Privacy Authentication using Deniable Key Exchange

Manali Mehta
B.E. Student
Department of Computer Engineering
Sinhgad Academy of Engineering, Pune, Maharashtra, India

Pratik Jambhale
B.E. Student
Department of Computer Engineering
Sinhgad Academy of Engineering, Pune, Maharashtra, India

S.S Pawar
Department of Computer Engineering
Sinhgad Academy of Engineering, Pune, Maharashtra, India

Abstract

Now-a-days for the network security, the main cryptographic tool used is the key exchange mechanism. Particularly the Diffie-Hellman key exchange (DHKE) is used. It is very important to achieve both the security and the privacy factor while exchanging the key through internet. In this paper we are making a collection that is a family of all the privacy authenticated DHKE. This group of authenticated DHKE is named deniable Internet key-exchange (DIKE). This is applicable both in the identity based setting and the customary PKI setting. This recently developed DIKE as per our facts and knowledge; it provides online efficiency, privacy and security to both the participant protocols. It has all the advantages of the forward deniability, session data and session key can be generated from DH-components and the exchangeable messages do not bear peer's identity.

Keywords: DHKE, Security Association (SA), DIKE, SIGMA

I. INTRODUCTION

The Internet Key Exchange is the protocol used to set up a security association (SA) in the IPsec protocol suite. The Internet Key-Exchange Protocols are the main cryptographic protocols that are used to achieve the Internet security, which specify the key exchange mechanisms that are used to establish shared keys for use in the Internet Protocol Security standards. A key-exchange (KE) protocol is run in a network of connected participants where each participant can be initiated to run an instance of the protocol called a session. Within a session a party can be start the session or reply to a received message. As a result of these activations, and according to the specification of the protocol, the participants creates and retains a session state, generates outgoing messages, and eventually completes the session by giving the session-key as output and eliminating the session state. We can also abort a session without creating the session key. The key session will be erased automatically, if the session gets expired. A KE session is associated with its holder or owner, a peer and a unique session identifier. Basically the IKE and IPsec are used to secure the information or data communicated in the IP layer 3 of the ISO-OSI model. IKE and IPsec can also be used to provide further protection, authentication and privacy for communication protocols in the higher layers of ISO-OSI model. Most IPsec applications consist of an IKE program that runs in user space and an IPsec stack in the kernel that processes the actual IP packets.

Programs running in the user-space have easy access to mass storage containing configuration information, such as the IPsec keys, endpoint addresses and certificates, as required. Kernel modules, on the other hand, can process packets efficiently, which is important for performance reasons. The IKE basically has two versions that are IKEv1 and IKEv2. The first generation that is IKEv1 provides authentication using public-key encryption. Whereas the second generation IKEv2 provides authentication using signatures that is basically the SIGMA protocol. In key-exchange protocol it is very important to meet certain goals related to privacy and security. This can be achieved using SIGMA protocol. When it comes to privacy concerns, deniability is an essential privacy property. DHKE provides this facility with nonmalleable zero knowledge.

A. IPsec:

Internet Protocol Security (IPsec) is a protocol suite for secure Internet Protocol (IP) communications by encrypting all the IP packet of a communication session. IPsec includes protocols for starting mutual authentication between agents at the beginning of the session and cooperation of cryptographic keys to be used during the session. IPsec can be used in protecting data flows between a pair of hosts, security gateways, or between a security gateway and a host. Internet Protocol security (IPsec) uses cryptographic security services to secure communications over Internet Protocol (IP) networks. IPsec is an end-to-end security protocol operating in the Internet Layer, while some other Internet security systems is extensively used, such as Transport Layer Security (TLS) and Secure Shell (SSH), operate in the higher layers at the Application layer. Hence, only IPsec keeps all application traffic over an IP network. Applications can be automatically protected by IPsec at the IP layer.

B. SIGMA:

SIGMA serves as the cryptographic basis for the Internet Key Exchange (IKE) protocol standardized to provide key-exchange functionality to the IPsec suite of security protocols. Basically, SIGMA is the signature-based authenticated key exchange in IKE. It is the most used mode of public-key authentication in IKE, and in IKEv2 it is the basis for the only mode of public-key authentication. The basic guiding requirements behind the design of SIGMA are to provide (a) a protected key exchange protocol based on the Diffie-Hellman exchange. (b) The use of digital signatures for public-key authentication of the protocol, and (c) to provide the option to protect the identities of the protocol peers from being identified by an attacker in the network. The design of SIGMA is strongly based on that of STS: both the strengths of the STS design principles (very well-articulated in as well as the weaknesses of some of the STS protocol choices have strongly influenced the SIGMA design.

