Wireless Netw (2018) 24:2915–2926
https://doi.org/10.1007/s11276-017-1512-3

Implementing a secure VoIP communication over SIP-based networks

Wen-Bin Hsieh[1] · Jenq-Shiou Leu[1]

Published online: 25 April 2017

Abstract Recent years the Session Initiation Protocol (SIP) is commonly used in establishing Voice over IP (VoIP) calls and has become the centerpiece for most VoIP architecture. As wireless and mobile all-IP networks become prosperous, free VoIP applications are utilized in all places. Consequently, the security VoIP is a crucial requirements for its adoption. Many authentication and key agreement schemes are proposed to protect the SIP messages, however, lacking concrete implementations. The performance of VoIP is critical for users' impressions. In view of this, this paper studies the performance impact of using key agreements, elliptic curve Diffie–Hellman and elliptic curve Menezes–Qu–Vanstone, for making a SIP-based VoIP call. We evaluate the key agreement cost using spongycastle.jce.provider package in Java running on android-based mobile phones, the effect of using different elliptic curves and analyze the security of both key agreements. Furthermore, we design a practical and efficient authentication mechanism to deploy our VoIP architecture and show that a VoIP call can be established in an acceptable interval. As a result, this paper provides a concrete and feasible architecture to secure a VoIP call.

Keywords Security · VoIP · SIP · ECDH · ECMQV

✉ Wen-Bin Hsieh
d9802106@mail.ntust.edu.tw

Jenq-Shiou Leu
jsleu@mail.ntust.edu.tw

[1] Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

1 Introduction

As wireless and mobile networks flourish, the properties of flexibility, convenient and low cost make VoIP applications be widely used in the enterprise and consumer markets. End-user get used to making a phone call over the public internet rather than via the public switch telephone network (PSTN). In addition to the transmission of voice, video and multimedia also benefit from this technology. The current mainstream of VoIP is Session Initiation Protocol [1] which uses text-based signals to establish, modify and terminate media transmission sessions. The media streams, such as voice and video over IP networks, are transmitted by utilizing the Real-Time Transport Protocol [2]. The maturity of VoIP standards and quality of service (QoS) on IP networks opens up lots of services like the IP Multimedia Subsystem [3], online video conferencing, and video on demand (VoD).

In view of numerous advantages in SIP application, the security become a prerequisite. In 2013, Edward Snowden, an American computer professional, former CIA employee, and former National Security Agency (NSA) contractor, revealed numerous global surveillance programs run by the NSA and the Five Eyes with the cooperation of telecommunication companies and European governments. This news makes governments and enterprises start to focus on the security of transmitting media over IP networks. Before that, many security threats had been studied [4, 5], such as Denial of Service (DoS) [6], SIP malformed message attacks [7] and Abusing SIP Authentication attacks [8]. Therefore, a lot of security frameworks and schemes [7, 9] were proposed. Most of the proposed schemes emphasize the authentication in the registration phase or the signal protection of SIP. There is no concrete and overall system being implemented.

In 2012, Shen et al. [10] present the impact of Transport Layer Security (TLS) on SIP Server. They implement TLS to establish a secure channel before sending SIP signals and show that using TLS reduces the performance compared to typical case of SIP-over-UDP. The cost of RSA operations used for session negotiation is the primary factor. The experiment is running on an Intel-based server, not a mobile device.

In [11], Ashok et al. proposed a mechanism for enhancing privacy of Voice Calls by using ECDH. The ECC Key agreement is implemented in Asterisk Gateway Interface (AGI) [12] server which locates between the asterisk servers. The mobile phones transmit voice packet to asterisk servers, then voice data are encrypted and decrypted between the asterisk servers using ECDH key. In this mechanism, end-to-end privacy does not be provided. The Diffie–Hellman key exchange is vulnerable to a man-in-the-middle attack. In this attack, an attacker intercepts public values and sets up two different session keys with both parties involved in communication. Thus the attacker can eavesdrop the call. Elliptic Curve Menezes–Qu–Vanstone (ECMQV) is an authenticated key agreement, it provides protection against Man in the Middle (MitM) attacks.

To secure SIP signals and media packets is one of the most important things in SIP-based VoIP environment. In view of this point, we realize a VoIP system to offer a secure communication environment.

In this paper, we make the following contributions.

- We present a feasible and secure SIP-based VoIP system. The system utilizes Java security package and Openssl library [13] to implement TLS that protects SIP signals. Furthermore, VoIP calls present key agreements by integrating ECDH and ECMQV, using the agreement key to secure voice packets which realize the encryption portion of SRTP [14]. In order to process the problem which firewalls blocks RTPs, we Ref. [15] to implement NAT traversal.
- We use android-based smart phones to run key agreements on different algorithms (ECDH or ECMQV) with elliptic curves recommended by National Institute of Standards and Technology (NIST) [16]. We present the performance by up to 20 combinations which are two algorithms with ten elliptic curves parameters. The result is a useful reference for users who want to implement elliptic curve cryptosystems (ECC) [17] on mobile devices.
- Only legal users can access our VoIP resources, thus we also propose an efficient and secure authentication mechanism in SIP registration process. We utilize the unique International Mobile Equipment Identity (IMEI) of the mobile device and the unique serial number, International Mobile Subscriber Identity (IMSI), of the subscriber identification module (SIM) card to generate the SIP REGISTER signal.

In past years, various security schemes are proposed, such as Password Authenticated Key Exchanged based (PAKE) schemes [18, 19], Hash and Symmetric Encryption based schemes [20, 21], Public Key Cryptography (PKC) based schemes [22, 23] and so forth. However, lacking implementations cannot provide concrete references to users. The proposed system integrates the key parameters into Session Description Protocol (SDP) [24] to achieve the key agreement and the experimental data make users aware of the overhead.

The remainder of this paper is organized as follows. The next section provides a brief background of SIP and TLS. Section 3 gives a brief overview of cryptosystems that is ECC, ECDH and ECMQV. In Sect. 4, we describe our secure VoIP system. Section 5 evaluates the performance on different combinations of elliptic curves and algorithms and the experiment results. Finally, we conclude this paper in Sect. 6.

2 Background

2.1 SIP overview

SIP is a signaling and an application-layer control protocol which is commonly used for VoIP communication. SIP defines two essential types of entities: user agents (UAs) and SIP servers. SIP servers are made up of registrar servers and proxy servers. Registrar servers are responsible for location management and proxy servers for message forwarding. SIP is based around request/response transactions, in a similar manner to the Hypertext Transfer Protocol (HTTP). Proxying, which means SIP message forwarding, is a critical function in the SIP infrastructure.

The standard application functionalities, such as authentication, authorization and media session setup, all require the proxy server to keep session state information.

Figure 1 shows a typical message flow of SIP proxying. User Agent Client (UAC) and User Agent Server (UAS) represent the caller and callee of a media session. First, the UAC and the UAS send register messages to the SIP proxy Server. The register message contains UAC/UAS's credentials that verify its claimed identity (e.g., generally base on MD5 digest algorithm [25]). After passing authentication, SIP proxy server responds 200 OK messages to the UAC and the UAS respectively. The authentication information is optional; however, it is commonly deployed between UAs and its first-hop SIP server for allowing legal UAs to access resources.

