An Improved Password Based EAP Method for WiMAX with Formal Verification

Anjani K. Rai
CSED
MNNIT
Allahabad, India

Vimal Kumar
CSED
MNNIT
Allahabad, India

Shivendu Mishra
CSED
MNNIT
Allahabad, India

ABSTRACT
IEEE 802.16e supports EAP (Extensible Authentication Protocol) for authentication, but do not specify the EAP method required for authentication. EAP-SPEKE and EAP-SRP are the strongest password based EAP methods. This paper examines these EAP methods and proposes an efficient password based authentication protocol for WiMAX. Proposed protocol is an improvement of EAP-SPEKE protocol and supports mutual authentication and key derivation. Protocol is verified using Automated Validation of Internet Security Protocols and Applications (AVISPA) which is a push button tool for the automated validation of security protocol and result shows that it does not have any security flaws. Proposed protocol uses only three message exchange for authentication and key derivation. Therefore, the number of exchanged message decreases by one and two compared with the EAP-SRP and EAP-SPEKE respectively.

General Terms
Security, Verification

Keywords
Authentication, AVISPA, EAP-SPEKE, EAP-SRP, IEEE802.16e, SPAN, WiMAX.

1. INTRODUCTION
IEEE 802.16 standard offers large bandwidth and high transmission speed to specify air interface of Wireless Metropolitan Area Network (Wireless MAN). IEEE 802.16-2004 [1] [2] is an amendment in IEEE 802.16, also known as WiMAX which is a forum promoting the IEEE 802.16 standard. IEEE 802.16e amendment [3] adds mobility functionality in IEEE 802.16. IEEE 802.16e support EAP (Extensible Authentication Protocol) for authentication. EAP provides the framework for authentication, but do not specify the mechanism by which to achieve authentication. In order to a client (supplicant) to establish a secure network connection with a Network Access Server (authenticator) such as a Wireless Access Point, it is necessary for the supplicant and the authenticator to authenticate each other. This means that there needs to be some method for verification of supplicant and authenticator to protect from non-repudiation and man-in-the-middle attack.

EAP (Extensible Authentication Protocol) is a universal authentication framework defined by RFC 3748 and supported by IEEE802.11i and IEEE 802.16e. The particular authentication methods supported by EAP are called EAP methods. The different vulnerabilities alleviated by EAP method are detailed in RFC 3748[4]. Originally EAP was developed for use with PPP or Point-to-Point Protocol connections and was later adapted for use by wired and then wireless IEEE 802 networks. In all of these situations, it is possible for an attacker to gain access to links over which EAP packets are transmitted. An attacker with access to the link may launch denial of service attacks, discover user identities, spoof EAP packets, recover passwords using a dictionary attack, and convince the peer to connect to an unsecured network by launching a man-in-the-middle attack as well as other types of attacks [4]. To avoid these kinds of attacks it is extremely important that the EAP method that is chosen is able to provide secure authentication so that a secure PMK (Pair-wise Master Key) can be established between the supplicant or user and the network access point (authenticator). EAP-SRP (EAP-Secure Remote Password protocol) [5] [6] and EAP-SPEKE (Simple Password-Authenticated Exponential Key Exchange) [7] are password authenticated key exchange protocols resist all the well-known passive and active attacks over the network

In our previous paper [22], we have analyzed the EAP-SRP and EAP-SPEKE extensible authentication protocol and proposed an efficient Password Authenticated Key exchange protocol. We refer to our proposed protocol as improved EAP-SRP protocol. This paper proposes an efficient password based authentication protocol which is an improvement of EAP-SPEKE protocol and is compliant with RFC 4017(an unofficial standard for EAP method requirement for wireless network). Formal verification is the methods for verification of security protocols to get the user confidence. There are many tools available for verification of the protocols. These tools are: NRL protocol analyzer [8], Murphi [9], Isabelle [10], ProVerif [11] [12] and Scyther [13]. CasperFDR [14] [15] and automated validation of internet security protocols and applications (AVISPA) [16] are the well known advanced tools AVISPA is a push button tool for the automated validation of security protocol. A modular and expressive formal language called HLSPL (High Level Protocols Specification Language) is used by AVISPA to specify the security protocol and their properties. A large number of protocols, including several variants of generic protocols like Kerberos and EAP have already modeled in HLPSL [17]. SPAN [18] interactively produces the Message Sequence Charts (MSC) [19] from an HLPSL speciation. Attack Simulation is a mode of SPAN for automatic building of MSC attacks. Analysis and verification of propose protocol has done using AVISPA and SPAN, result shows that protocol has no security flaws. Further, the number of exchange message decreases by one and two compared to EAP-SRP and EAP-

SPEKE respectively and hence is a very fast authentication technique.

