Design and Analysis on Trusted Network Equipment Access Authentication Protocol

Yingxu Lai[1,*], Yinong Chen[2], Qichen Zou[1], Zenghui Liu[3], Zhen Yang[1]

1. *College of Computer Science, Beijing University of Technology, Beijing 100124, China*

**Corresponding author. Tel.: +86 13439195095; fax: +86 01067391742. E-mail address: laiyingxu@bjut.edu.cn*

2. *School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe 85287, USA*

3. *Automation Engineering Institute, Beijing Polytechnic, Beijing 100176, China*

Abstract

Cloud security is a system engineering problem. A common approach to address the problem is to adapt existing Trusted Network Connection (TNC) framework in the cloud environment, which can be used to assess and verify end clients' system state. However, TNC cannot be applied to network equipment attached to the cloud computing environment directly. To allow the network devices to access the trusted network devices safely and reliably, we first developed a Trusted Network Equipment Access Authentication Protocol (TNEAAP). We use the BAN logic system to prove that TNEAAP is secure and credible. We then configure the protocol in an attack detection mode to experimentally show that the protocol can withstand attacks in the real network. Experiment results show that all the nine goals that decide the protocol's security have been achieved.

Keywords: Trusted network equipment; authentication; security protocol; BAN logic; attack detection model

1. INTRODUCTION

Cloud computing environment is emerging as a key platform supporting data-intensive processing and big data analysis in information technology. The advantages of convenience, economy, and high extensibility draw the attention of more and more researchers and practitioners. However, cloud computing is just like a double-edged sword, it brings us great convenience, and at the same time, it also carries additional problems, such as performance, reliability, trustworthiness, and security [1-2]. With the popularity of cloud computing, the significance of security issues is gradually increasing, and it has become an important factor to restrict cloud applications. The solution of cloud security is complex. A common approach is to adapt existing Trusted Network Connection (TNC) [3] solutions to the cloud computing paradigm. The basic objective of TNC from the perspective of endpoint integrity is to deny those network accesses to endpoints that do not meet certain minimum security criteria [4-6]. In the network access control research, many related studies have been conducted in the field of improving network connection protocol and improving TNC architecture.

1.1 Trusted network connection protocol

Luo et al. [7] followed the TNC framework to design an improved network access control system, T-NAC, which emphasized the platform authentication and communication security. A network access control system based on the network processor platform IXP2400 was designed. Latze [8] proposed a strong bidirectional authentication protocol, which was based on TPM (Trusted Platform Module) and EAP-TLS authentication method. Latze [9] later proposed a new authentication protocol: EAP-TPM. The first EAP-TPM system was built in Switzerland. Then Latze [10] improved EAP-TPM protocol based on zero-authentication. However, these protocols did not address the problem of Man-in-the-middle attacks. Wang et al. [11] analyzed the D-H (Diffie-Hellman) keys exchange protocol, and proposed a digital signature and signature verification end-to-end protocol to solve the problem of Man-in-the-middle attacks. Yu [12] studied the platform anonymous identity management defects of TNC and proposed a new method to improve these impairments. The new trusted certification generation method was based on ID (Identity) encryption mechanism and improved DAA (Direct Anonymous Attestation) protocol. The protocol was more flexible and security to manage identity on terminal platforms.

1.2 Trusted access model

To make the trusted network access mechanism more practicable, researchers proposed different solutions that focused on TNC architecture. Jungbauer [13] proposed a method to determine the integrity of endpoints which served as a basis for trustworthy communication. The model did not require specific hardware such as TPM (Trusted Platform Module) or special operating system structure. It also supported exiting network infrastructures. Rehbock [14] proposed a protocol stack that enabled the use of TNC in web-based environments and changed the TNC architecture to ensure additional security. The potential use of the TPM functionality within the TNC framework and experiences were given by Bente [15] and Heldenin [16], respectively. They further defined a conceptual model for client-side policies that was based upon TNC's IF-M (Interface-Measurement) protocol and showed that many policies can be enforced by extending the standard TNC framework [17]. Tang et al. [18] proposed a trusted network model based on the TPM, through which a trusted chain from terminals to network was established. Zhang [19] designed a TNC security model based on UCF (Universally Composable Framework) that can be extended to describe more security properties, such as anonymity. The model can be applied to analyze more protocols in the TNC architecture.