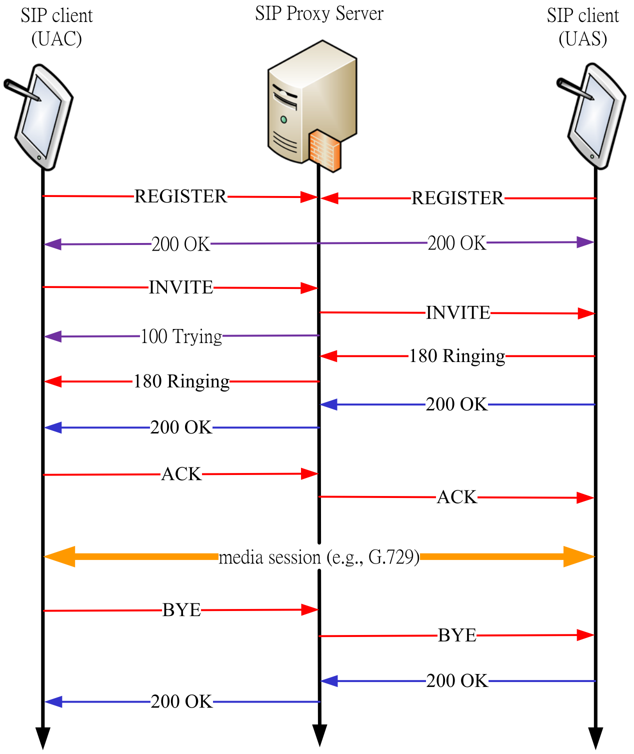


Fig. 1 SIP register and call setup flow

When the UAC wants to establish a session with the UAS, it first sends an INVITE message to the proxy server. Then the proxy server makes a response to the UAC with a 100 Trying message to inform the UAC that the message has been received. Then the proxy server checks the contact address for the SIP URI and forwards the message to the UAS. After receiving the INVITE message, the UAS acknowledges receipt with a 180 Ringing message and the callee's phone rings. When the callee picks up the phone, the UAS sends out a 200 OK message. Both the 180 Ringring and 200 OK messages are routed back to the UAC through the proxy server. Once receiving the 200 OK message, UAC generates an ACK message for response. Then the media session is established, both endpoints use a media protocol, such as RTP, to communicate directly. When the conversation is over, the UAC hangs up and sends UAS a BYE message which is forwarded by the proxy server. The UAS then sends a 200 OK message in response. Figure 1 presents a basic flowchart, but in real networks to have multiple proxy servers between UAs is common.

2.2 TLS overview

In this part, the brief depiction of the TLS protocol is given. For more detail, please read [26–28].

There are three subprotocols in the TLS protocol that are used to control the session connection [29]: the handshake, change cipher spec, and alert protocols. The TLS handshake protocol is used to negotiate the session parameters. The alert protocol is used to notify the other party of an error condition. The change cipher spec protocol is used to change the cryptographic parameters of a session. In this paper, we focus on the handshake protocol. The handshake protocol consists of a series of message exchanges between the client and the server, it allows the participants to negotiate a specific cipher suite which includes ey establishment, digital signature, confidentiality and integrity algorithms. For an example, TLS_RSA_WITH_AES_256_CBC_SHA is a cipher suite, indicating that the RSA public key algorithm is used for shared secret key exchange and authentication; 256-bit AES in Cipher Blocking Chaining mode is used for bulk data encryption; and SHA-1 [30] is used as the message digest algorithm to compute the Message Authentication Code.

In the TLS handshake protocol, there are three types: Normal TLS handshake, Mutual TLS Handshake and Resumed TLS handshake. The normal TLS handshake is the reduced version of the mutual TLS handshake, it do not request authentication of clients. Figure 2 presented the process of the mutual TLS handshake.

First, the client launches the handshake with a ClientHello message which contains the version of the protocol, the cipher suite list and the compression algorithms that the client supports. To prevent replay attacks, a

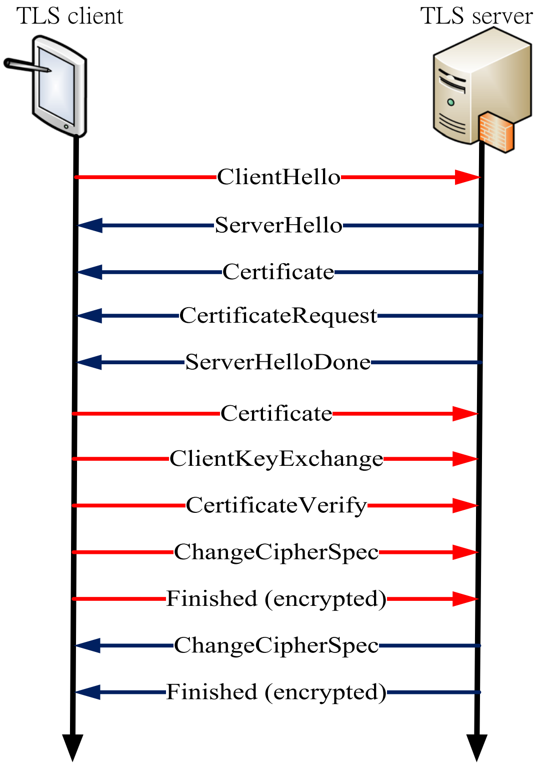


Fig. 2 Mutual TLS handshake process

random number and timestamp is included in the message. After receiving the ClientHello message, the server specifies the protocol version and chooses the cipher suite and the compression algorithms among those proposed by the client. Then the server sends a ServerHello message back indicating which cipher suite it accept. Also the ServerHello message contains a timestamp, a random number which is a part of the key material, and an optional session_id that can be used to resume the session by the client later. Next, the server sends the Certificate message which has the server's X.509 certificate containing its public key. Then the server transmits a CertificateRequest message to request the client's certificate. Finally the server sends ServerHelloDone message to indicate all the messages have been sent in this phase. Once receiving the server's CertificateRequest message, the client responds it with a Certificate message containing client's certificate with its public key. For receiving server's certificate, the client uses Certificate Authority (CA)'s public key to verify its certificate for authenticating the server. After the verification of server's certificate, the client gets the server's public key from the certificate. Thereupon the client generates a pre_master_secret and uses the server's public key to encrypt it. Next the client sends the server a ClientKeyExchange message with the encrypted pre_master_secret to and a CertifcateVerify message containing a digest signature signed by client's private key. The server can authenticate the client using client's public key and decrypt the encrypted pre_master_secret by its own private key. Based on the same pre_master_secret, the server and the client both can compute a common master_secret which is used to generate shared symmetric keys for message authentication and bulk data encryption. The ChangeCipherSpec message, both the server and the client exchange, is used to indicate the sender has switched to the newly negotiated algorithms. At last, the Finished message used to ensure the integrity of the handshake has been transmitted to the other party. The Finished message contains a MAC digest of the negotiated master_secret.

During a configured interval, the resumed TLS hanshake allows the server and the client to restore the session information including the chosen algorithms and the master_secret. The resumption mode reduces the cost of renegotiating a new pre_master_secret.

3 Cryptosystem

3.1 Elliptic curve cryptosystem

In 1985, Koblitz [17] and Miller [31] proposed public key cryptosystems using the group of points on an elliptic curve. The primary advantage that elliptic curve systems is that one can use an elliptic curve group that is smaller in size while maintaining the same level of security. The result is smaller key sizes, bandwidth savings, and faster implementations, features which are especially attractive for security applications where computational power and integrated circuit space is limited, such as smart cards and mobile devices. Table 1 [32] compares the cipher strength in both Rivest–Shamir–Adleman (RSA) and ECC.

Table 2 [33] compares the computing time in both ECC and RSA. The computation time needed to solve an ECC based on ECDLP with a length of 160 bits is equal to that of solving an RSA with a key length of 1024 bits. Today, in practice, elliptic curve groups over the finite field of F_p and F_2^m are used. Over the finite fields of F_p, an elliptic curve is defined by an equation of the form $y^2 = x^3 + ax + b$. Over the finite fields of F_2^m, an elliptic curve is defined by an equation of the form $y^2 + xy = x^3 + ax^2 + b$, where a and b are arbitrary constants and $4a^3 + 27b^2 \neq 0$. To qualify as an abelian group, an elliptic curve defines O, a point at infinity, which serves as the identity element for some operations. The operations include the addition of two points and the double of a point. The rules can refer to [17].

