SPIN-based Verification of Authentication Protocols in WiMAX Networks

Beth N. Komu, Mjumo Mzyece and Karim Djouani
Department of Electrical Engineering/French South African Institute of Technology
Tshwane University of Technology, Private Bag X680, Pretoria 0001, South Africa
Tel: +27 (0) 12 382 4191, Fax: +27 (0) 12 382 5294
Email: {bethkomu@gmail.com, mzyecem@tut.ac.za, djouanik@tut.ac.za}

Abstract- The rise in the use of the Internet and other communication technologies like WiMAX which are vulnerable to network attacks has necessitated the implementation of security protocols. Developing secure protocols has proved to be a difficult task as is evident from the presence of flaws in published protocols such as the Needham-Schroeder public key authentication protocol. Security protocols can be validated with formal verification techniques, increasing confidence in their use. The Initial Network Entry procedure in an IEEE 802.16 (WiMAX) network has security defects which can be exploited by the Man-in-the-Middle (MITM) attack, requiring the implementation of a suitable security protocol. In this paper, we analyse a modification of the Diffie-Hellman (DH) key exchange protocol proposed to mitigate this MITM attack and model the protocol and an intruder process with MITM capabilities in PROMELA formalism. We then use Linear Temporal Logic (LTL) to define the attributes the protocol should satisfy and carry out verification by use of the Simple PROMELA INterpreter (SPIN) model checker.

Index Terms— Formal modelling, Formal Verification, Initial Network Entry, LTL, Man-in-the-Middle, SPIN, WiMAX.

1. INTRODUCTION

A. Background

Security protocols, also referred to as cryptographic protocols, aim at providing security-related objectives in potentially hostile network environments such as the Internet and WiMAX networks. The Worldwide Interoperability for Microwave Access (IEEE WiMAX-802.16) is an emerging standard that offers last mile broadband wireless access (BWA). However, like all other wireless networks, WiMAX is vulnerable to network attacks. One such threat is the Man-in-the-Middle (MITM) attack that targets the unencrypted management messages at the Initial Network Entry point both in Fixed WiMAX (802.16d-2004) and Mobile WiMAX (802.16e-2005).

The Initial Network Entry procedure consists of security-sensitive unprotected Medium Access Control (MAC) management messages being exchanged between the Subscriber Station (SS) or Mobile Station (MS) and the Base Station (BS) for ranging, capabilities negotiations, authentication and key exchange and registration purposes. Unfortunately, the Advanced Encryption Standard (AES) implemented by Mobile WiMAX only protects the data messages after the Initial Network Entry procedure, leaving the MAC management messages to be sent in the clear and thus exposing them to the MITM attack [1].

The MITM attack strategically positions itself between the SS/MS and the BS and passively listens to an insecure channel of communication, in this case, the Initial Network Entry procedure. It then creates detailed profiles of the victim SS inclusive of its security settings and associations with the serving BS, imitates the legitimate stations and then modifies the management messages exposing the network to other destructive attacks like replay attacks, masquerade attacks and denial-of-service (DoS) attacks. Eventually, the MITM attack manages to fool the legitimate SS/MS and the BS into operating as if they are still communicating with each other, when in actual fact the intruder is controlling the communication process.

The vulnerability of the Initial Network Entry procedure in WiMAX makes the implementation of a security protocol vital. Although certain security protocols are simple and lightweight, they have still revealed themselves to be error prone, especially because of the difficulty in foreseeing all possible attacks. Consequently, a lot of research is being conducted on the use of formal verification techniques to discover subtle attacks in protocols that may otherwise be difficult to detect using traditional methods based on human inspection and testing.

Model checking [2] is an automatic formal approach for verifying finite state systems. Automatic verification provided by model checkers is not only useful for proving correctness of systems, but also of value in discovering bugs during the design of a new system. Model checking has so far been successfully used to check for correctness in communication protocols, software, and hardware comprising complex sequential circuit designs, significantly increasing the level of confidence in employing such systems. Model checking is gaining popularity due to its automatic system checking process, its quick speed of verifying systems and high efficiency.