The paper is organized as follows. Section 2 gives a basic overview of extensible authentication protocol with different password based EAP methods. Section 2 also describes the proposed protocol with security analysis. Section 3 verifies the protocol using AVISPA and SPAN and Section 4 concludes the paper.

2. OVERVIEW OF EXTENSIBLE AUTHENTICATION METHOD

Extensible Authentication Protocol (EAP) is an authentication framework which supports multiple authentication methods. EAP does not require IP and it runs over data link layers such as Point- to-Point Protocol (PPP) or IEEE 802. EAP method that is used in wireless network should be compliant with RFC 4017[20]. RFC 4017 is considered as an unofficial standard for the extensible authentication method that is used in wireless network. The requirements that are outlined in RFC 4017 can be summarized as:

All the EAP method must generates a master session key during authentication; the strength of generated key must be at least 128 bits of key strength. Any EAP method used in wireless network must provide mutual authentication. Method must resist dictionary attack and offline attack. Any EAP method used in wireless network may support fragmentation, reassembly, channel binding, end user identity and fast reconnect.

Two types of EAP model is specified in [4] [21] they are: pass through behavior model and multiplexing model. Pass through behavior model, consist three entities authentication, i.e. Supplicant, Authenticator, and Authentication Server, all of them exist in three separated devices. Supplicant exists in wireless client station, authenticator resides in access points, and authentication server resides in AAA (Authentication, Authorization, and Accounting) Authenticators, such as RADIUS and DIAMETER. Authenticator acts as a pass-through device.

Multiplexing model consist only two separated devices, where authenticator and authentication server exist in a single device. The authenticator will implement all authentication services. In this paper we have considered the multiplexing model where supplicant resides in MS/SS and authenticator resides in BS for wireless network.

The specific authentication mechanism that is supported by EAP is called EAP method. There are over three dozens EAP methods exist out of which EAP-SPEKE and EAP-SRP are the very strongest EAP methods. These methods are based on password that is secret between supplicant and authenticator. These methods prevent active network attacks (man-in-the-middle, replay, etc.) as well as passive attack. These Methods also prevents from password-sniffing and unconstrained brute force attack.

2.1 EAP-SRP

EAP-SRP stands for EAP-Secure Remote Password protocol [5] [23] [24] [25] is a member of the class of strong authentication protocols that defend against all the well-known passive and active attacks over the network. The SRP borrows some elements from other key-exchange protocol and adds some modification.

EAP-SRP may be considered as an alternative to TLS. The SRP uses User ID and password-based authentication which is easier to deploy than certificate-based authentication in organizations. EAP-SRP also generates a strong session key that is use for further encryption of application data. The procedure of EAP-SRP is shown in figure 1 and can be summarized as: p is a large prime in the form of ($p = 2q+1$, where q is prime), g is a generator modulo p, A, B is each public values of supplicant and authenticator respectively, Salt is Authenticator's salt, ID is identifier of supplicant, pd is Password, H () is strong One-way hash function, u is a 32-bit unsigned integer which takes its value from the first 32 bits of the SHA1 hash of (A, B), a, b is each Secret values. x is private key (derived from pd and Salt), v is the Password verifier. The authenticator stores passwords using the following formula: $x = H$ (salt, Pd). Authenticator computes Password verifier $v = g^{x} \bmod p$ and keeps (ID, Salt, v) in its password database. Now the EAP-SRP involves the following steps:

Supplicant computes $A = g^{a} \bmod p$ and sends (A, ID) to authenticator. Authenticator computes $B = (v + g^{b}) \bmod p$ and sends (Salt, B) to supplicant. Now supplicant calculates $x = H$ (Salt, Pd) and both supplicant and authenticator calculates $u = H$ (A, B), Supplicant also calculates $x = H$ (Salt, Pd) and then using x and u, calculates the session key $S = (B - g^{x})^{(a+ux)} \bmod p$ and then strong session key $K = H(S)$. Authenticator also calculates the same session key that is $S = (A\, v^{u})^{b} \bmod p$ and then both party computes strong session key $K = H(S)$.

Now the two parties have a shared, strong session key K and they can assure each other about possession of key as: Supplicant calculates Ma and sends it to authenticator. Authenticator calculates Mb and sends to supplicant. Where Ma= H (H (Pd) xor H (g), H (ID), Salt, A, B, K) and Mb = H (A, Ma, K).

EAP-SRP uses User ID and password-based authentication which is easier to set up than certificate-based authentication which causes long handshake latency and certificate management overheads associated with public key infrastructure [26].EAP-SRP protocol also has the added advantage of permitting the host to store passwords in a form that is not directly useful to an attacker. Even if the host's password database were publicly revealed, the attacker would - still need an expensive dictionary search to obtain any passwords. Guessing the password by an attacker is not easy since it requires much time than the hash currently used by most UNIX systems [26].

