LETTER Special Section on Deployment and Operation of New Internet Technology: Challenges and Approaches

Design and Implementation of One-Way Key Agreement Model for Enhancing VoIP Internet Phone Security

Kyusuk HAN[†a)], *Nonmember*, Taeshik SHON[††b)], *and* Kwangjo KIM[†], *Members*

SUMMARY The VoIP-based Internet Phonesystem is now seen as one of the killer applications in the high speed and broadband internet environment. Given the wide-spread use of the Internet Phone, it is necessary to provide security services for guaranteeing users' privacy. However, providing security service in Internet Phone has the possibility of incurring additional overheads such as call setup delay time. In this paper, we present a one-way key agreement model based on VoIP in order to reduce call setup time as well as protecting user privacy. The proposed approach decreases the delay time of the call setup in comparison with the previous models because our model enables the key generation in caller side without waiting the response from the receiver.

key words: *VoIP, ID-based cryptosystem, one-way key agreement, authentication, key agreement*

1. Introduction

These days Internet Phone is being widely used in daily life with high speed internet service and it is replacing legacy-type telephones with VoIP-based Internet Phone service. With the advent of new Internet technologies, Internet Phone service has become more prevalent in recent years. In order to support security feature in Internet Phone service, MIKEY [1] has become one of the recommended solutions for the standard VoIP implementation to provide the shared link key generation. In the standard, the key is called as Traffic Encryption Key (TEK), and is generated from shared TEK Generating Key (TGK). For generating TGK, several algorithms such as Diffie-Hellman key agreement or pre-sharing TGK are also recommended. Skype uses the proprietary key agreement model that each user generates 256-bit session key by exchanging their Identify Certificate and contributing 128 random bits in their security application [2]. Several studies such as [3], [4] have proposed lightweight solutions, but they still have the overhead of public key management.

Ring et al. [5] proposed an efficient model by applying identity based cryptography (IDBC) that uses the user's identity as the private key. Applying IDBC, their model does not require the public key management, and enables simplified process for the session key generation. However, the heavy cryptographic pairing computation delays call setup when generating TGK.

1.1 Our Contribution

Our contributions in this paper are as follows:

- We employed Okamoto et al.'s algorithm [6] that is the one-way key agreement model for reducing the call setup dely,
- also, combined with Hess's signature algorithm [7] for reducing overall SIP message sizes and the delay from initiating Secure Real-time Transport Protocol (SRTP).
- and, implemented the VoIP security service based on an open source Internet Phone called 'KPhone'.

Applying our protocol, the session key of caller side can be solely generated in the caller side. Thus, SRTP can be immediately initiated as soon as the response from the receiver arrived. Our novel design reduces delaying for the key generation and provides the explicit mutual authentication. In addition, the proposed approach reduces computational and communication overheads from public key management, signing number of messages by server and the SIP message sizes.

2. Reducing Call Setup Delay

In this section, we compare two approaches for key agreement of TGK between two entities and propose efficient VoIP key agreement design.

Let two entities Alice and Bob exchange a key for establishing secure call session. Using two-pass methods such as Ring et al.'s model [5] in Fig. 1(a), Alice and Bob mutually exchange key generating information T_A and T_B. Only after T_A and T_B are exchanged, Alice generates K_{AB} and Bob computes K_{BA}, where $K_{AB} = K_{BA}$. By contrast, using one-way method, Alice can generates K_{AB} when requesting key agreement as shown in Fig. 1(b). Only Alice's key generating information T_A is used for generating K_{AB} and K_{BA}.

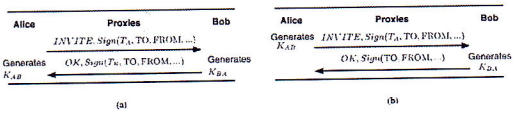
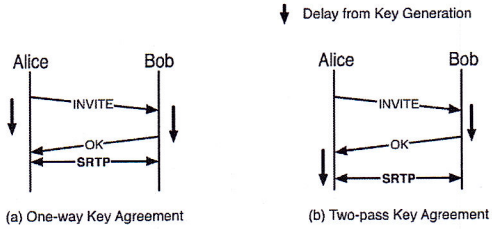


Fig. 1 Key agreement model for SIP: (a) Ring et al.'s model [5] (b) Our propose model.

