NEW KEY MANAGEMENT PROTOCOL FOR SSL/TLS*

Ibrahim Hajjeh[1], Ahmed Serhrouchni[1], Frédérique Tastet[2]
[1] Ecole Nationale Supérieure des Télécommunications; [2]Thales Communications

Abstract: In 1998, the IETF adopted ISAKMP (Internet Security association and key management protocol) as the standard key management protocol for IPSEC. ISAKMP was conceived to be a generic protocol, its features and capabilities were intended to provide a standard framework for the negotiation of security parameters for any security protocol. However, since then, ISAKMP has never been used outside the IPSEC framework. In this paper, we give a new dimension for ISAKMP by establishing a secure session for the SSL/TLS protocol. The main objective of this integration is to exploit the unused functionalities of ISAKMP and extend the work of SSL/TLS to support new services. We also analysé and discuss the recent successors to the ISAKMP protocol.

Key words: Key Management, Security Associations, ISAKMP, SSL Handshake, DOI

1. INTRODUCTION

Today SSL/TLS[3] (Secure Socket Layer / Transport Layer Secure) [1] [2] is the most deployed security protocol. This is due, mainly, to its native

* Work supported by the French program RNRT (Réseau National de Recherche en Télécommunications) under the Icare project (Infrastructure de Confiance sur des architectures de Réseaux Internet & Mobile)

[3] TLS protocol is based on SSL 3.0 Protocol published by Netscape. "No dramatic" difference exist between these two protocols. In this paper we mean by SSL, SSL/TLS.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35703-4_21
D. Gaïti et al. (eds.), *Network Control and Engineering for QoS, Security and Mobility II*

integration in web servers. However, this protocol has some limitations regarding the absence of some security services and access control mechanism. Currently SSL is made up of four modular and independent protocols, of which the Handshake protocol has to authenticate the parts concerned, negotiate security algorithms and generate a shared secret necessary to guarantee the services of integrity and confidentiality. The modular nature of SSL allows us to integrate ISAKMP [3] easily in the session establishment phase of SSL.

However, because ISAKMP and IKE were criticized as being too complex, the IETF IPSEC consortium have decided upon a new requirements for the creation of a new extensible and simple key management protocol that simplifies the use and deployment of IPSEC [4]. Between three candidates (JFK, IKEv2 and SIGMA) [5] [6] [7], IKEv2 was the front-runner even though the final version of IKEv2 integrates in part SIGMA and JFK. This proves in fact that the complexity came from the non-useful scenarios and the difficulties in managing the IKE configuration parameters and not from the ISAKMP architecture. We think that ISAKMP should be used outside the IPSEC consortium and that for three reasons:

1. Its capacity to offer a large scale of security services.
2. Its capacity to unify security solutions on different level of the network stack.
3. Its robustness

We thus propose a new architecture for SSL with ISAKMP to ensure, amongst other things, the service of authentication and identity protection.

The remainder of this paper is organized as follows. In the following section, we present the SSL protocol. In section 3 we present the ISAKMP architecture. In section 4 we describe the new proposed key exchange protocol then, in section 5, we discuss and analyse the advantage of these proposals and we explain why we use ISAKMP outside the IPSEC WG and specially with the SSL protocol. In section 6, we describe the integration of ISAKMP in SSL and we give the proposed TLS extension field. To conclude we propose an analysis of this solution and its prospects, in particular in experimentation and deployment.

2. SSL PROTOCOL

SSL (Secure Socket Layer) was originally designed by Netscape Company to meet the occurring needs of Internet Security at that time. In March 1996 TLS (Transport Layer Security) was approved by the IETF as the standard internet secure protocol. SSL and TLS provide a generic channel security mechanism that operate over a reliable transport protocol

like TCP to provide peer entity authentication, data confidentiality, data integrity, key generation and distribution, and security parameter negotiation.

The SSL protocol consists of four modular protocols (Figure.1): the SSL handshake, the SSL Record, the SSL Alert and the SSL Change Cipher Spec. The SSL handshake protocol allows the peer entities located at both ends of the channel to authenticate one another, to negotiate cryptographic algorithms and exchange secret session keys for encryption.