Cloud security is a system engineering problem. It requires additional security features not only for the endpoint security techniques, but also for the switches and routers, which are the core network equipment of Ethernet. In general, the existing TNC framework can be used to assess and verify end clients' system state, but cannot be applied to network equipment directly. There are certain differences between network equipment and the endpoints to join in a network. Network equipment undertakes forwarding packets within routing or switching function in the network. They are service providers, whereas, endpoints use network only, and they are service clients. If accessed network equipments failure, it will affect the part of the cloud network, whereas, a failure endpoint only affects itself. Thus, network equipment pays more attention on trusted boot up process and trusted network services than terminals do. Trusted boot up process ensures network equipments static trust, trusted network service ensures its dynamic trust.

On the other hand, existing authentication protocols of network equipment have various weaknesses and are difficult to be applied in the trusted network. For example, PAP (Password Authentication Protocol) is often used in router access authentication. However, PAP is not a strong and effective method of authentication. The password is transported in plain text. CHAP (Challenge Handshake Authentication Protocol) [20] is another router authentication protocol. The protocol is more secure than PAP. In CHAP, (1) the remote access server sends a challenge to the remote client that consists of a session ID and an arbitrary challenge string. (2) The remote client must return the user name and a Message Digest 4 (MD4) hash of the challenge string, the session ID, and the MD4-hashed password. (3) The authenticator checks the response against its own calculation of the expected hash value. If the values match, the authentication is acknowledged; otherwise the connection SHOULD be terminated. (4) At random intervals, the authenticator sends a new challenge to the peer, and repeats steps 1 to 3. However, CHAP [21] also has its weaknesses: (1) In the authentication server, the user's password was stored in plain text, which provided opportunities for intruders to obtain a user's password. (2) The protocol supports one-way authentication only. (3) The user password in CHAP is shared between two communication parties, and thus keys distribution and updating can cause inconsistency problems. (4) If a user used a simple password, the protocol could not prevent the dictionary attack. (5) In order to prevent the insertion channel attacks, the authentication server must reprint certification periodically. If the cycle time interval is too long, it can give the intruder opportunities.

This paper presents a network device access authentication protocol. In our protocol, in addition to equipment's platform authentication, the administrator also should be authenticated. The identification authentication process is actually the binding administrator to the trusted platform, to ensure the administrator is legal on trusted platform. Our protocol is to avoid malicious behavior from illegal administrator login trusted platform or legitimate user login illegal platform. The authentication protocol achieves these targets: safety, credibility, and low overhead. The security of the protocol is analyzed by a formalization method based on BAN (Burrows-Abadi-Needham) logic, which can reveals vulnerabilities and redundancy [22]. The protocol processes are formalized with HLPSL (High-Level Protocol Specification Language) [23]. The formalization of the protocol processes is tested by plugging into an attacking model of the safety testing tool to check whether the protocol is secure or not.

In the rest of the paper, section 2 shows the design of the authentication method for the trusted network devices to join in the network. Section 3 formalizes our protocol by BAN logic for safety analysis. Section 4 presents the experiment, which uses the attack model to attack the protocol and to demonstrate that protocol is secure. Section 5 gives the network equipment performance evaluation. We conclude the paper in section 6.

2. DESIGN OF THE AUTHENTICATION PROTOCOL

As shown in Fig. 1, the proposed trusted network equipment access authentication protocol (TNEAAP) consists of three major components: requester, boundary network equipment, and network authentication management server.

1) Requester(R): it is a piece of network equipment, in which a Trusted Platform Module (TPM) is embedded. The Requester sends the metric information to the authentication management server when it wants to be a member of the network.
2) Boundary network equipment (BNE): it is a strategy execution device that controls the requester to access the network.
3) Network authentication management server (NAMS): it is a decision builder that manages network equipment and an administrator in the trusted network.