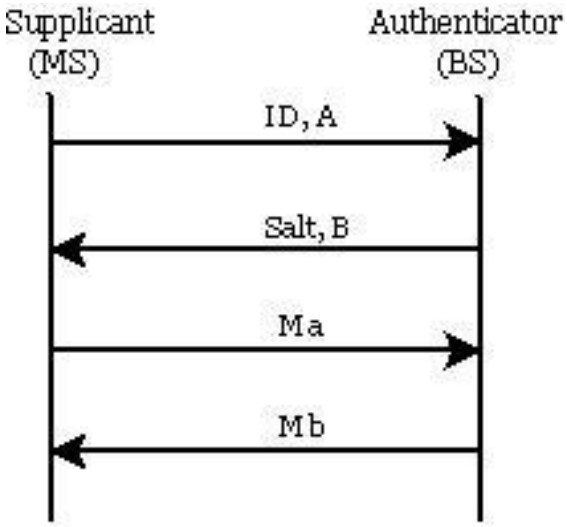

Figure 1. Message flow in EAP-SRP protocol

2.2 EAP-SPEKE

EAP-SPEKE (Simple Password-Authenticated Exponential Key Exchange) is an EAP method that is based on Diffie-Hellman key exchange. In this method a simple password is integrated with standard Diffie-Hellman to protect from malicious user [4].

In multiplexing model where the supplicant and authenticator exist, overall procedure of EAP-SPEKE involves the following steps: pd is a small common password for the supplicant and the authenticator. N= kM+1 is a suitable large prime number for Diffie- Hellman, where (N − 1)/2 is also prime. M is a large prime factor of N-1. h () is strong one way hash function. g is a suitable Diffie- Hellman base, either large prime order or primitive. f (pd) is a function that converts pd into a suitable Diffie-Hellman base(g). f (pd) = pd^k mod N. Xa, Ya are secret random numbers selected by supplicant and authenticator respectively. Supplicant sends exponential value A to authenticator and authenticator sends exponential value B to supplicant. EAP-SPEKE has two steps. In first stage both

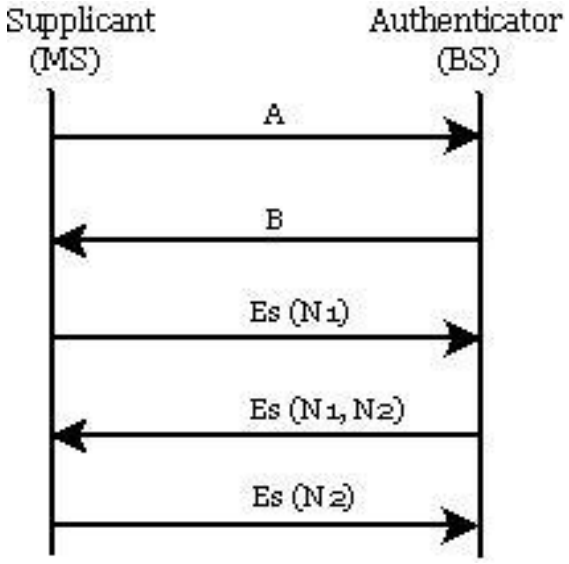

Figure 2. Message flow in EAP-SPEKE protocol

supplicant and authenticator negotiate a session key and in second stage they assure each other about the possession of same key.

Step 1: Supplicant calculates A= g^{Xa} mod N and send it to authenticator.

Step 2: Authenticator calculates B= g^{Ya} mod N and send it to supplicant.

Step 3: Both supplicant and authenticator calculates the secret: S= (B^{Xa} mod N) and S =h (A^{Ya} mod N) respectively, at this stage both have the same secret key S.

To assure each other about the possession of same key S,

Step 4: supplicant selects random nonce N1 and encrypt it with secret key S and sends ES (N1) to authenticator.

Step 5: Authenticator selects random nonce N2 and encrypts it with same secret key S and sends ES (N1, N2).

Step 6: Supplicant verifies that N1 is correct and encrypts N2 with secret key S and sends ES (N2) to authenticator

Authenticator verifies that N2 is correct and now both have mutually authenticated to each other and both consist the same secret key S.

In EAP-SPEKE, the password is too small but any passwords or password-crackable data does not send over the network, also the method integrates the act of authentication with session key negotiation therefore protocol prevent active network attacks (man-in-the-middle, replay, etc.) as well as passive attack. Method also prevents from password-sniffing and unconstrained brute force attack.

