Off-the-Record Communication with Location Hiding

Halil Kemal Taşkın[1], Murat Demircioğlu

***Abstract*—Cryptography is the main tool on the Internet for protecting private data. Systems such as PGP, SSL make use of long-lived keys for cryptographic purposes. However, a new protocol called Off-the-Record Messaging proposed by Borisov et al. [1] in 2004 changed the way of private communication. OTR is a system which provides end-to-end security with properties such as perfect forward secrecy and repudiability. In this paper, we give a two-party OTR messaging protocol with location hiding property. Besides basic requirements of OTR communication, location hiding enables parties to hide their physical locations namely IP addresses. Also, our design removes the requirement for trusted third parties. As a result, this allows it to be easily deployed on cloud networks.**

***Index Terms*—Private Social Communication, Location Hiding, Repudiability, Perfect Forward Secrecy, Authentication.**

I. Introduction

The invention of the internet has changed the way we communicate. E-mail is one of the newest communication tools coming into our lives. Less then two decades ago, many people had not heard about it, but today many of us use e-mail instead of writing letters. Every day billions of e-mail messages are sent around the world.

But sometimes e-mail is not practical enough. You may want to know if a person you want to send e-mail is online at that moment. Also e-mailing with someone intensively in a short time requires so many clicks. As a result of these kind of requests and difficulties, the instant messaging (IM) became so popular. By using IM, you can keep a list of people you want to communicate with. You can chat with that person as long as they are online.

On the other hand, IM does not provide the security properties available in a physical private conversation, such as repudiation. Online communication systems commonly provide confidentiality, authentication and non-repudiation. In 2004's Workshop on Privacy in Electronic Society (WPES), Borisov et. al. introduced the Off-the-Record (OTR) protocol [1], which allows two-party private conversation, aims to provide repudiation besides authentication, confidentiality and perfect forward secrecy. After OTR protocol was introduced it took attention of many researchers [3], [6], [7], [8]. Multi-party OTR protocol [2], [4] is an extension of OTR but we focus on only two-party communication.

H.K. Taşkın and M. Demircioğlu are with the Department of Cryptography, Institue of Applied Mathematics, Middle East Technical University.

E-Mails: {halil.taskin,demurat}@metu.edu.tr

[1]The author is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under the project number 112E101 titled "Blok Şifrelerin Olası Olmayan Kriptanalizi".

In this work, we present a new two-party OTR protocol which also provides location hiding. Using our protocol, communicating parties may not know their locations, namely their IP addresses. In our design, we used SHA-3 winner KECCAK [9] as a cryptographic building block for symmetric operations. By this way, having only implementation of KECCAK, it is possible to handle all symmetric operations.

In section II, we state the requirements of OTR protocol. Our new OTR protocol is given in section III. Finally, section IV concludes our work.

II. Cryptographic Requirements

The International Organization for Standardization (ISO) defines five typical security services to support the security in distributed applications. Those are identification/authentication, access control/authorization, confidentiality, integrity, and non-repudiation [12]. The parties using OTR protocol may need these services except non-repudiation. Instead of it, deniability (repudiation) is preferred. By this way, they will be able to deny their past conversations.

A. The OTR Protocol

The OTR protocol consists of four cryptographic primitives [1]; perfect forward secrecy, digital signatures, message authentication codes and malleable encryption.

1) Perfect Forward Secrecy: The past messages should not be recovered retroactively. In order to achieve this property, short-lived encryption/decryption key(s) are used for confidentiality. Used keys should be forgotten by the parties after they are used. It should be computationally infeasible to determine the used keys from the current key or the long-term keys.

2) Digital Signatures: It is used to make parties verify each other. Diffie-Hellman key exchange protocol does not provide the mutual authentication. Digital signatures produced via long term keys may solve this problem. However, attaching signatures to every IM messages leads to non-repudiation which we do not want. The solution is to use Message Authentication Code (MAC) on each message instead of user's digital signature. Authenticating the keys by using digital signature provides the identification service. Therefore, only the person who knows the right key can read the ciphertexts.

3) Message Authentication Codes: The parties shared a MAC key, and each message is sent with the MAC of the message. Deniability is achieved by this way. Since third parties do not know the MAC key, they cannot prove the source of the message. And also the recipient of the message cannot prove the source since it could be a message forged by himself.

4) Malleable Encryption: It will be used to provide forgeability of transcripts, repudiation of contents, and deniability. Forgeability is a little bit stronger than repudiation. Forgeability is achieved through malleable encryption which makes it easy to alter the ciphertext while it is still a proper ciphertext without knowledge of the key. Using stream ciphers can be an example for this.

B. Location (Source) Hiding

The communicating parties may want to hide their physical locations, namely IP addresses. For this purpose, anonymizing models should be used. Also, anonymizing without a trusted third party is an important issue to consider. Therefore, there will not be any direct connection between the messaging parties. So, they cannot obtain their locations.

