HIP_IKEv2: A Proposal to Improve Internet Key Exchange Protocol-based on Host Identity Protocol

S. Smaoui[1], F. Zarai[1], M. S. Obaidat[2], K. F. Hsiao[3] and L. Kamoun[1]
[1]*LETI laboratory, University of Sfax, Sfax, Tunisia*
[2]*Computer Science and Software Engineering Department, Monmouth University, West Long Branch, NJ 07764, U.S.A.*
[3]*Dep. of Information Management, Ming-Chuan University, Taoyuan County 333, Taiwan*

Keywords: IP Security, Simulation Analysis, Internet Key Exchange Version2, Host Identity Protocol, AVISPA, SPAN.

Abstract: IKEv2 offers authentication, authorization and key agreement services to establish a security association between two peers bound to IP addresses, but it is still vulnerable to some security problems such as denial of service (Dos) and man-in-the-middle attack. Host Identity Protocol (HIP) is also a security protocol that defines host identifiers for naming the endpoints and performs authentication and creation of IPsec security associations between them bound to identifiers. The purpose of HIP is to support trust systems, enhance mobility and greatly reduce the Denial of Service (Dos) attacks. We focus on an extension to IKEv2 in order to enhance authentication, eliminate man-in-the-middle attack and guarantee denial of service to provide better security between the two peers. In this paper, we describe our proposal that consists of combining the IKEv2 with the HIP to set up a security association based on two parameters which are location and Identity. This combination can provide better security properties than each protocol used alone. This scheme, named (HIP_IKEv2) couples location and identity to define a security association between two peers. We have used the Automated Validation of Internet Security Protocols and Applications (AVISPA) and its Security Protocol Animator (SPAN), and two powerful automated tools in order to formally specify and validate the HIP_IKEv2 protocol.

1 INTRODUCTION

With the development of wireless communication systems, the needs for security are increasingly become more important than ever before because these systems present a set of challenges which are distinct from those of wired based systems. A successful security strategy in wireless networks implies new needs such as, authentication of the entities, integrity of the exchanged messages, confidentiality of the transaction, and anonymity of the certificate owner.

Before explaining the necessary security mechanisms, it is important to enumerate some possible attacks techniques like eavesdropping, impersonation, man-in-the-middle, replay and denial of service attacks (Gurtov, 2008).

In order to tackle these attacks, a preventive mechanism is needed. Hence, it is desired to use cryptography in communication between users and different nodes in the network. In this context many protocols have been designed to assure security services such as authentication, authorization, accountability, data integrity, confidentiality, non-repudiation and privacy. Since the aim is to create security services at the network layer and since IPSec protocol is used to protect mobile Internet protocol version6 (MIPv6) data and signals, we choose to benefit from it to establish a security tunnel between two peers.

With IPSec both communication endpoints must agree on a set of algorithms and keys to achieve a secure connection (Gurtov, 2008). To establish a shared state, hosts can employ two different protocols which are the Internet Key Exchange IKEv2 (Kaufman et al., 2010) and the Host Identity Protocol (HIP) (Henderson and Gurtov, 2012). The shared security context is called a Security Association (SA).

SA is a central concept in IPSec that supports encryption, authentication, or both. They are unidirectional, so in order to protect a duplex

Smaoui S., Zarai F., S. Obaidat M., F. Hsiao K. and Kamoun L..
HIP_IKEv2: A Proposal to Improve Internet Key Exchange Protocol-based on Host Identity Protocol.
DOI: 10.5220/0004630804040411
In *Proceedings of the 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications* (SIMULTECH-2013), pages 404-411
ISBN: 978-989-8565-69-3

channel two SAs are necessary (an incoming and an outgoing one).

The IKEv2 exchange is designed to establish and manage SA in four messages. In the two first one, called IKE_SA_INIT, the communicating entities negotiate cryptographic algorithms, exchange nonce and make Diffie-Hellman exchange to obtain a shared key. In the last two messages, called IKE_AUTH, both entities authenticate the previous messages and exchanges Identity. Finally, the SA established by IKEv2 is essentially on based location.