2.3 Proposed Improved EAP-SPEKE Protocol

This section describes the proposed improved EAP-SPEKE protocol. Figure 3 shows the message flow of the proposed protocol in wireless network where Supplicant resides in mobile station and authenticator resides in access point (BS). Before the protocol starts, both the supplicant and authenticator set up several parameters. pd is a small common password for the supplicant and the authenticator. N =kq+1 is a suitable large prime number for Diffie- Hellman, where (N − 1)/2 is also prime. q is a large prime factor of N-1. h () is strong one way hash function. Kmb is a suitable Diffie- Hellman base, either large prime order or primitive. f (pd) is a function that converts pd into a suitable Diffie-Hellman base (Kmb). f (pd) = exp (pd, k) mod N. Na, Nb are secret random numbers selected by supplicant and authenticator respectively. Now the protocol involves following steps-

Step 1: Supplicant (MS) computes A= exp (Kmb, Na) mod N and selects a random nonce Ca and sends (A, ID of MS, Ca) to authenticator (BS).

Step 2: Authenticator (BS) computes B= exp (Kmb, Nb) mod N and K= exp (A, Nb) mod N, authenticator also selects a random nonce Cb and sends (B, {Ca, Cb} _ K) to supplicant (MS).Here {Ca, Cb} _ K is the encryption of Ca, Cb using symmetric key K.

Step 3: Supplicant (MS) computes K= exp (B, Na) mod N= exp (A, Nb) mod N and decrypts {Ca, Cb} _ K, since only-

authenticator has secret key K therefore received Ca confirms that the intended sender is authenticator.

Supplicant also sends Cb by encrypting it symmetric key K and authenticator decrypts it with the same key which confirms that intended sender is supplicant (MS). Now the mutual authentication between supplicant and authenticator has done and both parties have the same secret key K.

Proposed method performs mutual authentication and key agreement across an untrusted network while protecting passwords and negotiated authenticated keys. This method does not send any passwords or password-crackable data over the network; instead they integrate the password into a standard Diffie-Hellman exchange in a way that negotiates a mutually authenticated key. Method provides zero-knowledge proof of a password (ZKPP) which prevents unconstrained guessing from network attackers, prevent disclosure of password to wrong or spoofed server and performs safe mutual authentication. Integrated Diffie-Hellman key exchange provides forward and backward secrecy. Proposed method is a balanced password protocol, in that both parties share identical password-derived data and is scalable to EC group. This methods prevent passive and active network attacks (man in- the-middle, replay, etc.), as well as password-sniffing and unconstrained brute force attack from the network. The method is as efficient as a Diffie-Hellman key exchange (DH) computation, using any standard groups. Method is compatible with standard Diffie-Hellman as described in [IEEE 1363] and [ANSI X9.42], and is also aligned with the emerging IEEE [P1363.2] standard for password-based cryptography [7].

Proposed method computation involves only two symmetric key encryption and decryption whereas EAP-SPEKE requires three symmetric key encryption and decryption.).

Proposed protocol uses password-only credentials and stores only an ID parameter in its password database While EAP-SRP host needs to store ID, s, and v parameters in its password database therefore proposed method requires lesser amount of disk space compared to EAP-SRP.

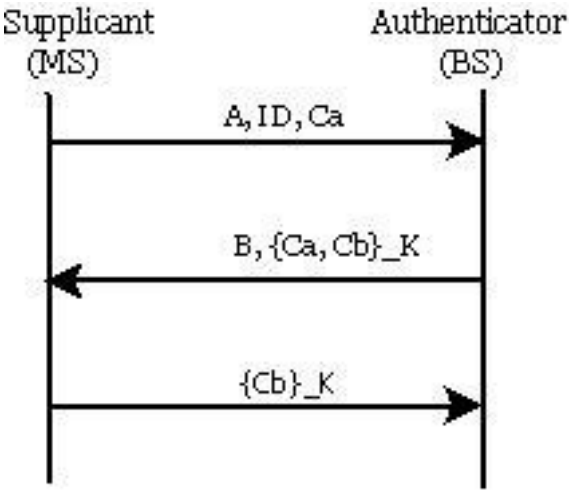

Figure 3. Message flow in proposed improved EAP-SPEKE protocol

Before the beginning of protocol, the server and client computes one modulo exponentiation. Once the protocol begins the server and client computes another exponentiation to perform mutual authentication. However, EAP-SRP needs to compute two modulo exponentiation during the protocol. Also, the number of exchange message reduces by one and two compared to EAP-SRP and EAP-SPEKE respectively, therefore method improves the performance of handshake latency (authentication delay). Also proposed method confirms the entire requirement for wireless network discussed above. Therefore proposed protocol is suitable for IEEE standard 802.16 (WiMAX).