III. The Protocol

In this chapter, we give the details of our protocol. Our protocol consists of two main parts: Two-party communication protocol and location hiding protocol.

In the constructions, we use Keccak [9] as a building block for hash function, stream cipher and generating message authentication code. Also, 2048-bit RSA is used as public key cryptosystem and 1024-bit Diffie-Hellman (DH) Key Exchange Scheme with generator g is used for public key exchange.

Our notation,

- A and B are the public pseudonyms for the parties.
- $||$ is concatenation.
- e_X and d_X are the public and private keys of part X, respectively.
- $[m]_{d_X}$ is the signature of message m signed by part X.
- $\{m\}_{e_X}$ is the encryption of message m with part X's public key.
- H is the Keccak-512 hash function.
- $ENC_{IV,K}(m)$ and $DEC_{IV,K}(m)$ are Keccak-256 in stream cipher mode encryption and decryption algorithms with message m, key K and initialization vector IV, respectively.
- $MAC_K(m)$ is the message authentication code for message m generated by Keccak-256 in MAC mode with key K.

A. Two-party Communication Protocol

Let A and B be the parties who want to communicate. Two-party off-the-record communication is established as follows:

1) Let n be the re-keying value.
2) A generates ephemeral public/private key pair (e_A, d_A).
   B generates ephemeral public/private key pair (e_B, d_B).
3) A sends its public key to B and B to A.
4) A and B verifies their public keys by an offline method such as voice communication, facebook, whatsapp etc. using the fingerprint of their corresponding public keys.
5) A initiates the counter: $ctr_A \leftarrow 1$
   B initiates the counter: $ctr_B \leftarrow 1$
6) Key exchange process is based on DH key exchange protocol.

$$A \xrightarrow{A, g^{x_1}, [H(g^{x_1})]_{d_A}} B$$

$$\xleftarrow{B, g^{y_1}, [H(g^{y_1})]_{d_B}}$$

   Both sides compute the shared secret $g^{x_1 y_1}$. It is hashed using the hash function H.

$$H(g^{x_1 y_1}) = K$$

   Let $K = (K_L, K_R)$ where K_L is the most significant half and K_R is the least significant half of the K.
   Hash of concatenation of least significant half of K_L and most significant half of K_R is used as session key ID (SKID).
7) If ctr_A mod $n = 0$ or ctr_B mod $n = 0$ then key exchange process is repeated as it is started from step 5. Then, after minimum $n-1$, maximum $2n-2$ messages later, there has to be a key exchange.
8) Part A sends message M_1 to B as follows:
   Let M_1^E be the encapsulated form of M_1. Than, M_1^E is computed as follows:

$$M_1^E = \big(SKID, ctr_A, IV, ENC_{IV,K_L}(M_1), MAC_{K_R}(SKID||ctr_A||IV||M_1)\big)$$

$$ctr_A \leftarrow ctr_A + 1$$

$$A \xrightarrow{M_1^E} B$$

   A random IV is generated for all new messages. Since both parties can check other party's counter value, none of them is able to delay the key exchange phase.
9) Part B decrypts and verifies M_1 as follows:
   It first computes M_1':

$$M_1' = DEC_{IV,K_L}(ENC_{IV,K_L}(M_1))$$

   Then, message authentication code MAC' is computed for M'.

$$MAC' = MAC_{K_R}(SKID||ctr_A||IV||M_1')$$

   If the value of MAC' is same as the one in the M_1^E then message is correctly decrypted and verified.
10) B does the same procedure to send a message to part A.

There is a trade off between perfect forward secrecy and efficiency of the protocol. To get over this issue the re-keying value n is used. The n value can be configured due to the requirements of the communication. For example, taking $n = 2$ forces the system to change keys at most every 2 packets, which is slower but more secure. But, for instant messaging taking higher n values should not be a problem.

B. Location Hiding Protocol

Our protocol which is explained in section III-A can be extended to have location hiding property. Our location hiding protocol makes use of some ideas from the anonymizing network protocols Tor [10] and I2P [11].

Location hiding protocol requires at least three different third party servers which is not required to have secure sockets. Plaintext communication is enough to handle the protocol. Every server should have two capabilities: Connection binding and location redirection.

Connection binding is used as a meeting point for A and B. Location redirection is used to redirect messages from a user to an other user.

Let N_i, N_j and N_k be the three servers.