In the same context, the Host Identity Protocol is used to establish a pair of IPSec security associations between two hosts through the HIP Base Exchange (HBE). The HBE consists also of four messages (I1,R1,I2,R2) based on a classic Diffie-Hellman key exchange with an inclusion of a puzzle by the responder node as a cryptographic challenge in order to avoid a Dos attack from an illegitimate node that wishes to saturate the responder node with HIP initiation messages (Arraez et al., 2011). Finally, the SA established by HIP is based essentially on the host identity name space introduced by this protocol.

Developing new security protocols is a difficult task and sometimes too difficult task for human mind. So, the idea is benefiting from existing protocols to create new one. We focus to extend the IKEv2 in order to enhance authentication, eliminate man-in-the-middle and reply attacks and guarantee Dos attacks in order to provide better security between the two peers. Hence in this paper, we describe a proposal that consists of combining the IKEv2 with HIP to set up a security association based on two parameters which are location and Identity. This combination may provide better security properties than each protocol used alone. This proposal, named (HIP_IKEv2) couple location and identity to define a security association between two peers. We have used the Automated Validation of Internet Security Protocols and Applications (AVISPA) and its Security Protocol Animator (SPAN), two powerful automated tools to formally specify and validate the HIP_IKEv2 protocol. The rest of this paper is structured as follows. Section II summarizes the state of the art related to this work. Section III describes integrating HIP with IKEv2. In Section VI, a formal specification and validation of the HIP_IKEv2 with AVISPA and SPAN are discussed. Finally, Section VII contains the conclusions and future works.

2 RELATED WORK

This section details related work focused on IKEv2 and its improvement. Having several advantages, IKEv2 still suffers from some deficiency, such as man-in-the-middle and Dos attack. Hence, the issue to protect peers form Dos attack has received the attention of researchers.

According to (Iso-Anttila et al., 2007) the resistance to Dos attacks is actually weaker in IKEv2 than in Just Fast Keying (JFK) or Full-SIGMA protocol in different networks. Therefore, the authors present a proposal to improve IKEv2 negotiation (Iso-Anttila et al., 2007), based on using cookies negotiation in order to detect a Dos attack, and present an improved cookies negotiation to remedy the weakness present in IKEv2. So the authors focus on preventing the traditional vulnerable cookies negotiation and adding a new challenge to the initiator without adding computational load. The proposed cookie negotiation delays the responder's calculation work to the last second and computational load is kept as low as possible.

Reference (Xiaowei et al., 2010) proposes an improvement of IKEv2, which is based on the shared secret and asymmetric distribution of calculations. By analyzing the improved IKEv2 with a cost-based framework, Iso-Anttila concludes that the improvement is robust against Dos attack. Furthermore, associated with cookie mechanism, the improvement can prevent flooding attack from spoofed IP addresses. And the improvement can also achieve the identity authentication in advance, resist man-in-the-middle attack and replay attack.

In (Zhou et al., 2010), a modified IKEv2 based on IP fragmentation, in which the authors design and implement an IKE application fragmentation protocol and put forward a series of other measures related to prevent lKEv2 from Dos attacks. Hence, they design a new IKEv2 header format called M-ISAKMP, and add a new type of Notification Payload and other related strategies. With the novel application-based fragmentation mechanism, the proposed solution achieves defending against Dos attack successfully and efficiently.

3 INTEGRATION HIP WITH IKEv2

This section describes a proposal that is based on making modification to the IKEv2 initial exchange

and then combining this modification with the HIP to set up a security association based on two parameters which are location and Identity. This combination may provide better security properties than each protocol used alone. This scheme, named (HIP_IKEv2), couples location and identity to define a security association between two peers. HIP_IKEV2 is dedicated to solve some secure problems. The first subsection describes the HIP_IKEV2 exchange, the following subsections describe how HIP_IKEV2 defends the man-in-the-middle, the Dos and the replay attack. Finally, the last subsection describes how HIP_IKEV2 assures the integrity of messages.