B. Related Work

Different tools for formal verification have been successfully implemented to expose vulnerabilities that may still exist in deployed security protocols, especially in networks like WiMAX. The authors of [3] applied the Scyther tool to expose vulnerabilities in the Privacy and Key Management version 1 (PKMv1) and Privacy and Key Management version 2 (PKMv2) authentication protocols implemented in WiMAX. They succeeded in discovering a breach in information confidentiality in both protocols. Similarly, the authors of [4] proposed the use of temporal

978-1-4673-1881-5/12/$31.00 ©2012 IEEE

logic of action (TLA+) to specify the properties of protocols in the authentication and ranging processes in WiMAX and used TLC as the model checking tool to check for denial-of-service (DoS) vulnerabilities. They identified some possible DoS attacks at the initial ranging process but failed to detect a DoS flaw in the PKMv2 authentication protocol using their attacker model. The authors of [5] implemented the Strand space verification technique to uncover the MITM attack in generic cryptographic authentication protocols.

Simple PROMELA INterpreter (SPIN) [6], one of the most powerful general-purpose model checkers, has been extensively used to identify defects in control systems, software systems and security protocols. Process/Protocol Meta-language (PROMELA) is the description language for SPIN used to implement concurrently executing processes in a protocol. In cryptographic protocols, the protocol participants and the behaviour of the intruder process are formalized in PROMELA, while the properties that the protocol should satisfy are specified in Linear Temporal Logic (LTL). Consequently, the protocol is executed in the presence of the modelled intruder process and in case a violation is detected, a counter-example is generated, aiding in the diagnosis of the error and in the improvement of the overall protocol. In [7] and [8] the authors implement, analyse and verify the Needham-Schroeder public key authentication protocol using the SPIN model checker and unearth a Lowe's attack that compromises the authentication and secrecy provided by the protocol. Likewise, the authors of [9] extended the approach used in [7] to analyse and check for correctness in the Helsinki protocol and were successful in discovering authentication vulnerabilities in the protocol.

C. *Focus of this paper*

Most research work done on the security of Mobile WiMAX does not establish any security leaks due to the extra security capabilities integrated in PKMv2 and thus assumes that the Initial Network Entry point is secure. A few authors have analysed and unmasked security holes in the Initial Network Entry process in WiMAX and subsequently proposed various security protocols aimed at mitigating the MITM attack. These protocols are believed to be correct and resistant to malicious manipulation.

In this paper, we focus on one of the few security protocols proposed to preclude the existing vulnerability at the Initial Network Entry point in WiMAX and perform protocol verification using the SPIN model checker to check for existing flaws in the protocol. We successfully uncover MITM vulnerabilities in the proposed protocol.

This paper is organized as follows. Section 2 explains basic and modified versions of the Diffie-Hellman (DH) key exchange protocol. Section 3 presents a PROMELA-based model of the modified DH key exchange protocol, its properties specifications, the modelled intruder process and its associated results. Section 4 concludes the paper.

2. DIFFIE-HELLMAN (DH) KEY EXCHANGE PROTOCOL

A. *Basic Version*

The fact that the DH key exchange protocol [10] supports unauthenticated key agreements between stations wishing to communicate, makes the protocol convenient to implement. The stations need not know each other's identities to establish a shared secret key through exchanging their public key messages in an open channel. However, this poses a threat that can be easily exploited by an attacker in the network. A malicious station can exchange its own public key with a legitimate BS or with a legitimate MS so as to generate the shared key used for encryption purposes. This compromises the security of the network and alterations to the DH protocol are necessary. The authors of [11] introduce the concept of entity authentication of stations willing to participate in a communication process prior to the implementation of the basic DH key exchange protocol as a solution towards mitigating the MITM attack at the Initial Network Entry point in WiMAX networks.

B. *Modified Version*

The implementation of the modified version [11] of the DH protocol involves entity authentication based on the use of cryptographic sealing functions and the International Subscriber Station Identifiers (ISSIs) for every MS so as to fit into the WiMAX environment. The first phase of the protocol is based on a challenge-response mechanism, as illustrated in Figure 1.

An MS claiming to be legitimate receives a challenge (Nb) from the serving base station (BS). The MS computes the solution to the challenge using its cryptographic function and then sends the result and its identity to the BS. The BS confirms the MS's solution and, if correct, it sends an acceptance token as proof of authentication. If the BS receives an incorrect solution, it disconnects from the respective MS. Upon receipt of an acceptance token, the MS sends a challenge (Na) to the BS which calculates the corresponding solution based on the MS's cryptographic function and sends it to the MS. The MS in turn verifies the solution and, if correct, sends back an acceptance token to the BS as proof of successful authentication. Similarly, if the MS receives an incorrect solution, the MS disconnects from that particular BS. Finally, successful mutual entity authentication is achieved.