Manuscript received November 30, 2010.
Manuscript revised March 14, 2011.

[†]The authors are with Korea Advanced Institute of Science and Technology, Daejeon, Korea.
[††]The author is with Division of Information and Computer Engineering, College of Information Technology, Ajou University, Suwon 443-749, Korea.
a) E-mail: hankyusuk, kkj@kaist.ac.kr
b) E-mail: tsshon@ajou.ac.kr (Corresponding author)

DOI: 10.1587/transcom.E94.B.2235



(a) One-way Key Agreement (b) Two-pass Key Agreement

Fig. 2 Comparison of (a) One-way key agreement model (Proposed) and (b) Two-pass key agreement model (Ring et al.).

In this model, the communication is required only once for key agreement. In order to reduce the delay from computing the session key used for SRTP encryption, we use the one-way key agreement model. In the figure, symbols 'TO' and 'FROM' are indications of the receive and the caller, respectively.

The comparison of our one-way key agreement and two-pass key agreement [5] employing in VoIP is shown in Fig. 2. Using one-way key agreement, Alice can precompute the session key when she sends the INVITE message to Bob. When Alice and Bob agree with the session key and send SRTP transaction, they can reduce the delay, which is shown in two-pass model. In two-pass model, Alice can compute the session key after Bob responds with OK message. Thus, our model enables the immediate SRTP initiation after OK message is received while two-pass key agreement models have the delay from key generation. Although the parameter for key generation is only chosen by caller, such disadvantage is not critical since the security strength is unharmed in practical applications.

2.1 Proposed Design

For the one-way key agreement protocol, we apply the scheme 1 in [6] that is based on ID-based cryptosystem. We assume a caller Alice, a receiver Bob, and a server in a certain VoIP service. To generate SIP message, Alice generates r, t, v, and u, where $r = e(P_1, P)^k$, $t = H^*(r) \cdot H(ID_{Alice})$, $v = h(m, t)$, and $u = v \cdot d_{Alice} + k \cdot P_1$. Here $h : \{0,1\}^* \times G_1 \rightarrow (Z/lZ)^\times$, $H : \{0,1\}^* \rightarrow G_1$, and others follow [6]. k is randomly selected by Alice. To generate t, r should be transformed from elliptic curve to finite fields. H^* is a map-to-point hash function, where $H^* : G_2 \rightarrow \{0,1\}$. To compute with $H(ID_{Alice})$, the transformation is necessary. $e : G_1 \times G_1 \rightarrow G_2$. G_1 is a cyclic additive group, generated by P with order q. G_2 is a cyclic multiplicative group with the same prime order q. d_{Alice} denotes Alice's private key, $d_{Alice} = sH(ID_{Alice})$. m is the SIP message that are fixed SIP headers including the sender's address, the receiver's address, message generated time and other necessary information. Session Description Protocol (SDP) information are not signed. Alice also generates the session key as follows.

$$k_{AB} = e(d_{Alice}, H(ID_{Bob}))H^*(r) \oplus e(d_{Alice}, H(ID_{Bob})) \quad (1)$$

Then, Alice sends u, v to Bob, where $(u, v) \in (G, (Z/lZ)\times)$. After receiving (u, v), Bob generates the following.

$$\begin{aligned} t &= H^*(r)H(ID_{Alice}) \\ &= H^*(e(u, P) \cdot e(H(ID_{Alice}), -sP)^v) \cdot (ID_{Alice}) \end{aligned} \quad (2)$$

After that, Bob verifies u and v with computing Equation (2) using m and t. After that Bob generates the session key as follows.

$$k_{BA} = e(t, d_{Bob}) \oplus e(H(ID_{Alice}), d_{Bob}) \quad (3)$$

The results of Eqs. (1) and (3) are same. Correctness of $k_{AB} = k_{BA}$ follows [6]. $\oplus$ is the additive operation in G_2. When a hash function $H' : G_2 \rightarrow \{0,1\}$ is used, $\oplus$ can be XOR operation in $k_{BA} = H'(e(t, d_{Bob})) \oplus H'(e(H(ID_{Alice}), d_{Bob}))$. K_{AB} is used as TGK, and TEK is generated using [8]. Thus, t is used for both SIP message signature and the key generation to reduce the additional communication only for the key generation.