3.1 HIP_IKEv2 Exchange

We note that the two protocols mentioned previously consist of four messages between the two peers which are in the form of two pairs in two round-trip times. So the idea is to combine each message from the first protocol with a message from the second protocol according to their hierarchical order. Thus the HIP_IKEV2 exchange is composed of four messages. The protocol is described with the notation summarized in Table1.

The HIP_IKEV2 protocol negotiations start with a combination between the I1 packet from the HIP protocol and the first message of a modified IKEv2 Initial Exchange using public key in one message named M1.

Table 1: Notation Used in HIP_IKEv2 Protocol.

Notation	Description
HDR	IKE header
SAi_n, SAr_n,	Cryptographic algorithms
KEx	Key exchange payload of x
Nx	Nonce of x
HITx	Host Identity tag of x
PK_x	Public Key of x
PR_x	Private Key of x
$\{M\}_K$	Encryption of message M with key K.
CERTREQ	Certificate Request Payload
CERT	Certificate Payload
||	Concatenation operation.
H(M)	Hash of message M
IDx	Identity of x
AUTHx	Authentication Payload of x
TSx	Traffic selectors of x
SK	Diffie-Hellman shared secret key
Ts	Time Stamp
puzzle	A cryptographic puzzle
Sol	Solution of puzzle

In M1 the initiator sends the IKE header an H-flag set up to indicate that the HIP extension (HDR contains the Security Parameter Indexes (SPIs), version numbers, and flags of various sorts), the next payload, which is a suite of cryptographic algorithms SAi_1, the third payload contains the Diffie-Hellman value $\{KEi\}_{PK\_R}$ followed by the nonce $\{Ni\}_{PK\_R}$, the host identity tag HITi of the initiator and a time stamp ticket Ts. The last payload includes the HASH_1 payload, which contains a hash function applied to all messages encrypted with the private key to protect the integrity of the message.

When the responder receives the M1 message, it replies with an M2 message which presents a combination between the R1 packet from the HIP protocol and the second message of IKE_SA_INIT. M2 contains the selected cryptographic algorithms in SAr_1, its Diffie-Hellman value $\{KEr\}_{PK\_I}$ which completes the Diffie-Hellman exchange followed by a random nonce $\{Nr\}_{PK\_I}$, its HIT (the responder's HIT), a cryptographic puzzle to be solved by the initiator, and optionally he can send a certificate Request Payload (CERTREQ), if he want get initiator's certificate to authenticate him. M2 contains also a timestamp ticket and a HASH_2 payload. After these first two messages, the hosts establish a secure tunnel. Until on receiving the M2 message, the Initiator can compute the keying material for the shared secret. So a symmetric key is in possession of the two peers.

Now, the initiator and the responder have all necessary information for a cryptographic protected conversation. All messages after M1 and M2 exchange are protected with the negotiated cryptographic algorithm. Only the headers of the next messages are not encrypted.

Once the puzzle is solved, the initiator can replies with an M3 message containing the HITs of the initiator and the responder, and the solution of the puzzle, also the initiator reveal its identity IDi and optionally he presents the certificate Payload (CERT) which is only necessary if the responder sent a CERTREQ. Also, he can sent a CERTREQ payload if he want to get a certificate of the responder. M3 contain also the second suite of cryptographic algorithms SAi_2, the authentication data (AUTHi) which is used by the initiator to authenticate its identity. The value of this AUTHi payload is calculated using a shared secret. In the last two payloads the initiator proposes traffic selectors (TSi, TSr) which include policies like the IP address range or the port range. This M3 message presents a combination between the first message of