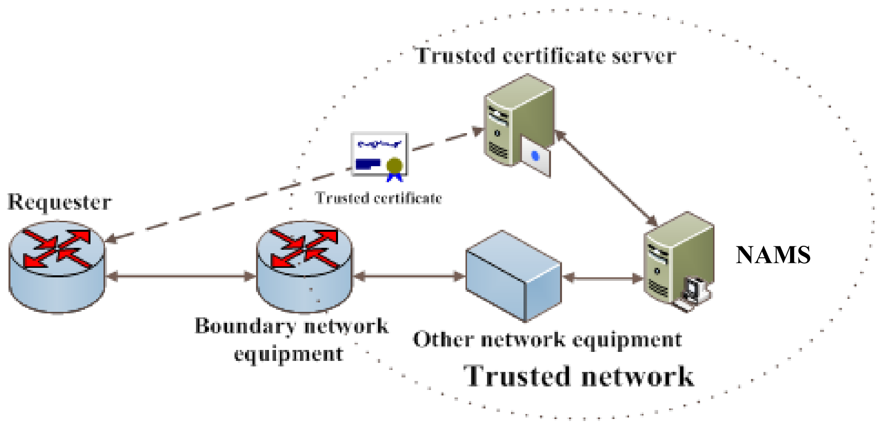


Fig. 1 TNEAAP architecture

2.1 Authentication Process

The proposed authentication process of TNEAAP is divided into the following three steps:

(1) Issuing of trusted certification: A requester R registers itself to the trusted certificate server, as shown in Fig.1, and then R applies for a trusted certificate from the server. R sends hardware information, software information, operating system version, and public key of R to the trusted certificate server. The server verifies information. If it is correct, the server will generate a trusted certificate and issue it to R.

(2) Platform authentication: This step performs the equipment platform authentication. R sends the Storage Measurement Log (SML) to the NAMS. NAMS validates whether SML accords with the trusted metric accessing rules. If information is certified by NAMS, the requester will be allowed to join into the network.

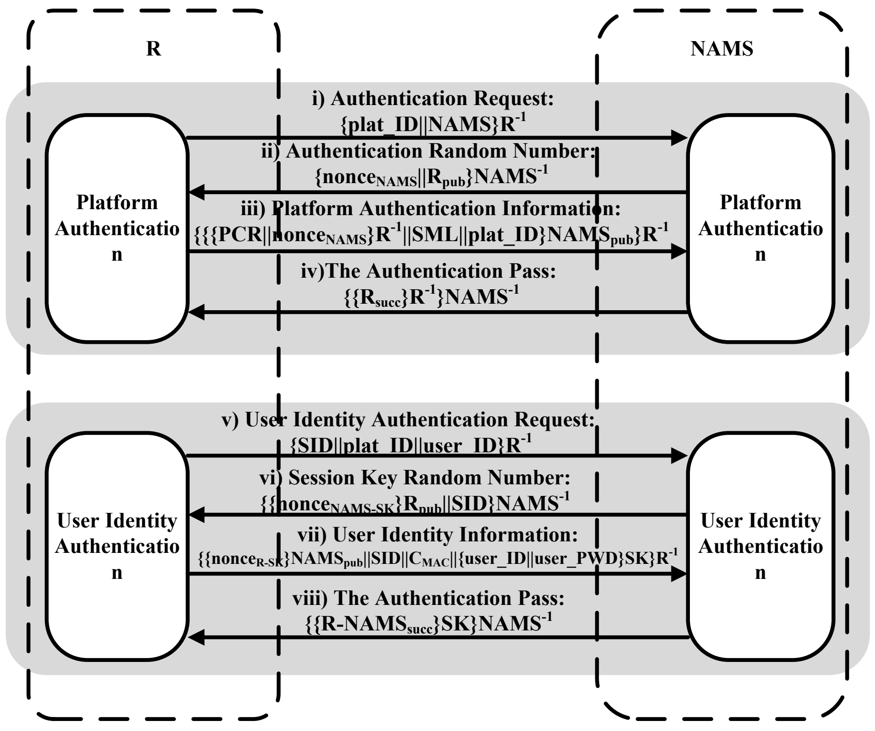
(3) Administrator authentication: When an administrator logs onto the network device (requester

R), R sends the administrator information to NAMS, which checks whether the administrator is legal or not.

In this paper, we do not discuss the trusted certificate server's strategy, and we focus on the security issues of the proposed authentication protocol.

2.2 Formalization of TNEAAP

The foregoing platform authentication and administrator authentication steps of TNEAAP are shown in Fig. 2, which elaborates the authentication process. The steps of the process listed in the figure are explained as follows.