II. DENIABLE KEY-EXCHANGE

Deniable encryption describes encryption techniques where the existence of an encrypted file or a message is deniable in the sense that an adversary cannot prove that the plain text data exists. The users may convincingly deny that a given piece of data is encrypted, or that they are able to decrypt given information of encrypted data or that some specific encrypted data exists. Now-a-days most of the communication takes place via digital media, so it is very important to provide these communication with “off-the-record” or deniability protocol to participants.

Conventionally deniability used to differentiate between the honest against a possibly malicious verifier. That is basically in a session text, the involvement of the honest participant cannot be detected by the malicious verifier. But what we want is that the honest participant during the protocol cannot claim the messages to be authenticate at later stages for the privacy of the verifier. This concept is known as forward deniability. That is whenever deniability of data or messages is required; we can just run the deniability protocol for each message to be sent. The advantage of using deniable key exchange is that if the key-exchange is deniable is that all the communications using session keys producing key-exchange protocol can be deniable both the participants.

Here we extend the above definition of deniability to the setting of (authenticated) key-exchange (KE) protocols. As in the case of authentication, we present a simplified definition of a KE protocol as a stand-alone procedure, and then define the deniability property in a concurrent-execution setting. A more general treatment of the subject, including a full integration of deniability with a model of KE security in a multi-party setting is left for future work. By simplifying the formal treatment in our presentation here, and focusing on the concurrent setting, we are able to highlight the technical and conceptual issues raised in the investigation of deniability. Also, by studying the deniability features of specific KE protocols that were proven secured. On these works we can build and form the deniability aspects. In a key exchange protocol, two parties, say *A* and *B*, are associated with public keys pk_A and pk_B respectively, for which they each own the matching secret key sk_A and sk_B. We assume that public/secret keys are generated according to a key generation algorithm KG which is part of the specification of the KE protocol, and these are used in the authentication steps of the KE protocol. The protocol specifies the interaction between *A* and *B* (one acting as “initiator” and the other as “responder”) and whose result is either a (session) key *K* or “error” if some of the operations/verifications in the protocol fail. The *B*asic (and simplified) security requirement in a KE protocol is that if *A* outputs a session key *K* and associates it to peer *B* then the only party that may possibly know *K* is *B*; and if *B* outputs the same session key then it associates it to peer *A*. Note however that this security guarantee is provided only for sessions (i.e., runs of the KE protocol) in which both peers are uncorrupted.

A. Non-Zero Malleable Knowledge:

Zero-knowledge (ZK) interactive proofs are fundamental constructs that allow the Prover to show the Verifier the validity with mathematical statement x ∈ L, while providing zero additional knowledge to the Verifier. In an asynchronous and concurrent setting, Concurrent Zero Knowledge considers the execution of zero-knowledge protocols. In this model, an opponent acts as verifiers in many concurrent executions of the zero-knowledge protocol, and launches a synchronized attack on multiple independent provers to gain information. Non-malleable Zero Knowledge also considers the synchronized execution of zero-knowledge protocols, but in a different manner. The execution occurs in two ways; in the first execution, also known as the left execution, acts as a verifier. The second execution, also known as the right execution, acts as a prover. The notion of Concurrent Non-malleable Zero Knowledge (CNMZK) considers both of the above attacks; the opponent may participate in an abundant number of synchronized executions, playing the role of a prover in some, and the role of a verifier in others. Despite the generality of such an attacks scenario, this idea of security seems most appropriate for modeling the execution of cryptographic protocols on internet that is in an open source.

III. OUR CONTRIBUTION

Basically, to ensure the security requirements for a deniable authentication protocol are satisfied or not, we check over enhanced ID-based deniable authentication protocol in order. Security and privacy, both are desired parameters for key-exchange protocols. One of the major criteria is to provide privacy protection. That is to provide the development of a list of important industrial standards of KE protocols, which is particularly witnessed by the evolution of IKE. Deniable key exchange is when the key-exchange protocol can be denied. Hence both the protocol participants can deny all the transactions made using the session key produced by the key-exchange protocol. It is comparatively very advantageous for privacy preservation and has been proved by Meng-Hui Lim.