3. FORMAL VERIFICATION AND VALIDATION OF PROPOSED PROTOCOL

3.1 AVISPA

AVISPA [16] is a tool for the automated validation of security protocol. No other tool exhibits the same scope and robustness while enjoying the same scalability and performance. In particular, the AVISPA Tool has detected a number of previously unknown attacks on some of the protocols analyzed, e.g., on the IKEv2 protocol with digital signatures, on some protocols of the ISO-PK family, ASW protocol, on the SET protocol and on the H.530 protocol. A modular and expressive formal language called HLSPL (High Level Protocols Specification Language) is used by AVISPA to specify the security protocol and their properties. A large number of protocols, including several variants of generic protocols like Kerberos and EAP have already modeled in HLPSL. HLPSL is a role-based language, meaning that we first specify the sequence of actions of each kind of protocol participant in a module, which is called a basic role. This specification can later be instantiated by one or more agents playing the given role, and we further specify how the resulting participants interact with one another by "gluing" multiple basic roles together into a composed role. The structure of the AVISPA tool is shown in Figure 4. A HLPSL specification is translated into the Intermediate Format (IF), using a tool called hlpsl2if. Note that this intermediate step is transparent to the user, as the translator is called automatically. The intermediate format specification is then processed by model-checkers to analyze if the security goals are violated. There are four different verification back end tools use to analyze the IF specification. These tools are: OFMC (On-the-Fly Model- Checker), CL-AtSe (Constraint-Logic-based Attack Searcher), SATMC (SAT-based Model-Checker), TA4SP (the Tree Automata tool based on Automatic Approximations for the Analysis) of Security Protocols. Possible flaws in a protocol can be identified using these back end tools. OFMC employs symbolic techniques to perform bounded analysis and protocol falsification. OFMC provides a translation which is use to find attack (if exist) in any protocol. Translation and checking are fully automatic and Performed by OFMC without use of external tool. SPAN interactively produces the Message Sequence Charts (MSC) from an HLPSL specification. Attack Simulation is a mode of SPAN for automatic building of MSC attacks from the output of OFMC tool. We uses OFMC back end tool with AVISPA and SPAN to analyze the proposed protocol.