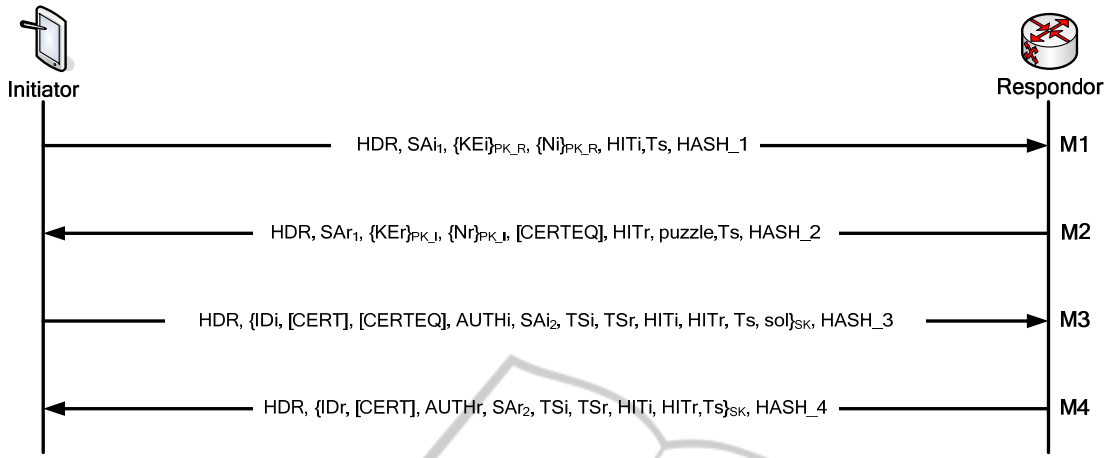

Figure 1: The HIP_IKEv2 Exchange.

IKE_AUTH from the IKEv2 protocol and the I2 packet from the HIP protocol.

Finally, the responder send an M4 message which couple the second message of IKE_AUTH and the R2 packet containing also the HITs of the initiator and the responder, its identity IDr, the selected algorithms SAr_2, the authentication data (AUTHr) and traffic selectors (TSi, TSr). The M3 and M4 contain also a time stamp ticket and a HASH_3 and HASH_4 payloads respectively.

Now, during the last two messages the hosts identify each other and establish a security association bound to IP addresses and HITs.

3.2 Defend Man-in-the-Middle Attack

Although IKEv2 is designed to improve IKEv1 and solve related security problems, it still faces the man-in-the-middle attack, especially in the IKEv2 initial exchange. In order to resist man-in-the-middle attack, the responder and the initiator should confirm the identity of each other before distributing resources. Hence, the idea in our improvement is based on public key cryptographic mechanism. In the initial exchange of IKEv2 protocol, the initiator Diffie-Hellman value (KEi), the initiator nonce (Ni), the responder Diffie-Hellman value (KEr) and the responder nonce (Nr) are transferred without any kind of security. So, an attacker can easily intercept. For example, the attacker can intercept in the first message of this initial exchange by replacing KEi with KEa1 also Ni with Na1 and send the replaced payloads to the responder. So the responder instead of receiving parameters of the initiator, it receives those of the attacker. The same for the second message, the attacker can also intercept by replacing KEr with KEa2 and Nr with Na2 and send the replaced payloads to the initiator. Thereby the initiator instead of receiving parameters of the responder, it receives those of the attacker. To eliminate this thread, we propose to use mutual public key in M1 and M2 message to encrypt KEi, Ni, KEr and Nr. We also propose to sign these messages with HASH payloads.

First, the initiator encrypt KEi and Ni with the public key of the responder and add a HASH_1 payload that contains a hash function applied to the all message encrypted with its private key to protect the integrity of the message.