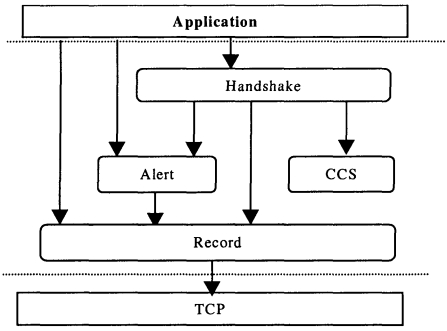

Figure 1 SSL Handshake

Once The Handshake protocol establish the shared secret, the SSL Change Cipher Spec protocol notify the SSL Record protocol that subsequent records should be protected under the newly negotiated cipher spec and keys. If an error is detected in the secured channel, the SSL Alert Protocol in the detecting party sends an alert message containing the occurred error.

When an SSL client initiates a connection with an SSL server, they first run the SSL Handshake protocol to negotiate security algorithms, to authenticate each other and to establish a shared cryptographic secret.

SSL handshake supports two different key exchange methods: Key exchange with RSA keys and Key Exchange with Diffie-Hellman (DH) keys (See [8] for a description of RSA and DH algorithms). These two key exchange methods are used in three different SSL authentication modes: authentication of both parties, server authentication with an unauthenticated client, and total anonymity. In the anonymous key exchange mode, the public RSA key or the DH components are exchanged without authentication. In this mode the shared secret is protected from eavesdropping but since the communicating parties are not authenticated, active man-in-the middle attack are possible.

In the case where only the server is authenticated, the server's public key can be verified by the client using the certificate sent by the server. This is the most used mode of SSL where only the server has to prove his identity.

In the mutual authentication mode, the client and the server are authenticated based on theirs certified public keys. In this mode the client should sign a hash value derived from the shared secret and all proceeding handshake messages. The verification of the signature by the server using the client's

certified public key proves the client's identity. The identity of the SSL server is also verified as the client did in the previous case.
SSL was designed to provide server authentication to clients easily and efficient encryption negotiation for any application layer program. Nevertheless HTTP is the protocol the most frequently used with SSL and so it is so natural to think that HTTP runs inside SSL. SSL shows its advantage when performing many secure and small connections. Unfortunately it is non trivial to predict the exact result of SSL on movies, audio or multi-cast services. The next sections will introduce to ISAKMP protocol and Security Association negotiation. Our main objective will remain to keep interoperability with current SSL handshake.

3. ISAKMP PROTOCOL

The Internet Security Association and Key Management Protocol (ISAKMP), defined in the RFC 2408 of November 1998, is a framework that defines procedures and packet formats for establishing, negotiating, modifying and deleting Security Association (SA). It also allows two peers to authenticate each other and to perform key exchange in a protocol- and algorithm- independent way. In this part, we present the general architecture and key negotiation mechanism of ISAKMP without going into the details of various ISAKMP payloads nor different exchange types.

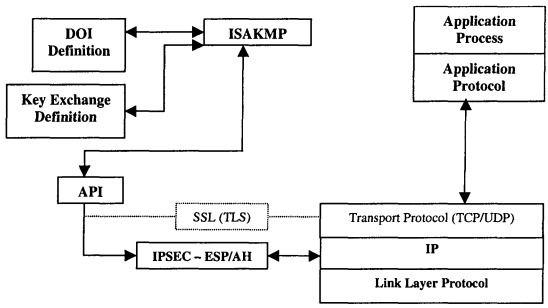

Figure 2. ISAKMP Architecture

ISAKMP can be implemented over any transport protocol or over IP itself. Implementations must however support at least UDP. Because ISAKMP does not impose anything on the parameters that compose the SAs, a document called Domain of Interpretation or DOI [9] must define the negotiated parameters. The DOI plays an essential role in key management. DOI defines payload formats, exchange types and some security information such as security policies or cryptographic algorithms. There is also a DOI identifier used to interpret the payloads of the ISAKMP messages. For example, the IPsec protocol has number 1 as its DOI identifier. A list of the