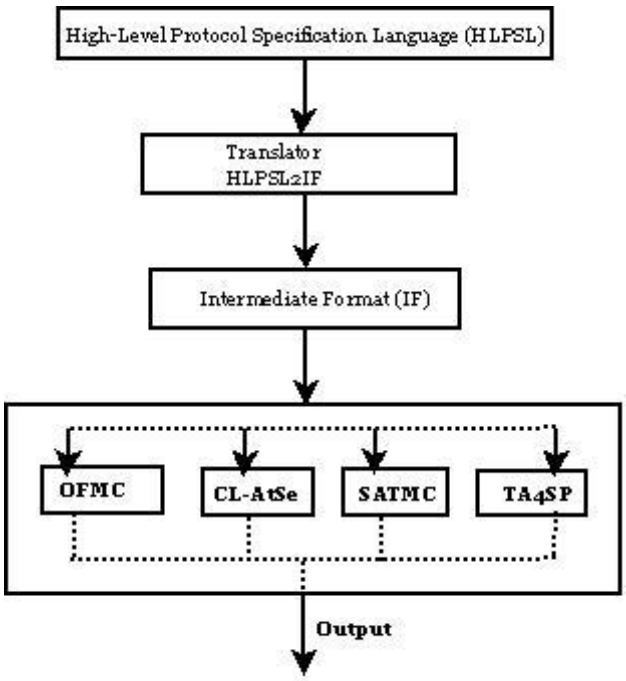

Figure 4. AVISPA tool v.2 architecture

3.2 Specification and Verification of Proposed Protocol Using AVISPA and SPAN

We use two basic roles ispeke_Init played by MS and ispeke_Res played by BS. Each basic role consist the initial information known by the participant, its initial state and the transition by which state can change. After defining the basic roles, we have to define composed roles describing the sessions of the protocol. Finally a top level role "environment role" containing global constant, a statement describing the initial knowledge of intruder and composition of one or more session is defined. Each role communicates with other role using channel and they are independent from each others. SPAN is use to symbolically execute the HLPSL protocol specification and hence provides a better understanding of the specification, SPAN also checks that the protocol is executed and it corresponds to expected output. Fraction of HLPSL syntax used for proposed protocol is shown below-

```
role ispeke_Init (MS,BS: agent,
                  Kmb: symmetric_key,
                  Snd,Rcv: channel(dy))
played_by MS
def=
local  State: nat,
Na,Nb: text,
Ca, Cb   : text,
K,X  : message
const  ca,cb, sec_i_Cb : protocol_id
init   State := 0
transition
1. State = 0 /\ Rcv(start) =|>
   State':= 1/\ Snd(MS.Ca'.exp(Kmb, Na'))
2. State  = 1 /\ Rcv(X'.{Ca.Cb'}_exp(X',Na)) =|>
   State':= 2 /\ Snd({Cb'}_K')
             /\ secret(Cb',sec_i_Cb,{MS,BS})
             /\ witness(MS,BS,cb,Cb')
             /\ request(MS,BS,ca,Ca)
end role
role ispeke_Resp (BS,MS: agent,
                  Kmb: symmetric_key,
                  Snd,Rcv: channel(dy))
played_by BS
def=
local State: nat,
      Nb,Cb: text,
      Ca   : text,
      Y,K  : message
const ca,cb,sec_r_Cb : protocol_id
init  State := 0
transition
1. State  = 0 /\ Rcv(MS.Ca'.Y') =|>
  State':= 1   /\ Snd(exp(Kmb, Nb').{Ca'.Cb'}_K')
               /\ secret(Cb',sec_r_Cb,{MS,BS})
               /\ witness(BS,MS,ca,Ca')
2. State  = 1 /\ Rcv({Cb}_K) =|>
   State':=2   /\ request(BS,MS,cb,Cb)
end role
role environment()
def=
const i, ms, bs         : agent,
      kmb, kmi, kbi: symmetric_key,
      ca, cb       : protocol_id
intruder_knowledge = {ms, bs, kmi, kbi}
goal
secrecy_of sec_i_Cb, sec_r_Cb
authentication_on cb
authentication_on ca
end goal
environment()
```

As shown above each role has a number of parameters. MS, BS are of type agent, Kmb is of type symmetric key. The Snd and Rcv parameters are of type channel, indicating that these are channels through which the agent playing role ispeke_Init and ispeke_Res will communicate. The attribute to the channel type, in this case (dy) for Dolev- Yao, denotes the intruder model to be considered for this channel. The parameter MS and BS appears in the played by section, which means that MS denotes the name of the agent who plays the role of ispeke_Init and BS denotes the name of agent who plays the role of ispeke_Res. The local section declares local variables of ispeke_Init and ispeke_Res, such as the local variable called State, which is a natural number and is initialized to "0" in the init section.

The transition section of HLPSL specification for proposed protocol contains a set of transitions. Each transition represents the receipt of a message and the sending of a reply message. A transition consists of a trigger, or precondition, and an action to be performed when the triggering event occurs. Generated nonce Cb is kept secret among MS and BS. So in the creating role, we place these lines (where the primes are required there to refer to the new value of Cb): secret(Cb',sec_i_Cb,{MS,BS}), secret(Cb',sec_r_Cb,{MS,BS}) indicating that MS allows that the two values are shared between (only) MS and BS and vice versa. Constant i is used to refer to the intruder. There is a statement which describes the initial knowledge of the intruder. This includes the names of all agents and symmetric keys (kmi, kbi) he shares with others. Finally, we have declared the goal(s) of the protocol, we model the goals of the protocol by labeling several transitions in the HLPSL specification with special events that express the meaning of the transition with respect to for secrecy, the goal facts assert which values should be secret between whom, and the goal declaration in the goal section (e.g., secrecy_of sec_i_Cb, sec_r_Cb) specifies that if the intruder learns a secret value that is not explicitly a secret between him and someone else, then the intruder has successfully attacked the protocol. Similarly, HLPSL provides for the specification of goal facts related to authentication (e.g., authentication_on cb authentication_on ca) which are for instance used to check that a principal is right in believing that his intended peer is present in the current session, has reached a certain state, and agrees on a certain value, which typically is fresh. Figure 5 shows the protocol simulation of proposed protocol.