To achieve the basic privacy preserved authentication using key exchange, we basically take into consideration of two papers together that is “Cryptographic Algorithms for Secure Data Communication” by Zirra Peter Buba and “Key Agreement uses Diffie-Hellman Method and Secure Chatting Program**”** by Jain Rupesh R.

A. “Cryptographic Algorithms For secure Data Communication” by Zirra Peter Buba:

Personal privacy is become a must in the global networked world and is of high importance. One of the best tools to help people secure their personal data is by the use of cryptography. In this paper, new cryptographic algorithms have been employed that make use of the asymmetric keys. In the proposed algorithms, by using public key we can encipher the message into nonlinear equations and by using the private key we can decipher by the intended party. If a third party interrupted the message, due to the multilevel ciphers it will become very difficult to decipher of the proposed application.

The proposed system by Zirra Peter Buba in this paper is an encryption algorithm system which consists of three levels of cipher attempt to keep the data secure. The first level could be achieved by word compression flowchart. The second level consists of transforming the compressed words into systems of nonlinear equations and the third level consists of applying the delta encoding principles.

B. “Key Agreement using Diffie-Hellman Method and Secure Chatting Program” by Jain Rupesh R:

Secure chat program is to be made using java. Using Diffie-Hellman key exchange; we have to transfer server public key to client and client public key to server.

The Diffie-Hellman key agreement protocol is also called exponential key agreement. It was developed by Diffie and Hellman in 1976 and published in the ground-breaking paper “New Directions in Cryptography.” The protocol basically lets the two parties exchange a secret key to each other on an unsecured medium without any prior knowledge. The protocol has two system parameters *a* and *b*. Both of these parameters are public and can be used by any user in a system. Parameter *a* is a prime number and parameter *b* (usually called a generator) is an integer less than *a*, which is capable of generating every element from 1 to *a-1* when multiplied by itself a certain number of times, modulo the prime *a*.

Suppose that Henry and Mary want to agree on a shared secret key using the Diffie-Hellman key agreement protocol. First, Henry generates a random private value *p* and Mary generates a random private value *g*. Then they derive their public values using parameters *a* and *b* and their private values. Henry’s public value is $b^p \bmod a$ and Mary’s public value is $b^g \bmod a$. After that their public values are exchanged. Finally, Henry computes $k_{pg}= (b^g)^p \bmod a$, and Mary computes $k_{gp}=(b^p)^g \bmod a$. Since $k_{pg}=k_{gp}=k$, Henry and Mary now have a shared secret key *k*.

The protocol depends on the discrete logarithm problem for its security. It assumes that it is computationally infeasible to calculate the shared secret key $k=b^{pg} \bmod a$ given the two public values $b^p \bmod a$ and $b^g \bmod a$ when the prime *a* is sufficiently large. The breaking of the Diffie-Hellman protocol was equivalent to computing discrete logarithms under certain assumptions was shown and proved by Maurer.

The Diffie-Hellman key exchange protocol is helpless when it comes to middleperson attack. In this attack, an opponent, Alice, intercepts Henry’s public value and sends her own public value to Mary. When Mary transmits his public value, Alice substitutes it with her own and sends it to Henry. Alice and Henry thus agree on one shared key and Alice and Mary agree on another shared key. After this exchange, Alice simply decrypts any messages sent out by Henry or Mary, and then reads and possibly alters them before re-encrypting with the proper key and transmitting them to the correct party. This vulnerability is due to the disadvantage that Diffie-Hellman key exchange does not authenticate the participants which could lead to privacy issues. One of the solutions to this problem includes the use of digital signatures and other protocol variants.

The main objective of this paper is to create secure chat program that lets the participants communicate securely with each other. The AES encryption/decryption will provide us with the security of the data that is send. But before the sending transaction of any data takes place, both the sides must agree on a key; this key distribution problem can achieved with Diffie Helman key exchange method. The AES encryption/decryption will make use of this agreed key.