DOIs is defined in [9]. A new DOI Identifier for SSL should be added. "Figure 2" illustrates a high level-level view of the placement of ISKAMP within a system context in network architecture.
ISAKMP comprises two phases that allow a clear separation of the SA negotiation for a specific protocol and traffic protection.
During the first phase, all attributes regarding the security are negotiated, the identities of the thirds are authenticated and the keys are generated. These elements constitute a first "Security Association", known as SA ISAKMP.
Phase 1 is concerned only with establishing the protection suite for the ISAKMP messages and does not establish any security associations or keys for protecting user data.
The second phase makes it possible to negotiate the security parameters related to the SAs to another security protocol like Authentication Header (AH) [10] and Encapsulating Security Payload (ESP) [11] to protect user data exchanges. The exchanges of this phase are protected (confidentiality, authenticity...) by the SA's ISAKMP established in phase 1. Phase 1 negotiations are executed once a day or maybe once a week but phase 2 negotiations are executed once every minute.

4. IKEV2 PROTOCOL

In 2002, the IPsec Working Group considered a new successor to the IPSec key exchange framework. The proposed protocols try to give a simple, secure and inexpensive key agreement protocol. After a long discussion in the IPSec mailing list, the result was IKEv2 that tried to include all the advantages in the three proposed protocols. IKEv2 is an attempt to simplify the standard, remove the un-needed requirements, and incorporate new standard IPSec functionalities currently contained within other documents.

The IKEv2 is very similar to IKEv1 in performing mutual authentication and establishing an ISAKMP security associations. IKEv2 first replaces the eight possible phase 1 exchanges with a single exchange that provides identity protection and is based on either public signature or shared secret keys. In addition, IKEv2 is the only proposal that was conceived to be simply extensible. In a simple manner, IKEv2 propose adapting a simple hash function over all payloads, no matter which. This has a clear implication to extensibility, i.e. future payloads (in JFK, it is very difficult to add a future payload and to put it under the encrypted or non-encrypted part). As shown in Figure. 3, IKEv2 can assure identity hiding with two round trips.

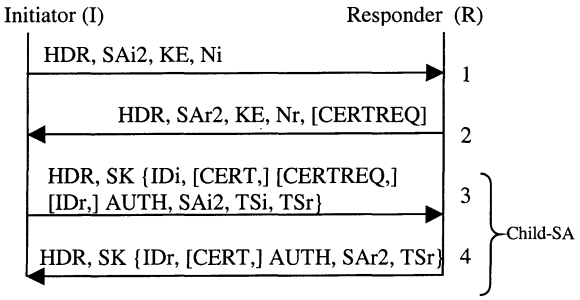

Figure 3. IKEv2 protocol exchange

In the first exchange, the initiator sends a list of proposed cryptographic algorithms in the SA payload, his Diffie-Hellman public value and a random nonce Ni. The responder will reply by sending in the SA payload his accepted algorithm suite, complete the DH exchange with the KE payload and sends a nonce value in the Nr payload. At this point, the two endpoints begin generating the master secret SKEYSEED and the derived keys SK_e, SK_a and SK_d. now, all messages in the second round trip (without the HDR payload) will be encrypted using the encryption key. The initiator can now send his identity with the ID payload, and a hash of the first round trip messages using the AUTH payload. The initiator can optionally send his certificate that contains his public key that proves his real identity. The initiator can also send a certificate request and the identity of the responder that can host multiple services. The second exchange contains also the SA2 that can serve for the child-SA negotiation and the TS payloads. In the last message the responder will assert his identity in the IDr payload, optionally sends his CERT that contain his public key, hash the 3 messages in AUTH payload to assure an integrity protection and complete the negotiation of a child SA. TSi and TSr are used to assure the description of traffic to be sent.