3. Implementation and Analysis

3.1 Implementation

We implemented the one-way key agreement model based on the open source VoIP client, 'KPhone' (http://sourceforge.net/projects/kphone) for user terminal, 'SIP Express Router' of iptel.org (http://www.iptel.org/ser) for the SIP gateway that includes SIP registrar, SIP proxy, SIP redirection and SIP location server. We implemented our signing and key agreement protocol in 'S-INVITE' of call setup phase as in Fig. 3. Compare to previous designs, the caller does not have to wait the response '200 OK' for the key generation due to the one-way key agreement in our protocol.

The caller only sign the first indicator and fixed SIP header parts, since other parts such as SDP can be continually changed during the communication. The signature is attached after SDP parts as in Fig. 4.

After receiving 'S-INVITE', the receiver generates the public key from the caller's ID '*109@220.69.191.100*' as in Fig. 5. Since ID of an entity is used to generate the public key in ID-based cryptosystem, there is no need to verify the public key of each entity. The receiver also finds u, v, and recovers t. The receiver also collects the fixed parameters from the initial SIP message and recovered t and generates the hashed output of two parts. And then the receiver compares the hashed output with the received v for integrity check as in Fig. 6. Finally, the receiver generates the TGK first and the traffic encryption key (TEK) with the method of [8] from the value t as in Fig. 7. Since caller already generated 163 bits TGK and 163 bits TEK, the receiver can directly use TEK for the secure communication.

The result of key generation in both sides should be same as shown in Fig. 8. The caller's TGK, 'TGK_A' and the receiver's TGK, 'TGK_B' are same in the figure.

3.2 Security Analysis

We analyze our design that holds security requirements defined in [6] as follows.

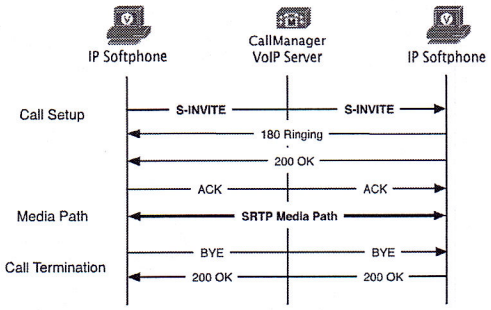


Fig. 3 The overall calling process. We implemented our protocol as 'S-INVITE' in call setup phase.

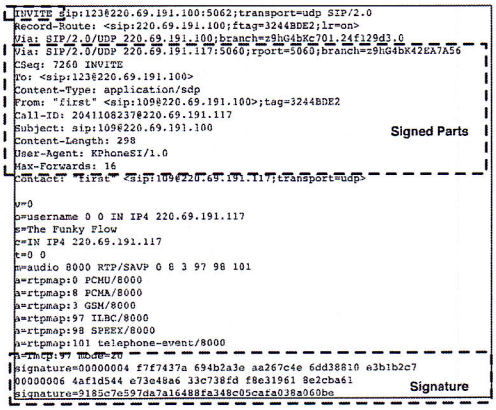


Fig. 4 Generate signature of SIP message. (First 'INVITE' and SIP header parts)

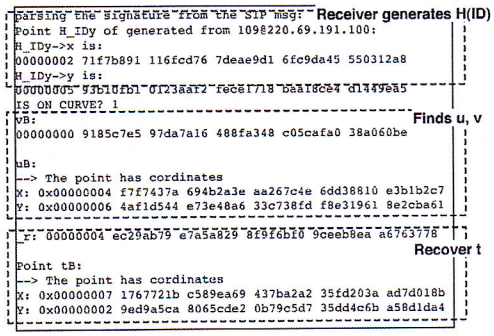


Fig. 5 After receiving S-INVITE, receiver generates H(ID) from caller ID, finds u, v and recovers t.