The total number of points on a curve, described mathematically as $\#E(F_p)$ or $\#E(F_2^m)$, is referred to as the order of a curve. The ECDLP is defined as follows: given P $\in \#E(F_p)$ and $Q = [a]P$, find a.

3.2 Elliptic curve Diffie–Hellman key agreement protocol

The D–H key agreement protocol is one of the earliest practical methods of exchanging keys over an insecure channel. The original D–H was based on discrete logarithm problem. In this protocol, if Alice and Bob want to set up a random secret (session) key for their private key system, they first decide on a cyclic group, G, of order n and a generator, g, of the group in public. Then, Alice randomly

Table 1 Comparison of security strength

Security strength	IFC (RSA)	ECC (ECDH, ECDSA, ECMQV)
80	1024	160–223
112	2048	224–255
128	3072	256–383
192	7860	384–511
256	15,360	512+

ECC elliptic curve cryptography, *ECDH* elliptic curve Diffie–Hellman, *ECDSA* elliptic curve digital signature algorithm, *ECMQV* elliptic curve Menezes–Qu–Vanstone, *IFC* integer factorization cryptography, *RSA* Rivest–Shamir–Adleman

Table 2 Comparison of computation time

ECC[a]		RSA[b]	
Key length (bits)	Computing time (MIPS-years)	Key length (bits)	Computing time (MIPS-years)
150	3.8×10^{10}	1024	3×10^{11}
205	7.1×10^{18}	1280	1×10^{14}
234	1.6×10^{28}	1536	3×10^{16}
		2048	3×10^{20}

ECC elliptic curve cryptography, *MIPS* million instructions per second, *RSA* Rivest–Shamir–Adleman

[a] Uses Pollard's rho method [34] to solve the problem of elliptic curve discrete logarithm

[b] Uses the generalized number field sieve method [35] to solve the problem of factoring two large primes

Table 3 Elliptic curve cryptography domain parameters

q	The field size (may be either an odd prime p or 2^m, where m is a prime.
FR	An indication of the basis
a, b	Two field elements that define the equation of the curve
SEED	An optional bit string that is included if the elliptic curve was randomly generated in a verifiable fashion
G	A base (generating) point consisting of (X_G, Y_G) of prime order on the curve
n	The order of the point G
h	The cofactor, which is equal to the order of the curve divided by n

chooses a prime integer, $a \in [1, n - 1]$, and sends g^a to Bob. Likewise, Bob compute g^b for a random prime number, $b \in [1, n - 1]$, and sends it to Alice. The secret key, g^{ab}, is then set up, which Alice computes as $(g^b)^a$ and Bob computes as $(g^a)^b$.

The ECDH key agreement protocol uses the D–H key agreement protocol based on ECDLP to computes the session key $[ab]P$. Table 3 defines the domain parameters for the ECC schemes [36].

The process of the elliptic curve Diffie–Hellman key exchange protocol refers to [37].

3.3 Elliptic curve Menezes–Qu–Vanstone key agreement protocol

In the ECMQV protocol [38], both parties are assumed to have long-term public and private key pairs. For example, Alice has the static key pair, $[a]G$ as the public key and a as the private key. Bob has the static key pair $[b]G$ and b likewise. To agree on a shared secret, Alice and Bob both generate two transient key pairs that are $([c]G, c)$ and $([d]G, d)$. After that, they exchange the public keys of these transient keys as in the standard ECDH protocol shown in Fig. 3.

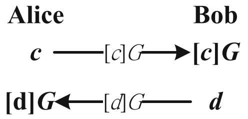


Fig. 3 Elliptic curve Diffie–Hellman key agreement protocol

After exchanging the public keys, Alice knows

$$a, c, [a]G, [c]G, [b]G \text{ and } [d]G$$

and Bob knows

$$b, d, [b]G, [d]G, [a]G \text{ and } [c]G$$

The shared secret is then computed by Alice according to the following algorithm:

```
ECMQV key derivation
INPUT: A set of domain parameter ($\#E(F_p)$, $q$, $h$, $G$) and $a$, $c$, $[a]G$, $[c]G$, $[b]G$ and $[d]G$
OUTPUT: A shared secret $Q$
1. $n \leftarrow \lceil log_2(\#E(F_p)) \rceil /2$
2. $u \leftarrow (x([c]G)(\text{mod } 2^n)) + 2^n$,    Convert the x-coordinate of the public key $[c]G$ to an integer
3. $s \leftarrow c + ua \ (\text{mod } q)$
4. $v \leftarrow (x([d]G)(\text{mod } 2^n)) + 2^n$,    Convert the x-coordinate of the public key $[d]G$ to an integer
5. $Q \leftarrow [s]([d]G + [v]([b]G))$
6. If $Q$ is an infinity point goto step 1.
7. Output $Q$.
```

Bob also compute the same point of Q by changing the parameters (a, c, $[a]G$, $[c]G$, $[b]G$ and $[d]G$) in the above algorithm with b, d, $[b]G$, $[d]G$, $[a]G$ and $[c]G$. Then the shared secret Q is agreed.

4 The proposed VoIP system

The VoIP system we proposed is presented in Fig. 4. User equipments such as mobile devices can access VoIP services on the internet provided by 3G/4G base stations or WiFi access points. SIP Proxy servers and RTP Relay Servers are deployed behind the firewall which defends malicious internet attacks. The certificate authority and the database server, which are the kernel of the system, are allocated behind the second firewall and the intrusion prevention system.

In this architecture, we assume that CA and DB are well-protected (In reality, a successful intrusion will make the in-use certificates be revoked and new one be issued). Clients are equipped with requisite certificates containing server's and callees' public keys.

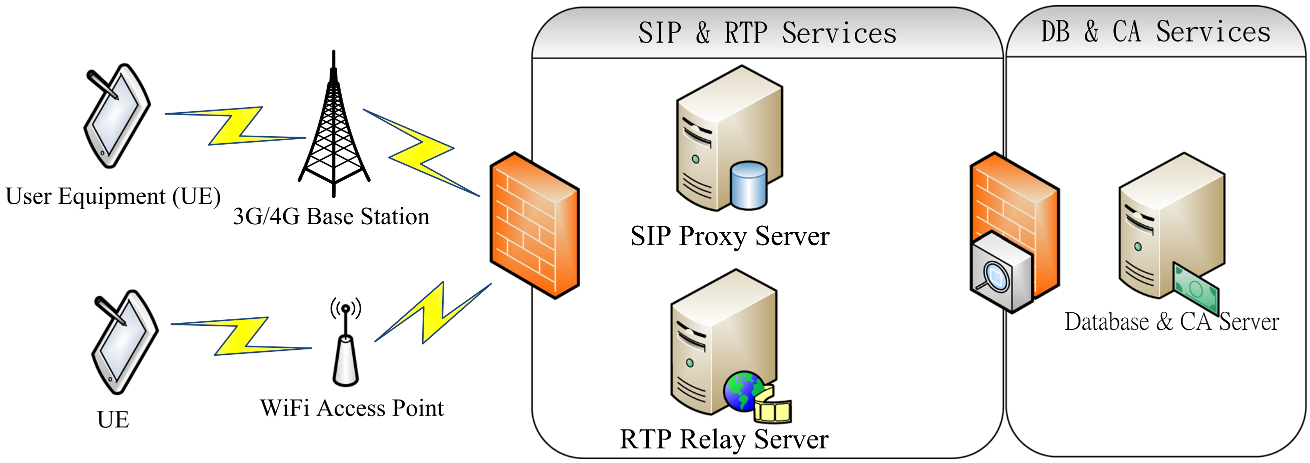


Fig. 4 The architecture of the proposed VoIP system

First, the SIP client must register and be authenticated by the SIP server using the proposed authentication mechanism. Then when two SIP clients want to establish a media session, they use the SIP messages integrating ECMQV protocol to achieve a key agreement and use the agreed session key to encrypt the RTP packets with AES256. The SIP messages are protected by a secure communication channel which is provided by Transport Layer Security. The authentication mechanism and the integration of SIP and ECMQV key agreement protocol will be described in the following.