After successful authentication, the basic Diffie-Hellman key exchange protocol is adopted to generate a shared encryption key with trust having been established between the MS and the BS. The symmetric key generated together with the use of the Vernam Cipher encryption, is used to encrypt the unprotected MAC management messages.

In this model, it is assumed that it is only the legitimate BS and the legitimate MS that have knowledge of the cryptographic function used to compute the challenge sent in the protocol run. Therefore, an attacker in the network is not able to bring forth the correct value to the given challenge and is thus isolated as an intruder to the network.

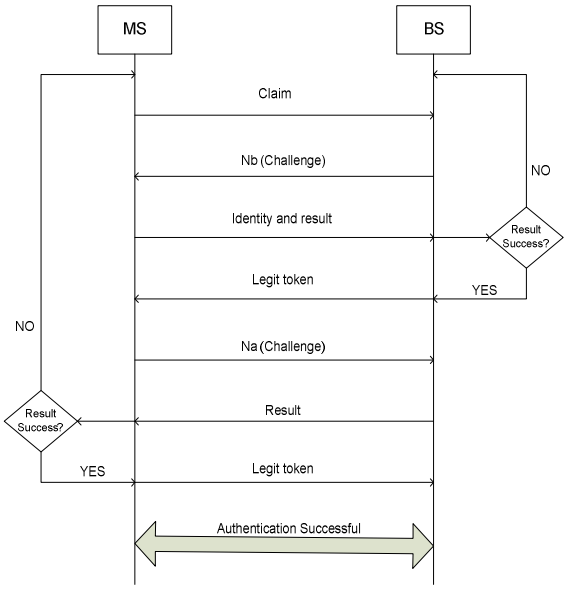

Figure 1: Entity Authentication of principals in the modified DH key exchange protocol

3. METHODOLOGY AND RESULTS

We employ the SPIN model checker tool to check for vulnerabilities in the modified DH key exchange protocol and the iSPIN graphical user interface for visualization of simulation runs.

A. The PROMELA Equivalent of the Modified DH protocol

We first implement the MS and the BS in the proposed protocol as processes in PROMELA and model their protocol sessions and respective assumptions, without interference from an intruder station as illustrated in Figure 2.

In the PROMELA model in Figure 2, the International Subscriber Station Identity (ISSI) as specified in [11] is represented by process identifiers like "0" for the MS and "1" for the BS, which represent the two stations' identities respectively. The cryptographic function implemented herein is set to a constant value only known to the MS and the BS, while the respective nonces (challenges) in the protocol run are assigned different values in each process by invoking a random function. This model of the DH protocol assumes that identity authentication of principals that want to communicate in the WiMAX network is securely achieved through the challenge-response procedure illustrated in Figure 2. The basic DH protocol is then implemented after successful entity authentication at step 146 of Figure 2 so as to establish a shared encryption key that will be used to encrypt the management messages. The Legit token in the respective principals' processes is an indicator that successful authentication has been achieved.
The frame transmitted in the protocol assumes the structure in Figure 3.

Destination		Source		Payload (either Message and/or Integer)
MAC Address	Identity	MAC Address	Identity	

Figure 3: Protocol's Frame Structure

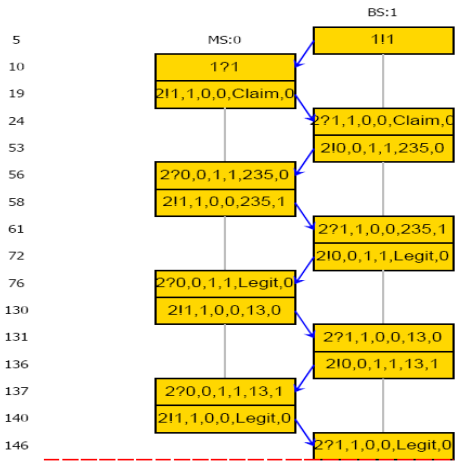

Figure 2: Entity authentication of principals in the modified DH key exchange protocol in PROMELA

B. Defining Correctness Claims

LTL, first introduced by Pnueli in [12], is employed to describe the properties the protocol proposed in [11] should satisfy. The LTL formulae are then included within the PROMELA model as inline specifications as follows.