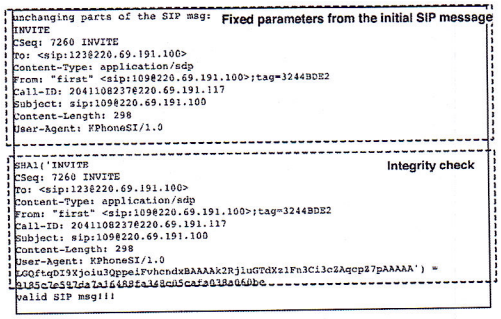


Fig. 6 Integrity check of received message.

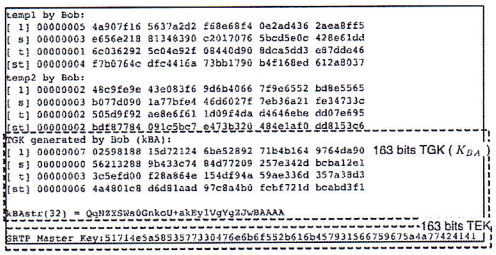


Fig. 7 Key agreement with recovered t.

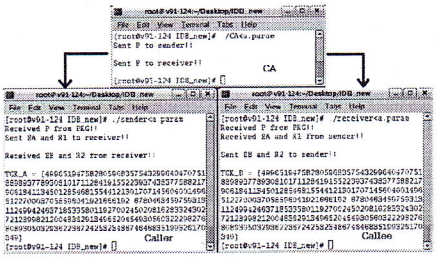


Fig. 8 The result shows that the both sides have the same result.

3.2.1 Known-Key Security

The caller Alice randomly choose P_1 and k in each session in order to generate r, where $r = e(P_1, P)^k$. The leakage of P_1 or k doesn't affect the previous session. Although an adversary $\mathcal{A}$ obtains P_1 and k, $\mathcal{A}$ cannot know the TGK used in the previous session.

3.2.2 Unknown Key-Share

In order to generate the session key, Bob firstly generates t. t is used to verify the signature of Alice. Also, Alice self-generates the session key without any information from Bob. Therefore, any other entities except Alice and Bob cannot exchange the key. To succeed the attack, the adversary should be able to generate the signature of Alice or know the private key of Bob.

3.2.3 Key Control

Since Alice selects the key generating parameter, and the process is done in one-way, Bob cannot control the session key, also it is difficult for Alice to pre-compute the random integer r and the generator P_1 to control t.

3.2.4 Attacks to Sender

When Alice's private key is leaked, the adversary can impersonate Alice, since r is known to Alice, while it is not possible to impersonate other entity. Sender's forward security is guaranteed, since k and P_1 are randomly selected in each session by Alice.

3.2.5 Random Number Compromise

The random integer r is easily known from (u, v). However it is difficult know Alice and Bob's private keys or session key from public parameters P, sP, and r. To attack the session key, the knowledge of Alice or Bob's private key is necessary. The success of attack with P, sP and r is the same as the success of attack on the signature.

3.3 Performance Analysis

Our design requires one exponentiation operation in G_2, two hash operations, two multiplications in G_1 for the signature generation, and one exponentiation operation in G_2, two pairing operations, and one multiplication operation for the verification. When several messages are sent by the same identity, the sender can reduce one pairing operation by pre-computing $e(H(ID), -sP)$. For the key agreement, one pairing operation of the sender, one multiplication over elliptic curve, one exponentiation operation, and two pairing operations of the receiver are required. Computing the hash operation takes less than 600 microseconds, and the generation of signature and key agreement takes approximately 26 milliseconds on the Intel Core2 Duo 2.0 GHz. Recent studies on optimal pairing implementations [9], [10] show that the pairing computation takes approximately 3 seconds in smart card, and 14.5 milliseconds in Core2 Duo 1.66 GHz. The open source cryptographic pairing library [11] showed similar results. Recent benchmark [12] shows that the performance of current smart phones have about a half of Core2 Duo processor. While Ring et al.'s protocol requires about 10–26 milliseconds for key generation after receiving 'OK' message in caller side, our proposed protocol immediately initiate SRTP encryption without such delay. In this sense, our proposed design enables practical adaptation of IDBC in smart phones than previous models where ITU-T G.114 recommendation specifies that for good voice quality, no more than 150 milliseconds of one-way, end-to-end delay should occur [13].