4.1 Authentication mechanism

Clients installed our SIP application will first set up a TLS-secured channel with the SIP server. Then the UE will send its *IMSI* and *IMEI* via the secure communication channel to the server. After receiving the *IMSI* and *IMEI*, the server encrypts and stores this information with the corresponding SIP account into database server. When the client wants to register to the SIP server, it starts with computing the authentication code as follows:

$$\mathrm{SHA256}(IMSI|CSeq)|\ \mathrm{SHA256}(IMEI|CSeq)|\ \mathrm{SHA256}(CSeq)|SIP\_ACCOUNT$$

Then the client appends the authentication code to the SIP REGISTER message and sends it to the SIP server. The SIP REGISTER message is illustrated as Fig. 5 which is captured by WireShark packet analyzer. The first picture in Fig. 5 presents the encrypted application data in TLS which is unable to read. In order to explain the modified register message, we temporarily halt TLS protocol to show the content.

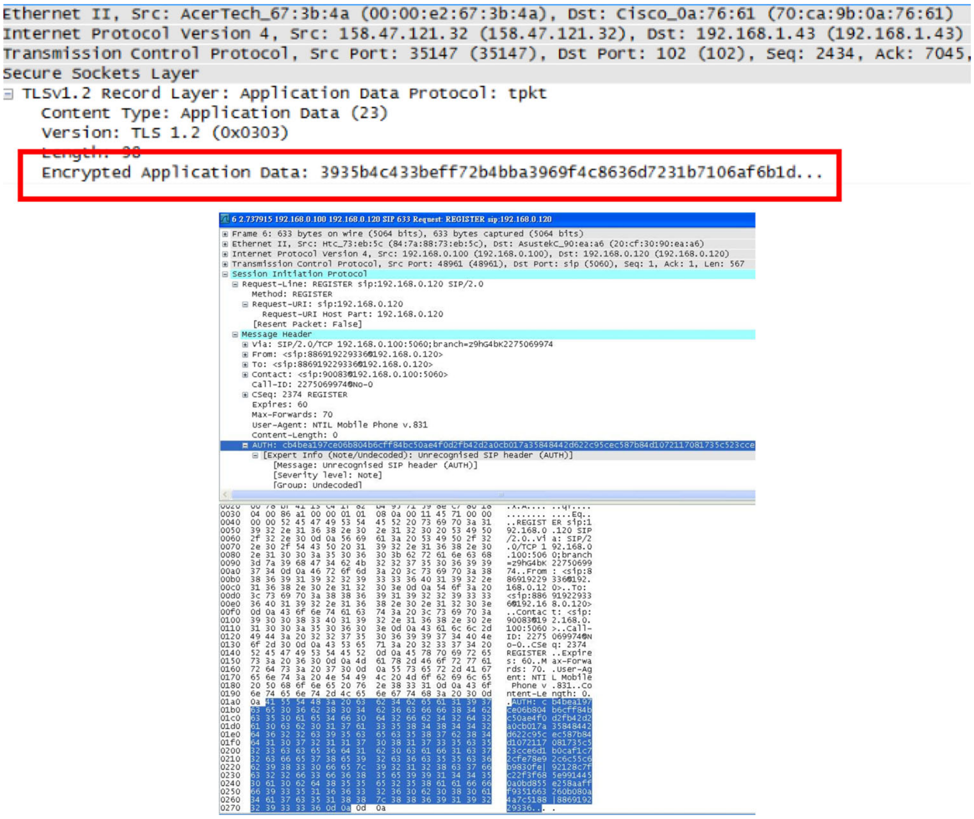
The registration mechanism incorporates *IMSI* and *IMEI* to bind the application with the user's mobile device, in order to avoid an attacker installing the application on other devices to pretend users. The pervasive authentication that uses SMS may be broken by redirecting the SMS authentication code. New accounts can register the system with new IMSIs and IMEIs, the origianl users can request to the existing IMSIs and IMEIs from the setting.

The SIP server will store *CSeq* into the database at first time and extract *IMSI* and *IMEI* from the database in accordance with the SIP account. Next the SIP server computes the authentication code in the same way and compares it with the client's. If the result is the same, the SIP server will respond 200 OK message to the client; otherwise, 403 FORBIDDEN message will be sent.

After the first registration, the subsequent register messages will adhere to the following process:

```
if(exists($SIP\_ACCOUNT$) == true){
  if ($CSeq\ != 0\ \&\&\ CSeq < (CSeq[DB] + 250)$){
    $AUTH_{\mathrm{server}} = (\mathrm{SHA256}(IMSI[DB]|CSeq)|$
      $\mathrm{SHA256}(IMEI[DB]|CSeq)|$
                    $\mathrm{SHA256}(CSeq)|SIP\_ACCOUNT)$;
    if(!strcmp($AUTH_{\mathrm{client}}, AUTH_{\mathrm{server}}$))
        send(200_OK);
    else
        send(403_FORBIDDEN);
  }else
        send(403_FORBIDDEN);
}else
        send(403_FORBIDDEN);
```