$HASH\_1 = \{H(SAi_1||KEi||Ni||HITi||Ts)\}_{PR\_I}$

Once received the responder uses its private key to decrypt KEi and Ni payloads. Moreover, it uses the public key of the initiator to decrypt HASH_1 and therefore obtain $H(SAi_1||KEi||Ni||HITi||Ts)$. After that the responder applies the same hash function on $(SAi_1||KEi||Ni||HITi||Ts)$ and compares it with the previous one. If correct, it means that the M1 comes from a legal initiator and there is not an attacker. If not, the responder must terminate this exchange.

Such a treatment is applied to the second message. The responder encrypts KEr and Nr with the public key of the initiator and adds a HASH_2 payload that contains a hash function applied to the message encrypted with its private key to protect the integrity of the message.

$HASH\_2 = \{H(SAr_1||KEr||Nr||[CERTEQ]||HITr||puzzle||Ts)\}_{PR\_R}$

Once received the initiator uses its private key to decrypt KEr and Nr payloads. Moreover, it uses the public key of the responder to decrypt HASH_2 and therefore obtain

$H(SAr_1||KEr||Nr||[CERTEQ]||HITr||puzzle||Ts)$.

After that the initiator applies the same hash

function on H(SAr_1, KEr, Nr,[CERTEQ], HITr, puzzle, Ts) and compares it with the previous one. If correct, it means that the second message of this initial exchange comes from a legal responder and there is no attacker. If not, the initiator must terminate this exchange.

So, this HIP_IKEv2 protocol is able to defend against man in the middle attack by using public key cryptography mechanism, since with the private key of the initiator the attacker is unable to intercept.

3.3 Defend DoS Attack

As mentioned before although IKEv2 is designed to improve IKEv1 and solve related security problems, it still faces the Dos attack.

But unlike other attacks, Dos is a matter of degree, so no protocol can completely protect against Dos attack. However, we try to give a solution more secure against the original one which is basically the use of cookies negotiation.

The cryptographic puzzle exists to protect the responder from Dos attacks. Hence, the idea is to benefit from the Host Identity Protocol which uses a cryptographic puzzle in its base exchange.

Before committing resources, the responder should ask the initiator to solve a cryptographic puzzle. To require a correct solution of the cryptographic puzzle before allocating resources as precondition reduces the attack rate as it is a brute force computation. The puzzle is based on a cryptographic Hash function and it is composed of three components: the puzzle nonce I, the solution J and the difficulty level K. Hence, if the responder wants to defend against a Dos attack, he needs to send a nonce I to the initiator. The initiator is asked to find the solution J for which the K lowest order bit for the binary representation of the result H(I||J) is equal to zero. Hence, the initiator must vary J and apply the hash function to the concatenation of the nonce I with the new J every time until an appropriate solution is found. The difficulty of a cryptographic puzzle depends on the responder and on its level of trust of the Initiator.

Therefore, the puzzle is used to reduce the effect of Denial of Service attack without adding an extra round trip to the exchange. We see that the HIP_IKEv2 consists of four messages only and not six as is the case in the solution of cookies negotiation.

3.4 Defend Replay Attack

Since Timestamping is a way of preventing the replay attack, we propose to add a time stamp payload to all HIP_IKEv2 messages. So the initiator and the responder accept only messages for which the timestamp is included within a reasonable tolerance. The advantage of this method is that there is no need to generate random numbers.

3.5 Integrity of Messages

The HASH payload included in the four messages of HIP_IKEv2 exchange assures the integrity of the messages. The HASH payload presents always a hash function applied to all messages encrypted with its private key. Hence, when the communication node receives the message, it can calculate the Hash value, and then compare the result with the HASH payload after decrypting it.

4 FORMAL SPECIFICATION AND VALIDATION OF THE HIP_IKEv2

We have chosen to verify the HIP_IKEv2 using the "Automated Validation of Internet Security Protocols and Applications" AVISPA, since it is the most effective tools according to many comparatives studies (Cheminod et al., 2009; Lafourcade et al., 2010).

4.1 AVISPA and SPAN

AVISPA is a push-button tool for building and analyzing security protocols. The AVISPA Tool is equipped with a web-based graphical user interface (www.avispa-project.org/software) that supports the editing of protocol specifications and allows the user to select and configure the different back-ends of the tool (Armando, 2005).