1) Authentication Checking

We express the fact that the MS and the BS correctly authenticate each other in the challenge-response process of the model in Figure 1 with the following predicates: a BS commits to a session with the MS if and only if the BS receives the correct response to its challenge from the MS, in turn correctly authenticating it. This condition is expressed as *BSCommit.* Moreover, for the MS to be authenticated, it must have accepted to share a protocol instance with the BS by sending a claim, defined by the *MSRunning* variable in the implementation of the protocol. A similar predicate expresses the reciprocal property, that a BS was correctly authenticated by the MS expressed as *MSCommit*. Similarly, for the BS to be authenticated, it must have received a challenge from the MS, defined by the *BSRunning* variable in the implementation of the protocol. To check for authentication of MS to BS and vice versa, the precedence property using LTL is expressed as:

[] ((! MSCommit) U (! BSRunning) U (! BSCommit) U (MSRunning))

For correct mutual authentication to be achieved, the above LTL formula should always hold during the verification process of the protocol.

2) Secrecy Checking

The confidentiality criteria of messages in the protocol run between the legitimate MS and the BS are validated in the presence of an intruder process by checking that the result generated by both the MS and the BS through the implementation of a basic cryptographic function, is not intelligible to the intruder. In the intruder process, two Boolean variables, namely *KMSResult* and *KBSResult,* are initialized to false and become true once the intruder manages to intercept the contents of the messages exchanged in the protocol. To verify the confidentiality of the respective results computed, the following LTL formula

is applied:

[] ((! KMSResult) && (! KBSResult))

Likewise, for the secrecy of the MS and BS results to be achieved, the above LTL formula must hold during the verification process of the protocol.

C. *The PROMELA Model with the Intruder's Interference*

To introduce the intruder process with MITM capabilities to the already modelled protocol, the general concept used in [13] was borrowed taking into account three assumptions: first, that the intruder station is strategically positioned between legitimate communicating stations in the WiMAX network; second, that the intruder station has two wireless network cards; and third, that the intruder already knows the general flow of the protocol. This process is illustrated in Figure 4.

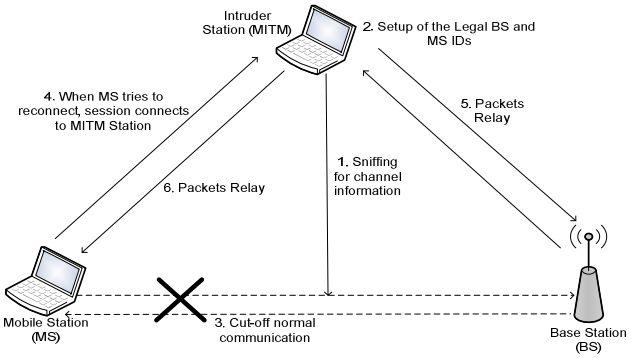

Figure 4: The MITM attack intrusion into the network

By first listening to an already established connection between communicating parties, the intruder process intercepts the frames being transmitted in the execution of the protocol, retrieves the legitimate stations' identities and maps them to its MAC address, increasing its initial knowledge necessary to launch attacks at Step 2. After successful mapping, the intruder process attempts to disconnect or break the connection between the MS and BS, forcing the MS to go back to the claiming stage of Figure 1. The intruder takes advantage of this opportunity and poses as the legitimate BS, redirecting data traffic to itself from the MS in Step 4. Similarly, to the BS, the intruder poses as the legitimate MS, forwarding traffic to and from the BS in Step 5. Eventually, the intruder process manages to take control of the communication process unobserved by the MS and the BS, and thus a MITM attack is launched.

The modelled protocol with the intruder process is illustrated in Figure 5. At step 113, the intruder process whose identity is "2" sends a message with a random number "10" as an attempt to disrupt the protocol run between the MS and the BS. This random number is taken by the BS as the response to the challenge it had sent earlier in step "105". This message from the intruder process has the identity of the MS and arrives at the BS prior to the result from the legitimate MS. As a result, the BS verifies it and finds out it is incorrect, consequently disconnecting from the MS. At this point, the intruder process has successfully broken the connection between the MS and the BS. At step 182, the MS selects the intruder process posing as a BS as it tries to reconnect again and sends a claim. The intruder process in turn forwards the claim message to the legitimate BS with the identity of the MS. The BS then sends a challenge to the intruder posing as the MS at step 255, which it forwards to the MS posing as the BS. This process continues until the protocol ends with all the messages passing through the intruder process before being forwarded to the respective recipients. The BS and the MS authenticate each other oblivious of the fact that the protocol was controlled by another process the entire time.