where is *CSeq* received from the client and *CSeq*[*DB*], *CSeq*[*DB*] and *CSeq*[*DB*] is extracted from the database. Considering packets may lose over the internet, the SIP server will accept the tolerance difference of *CSeq* in 250.



```
Ethernet II, Src: AcerTech_67:3b:4a (00:00:e2:67:3b:4a), Dst: Cisco_0a:76:61 (70:ca:9b:0a:76:61)
Internet Protocol Version 4, Src: 158.47.121.32 (158.47.121.32), Dst: 192.168.1.43 (192.168.1.43)
Transmission Control Protocol, Src Port: 35147 (35147), Dst Port: 102 (102), Seq: 2434, Ack: 7045,
Secure Sockets Layer
⊟ TLSv1.2 Record Layer: Application Data Protocol: tpkt
    Content Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 98
    Encrypted Application Data: 3935b4c433beff72b4bba3969f4c8636d7231b7106af6b1d...
```

```
6 2.737915 192.168.0.100 192.168.0.120 SIP 633 Request: REGISTER sip:192.168.0.120
⊞ Frame 6: 633 bytes on wire (5064 bits), 633 bytes captured (5064 bits)
⊞ Ethernet II, Src: Htc_73:eb:5c (84:7a:88:73:eb:5c), Dst: AsustekC_90:ea:a6 (20:cf:30:90:ea:a6)
⊞ Internet Protocol Version 4, Src: 192.168.0.100 (192.168.0.100), Dst: 192.168.0.120 (192.168.0.120)
⊞ Transmission Control Protocol, Src Port: 48961 (48961), Dst Port: sip (5060), Seq: 1, Ack: 1, Len: 567
⊟ Session Initiation Protocol
  ⊟ Request-Line: REGISTER sip:192.168.0.120 SIP/2.0
      Method: REGISTER
    ⊟ Request-URI: sip:192.168.0.120
        Request-URI Host Part: 192.168.0.120
      [Resent Packet: False]
  ⊟ Message Header
    ⊞ Via: SIP/2.0/TCP 192.168.0.100:5060;branch=z9hG4bK2275069974
    ⊞ From: <sip:886919229336@192.168.0.120>
    ⊞ To: <sip:886919229336@192.168.0.120>
    ⊞ Contact: <sip:90083@192.168.0.100:5060>
      Call-ID: 2275069974@No-0
    ⊞ CSeq: 2374 REGISTER
      Expires: 60
      Max-Forwards: 70
      User-Agent: NTIL Mobile Phone v.831
      Content-Length: 0
    ⊟ AUTH: cb4bea197ce06b804b6cff84bc50ae4f0d2fb42d2a0cb017a35848442d622c95cec587b84d1072117081735c523cce
      ⊟ [Expert Info (Note/Undecoded): Unrecognised SIP header (AUTH)]
          [Message: Unrecognised SIP header (AUTH)]
          [Severity level: Note]
          [Group: Undecoded]

0030  04 00 86 a1 00 00 01 01  08 0a 00 11 45 71 00 00   ........ ....Eq..
0040  00 00 52 45 47 49 53 54  45 52 20 73 69 70 3a 31   ..REGIST ER sip:1
0050  39 32 2e 31 36 38 2e 30  2e 31 32 30 20 53 49 50   92.168.0 .120 SIP
0060  2f 32 2e 30 0d 0a 56 69  61 3a 20 53 49 50 2f 32   /2.0..Vi a: SIP/2
0070  2e 30 2f 54 43 50 20 31  39 32 2e 31 36 38 2e 30   .0/TCP 1 92.168.0
0080  2e 31 30 30 3a 35 30 36  30 3b 62 72 61 6e 63 68   .100:506 0;branch
0090  3d 7a 39 68 47 34 62 4b  32 32 37 35 30 36 39 39   =z9hG4bK 22750699
00a0  37 34 0d 0a 46 72 6f 6d  3a 20 3c 73 69 70 3a 38   74..From : <sip:8
00b0  38 36 39 31 39 32 32 39  33 33 36 40 31 39 32 2e   86919229 336@192.
00c0  31 36 38 2e 30 2e 31 32  30 3e 0d 0a 54 6f 3a 20   168.0.12 0>..To: 
00d0  3c 73 69 70 3a 38 38 36  39 31 39 32 32 39 33 33   <sip:886 91922933
00e0  36 40 31 39 32 2e 31 36  38 2e 30 2e 31 32 30 3e   6@192.16 8.0.120>
00f0  0d 0a 43 6f 6e 74 61 63  74 3a 20 3c 73 69 70 3a   ..Contac t: <sip:
0100  39 30 30 38 33 40 31 39  32 2e 31 36 38 2e 30 2e   90083@19 2.168.0.
0110  31 30 30 3a 35 30 36 30  3e 0d 0a 43 61 6c 6c 2d   100:5060 >..Call-
0120  49 44 3a 20 32 32 37 35  30 36 39 39 37 34 40 4e   ID: 2275 069974@N
0130  6f 2d 30 0d 0a 43 53 65  71 3a 20 32 33 37 34 20   o-0..CSe q: 2374 
0140  52 45 47 49 53 54 45 52  0d 0a 45 78 70 69 72 65   REGISTER ..Expire
0150  73 3a 20 36 30 0d 0a 4d  61 78 2d 46 6f 72 77 61   s: 60..M ax-Forwa
0160  72 64 73 3a 20 37 30 0d  0a 55 73 65 72 2d 41 67   rds: 70. .User-Ag
0170  65 6e 74 3a 20 4e 54 49  4c 20 4d 6f 62 69 6c 65   ent: NTI L Mobile
0180  20 50 68 6f 6e 65 20 76  2e 38 33 31 0d 0a 43 6f    Phone v .831..Co
0190  6e 74 65 6e 74 2d 4c 65  6e 67 74 68 3a 20 30 0d   ntent-Le ngth: 0.
01a0  0a 41 55 54 48 3a 20 63  62 34 62 65 61 31 39 37   .AUTH: c b4bea197
01b0  63 65 30 36 62 38 30 34  62 36 63 66 66 38 34 62   ce06b804 b6cff84b
01c0  63 35 30 61 65 34 66 30  64 32 66 62 34 32 64 32   c50ae4f0 d2fb42d2
01d0  61 30 63 62 30 31 37 61  33 35 38 34 38 34 34 32   a0cb017a 35848442
01e0  64 36 32 32 63 39 35 63  65 63 35 38 37 62 38 34   d622c95c ec587b84
01f0  64 31 30 37 32 31 31 37  30 38 31 37 33 35 63 35   d1072117 081735c5
0200  32 33 63 63 65 36 64 31  62 30 63 61 66 31 63 37   23cce6d1 b0caf1c7
0210  32 63 66 65 37 38 65 39  32 63 36 63 35 35 63 36   2cfe78e9 2c6c55c6
0220  62 39 38 33 30 66 65 7c  39 32 31 32 38 63 37 66   b9830fe| 92128c7f
0230  63 32 32 66 33 66 36 38  35 65 39 39 31 34 34 35   c22f3f68 5e991445
0240  30 61 30 62 64 38 35 35  65 32 35 38 61 61 66 66   0a0bd855 e258aaff
0250  66 39 33 35 31 36 36 33  32 36 30 62 30 38 30 61   f9351663 260b080a
0260  34 61 37 63 35 31 38 38  7c 38 38 36 39 31 39 32   4a7c5188 |8869192
0270  32 39 33 33 36 0d 0a 0d  0a                        29336... .
```

Fig. 5 SIP register message

4.2 Integration of SIP and ECMQV

If two SIP clients want to establish a media session. The caller (UAC) first generates the ephemeral key pair that is ($[c]G$, c). Then the caller attaches the public key ($[c]G$) to Session Description Protocol (SDP) [39] information in the SIP INVITE message and sends the message to the SIP server. After receiving the SIP INVITE message, refer to [40, 41], the SIP server queries the RTP server to get two audio communication ports which is allowed through the firewall, modifies the SDP as follows:

Original SDP:
c = IN IP4 10.197.134.175
m = audio 49170 RTP/AVP 18
a = rtpmap:18 G729/8000/1
k = PK:ce15ad9708e3c406255afc01784e480681d22a6b225a8148465d4b6118e047a43f3777c44752bbb61ae5f264deab64c9916b9890d17179abbd606b92bf52480830de6ea686ad1e2592f32235426446d1246f7410c962179ae14b77d62cec81fb56570877b397f03045c6c432a22616b3d31d033f3cedf9ee9c72f157fe99580cc03d

Modified SDP:
c = IN IP4 140.118.122.145
m = audio 20000 RTP/AVP 18
a = rtpmap:18 G729/8000/1
k = PK:ce15ad9708e3c406255afc01784e480681d22a6-b225a8148465d4b6118e047a43f3777c44752bbb61ae5f264deab64c9916b9890d17179abbd606b92bf52480830de6ea686ad1e2592f32235426446d1246f7410c962179ae14b77d62cec81fb56570877b397f03045c6c432a22616b3d31d033f3cedf9ee9c72f157fe99580cc03d

Next, the SIP server forwards the SIP INVITE message to the callee (UAS). The callee also generates the ephemeral key pair that is ($[d]G$, d) and attaches the public key ($[d]G$) to Session Description Protocol (SDP) as the caller does. Furthermore, the callee computes the shared secret point Q and calculates $H(Q) = (k, k')$ where H is a hash function. The callee uses k' to generate a Message Authentication Code (MAC) and appends MAC to the SDP either.

$$M = MAC_{k'}(2,\ \text{Callee},\ \text{Caller},\ [d]G,\ [c]G)$$

Thus the callee transmits the SIP 200 OK message to the SIP server. The SIP server modifies the communication IP address and the port number in SDP and conveys the SIP 200 OK message to the caller.

After receiving the callee's public key, the caller also computes shared secret point Q and calculates $H(Q) = (k, k')$. Then the caller to k' to verify the MAC contained in SDP. If the value is not the same, terminate this session.

Otherwise, the caller generates $M' = MAC_{k'}(3$, Caller, Callee, $[c]G$, $[d]G)$ and attaches it to SDP of the SIP ACK message. After that, the caller sends out the SIP ACK message to the SIP server and the message is forwarded to the callee by the server.