5. ANALYSIS

In the previous section, we have described the recent proposed key exchange protocol for the IPSec consortium. Even though the three proposals (JFK, IKEv2, and SIGMA) aim at simplicity as their main objective, there is still the real cost and need of developing and implementing a new protocol. The first objective that the IPsec consortium tried to obtain is the compatibility and code reuse with the existing solution. This has eliminated JFK from being the next IPSec KE standard. H. Krawczyk also noticed other disadvantages of JFK such as: the unsecured JFK KE mechanisms, forcing the non-repudiation service between the two communicators and the long input to HMAC. On the other hand, the SIGMA protocol was extremely compatible with IKEv1, it has been proven that it is

a secure protocol in its core cryptographic design. The main difference with IKEv2 is that SIGMA proposed a single-phase exchange. As described in different documents, the two phases were proved useful in the IKEv1 and for this reason this option is maintained in the new proposal. Finally, the IPsec WG has preferred to combine all the advantage of these proposals under the IKEv2 protocol.

There is a general impression in the IPsec WG thinking that ISAKMP is an overly generalized protocol, hard to evaluate and to understand. However, the next IPSec KE protocol will be totally based and dependent on the ISAKMP framework with fewer functionalities and fewer unused IKEv1 scenarios. However, ISAKMP was first conceived to support the negotiation of Security Associations (SA) for security protocols at all layers of the network stack (e.g., IPsec [4], SSL...). Contrary to the existing proposals, ISAKMP was considered to be generic and able to support all types of protocols. A manner to unify all these proposals is a real implementation of ISAKMP in SSL. Our main reasons for adapting ISAKMP in SSL are as follows:

- Supporting Identity Protection of the two communicators.
  Identity Protection is primarily useful where one host has multiple identities and wishes to mask who is behind a specific handshake (unlike other protocols, identity in ISAKMP does not necessarily bear any relationship with an IP address, but it can be related to various information existing in certificates).
- Separating the functionality of key exchange from security association management.
  This is critical for interoperability between systems with differing security requirements. It also simplifies the analysis of further evaluation of an ISAKMP server.
- Parameterized Protocol
  ISAKMP is a full-parameterized protocol by a domain of interpretation (DOI) [9] in which specific key determination mechanisms are defined. The extended use of ISAKMP with SSL can be realized through the specification of a new DOI that we call it TLS DOI.
- Modularity of ISAKMP Protocol.
  Since ISAKMP is an application layer protocol, then it benefits from all the advantages of this layer; including data comprehension, user authentication related to application and also the advantage of non-repudiation, which can cause an overhead in other sub layers.
- Interoperability between different security protocols.
  Several protocols (e.g. TLS, IPsec...) could share the same key management code. This simplifies migration from one protocol to

another and reduces the amount of duplicated functionality within each security protocol.

- Sending initiator or responder certificate is optional.
  In ISAKMP, sending a certificate to a responder is optional. The sender can transmit a URL to an LDAP (Lightweight Directory Access Protocol) Directory pointing to his certificate. This is significant to reduce the time of ISAKMP exchange in the case where several certificates are transmitted between the two communicators.

6. INTEGRATING ISAKMP IN SSL

6.1 Architecture

The integration of ISAKMP in SSL (Figure. 5) can begin after an ISAKMP phase one negotiation. ISAKMP phase two is intended to create a Security Association SA for any protocol specified and defined in the DOI database. Unlike ISAKMP SA, SA phase 2 is unidirectional. This gives the two communicated entities different authentication methods. Any SSL ISAKMP integration should respect the three main definitions:
The exchanged message should respect the ISAKMP message format. In our case, *Oakley quick mode* [13] (Figure. 4) and other optional ISAKMP payload combination are enough to assemble all four SSL handshake scenarios (Server Authentication, client and server authentication, RSA or DH authentication And resumed handshake) even though a normal phase two exchange is compared to SSL session resumption.

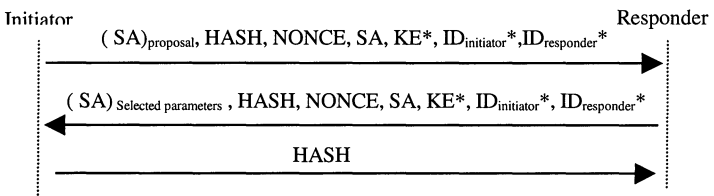

Figure 4. Oakley Quick Mode.