4. Conclusion

In this paper, we proposed an efficient authentication and key agreement model for VoIP security service in order to show the possibility of implementing an open source-based secure and efficient VoIP client. The proposed model enables the practical application of IDBC in VoIP by deploying one-way key agreement model because the cryptographic pairing computation in ID based cryptography (IDBC) is still heavy. By Pre-computing the secret key in the caller side, our proposed model significantly reduces the call setup delay. Moreover, our analysis of performance and security showed that the proposed approach decreases not only the cost for the public key management but also additional process for the key generation with using the parameter for the signature verification and reducing the delay for the initiating SRTP media path.

Acknowledgement

Related results of our theoretical part in this paper were presented as a part of [14]. The practical approach and the evaluation including security and performance analysis are fully revised.

References

[1] P. Thermos and A. Takanen, Securing VoIP Networks: Threats, Vulnerabilities, and Countermeasures, Addison-Wesley Professional, 2007.
[2] L.D. Cicco and S. Mascolo, "A mathematical model of the Skype VoIP congestion control algorithm," IEEE Trans. Autom. Control, vol.55, no.3, pp.790–795, 2010.
[3] J. Peterson and C. Jennings, "Enhancements for authenticated identity management in the session initiation protocol (SIP)," RFC4474, Aug. 2006.
[4] L. Kong, V.A. Balasubramaniyan, and M. Ahamad, "A lightweight scheme for securely and reliably locating SIP users," 1st IEEE Workshop on VoIP Management and Security (VoIP MaSe 2006), 2006.
[5] J. Ring, K.R. Choo, E. Foo, and M. Looi, "A new authentication mechanism and key agreement protocol for SIP using identity-based cryptography," Proc. AusCERT Asia Pacific Information Technology Security Conference, 2006.
[6] T. Okamoto, R. Tso, and E. Okamoto, "One-way and two-party authenticated ID-based key agreement protocols using pairing," Modeling Decisions for Artificial Intel ligence, Second International Conference, MDAI 2005, vol.3558, no.2005, pp.122–133, Tsukuba, Japan, 2005.
[7] F. Hess, "Efficient identity based signature schemes based on pairings," 9th Annual International Workshop, SAC 2002, vol.2595, no.2003, pp.310–324, 2002.
[8] M. Baugher, R. Canetti, L. Dondeti, and F. Lindholm, "Multicast security (MSEC) group key management architecture," IETF RFC 4046, 2005.
[9] A.J. Devegili, M. Scott, and R. Dahab, "Implementing cryptographic pairings over Barreto-Naehrig Curves," Pairing-Based Cryptography — Pairing 2007, LNCS., vol.4575, no.2009, pp.197–207, 2007.
[10] D. Hankerson, A. Menezes, and M. Scott, "Software implementation of pairings," in Identity-Based Cryptography, ed. M. Joye and G. Neven, May 2008.
[11] B. Lynn, On the implementation of pairing-based cryptosystems, Ph.D. Thesis, Stanford University, 2007.
[12] Nvidia, "Coremark performance on Kal-El (Kal-El vs. Tegra 2 vs Core2Duo T7200)," (Youtube), Feb. 2011.
[13] J. Davidson, J. Peters, M. Bhatia, S. Kalidindi, and S. Mukherjee, VoIP: An In-Depth Analysis, in Voice over IP Fundamentals, 2nd ed., Cisco Press, 2006.
[14] C. Yeun, K. Han, and K. Kim, "New novel approaches for securing VoIP applications," Proc. Sixth International Workshop for Applied PKI Conference (IWAP'07), Perth, Australia, Dec. 2007.