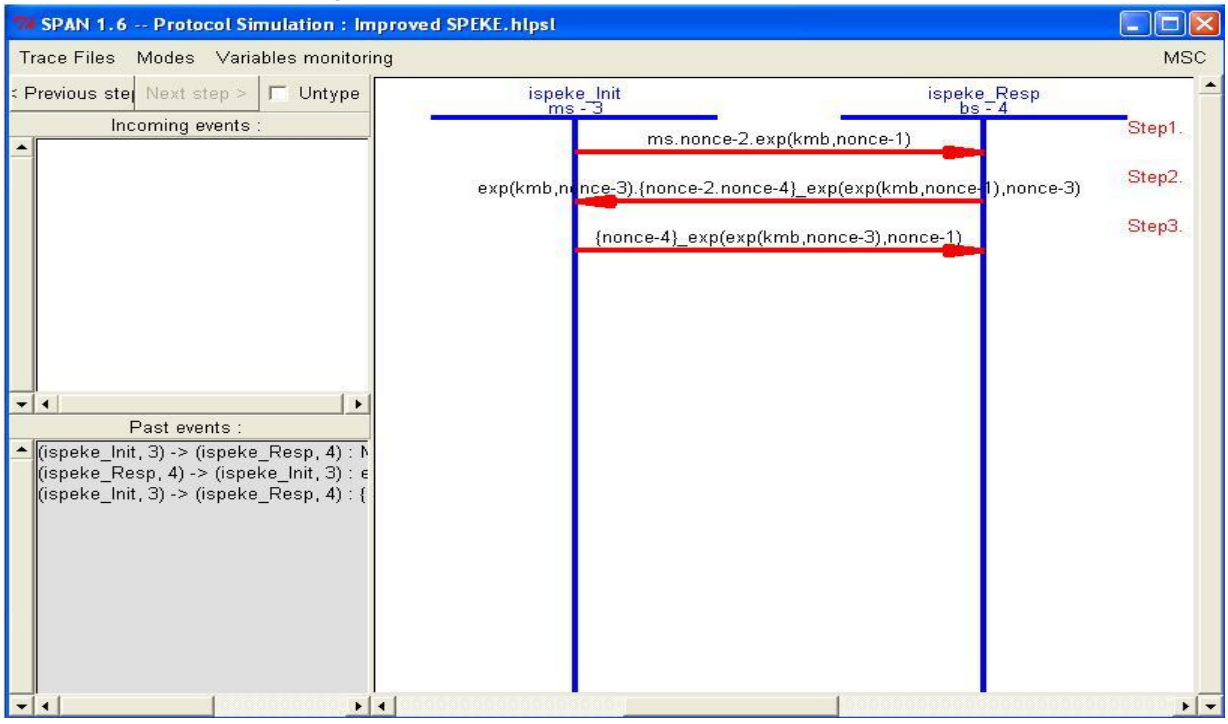

Figure 5. Simulation of proposed protocol using SPAN

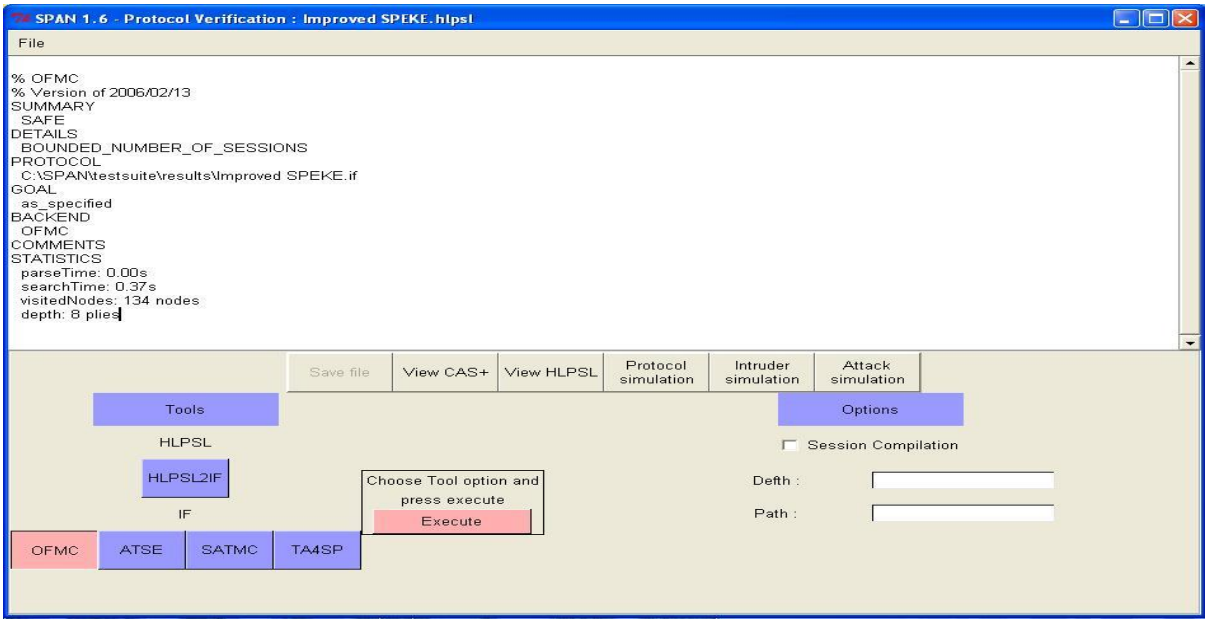

Figure 6. Result obtained for the proposed protocol using SPAN.

After executing the specification, we did not find any attack on the events: secret(Cb',sec_i_Cb,{MS,BS}), witness(MS,BS,cb,Cb'), request(MS,BS,ca,Ca) and secret(Cb',sec_r_Cb,{MS,BS}, witness(BS,MS,ca,Ca') , request(BS,MS,cb,Cb) which shows that protocol is safe against all type of attacks. Figure 6 shows the result of proposed protocol using SPAN.

