time to generate session keys depending on this statistical data.

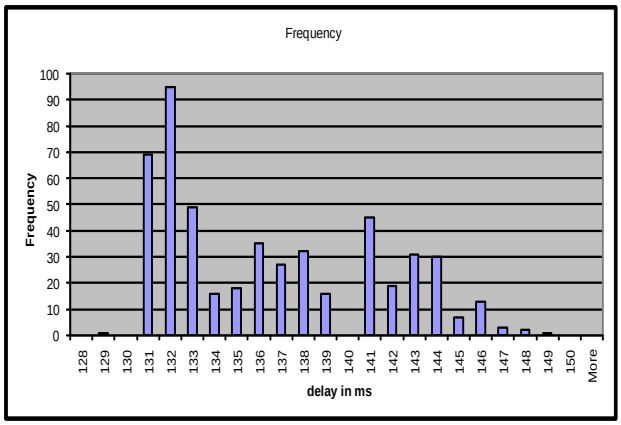

Figure 9: Frequency vs. delay for Generating Session Keys (adapted from [2])

Similarly, mean time required to decrypt a SRTP packet is ~19 micro seconds using the generated session keys as shown in Figure 10.

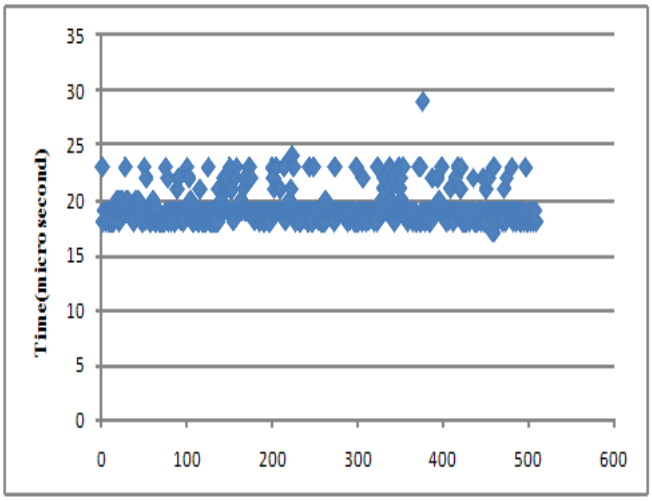

Figure 10: Time required decrypting an SRTP packet using session keys (adapted from [2])

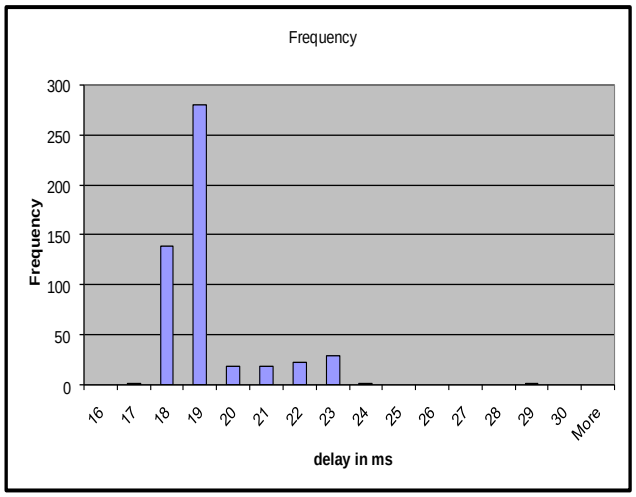

Figure 11: Frequency vs. delay for decrypting SRTP packets (adapted from [2])

Table 3 presents the statistical data for required time decrypting SRTP packets where minimum required time is 17 micro seconds and maximum required time is 29 micro seconds.

Table 3: Data analysis on required time for decrypting an SRTP packet

Mean	19.2
Median	19
Mode	19
Standard Deviation	1.4
Sample Variance	1.96
Range	9
Minimum	17

Besides, Figure 11 shows the frequency vs. delays to decrypt the recorded session for regenerating session keys. Earlier it is mentioned that there are other required delays such as time for getting court order, getting escrow information from the Escrow Agent, transmission delay and the time to generate session keys. Therefore required for decrypting a session depends on the specific scenario. For instance, to get a court warrant depends on the co-ordination between the judicial department and the crime investigation department of that specific country.

X. Conclusion & Future Works

Lawful Interception is an effective tool for protecting national security. In contrast, individual's privacy is also a big issue which must be protected while lawful interception. In this paper, we have shown how a recorded session can be modified for ill-intention and also presented a mechanism to prevent this kind forgery. Proposed LI model could balance between national security and individual privacy, although there more tasks to be done to implement the proposed LI model in the real filed or commercial purposes.

References

[1] Office of the Inspector General, "The implementation of the Communications Assistance For law Enforcement Act", U.S. Department of Justice, Audit Division, Audit report, 6-13 March 2006.

[2] Morshed, M., S., J., "Voic Over IP and Lawful Intercept: Good Cop /Bad Cop", Masters Thesis, Royal Institute of Technology, Sweden, 2010.

[3] Minisip, http://www.Minisip.org/, last accessed on 28-09-2011.

[4] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman "The Secure Real-Time Transport Protocol (SRTP)", IETF, Network Working Group, RFC3711, March 2004.http://www.ietf.org/rfc/rfc3711.txt

[5] C. Hett, N. Kuntze, A.U.Schmidt, "Non-repudiation of Coice-over-IP", Fraunhofer SIT, Darmstadt, Germany. Last Access on 05-10-2009

[6] Rafael Accorsi, "Log Data as Digital Evidence: What Secure Logging Protocols Have to Offer?", 2009 33rd Annual IEEE International Computer Software and Applications Conference, Seattle, Washington, USA, July 20-July 24, 2009.

[7] Vassilios Stathopoulos, Panayiotis Kotzanikolaou, and Emmanouil Magkos," A Framework for Secure and Verifiable Logging in Public Communication Networks ", Critical Information Infrastructures Security, Volume 4347/2006, Springer Berlin/Heidelberg, 2006.

[8] Electronic Privacy information Center, The Clipper Chip, "http://epic.org/crypto/clipper/", last accessed on29-09-2009.

[9] Md. Sakhawat Hossen, A session Initiation Protocol User Agent with Key Escrow: providing authencity for recording session, Masters Thesis, School of Information and Communication Technology, Royal Institute of Technology (KTH), Stockholm, Sweden, January 2010.

[10] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman, "MIKEY: Multimedia Internet KEYing", IETF, Network Working Group, RFC 3830, August 2004. http://www.ietf.org/rfc/rfc3830.txt last Accessed on 13-09-2009.

[11] http://msdn.microsoft.com/en-us/library/aa380513 (VS.85).aspx.Last Accessed on 30-12-2009.

[12] http://en.wikipedia.org/wiki/Transmission_delay, Last Accessed on 30-12-2009.

[13] Wireshark, http://www.wireshark.org/. Last accessed on 26-09-2009.

[14] T. Berners-Lee, R. Fielding, U. C. Irvine, and L. Masinter, "Uniform Resource identifiers (URI)", IETF, Network Working Group, RFC 2396, August 1998, http://www.ietf.org/rfc/rfc2396.txt. , Last Accessed on 23-10-2009.

[15] J. Postel, "User Datagram Protocol", RFC 768, IETF, August 1980.