1) Parties A and B agree on a meeting server N_i and a shared secret passphrase.
2) A selects its meeting server except N_i which is N_j. Similarly, B selects N_k.
3) A opens a connection to N_j. N_j stores the id of A and its corresponding connection session. Than, A is connected to N_i and does a connection binding to N_j with the shared secret passphare as follows:
$$A \xrightarrow{A, ENC_W(A||"Location:"||N_j)} N_i$$
4) Similarly, B does the same process:
$$B \xrightarrow{B, ENC_W(B||"Location:"||N_k)} N_i$$
5) For a party, only the last connection binding session data is stored on a server. So, when a request is sent for a user, only the last binding session data will be responded.
6) A requests B's binding point using the id of B. Encrypted location data is sent to A. Than A decrypts the location using shared secret passphrase to meet with it. If passphrase is validated B's meeting point N_k is recovered.
   If $N_k = N_i$ or $N_k = N_j$ then A rejects the communication. Process is repeated until agreed on different servers.
   Similarly, B gets A's meeting point N_j.
7) A sends message M_a to B using location redirection on server N_k.
$$A \xrightarrow{B, M_a} N_k \xrightarrow{M_a} B$$
8) B sends message M_b to A using location redirection on server N_j.
$$A \xleftarrow{M_b} N_j \xleftarrow{A, M_b} B$$

By design, only 1 hop is used between A and B, but to get higher levels of location hiding it it possible to have more than one hops between two parties. Having more hops decreases speed of communication.

C. Why it is OTR?

In this section, we show that our protocol satisfies the requirements for the OTR communication.

- **Perfect Forward Secrecy:** In our design we do not use any long-lived keys, instead of this we use ephemeral RSA keys for public encryption and signing. Also, rekeying value n is used to force parties to change session keys after at most $2n - 2$ messages.
- **Authentication:** Parties A and B authenticate each other in an offline fashion. This removes the requirement for a long-lived key. This authentication mechanism enables to sign key exchange values without compromising repudiability.
- **Repudiation (Deniability):** Only the hash of key exchange values (i.e. DH key exponents) are signed by parties, it is not computationally possible to show the session key used in encryption is derived from this particular key exchange process.
- **Forgeability (Malleability):** Malleability is stronger form of forgeability. Our design uses a stream cipher for encryption and decryption of messages. So, it is possible to modify ciphertext and it will still be a proper ciphertext due to XOR'ing property of stream ciphers. This property allows us to modify any encapsulated message properly. Anyone who has decryption key can properly decrypt the modified ciphertext. But only recipient will be able to detect this modification because of the MAC value of the encapsulated message.
- **Location Hiding:** We also show that our design enables location hiding when it is used with the protocol given in section III-B. Since authentication is done offline and any message is binded to its source, any server between two parties would not be able to mount a man-in-the-middle attack.

IV. Conclusion

We presented a new off-the-record communication protocol with location hiding property that removes the requirement for a trusted third party. Our main assumption for the security is that it is not feasible for someone to observe both OTR and offline communication channels. Our design is resistant against impersonation attacks for all parties and servers due to authentication is done in an offline fashion. Man-in-the-middle attacks are also not feasible because of two main reasons; first one is that authentication is offline and second one is that every type of packet sent to other party has binding for its source. Using any third party server without considering the security of them allows our design to be easily established on a cloud system. In this way, it is possible to have many servers for location hiding based on plaintext communication.

One of the problems in the OTR is to tell users how to initiate the protocol since the protocol itself is not user-friendly [5], implementations should care about a user friendly interface. Also, location hiding in multi-party OTR is another research area. These are possible future works for our problem.

Acknowledgment

The authors would like to thank Cihangir Tezcan for his comments and guidance.

REFERENCES

[1] N. Borisov, I. Goldberg, E. A. Brewer, *Off-the-record communication, or, why not to use PGP*. WPES 2004: 77-84.

[2] I. Goldberg, B. Ustaoğlu, M. V. Gundy, H. Chen, *Multi-party off-the-record messaging*. ACM Conference on Computer and Communications Security 2009: 358-368.

[3] M. Di Raimondo, R. Gennaro, H. Krawczyk, *Secure off-the-record messaging*. WPES 2005: 81-89.

[4] J. Bian, R. Şeker, Ü. Topaloğlu, *Off-the-Record Instant Messaging for Group Conversation*. IRI 2007: 79-84.

[5] R. Stedman, K. Yoshida, I. Goldberg, *A user study of off-the-record messaging*. SOUPS 2008: 95-104.

[6] I. Goldberg, D. Stebila, B. Ustaoğlu, *Anonymity and one-way authentication in key exchange protocols*. Des. Codes Cryptography 67(2): 245-269 (2013).

[7] M. Di Raimondo, R. Gennaro, H. Krawczyk, *Deniable authentication and key exchange*. ACM Conference on Computer and Communications Security 2006: 400-409.

[8] C. Alexander, I. Goldberg, *Improved user authentication in off-the-record messaging*. WPES 2007: 41-47.

[9] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche. *Keccak sponge function family main document*. Submission to NIST, 2011. Version 3.0.

[10] R. Dingledine, N. Mathewson, P. Syverson. *Tor: The second-generation onion router*. In Proceedings of the 13th USENIX Security Symposium, pages 303-320, August 2004.

[11] Invisible Internet Project (I2P), *http://www.i2p2.de*

[12] I. O. for Standardization. *Information processing systems - Open Systems Interconnection - Basic Reference Model Part 2: Security Architecture*. Number ISO 7498-2. 1988.