4. CONCLUSION

This paper proposed an improved password based EAP method for IEEE 802.16 (WiMAX) which is an improvement of EAP-SPEKE protocol. Proposed protocol inherits all the security feature of EAP-SPEKE and provides mutual authentication between MS and BS. Protocol is safe against all type of passive and active attack and with only three message exchanges; supplicant and the authenticator authenticate each other which reduces the authentication delay and encryption/decryption time. Proposed protocol is specified and verified using AVISPA and SPAN, no new vulnerability or attack has been surfaced.

5. REFERENCES

[1] "IEEE std 802.16–2004: Air interface for fixed broadband wireless access system," IEEE, 2004.

[2] IEEE 802.16 and WiMax: Broadband Wireless Access for everyone, Intel White Paper, 2004.

[3] "IEEE std 802.16e–2005: Air interface for fixed broadband wireless access system – amendment: Physical and medium access control layers for combined fixed and mobile operation in licensed bands," IEEE, 2006.

[4] Adoba, B., Blunk, L., Vollbrecht, J., Carlson, J. and Levkowetz, E. 2004. Extensible authentication protocol (EAP). RFC 3748.

[5] T. Wu: The SRP Authentication and Key Exchange System, RFC 2945 (2000).

[6] Su Jung Yu and Joo Seok Song, "An Improved Password Authentication Key Exchange Protocol for 802.11 Environments", V. Kumar et al. (Eds.): ICCSA 2003, LNCS2668, pp. 201–209, 2003

[7] D. Jablon: The SPEKE Password-Based Key Agreement Methods, IETF draftjablon-speke-02.txt (2003).

[8] Meadows C., The NRL Protocol Analyzer: an overview, Journal of Logic Programming, February 1996.

[9] Mitchell J.C. and others, Automated Analysis of Cryptographic Protocols Using Murphi, Proceedings of the 1997 IEEE Symposium on Security and Privacy (1997) pages 141-151, IEEE Computer Society Press.

[10] University Of Cambridge, http://www.cl.cam.ac.uk/ research/hvg/Isabelle/overview.html, updated 12-07-2006.

[11] B. Blanchet. An efficient cryptographic protocol veri_er based on prolog rules. In Proc. CSFW'01, pages 82-96. IEEE Comp. Soc. Press, 2001.

[12] B. Blanchet. Cryptographic Protocol Veri_er User Manual, 2004.

[13] C.J.F. Cremers. The Scyther Tool: Verification, falsification, and analysis of security protocols. In Computer Aided Verification, 20th International Conference, CAV 2008, Princeton, USA, Proc., volume 5123/2008 of Lecture Notes in Computer Science, pages 414-418. Springer, 2008.

[14] Formal Systems (Europe) Ltd, "FDR2 user manual: Failure–divergence efinement," May 2000.

[15] G. Lowe, "Casper: A compiler for the analysis of security protocols," ournal of Computer Security, vol. 6, pp. 53–84, 1998.

[16] Avispa – a tool for Automated Validation of Internet Security Protocols. http://www.avispa-project.org

[17] D6.2: Specification of the Problems in the High-Level Specification Language. http://www.avispa-project.org.

[18] SPAN – a Security Protocol Animator for AVISPA. http://www.irisa.fr/lande/genet/span

[19] D. Harel and P. S. Thiagarajan. Message sequence charts. UML for Real: Design of Embedded Real-time Systems, 2003.

[20] Stanley, D., Walker, J., and Aboba, B. 2005. Extensible authentication protocol (EAP) method requirements for wireless LANs. RFC 4017

[21] Anjani K .Rai, Vimal Kumar, Shivendu Mishra, " Strong Password Based EAP-TLS Authentication Protocol for WiMAX" , Anjani K. Rai et al./(IJCSE) International Journal on Computer Science and Engineering, Vol. 02, No. 02,2010, 2736-2741

[22] Anjani K. Rai, Vimal Kumar, Shivendu Mishra,"An Efficient Password Authenticated Key Exchange Protocol for WLAN and WIMAX" International Conference and Workshop on Emerging Trends and Technology (ICWET), 2011 proceedings by ACM at Mumbai Maharashtra. (Accepted)

[23] T. Wu: The Secure Remote Password Protocol, In Proceedings of the Internet Society Symposium on Network and Distributed Systems Security, San Diego,

[24] D. Taylor: Using SRP for TLS Authentication, IETF draft-ietf-tls-srp-01.txt (work in progress) (2001)CA,(1998)97-111.

[25] D.P. Jablon: Strong Password-only Authenticated Key Exchange, ACM SIGCOMM Computer Communications Review (1996)

[26] David Q. Liu, Mark Coslow, "Extensible Authentication Protocols for IEEE Standards 802.11 and 802.16".