The result in derived keys and exchanged algorithm names should be transparent to the SSL Record protocol. This is automatically provided

because any phase 2 exchanges will be based on Diffie-Hellman public keys, SKEYID[1] and the NONCEs values.
For backward compatibility with TLS Protocol, a new TLS extension should be added allowing the SSL client to choose his key negotiation mechanisms (SSL Handshake, ISAKMP...). In this paper, we present a new extension type named "isakmp_key_negotiation" based on [12].

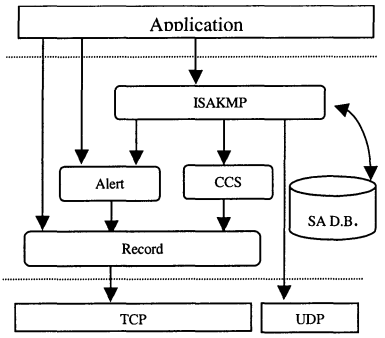

Figure 5. SSL ISAKMP Architecture

Integrating ISAKMP in SSL brings notably new authentication methods, identity protection, and fast algorithm negotiations. These points will be detailed in this section.

1. New authentication methods

ISAKMP makes no distinction between client and server because the first use of ISAKMP was with IPsec and this distinction does not exist at the IP layer. Instead, we call the sender of the first packet the initiator and the second the responder. To preserve interoperability with the SSL handshake we will explain how ISAKMP can re-generate the SSL handshake scenarios.
First, the two SSL Key Exchange methods (RSA and DH) are ensured by ISAKMP with the *Key Exchange (KE)* payload. ISAKMP have also a pre-shared key for a quick negotiation.
The random client and server in SSL are replaced with $NONCE_{initiator}$ and $NONCE_{responder}$ payloads in SSL/ISAKMP.
The list of negotiated algorithms in SSL/ISAKMP is ensured using *proposal* and *transform* payloads exchanged in the *Security Association (SA)* payload.
At this point, the server authentication in SSL can be replaced with an X509 certificate for the responder and a NONE (defined as one of the seven certificate formats in rfc. 2408) certificate for the initiator. The certificates are transmitted in the *Certificates* payload. The *HASH* of all exchanged messages encrypted with their public keys are exchanged in the

[1] SKEYID provides the raw input from which cryptographic keys will be derived later. It obtained by applying the agreed-to pseudorandom function to the known inputs: (N_i, N_r, g^{xy}) where N_i and N_r are respectively the two nonces of the initiator and responder. g^{xy} is the shared DH secret.

authentication payload. A *signature payload* can also be added to sign to hash data.
Like server authentication, client and server authentication is a bi-directional authentication with X509 certificates and signature payload.
In SSL, anonymity authentication require only *KE* payload with Diffie-Hellman or RSA key exchange and a *HASH* Payload;
The last one in SSL is session resumption. This is the fastest key exchange in SSL/ISAKMP where the two entities generate new keying material from the original SA without a new key exchange.
In addition, other authentication methods can be obtained by ISAKMP like authentication based on delegated attribute certificate, user information, manual shared secret, and distributed secret by a key distribution center (KDC).
2. Identity protection
Identity protection is primarily useful for multi-users using a shared station or IP address. An ISAKMP long life phase 1 exchange can be established with authentication based on the IP address or station certificate. Every user can afterwards use his proper authentication methods that can even be a combination of IP address and certificates.
3. Fast algorithm negotiations
Unlike the SSL handshake, the SSL/ISAKMP exchange is based on shared and authenticated keys. A new process can be added to SSL for a fast handshake. If both hosts have defined the authentication method with X509 certificate, the initiator can send his certificate in the first message exchanged, with a signature payload (the signed data can be all the exchanged messages in phase 1).
Finally, the negotiation of SSL ISAKMP will not be possible without DOI values assigned by the Internet Assigned Numbers Authority (IANA). This part will be defined in a separate document.