AVISPA provides a modular and expressive formal language called the High Level Protocol Specification Language (HLPSL) for specifying intended protocols and formally validating them (Lim et al., 2007).

In order to help protocol designers in designing and debugging HLPSL specifications, a new feature "Security Protocol Animator" (SPAN) (Cheminod et al., 2009) was created to facilitate the specification phase by allowing the animation of the language HLPSL (Armando et al., 2005).

We specify the HIP_IKEv2 between the initiator and the responder with HLPSL language. Two different agents have been defined in the modeling

process: I which acts as the initiator and R which acts as the responder.

4.2 Intruder Model

The AVISPA tool assumes that the protocol messages are exchanged over a network that is under the control of the Dolev_Yao (DY) intruder model. This intruder has many capacities over the communication channel. Hence, it can read all messages exchanged between the agents and written in the channel. It can also derive new messages from its initial knowledge and the messages received from honest principals during protocol runs. To derive a new message, the intruder can encrypt and decrypt messages, compose and decompose, in case he knows the key. The knowledge of the intruder is declared in the environment role, which is the top level role. In our protocol, the intruder knows the two communicating agents (I, R), their public keys (PK_I, PK_R), and the solution of the puzzle (sol). It can compute the Hash payload if it knows the key and finally it possesses a public and private key (PK_i, PR_i). So, the intruder knowledge is summarized as follow:

```
intruder_knowledge= {I, R, PK_I,
HITi, HITr, PK_R, soln_, Hash, PK_i,
PR_i}
```

4.3 Security Goals

For security goal we are able to check the mutual authentication of the agents and the secrecy of Diffie-Hellman shared key.

4.3.1 Mutual Authentication

The two agents are authenticated on the Diffie-Hellman shared key. The witness and request events are goals related to authentication. So we have modeled this goal in HLPSL as shown below:

```
role initiator (I,R: agent,……)
played_by  I
transition:
.
.
/\ request (I,R,SK1,SK)

role responder (I,R: agent,……)
played_by  R
transition:
.
.
/\ witness (R,I,SK1,SK)
```

Then, in the goal section of the protocol, we write:

```
authentication_on SK1
```

That is, the initiator requests a check of the shared key agreed with the responder and identified by SK1.
The same procedure is adopted to authenticate the initiator by the responder. So we have modeled this goal in HLPSL as shown:

```
role responder (I,R: agent,……)
played_by  R
transition:
.
.
/\ request (R,I,SK2,SK)

role initiator (I,R: agent,……)
played_by  I
transition:
.
.
/\ witness (I,R,SK2,SK)
```

Then, in the goal section of the protocol, we write:

```
authentication_on SK2
```

That is, the initiator requests a check of the shared key agreed with the responder and identified by SK2.

4.3.2 Secrecy of the shared key between I and R

The Diffie-Hellman shared key must only be known by authenticated entities: the initiator and the responder. The secret is the goal fact related to secrecy.

This goal has been modeled in HLPSL as follows:

```
role initiator (I,R: agent,……)
played_by  I
transition:
.
.
/\ secret (SK, sec_a_SK,{I,R})

role responder (I,R: agent,……)
played_by  R
transition:
.
.
/\ secret (SK, sec_b_SK,{I,R})
```

In the goal section of the protocol, we just write:

```
secrecy_ofsec_a_SK, sec_b_SK,
```

That is, I declares the SK key as a secret shared between I and R and identified by sec_a_SK. Moreover, R declares it as a secret shared between R and I; it is identified by sec_b_SK when sec_a_SK and sec_b_SK are declared as a protocol_id.