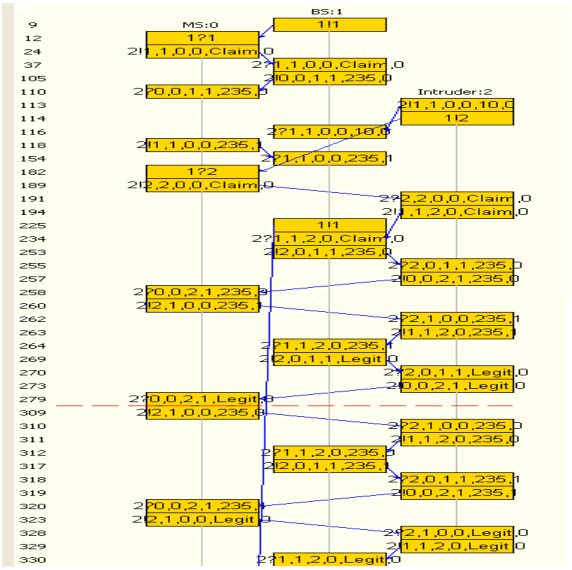

Figure 5: The MITM attack on the proposed protocol

D. *Verification Results*

Formal Verification of the protocol in [11] is performed on a laptop with a 2.0 GHz Intel Pentium Processor and 1.99 GB of RAM. We automatically verify the protocol with the stated LTL properties using an exhaustive search that includes a Depth First Search (DFS) algorithm of the state space and the use of the partial order reduction strategy aimed at reducing the number of states that need to be visited, so as to verify the stated properties. During the verification process, SPIN executes the generated never claim from the specified LTL formulae in Section 3.B with the given PROMELA model looking for a match between the claim and the model. A match signifies a violation of the LTL property under test which can either correspond to the detection of an acceptance cycle within the never claim or the termination of the never claim. A violation leads to the generation of a counter-example for the LTL formula. The results of the verification process are displayed below:

1) *Authentication Property:*

```
verification result:
ltl AuthProperty: [] ((((! (MSCommit)) U (! (BSRunning))) U (! (BSCommit))) U
(MSRunning))
./pan -m10000  -a
pan:1: acceptance cycle (at depth 495)
pan: wrote FirstProtocolScratch.pml.trail

Full statespace search for:
        never claim             + (AuthProperty)
        assertion violations    + (if within scope of claim)
        acceptance   cycles     + (fairness disabled)
        invalid end states      - (disabled by never claim)
State-vector 568 byte, depth reached 606, errors: 1
    5017 states, stored (9823 visited)
    1930 states, matched
   11753 transitions (= visited+matched)
     218 atomic steps
hash conflicts:        34 (resolved)
Stats on memory usage (in Megabytes):
     4.004          total actual memory usage
pan: elapsed time 0.042 seconds
```

Figure 6: The Verification Output with LTL Authentication Property

2) Secrecy Property:

```
verification result:
ltl SecrecyProperty: [] ((! (K_MSResult)) && (! (K_BSResult)))
./pan -m10000 -a
pan:1: end state in claim reached (at depth 877)
pan: wrote FirstProtocolScratch.pml.trail
Full statespace search for:
        never claim             + (SecrecyProperty)
        assertion violations    + (if within scope of claim)
        acceptance   cycles     + (fairness disabled)
        invalid end states      - (disabled by never claim)
State-vector 552 byte, depth reached 877, errors: 1
    7454 states, stored
    1649 states, matched
    9103 transitions (= stored+matched)
     220 atomic steps
hash conflicts:        13 (resolved)
Stats on memory usage (in Megabytes):
      4.687        total actual memory usage
pan: elapsed time 0.037 seconds
```

Figure 7: The Verification Output with LTL Secrecy Property

E. Discussion

By executing an exhaustive search for the verification of the model with the LTL authentication property stated in Section 3.B, an acceptance cycle was detected as seen in Figure 6, meaning that there exists a state in the model that is visited infinitely often, resulting in a violation of the stated authentication property. On replaying the error trail file generated by SPIN, a cycle exists when the intruder process is attempting to break the normal protocol run between the MS and BS rendering the *BSCommit* variable to false, which leads to a violation of the LTL property. Nonetheless, eventually, the modelled intruder process was able to break the normal protocol run between the BS and the MS by redirecting messages from the MS and BS to itself and relaying them to the intended recipients as seen in Figure 5. By doing this, the intruder process is successful in penetrating the network consisting of the MS and the BS and acts as a "middle man".