IV. ADVANTAGES OF DIKE

The DIKE protocol provides privacy to both the participants. In this all the authentic messages, $POK(\hat{Y}, b)$ and $NMZK(y, b)$ (resp., $NMZK(x, a)$), from $\hat{Y}$ (resp., $\hat{X}$) can simply be calculated from its peer’s DH-exponent *a* (resp., *b*) one’s own public messages. When one party is sure of the fact that the peer knows the conforming DH-component, then it sends the authenticate message including the secret-key. This guarantees deniability for both of the protocol participants simultaneously. As per our information, this is the first provably secure DHKE protocol that provides deniability for both protocol participants simultaneously. We can also interpret the authentic message $POK (\hat{Y}, b)$ as a proof-of-knowledge (POK) of the DH-exponent *b* for the DH-component *B* sent by $\hat{Y}$, that is in turn bounded to the identity $\hat{Y}$. In this way, proving the combined knowledge of both *x* and *a*, the first three round message can be viewed as a nonmalleable zero-knowledge (NMZK) and the combination of the first, the third and the fourth rounds can be viewed as an NMZK for proving the combined knowledge of both *y* and *b*. IKev2 and SIGMA protocol use signature as the core protocol and hence cannot utilize this privacy features. In DIKE protocol the messages from one party do not contain of any information about the peer's ID and public key. The DIKE protocol works in the post-specified-peer setting.

V. Comparison of Proposed Deniable IKE with SIGMA

In comparison with SIGMA protocol (i.e., the basis of IKEv2), our deniable IKE has the following advantages:

- Our deniable IKE provides full synchronized and forward deniability with nonmalleable zero knowledge, while SIGMA does not provide us with this facility (due to the underlying signatures used). Besides full deniability, our deniable IKE also provides apparently better privacy protection for player role participants than SIGMA protocol.
- Our deniable IKE is of very simple conceptuality and clarity.
- Our deniable IKE is basically taken more in practice rather than SIGMA protocol. Basically, in our deniable IKE, the player need not show or use signature to prove their possession. Rather, it shows the stronger proof-of-knowledge of its secret-key via an approach such that the proof-procedure and the verification-procedure respectively, only need one modular exponentiation respectively (we do not know, to our knowledge, so practical secure signature scheme even in the RO model).
- Our deniable IKE is more protected and secured as compared to SIGMA protocol.

VI. Conclusion

In this paper, we have shown various security services that can be used to improve the security and the privacy features for key-exchange over the internet. Advantages of DIKE over SIGMA protocol have also been explained. Deniability, Internet Key-exchange (IKE), Internet Protocol Security are the basic building blocks that ensure security in accordance with various security services like confidentiality, integrity, authentication and privacy preservation while communication.

References

[1] H. Krawczyk, "SIGMA: The 'SIGn-and-MAc' approach to authenticated Diffie-Hellman and its use in the IKE-protocols," in *Proc. CRYPTO 2003*, pp. 400–425.
[2] H. Krawczyk, "HMQV: A high-performance secure Diffie-Hellman protocol," in *Proc. CRYPTO 2005*, pp. 546–566.
[3] C. Kudla and K. Paterson, "Modular security proofs for key agreement protocols," in *Proc. Asiacrypt 2005*, pp. 549–565.
[4] W. Mao, *Modern Cryptography: Theory and Practice*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2004.
[5] U. Maurer and S. Wolf, "Diffie-Hellman oracles," in *Proc. CRYPTO 1996*, pp. 268–282.
[6] M. Naor and O. Reingold, "Number-theoretic constructions of efficient pseudo-random functions," *J. ACM*, vol. 1, no. 2, pp. 231–262, 2004.
[7] K. Neupane, R. Steinwandt, and A. S. Corona, "Scalable deniable group key establishment," in *Proc. FPS 2012*, pp. 365–373.
[8] T. Okamoto and D. Point cheval, "The gap-problems: A new class of problems for the security of cryptographic schemes," in *Proc. PKC 2001*, pp. 104–118.
[9] R. Pass, "On deniability in the common reference string and random oracle models," in *Proc. CRYPTO 2003*, pp. 316–337.
[10] R. Pass and A. Rosen, "New and improved constructions of nonmalleable cryptographic protocols," in *Proc. STOC 2005*, pp. 533–542.
[11] A. P. Sarr and P. E. Vincent, "A complementary analysis of the (s)YZ and DIKE Protocols," in *Proc. Africacrypt 2012*, pp. 203–220.
[12] M. Yung and Y. Zhao, "Interactive zero-knowledge with restricted random oracles," in *Proc. TCC 2006*, pp. 21–40.
[13] A. C. Yao and Y. Zhao, "Deniable Internet key-exchange," IACR (The International Association for Cryptologic Research), San Diego, CA, USA, Tech. Rep. 2011/035, Jan. 2011.