Finally, the callee uses k' to verify the M' in SDP. If the verification fails, the call will not be set up. If not, the session is established and both clients use k to encrypt the subsequent RTP packets. The whole process is presented in Fig. 7. The TLS handshake phase refers to Fig. 2. Before the media session, the SIP messages are protected by the TLS-secured channel that SIP payloads are encrypted by the agreement key of the server and the client. After the media session is established, clients use the agreed session key to encrypt RTP payloads that even the SIP server or the RTP server cannot eavesdrop.

5 Performance analysis

In this section, we evaluate the consuming time of establishing a TLS-secured channel, elliptic curve point addition, multiplication, ECDH and ECMQV. The experimental devices are hTC Butterfly which equipped with quadcore Snapdragon APQ8064 CPU at up to 1.5 GHz per core and 2G RAM and runs on 4.2.2 Android platform. The servers have a 1.9 GHz Intel i3-3227U CPU and runs Windows 8.1 operating system.

First, we use openssl and Java keytool to generate essential certificates and deploy them to the servers and the clients. Figure 6 illustrates a sample certificate which contains 4096 RSA public key and be used to proceed the TLS handshake. We evaluate TLS handshakes with different RSA key lengths in a pure WiFi environment. The result is shown in Table 4. The cipher suite is set to TLS_RSA_WITH_AES_256_CBC_SHA.

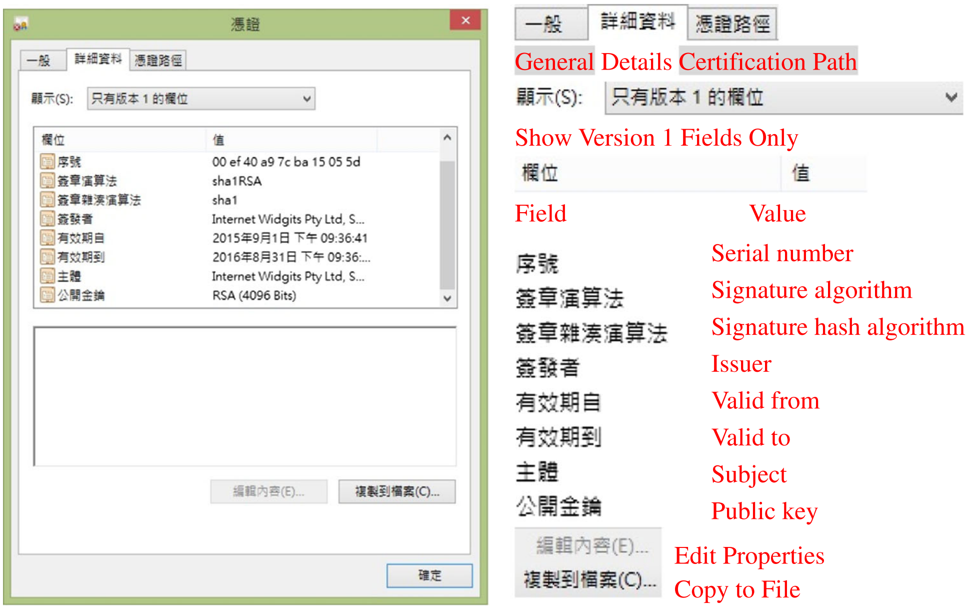
Table 4 TLS 1.2 handshake

Key length (bits)	Time cost (s)
1024	13.124
1536	13.187
2048	13.266
3072	13.359
4096	13.547

From Table 4, we see the time cost of establishing a TLS for mobile devices is high. However, the process of establishing a TLS can be carried out at startup of the application. Thus, the clients will not be conscious of it when making a VoIP call. Besides, Table 4 shows that the difference of time at different key lengths is small.

Next we present the consuming time of ECDH and ECMQV key agreement in Table 5. The elliptic curve refers to the recommendation of NIST.

From Table 5, it's clear that the time cost of ECC is extremely low than that of f Integer Factorization Cryptography (IFC). In NIST's recommendation, there are the other two curves sect409r1 and sect409k1 respectively; however, compared to others, the performance of these two curves are relatively bad. The results may be influenced by



憑證

顯示(S): 只有版本 1 的欄位

欄位	值
序號	00 ef 40 a9 7c ba 15 05 5d
簽章演算法	sha1RSA
簽章雜湊演算法	sha1
簽發者	Internet Widgits Pty Ltd, S...
有效期自	2015年9月1日 下午 09:36:41
有效期到	2016年8月31日 下午 09:36:...
主體	Internet Widgits Pty Ltd, S...
公開金鑰	RSA (4096 Bits)

Fig. 6 The client certificate contains 4096 public key

Table 5 ECC time cost (s)

Elliptic curve	EC_add	EC_mul	ECDH	ECMQV
Secp256r1	0.0001	0.0213	0.152	0.176
Secp256k1	0.0001	0.022	0.111	0.182
Sect283k1	0.0006	0.092	0.539	0.584
Sect283r1	0.0006	0.158	0.483	0.694
Secp384r1	0.0004	0.1575	0.554	0.705
Sect409r1	0.001	0.3025	0.936	1.285
Sect409k1	0.001	0.1775	0.884	1.145
Secp521r1	0.0005	0.24	0.806	0.9875
Sect571r1	0.0015	0.57	1.714	2.265
Sect571k1	0.0015	0.32	1.62	2.12

the parameters of the domain. We do not discuss it in this paper. Therefore, we remove these two curves from Table 4. We see that the time cost of ECDH and ECMQV is not the multiples of the addition and multiplication. That is because the consuming time of the function call should be taken into consideration. This is a critical point to implement a cryptographic system.

Finally, we show the voice quality of our system compared with popular VoIP applications such as Skype and Line (Fig. 7). We utilize Spirent Communications to measure the performance and present the results in Table 6.

The Spirent software records the output audio and uses it with reference to its original audio file to compute PSEQ (MOS-LQO) score. Perceptual Evaluation of Speech Quality (PSEQ) [42] is a worldwide applied industry standard for objective voice quality testing, Mean Opinion Score-Listening Quality Objective (MOS-LQO) scale is in the range 1–5. Our application adopts G.729 audio compression standard with the proposed security mechanism. Skype uses self-made SILK as its Codec and RSA for key negotiation and the Advanced Encryption Standard to encrypt conversations. Line does not release the information of its codec and the audio packets are not encrypted. Our experimental environment is full of WiFi access points