6.2 TLS Extension

[12] has proposed a new extension to extend the work of TLS protocol to new environments. This work has reinforced our proposition and resolves all compatibility problems between TLS entities that want to open an ISAKMP phase 2 negotiation instead of the normal TLS handshake and TLS servers that do not support this option, and vice versa.
In order to allow a TLS client to negotiate an ISAKMP Exchange instead of a normal TLS handshake, a new extension type should be added to the Extended Client Hello and Server Hello messages. TLS clients and server may include an extension of type "isakmp_key_negotiation" in the (extended) Client Hello and Server Hello messages. The "extension_data"

[12] field of this extension will contain an "ISAKMP_KN_Request" where the TLS presentation for isakmp_key_negotiation":

```
enum {
      false(0), true(1);
       } Boolean;
struct{
      Boolean ISAKMPSA_Present;
      Select (ISAKMPSA_Present) {
            Case true: Begin_ISAKMP_Phase2
            Case false: Begin_ISAKMP_Phase1
           } request;
       } ISAKMP_KN_Request;
struct {
  ISAKMP_phase2_Exch  ISAKMP_phase2_Exch_list<1..2^16-1>;
      } Begin_ISAKMP_Phase2;
struct {
  ISAKMP_phase1_Exch  ISAKMP_phase1_Exch_list;
      } Begin_ISAKMP_Phase1;
struct {
      DOI doi_id ;
      } ISAKMP_phase1_Exch;
struct {
      DOI doi_id ;
      SPI_value spi-value;
      } ISAKMP_phase2_Exch;
opaque SPI_value<1..2^16-1> ;
opaque Proposal_number[1] ;
opaque doi_id[4] ;
```

The presence of the isakmp_key_negotiation extension gives TLS client the potential to use an existing ISAKMP phase 1 negotiation and begin directly with a new SSL ISAKMP exchange or renegotiate a new ISAKMP phase 1 and 2 if there is no ISAKMP SA between client and server.

- If "isakmp_present" contain the Boolean "false", TLS client requests to establish an ISAKMP SA and an SSL ISAKMP SA with the TLS server. TLS client will just send the DOI identifier (e.g. "5" for TLS DOI). In this case, TLS client will be forced to reject communications with servers that do not support this extension or this DOI identifier.

- If "isakmp_present" contain the Boolean "true", TLS client will begin an SSL ISAKMP SA based on an established ISAKMP SA. The TLS client may send a list of established ISAKMP SA. The client will send to Identifier of ISAKMP SA in "SPI_id" and the DOI he want to employ.

7. CONCLUSION

Protocols for authentication and key management are the most essential part for the future growth of security protocols. Nowadays, each protocol brings its own key exchange mechanism. We have considered some recently proposed protocols under the ISAKMP framework. We are aware that ISAKMP exceeds the capacity of IPSec and we recommend that ISAKMP be used outside the IPSec working group to meet its goal of supporting the establishment of security associations for all possible security protocols and applications. For SSL, ISAKMP brings new services like authentication, fast algorithm negotiations and authorization mechanisms based on attribute certificates. Currently, an experimental implementation of ISAKMP in SSL is in progress.

REFERENCES

[1] T. Dierks, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[2] A. Freier, P.Karlton and P.Kocker, "The SSL Protocol, Version 3.0", Internet Draft, 1996.

[3] D. maughan, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

[4] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998

[5] W. Aiello and al, "Just Fast Keying (JFK)", IETF Draft, draft-ietf- ipsec-jfk-00.txt

[6] H. Krawczyk, "The IKE-SIGMA Protocol", IETF Draft, draft- krawczyk-ipsec-ike-sigma-00.txt. November 2001.

[7] C. Kaufman, "Internet Key Exchange (IKEv2) Protocol", IETF Draft, draft-ietf-ipsec-ikev2-08.txt, May 2003

[8] B. Schneier, "Applied Cryptography", 2nd Edition, J. Wiley & Sons, NY 95

[9] D. Piper, "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[10] S. Kent and al., "IP Authentication Header (AH)", RFC 2402, Nov. 1998.

[11] S. Kent and al., "IP Ecapsulating Security Payload (ESP)", RFC 2406, Nov. 1998.

[12] S. Balke-Wilson and al., "Transport Layer Security (TLS) Extensions", IETF Draft, draft-ietf-tls-extensions-06.txt, February 2003.

[13] H. Orman, Univ. of Arizona, "The OAKLEY Key Determination Protocol", RFC 2402, November. 98.