4.4 Verifying the HIP_IKEv2 Protocol

We check the specification by the SPAN tool. For our verification, we have used the OFMC and the CL-AtSe back-end to search for the attacks on the tunnel IPSec arrangement. The output gives details about whether the specification is safe or not. If not then it also gives the trace of the attack found, to indicate secrecy attack or authentication attack. So even though many properties of the protocol are to be checked, only a few can be verified using SPAN tool. Only authentication and secrecy goals are supported by AVISPA and SPAN.

As already mentioned HIP_IKEv2 was tested using OFMC and CL-AtSe verification techniques, which assured its security. No attacks or vulnerabilities were found.

Figures 2 and 3 demonstrate the messages returned by OFMC and CL-AtSe verification technique, respectively. As shown in the two figures below, the proposed solution is safe to use and no attacks were found.

Figure 2: Message Returned by AVISPA for HIP_IKEv2 with OFMC.

4 CONCLUSIONS

HIP_IKEv2 protocol makes security modification to the initial exchange of IKEv2 protocol and then integrates this modification with the HIP protocol to

Figure 3: Message Returned by AVISPA for HIP_IKEv2 with CL-AtSe.

enhance security and establish a security association based on location and identity. HIP_IKEv2 has effectively solved some secure problems such as man in the middle, Dos and replay attack. It also assures the integrity of messages. The solution has been validated with the Automated Validation of Internet Security Protocols and Applications (AVISPA) and its Security Protocol Animator (SPAN), which proves that the solution is safe and no attacks were found.

Future works will be focused on trying to give solution against some other types of attacks such as connection hijacking, and impersonation.

REFERENCES

Gurtov, A., 2008. *Host Identity Protocol (HIP): Towards the Secure Mobile Internet*. Helsinki Institute for Information Technology (HIIT), Finland, Wiley

Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., 2010. Internet Key Exchange Protocol Version 2 (IKEv2), *IETF RFC5996; www.rfc-editor.org/info/rfc5996*

Henderson, T., Gurtov, A., 2012. The Host Identity Protocol (HIP) Experiment Report, *IETF RFC6538; www.rfc-editor.org/info/rfc6538*

Arraez, L., Chaouchi, H., G.Ayadin, Z., 2011. "Performance Evaluation and Experiments for Host Identity Protocol", *IJCSI International Journal of Computer Science Issues*, Vol 8, Issue 2, pp 74 - 83

Iso-Anttila, L., Ylinen, J., Loula, P., 2007. "A Proposal to Improve IKEv2 negotiation", *International Conference on Emerging Security Information Systems and Technologies* (IEEE ICESIST), pp 169 - 174.

Xiaowei, Z., Zhou, H., Jun, L., 2010. "Analysis and improvement of IKEv2 against denial of service attack", *International Conference on Information Networking and Automation* (IEEE ICINA), pp 350 - 355.

Zhou, P., Qin, Y., Xu, C., Guan, J., Zhang, H., 2010. "Security investigation and enhancement of IKEV2

protocol", *3rd International Conference on Broadband Network and Multimedia Technology* (IEEE IC-BNMT), pp 65 - 69.

Cheminod, M., Bertolotti, I., Durante, L., Sisto, R., Valenzano, A., 2009. "Tools for cryptographic protocols analysis: A technical and experimental comparison", *Elsevier Computer Standards & Interfaces*, Vol 31, Issue 5, pp 954 - 961

Lafourcade, P., Terrade, V., Vigier, S., 2010. "Comparison of Cryptographic Verification Tools Dealing with Algebraic Properties", *Springer Formal Aspects in Security and Trust*(FAST), Vol 5983, pp 173-185.

Armando, A., Basin, D., Cuellar, J., Rusinowitch, M., Viganò, L., 2005. "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications", *Springer Computer Aided Verification*(CAV), Vol 3576, pp 281- 285.

Lim, S., Bang, K., Yi, O., Lim, J., 2007. "A Secure Handover Protocol Design in Wireless Networks with Formal Verification", *Springer Wired/Wireless Internet Communications*(WWIC), Vol 4517, pp 67 - 78.