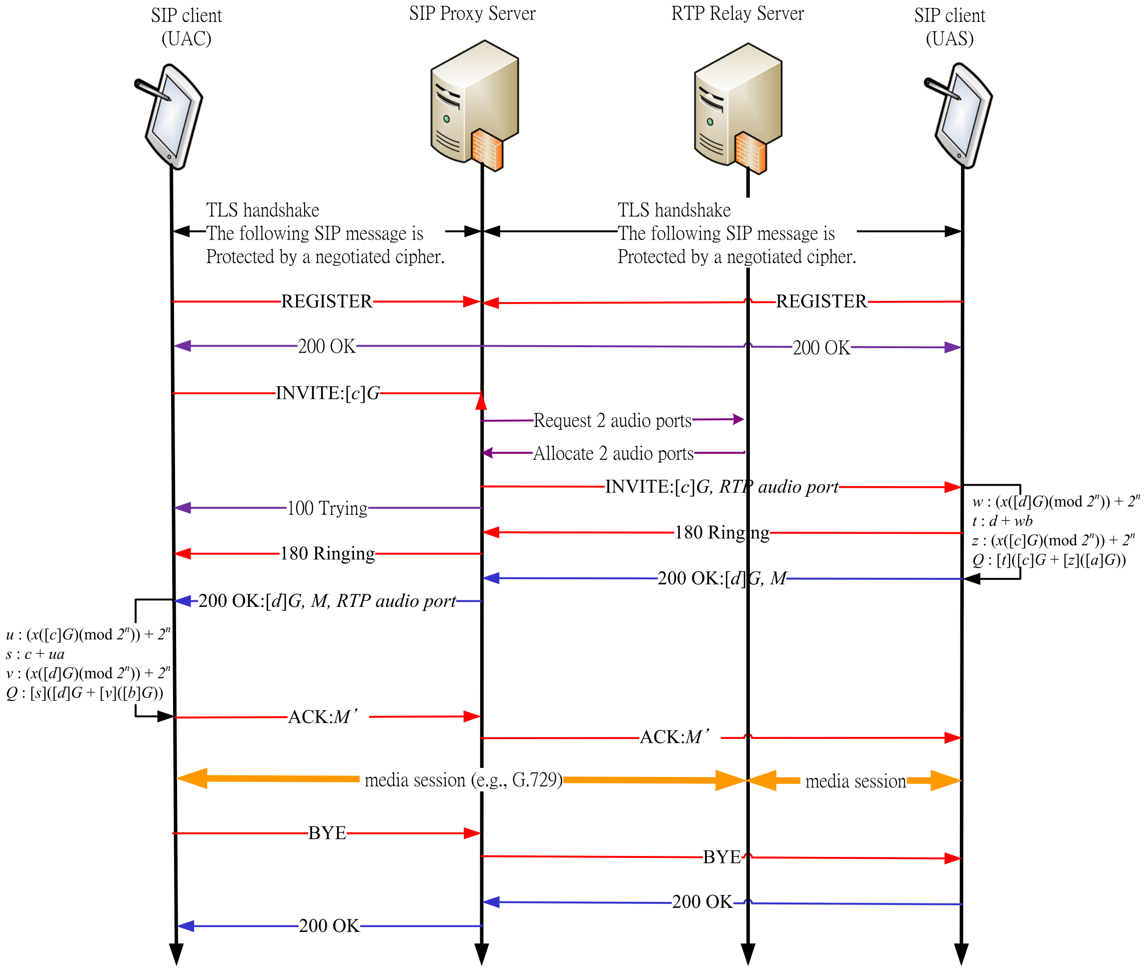

SIP client
(UAC)
SIP Proxy Server
RTP Relay Server
SIP client
(UAS)
TLS handshake
The following SIP message is
Protected by a negotiated cipher.
TLS handshake
The following SIP message is
Protected by a negotiated cipher.
REGISTER
REGISTER
200 OK
200 OK
INVITE:[c]G
Request 2 audio ports
Allocate 2 audio ports
INVITE:[c]G, RTP audio port
w : (x([d]G)(mod 2^n)) + 2^n
t : d + wb
z : (x([c]G)(mod 2^n)) + 2^n
Q : [t]([c]G + [z]([a]G))
100 Trying
180 Ringing
180 Ringing
200 OK:[d]G, M
200 OK:[d]G, M, RTP audio port
u : (x([c]G)(mod 2^n)) + 2^n
s : c + ua
v : (x([d]G)(mod 2^n)) + 2^n
Q : [s]([d]G + [v]([b]G))
ACK:M'
ACK:M'
media session (e.g., G.729)
media session
BYE
BYE
200 OK
200 OK

Fig. 7 The process of SIP integrating ECMQV

Table 6 Voice quality comparison

Mode	Packet loss	Latency	App	PESQ (MOS-LQO)		
				Minimum	Average	Maximum
3G	Low	Low	Ours	2.445	2.777	3.110
			Skype	1.494	1.494	1.494
			Line	1.656	1.656	1.656
WiFi	High	High	Ours	1.942	1.942	1.942
			Skype	2.092	2.092	2.092
			Line	1.511	1.511	1.511

that generate many interference signals. The packet loss rate and latency is high. On the contrary, 3G signal strength is much better. Thus, we found the voice quality with 3G signal has superior performance than the voice quality with WiFi signal. The outcome shows it is better to utilize 3G rather than WiFi in a chaotic WiFi environment (In our laboratory, we do not have instruments to measure the signal strength). From Table 6, the result shows that our application has best performance in 3G mode. In WiFi mode, the voice quality of our application is a little bit worse than Skype. However, the security of our system is much stronger than that of the other two. Moreover, the VoIP system is controlled by ourselves rather than it is under others' control.

Compared to other application, MicroSIP which is an open source portable SIP softphone based on PJSIP stack can only be installed in Windows OS. Xlite is a desktop application that runs on Windows or MacOSX. Zoiper is a free SIP client that supports both SIP calls over 3G or WiFi connections. However, most applications adopt the Diffie–Hellman cryptographic key exchange that lacks authentication and suffers the Man in the Middle (MitM) attack, especially the servers are not under control. The attacker can impersonate a server to eavesdrop communication channels. The cryptographic algorithms they utilized are opaque. In the proposed system, certificates are signed and issued by ourselves. We adopt the relatively secure and efficient elliptic curve and algorithms to implement cryptography. On top of that, from cryptography security to information security, the whole system is under our management.

The Fig. 8 illustrates our experiment environment which floods with many WiFi APs. However, the lab located under the coverage of signal of one base station. From the In 3G mode, the signal strength of the mobile phone is about $-85 \sim -90$ dBm. (When the signal strength exceeds -70 dBm, the quality of mobile network is excellent. When the signal strength is between -70 and -102 dBm, mobile network service is good. When the signal strength is lower than -102 dBm, the performance of mobile network is bad.) Thus we obtain a good QoS. However, in WiFi mode, many WiFi APs interfere with each other in the experiment environment. It influences the signal strength, lower than -102 dBm, and makes the connection unstable. For this reason, it results in bad performance. It concludes, in a environment full of WiFi APs, users should choose 3G/4G mode rather than WiFi mode.

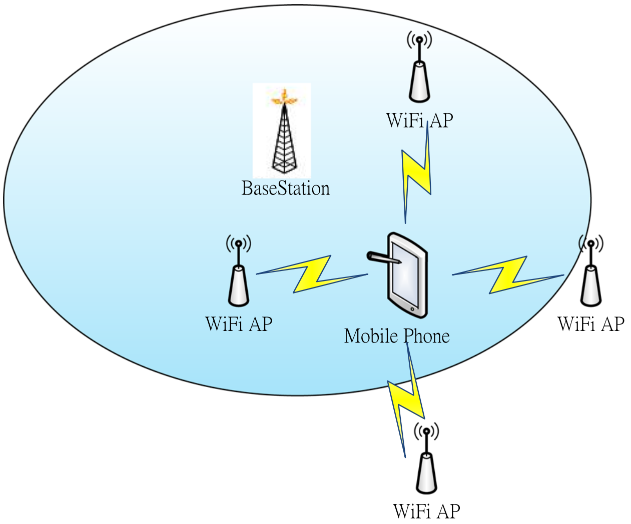


Fig. 8 The illustration of our experiment environment

6 Conclusion

As the events of surveillance and eavesdropping are disclosed, more and more governments and enterprises focus on the privacy of data transmitted on the internet. Nowadays, there are many free VoIP applications can be downloaded from internet; however, the security of these applications is not guaranteed. Many studies are proposed to provide the security of VoIP, but there is no concrete implementation. Besides, the voice quality after encryption should be taken into consideration. In this paper, we present a completely cryptographic VoIP system and show that the quality of voice is superior. In future research, we will study the QoS affected by various factors, especially packet loss, latency and jitter.