On verifying the protocol with the LTL secrecy property stated in Section 3.B, an end state of the claim was reached meaning that there was a match between the model and the generated never claim, resulting in a violation of the secrecy property as seen in Figure 7. On replaying the error trail file generated by SPIN, the error is traced back to the fact that any measure of confidentiality in the critical information is lacking in the protocol proposed in [11] and as a result, the intruder process was able to capture both the MS and BS results effortlessly. This in turn introduces a security flaw, since the intruder can easily manipulate the messages so as to lower the security capabilities of the network.

The modelled intruder process succeeded in eavesdropping the communication process between the MS and the BS, impersonation of both the MS and BS identities, injecting self-generated messages and capturing all the transmitted messages between the MS and the BS, all of which are characteristics of a MITM attack. Based on the results achieved, it is evident that the protocol proposed by [11] has traces of weaknesses against the MITM attack and therefore cannot be recommended as an extra step towards securing the initial network entry procedure in WiMAX.

4. Conclusion

Although security flaws in the protocol under consideration were perceptible through inspection, it was necessary to apply formal methods to explicitly prove or disprove the correctness claim made in [11]. We therefore modelled the modified cryptographic protocol using PROMELA and specified the properties that the protocol must satisfy in LTL. We further modelled an intruder process with MITM capabilities and introduced it to the specified protocol for behaviour analysis. Eventually, verification of the protocol was performed. We succeeded in identifying traces of vulnerability in the identified protocol that compromised the confidentiality of the messages being exchanged between legitimate stations, thus confirming its fallibility. Our future work includes formally modelling and verifying similar protocols implemented to curb the MITM at the initial network entry point in WiMAX.

REFERENCES

[1] T. Nguyen. (2009) A Survey of WiMAX Security Threats.[Online]. http://www.cse.wustl.edu/~jain/cse571-09/ftp/wimax2.pdf

[2] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*.: MIT Press, 1999.

[3] A. M. Taha, A. T. Abdel-Hamid, and S. Tahar, "Formal Verification of IEEE 802.16 Security Sublayer Using Scyther Tool," in *IEEE Int. Conf. Network and Service Security, N2S '09*, Paris, France, 2009.

[4] P. Narayana, R. Chen, Y. Zhao, Y. Chen, Z. Fu, and H. Zhou, "Automatic Vulnerability Checking of IEEE 802.16 WiMAX Protocols through TLA+," in *Proc. 2nd IEEE Workshop on Secure Network Protocols (NPsec)*, Santa Barbara, California, November, 2006.

[5] R. K. Guha, Z. Furqan, and S. Muhammad, "Discovering Man-In-The-Middle attacks in authentication protocols," in *MILCOM 2007*, Orlando, FL, October 29-31, 2007.

[6] G. J. Holzmann, *The Spin Model Checker, The: Primer and Reference Manual*.: Addison Wesley, 2003.

[7] P. Maggi and R. Sisto, "Using SPIN to Verify Security properties of Cryptographic Protocols," in *Proc. 9th Int. SPIN Workshop on Model Checking of Software*, Grenoble, France, 2002.

[8] A. S. Khan, M. Mukund, and S. P. Suresh. Generic Verification of Security Protocols. [Online]. http://spinroot.com/spin/Workshops/ws05/025_paper.pdf

[9] M. Xiao and J. Li, "The Modeling Analysis of Cryptographic Protocols Using Promela," in *Proc. 6th World Congr. on Intelligent Control and Automation (WCICA)*, Dalian, China, 23 October, 2006.

[10] W. Diffie and M. E. Hellman, "New Directions in Cryptogaphy," *IEEE Trans. Inform. Theory*, vol. 22, no. 6, 1976.

[11] M. S. Rahman and M. Md. S. Kowsar, "WiMAX Security Analysis and Enhancement," in *Proc. 2009 12th Int. Conf. Comput. and Inform. Technology (ICCIT 2009)*, Dhaka, Bangladesh, 2009.

[12] A. Pnueli, "The Temporal Logic of Programs," in *Proc.18th IEEE Symp. on Foundations of Comput. Sci.*, Providence, Rhode Island, 1977.

[13] H. Hwang, G. Jung, K. Sohn, and S. Park, "A Study on MITM (Man in the Middle) Vulnerability in Wireless Network Using 802.1x and EAP," in *Int. Conf. Inform. Science and Security*, Seoul, Korea, 2008.