References

1. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., et al. (2002). SIP: Session initiation protocol. Internet Engineering Task Force, RFC 3261.
2. Schulzrinne, H., Casmer, S., Frederick, R., & Jacobson, V. (2003). RTP: A transport protocol for real-time applications. Internet Engineering Task Force, RFC 3550.
3. Geneiatakis, D., Lambrinoudakis, C., & Kambourakis, G. (2008). An ontology based-policy for deploying secure sip-based VoIP services. *Computer and Security, 27*(7–8), 285–297.
4. Geneiatakis, D., Dagiuklas, T., Kambourakis, G., Lambrinoudakis, C., Gritzalis, S., Ehlert, S., et al. (2006). Survey of security vulnerabilities in session initiation protocol. *IEEE Communications Surveys & Tutorials, 8*(3), 68–81.
5. Salsano, S., Veltri, L., & Papalilo, D. (2002). SIP security issues: The SIP authentication procedure and its processing load. *IEEE Network, 16*(6), 38–44.
6. Rafique, M. Z., Akbar, M. A., & Farooq, M. (2009). Evaluating DoS attacks against SIP-based VoIP systems. In *Proceedings of IEEe global telecommunications conference GLOBECOM'09*, Nov 30 2009–Dec 4 (pp. 1–6).
7. Geneiatakis, D., Kambourakis, G., Lambrinoudakis, C., Dagiouklas, A., & Gritzalis, S. (2007). A framework for protecting SIP-based infrastructure against malformed message attacks. *Computer Networks, 51,* 2580–2593.
8. Abdelnur, H., Avanesov, T., Rusinowitch, M., & State, R. (2008). Abusing SIP authentication. In *Proceedings of the international conference on information assurance and security* (pp. 237–242).
9. Xie, Q. (2012). A new authenticated key agreement for session initiation protocol. *International Journal of Communication Systems, 25,* 47–54.
10. Shen, C., Nahum, E., Schulzrinne, H., & Wright, C. P. (2012). The impact of TLS on SIP server performance: Measurement and modeling. *IEEE/ACM Transactions on Networking, 20*(4), 1217–1230.
11. Ashok, S., Arjun, A., & Subashri, T., Dynamic ECDH mechanism for enhancing privacy of voice calls on mobile phones over VoIP server. In *2014 international conference* on *advanced communication control and computing technologies* (ICACCCT) (pp. 1179–1184), 8–10 May, 2014.
12. Asterisk. Asterisk (2012). http://www.asterisk.org.
13. Cryptography and SSL/TLS Toolkit. OpenSSL (2015). https://www.openssl.org.
14. McGrew, D., Naslund, M., Norman, K., Blom, R., Carrara, E., & Oran, D. (2004). The secure real time transport protocol (SRTP), RFC 3711, March 2004.
15. Boulton, C., Rosenberg, J., Camarillo, G., & Audet, F. (2011). NAT traversal practices for client–server SIP, RFC 6314, July 2011.
16. National Institute of Standards and Technology. Recommended elliptic curves for Federal Government Use, July 1999.
17. Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation, 48*(177), 203–209.
18. Yang, C.-C., Wang, R.-C., & Liu, W.-T. (2005). Secure authentication scheme for session initiation protocol. *Computers & Security, 24,* 381–386.
19. Jo, H., Lee, Y., Kim, M., Kim, S., & Won, D. (2009). Off-line password-guessing attack to Yang's and Huang's authentication schemes for session initiation protocol. In *Fifth international joint conference on INC, IMS and IDC, Los Alamitos, CA, USA* (pp. 618–621).
20. Geneiatakis, D., & Lambrinoudakis, C. (2007). A lightweight protection mechanism against signaling attacks in a sip-based VoIP environment. *Telecommunication Systems, 36*(4), 153–159.
21. Tao, C., Qiang, G., & Baohong, H. (2008). A lightweight authentication scheme for session initiation protocol. In *Proceedings of IEEE international conference on communications, circuits and systems (ICCCAS)* (pp. 502–505).
22. Srinivasan, R., Vaidehi, V., Harish, K., LakshmiNarasimhan, K., LokeshwerBabu, S., & Srikanth, V. (2005). authentication of signaling in VoIP applications. In *11th Asia Pacific conference on communication (APCC), Perth, Australia*, October 2005.
23. Kong, L., Balasubramaniyan, V. A., & Ahamad, M. (2006). A lightweight scheme for securely and reliably locating sip users. In *IEEE/IFIP network operations and management symposium, Vancouver, Canada*, April 2006.
24. Handley, M., & Jacobson, V. (1998). SDP: Session Description Protocol. RFC 2327, April 1998.
25. Rivest, R. (1992). The MD5 message digest algorithm. RFC 1321.
26. Dierks, T., & Rescorla, E. (2008). The Transport Layer Security (TLS) Protocol version 1.2. RFC 5246.
27. Rescorla, E. (2000). *SSL and TLS: Designing and building secure systems*. Reading, MA: Addison Wesley.
28. Schneier, B. (1996). *Applied cryptography* (2nd ed.). New York: Wiley.
29. Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations. NIST, Tim Polk, Kerry McKay, Santosh Chokhani, 2014. http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r1.pdf.
30. Eastlake, D. E., & Jones, P. E. (2001). US Secure Hash Algorithm 1 (SHA1). RFC 3174.
31. Miller, V. (1986). Uses of elliptic curves in cryptography. In *Advances in cryptology*, *CRYPTO'85*. Lecture Notes in Computer Science (Vol. 218, pp. 417–426). Berlin: Springer.
32. NIST. Recommendation for key derivation throughextraction-then-expansion, 2011. Second Draft NIST Special Publication 800-56C.
33. Wenbin, H., & Jenqshiou, L. (2014). Anonymous authentication protocol based on elliptic curve Diffie–Hellman for wireless access networks. *Wireless Communications and Mobile Computing, 14*(10), 995–1006.
34. Pollard, J. M. (1978). Monte Carlo methods for index computation (mod p). *Mathematics of Computation, 32*(143), 918–924.
35. Pollard, J. M. (1993). Factoring with cubic integers. In *The development of the number field sieve*. Lecture notes in mathematics (Vol. 1554, pp. 4–10). Heidelberg: Springer.
36. Digital Signature Standard (DSS), FIPS PUB 186-3, 2009.
37. Standards for Efficient Cryptography Group (SECG),SEC 1: Elliptic curve cryptography, version 1.0, September 2000.
38. Blake, I., Seroussi, G., & Smart, N. (2005). *Advances in elliptic curve cryptography*. London Mathematical Society lecture note series (Vol. 317). Cambridge: Cambridge University Press.
39. Handley, M., & Jacobson, V. (1998). SDP: Session description protocol, RFC Editor.
40. Hwang, S.-H., & Yao, B.-C. (2014). SIP communication protocol. U.S. 8700785, April 15, 2014.
41. Hwang, S.-H., Chen, K.-L., Chang, S.-C., Huang, C.-J., Shen, L.-T., & Liu, B.-C. (2014). NAT traversal method in session initial protocol. U.S. 8676933, March 18, 2014.
42. ITU-T Recommendation P.862: Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, February 2001.

Wen-Bin Hsieh received his B.S. degree in Computer Science and Information Engineering from Tamkang University, Taipei, Taiwan, his M.S. degree and Ph.D. in Electronic Engineering from National Taiwan University of Science and Technology, Taipei, Taiwan. He is currently a postdoctoral researcher in the Electronic Engineering Department of National Taiwan of Science and Technology, Taipei, Taiwan. He worked in the information Department of Landbank from 2006 to 2009, as a programmer. His research interests include communication protocol, security control and cloud computing.

Jenq-Shiou Leu received his B.S. degree in Mathematics and his M.S. degree in Computer Science and Information Engineering from National Taiwan University, Taipei, Taiwan, in 1991 and 1993, respectively. He was with Rising Star Technology, Taiwan, as a R&D Engineer from 1995 to 1997, and worked in the telecommunication industry (Mobitai Communications and Taiwan Mobile) from 1997 to 2007, as an Assistant Manager. He obtained the Ph.D. degree on a part-time basis in Computer Science from National Tsing Hua University, HsingChu, Taiwan, in 2006. In Feb. 2007, he joined the Department of Electronic Engineering at National Taiwan University of Science and Technology, Taipei, Taiwan, as an Assistant Professor. He becomes an Associate Professor since Feb. 2011. His research interests include mobile services over heterogeneous networks, heterogeneous network integration and P2P networking. He has published 52 SCI paper including IEEE Transaction on Computer, IEEE Transaction on Services and so on.