R
NAMS
Platform Authenticatio n
Platform Authenticatio n
i) Authentication Request: {plat_ID||NAMS}R⁻¹
ii) Authentication Random Number: {nonce_NAMS||R_pub}NAMS⁻¹
iii) Platform Authentication Information: {{{PCR||nonce_NAMS}R⁻¹||SML||plat_ID}NAMS_pub}R⁻¹
iv)The Authentication Pass: {{R_succ}R⁻¹}NAMS⁻¹
User Identity Authenticatio n
User Identity Authenticatio n
v) User Identity Authentication Request: {SID||plat_ID||user_ID}R⁻¹
vi) Session Key Random Number: {{nonce_NAMS-SK}R_pub||SID}NAMS⁻¹
vii) User Identity Information: {{nonce_R-SK}NAMS_pub||SID||C_MAC||{user_ID||user_PWD}SK}R⁻¹
viii) The Authentication Pass: {{R-NAMS_succ}SK}NAMS⁻¹

Fig. 2 Platform authentication and administrator authentication of TNEAAP

(1) Platform authentication

i) R sends a request to NAMS. R signs the NAMS's public key NAMS$_{pub}$ and R's platform identity plat_ID by applying the private key of R (R^{-1}), and R sends the signed message to the server NAMS.

$$R \to NAMS: \{plat\_ID \parallel NAMS_{pub}\}R^{-1} \qquad (1)$$

ii) NAMS sends a random number to R. In the step (i), NAMS has obtained R's ID and NAMS$_{pub}$, which means that R trusts NAMS. Then NAMS generates a random number nonce$_{NAMS}$. NAMS sends nonce$_{NAMS}$ and R$_{pub}$ signed by the private key of NAMS (NAMS^{-1}) to R, where R$_{pub}$ is the public key of R.

$$NAMS \to R: \{nonce_{NAMS} \parallel R_{pub}\}NAMS^{-1} \qquad (2)$$

iii) R sends a platform authentication message to NAMS. R obtains the random number nonce$_{NAMS}$ and R$_{pub}$, which indicates that NAMS knows the public key of R. The hash digest of boot up processes is stored in PCR (Platform Configuration Register), which cannot be deleted or modified. PCR's value and nonce$_{NAMS}$ are signed by R's private key R^{-1}, that is, N = (PCR, nonce$_{NAMS}$)R^{-1}. R sends N, SML, and plat_ID signed by R^{-1} to NAMS.

$$R \to NAMS: \{\{\{PCR \parallel nonce_{NAMS}\}R^{-1} \parallel SML \parallel plat\_ID\}NAMS_{pub}\}R^{-1} \qquad (3)$$

iv) NAMS sends an acknowledgement message to R. NAMS unpacks the received message by R$_{pub}$ and NAMS^{-1}, obtains the PCR's value. NAMS compares the hash digest of SML with

PCR's value. If the result is consistent, SML is not tampered. Then NAMS considers the boot up process is trusted. NAMS sends ACK (Acknowledgement) of platform authentication success to R and allows R to access network.

$$NAMS \rightarrow R : \{\{ACK\}R_{pub}\}NAMS^{-1} \tag{4}$$

(2) Administrator authentication (requires negotiation session key)

v) R sends a user authentication request to NAMS. The request includes plat_ID, username (user_ID), and the current session identifier (SID). R sends the request signed by R^{-1} to NAMS.

$$R \rightarrow NAMS : \{SID \,||\, plat\_ID \,||\, user\_ID\}R^{-1} \tag{5}$$

vi) NAMS generates and sends a session key random number to R. NAMS determines whether the user is legal and valid through the plat_ID and user_ID. If the information is valid, NAMS generates a session key random number $nonce_{NAMS-SK}$ and encrypts it by R_{pub}. NAMS signs the cryptograph and SID, then sends this signature information to R.

$$NAMS \rightarrow R : \{nonce_{NAMS-SK}\}R_{pub} \,||\, SID\}NAMS^{-1} \tag{6}$$

vii) R sends user identity information to NAMS. R decrypts SID by the public key of NAMS and the random number $nonce_{NAMS-SK}$ using its own private key. Then, R generates a new random number $nonce_{R-SK}$ and message authentication codes C_{MAC}, $C_{MAC} = HMAC - SHA1_{SK}(nonce_{NAMS-SK}, SID)$, which ensures information integrity. R uses the pseudo-random number generation function PRGF to calculate this session key SK using the $nonce_{R-SK}$, $nonce_{NAMS-SK}$ and the current session identifier SID. SK's calculation equation is $SK = PRGF(nonce_{NAMS-SK}, nonce_{R-SK}, SID)$. R encrypts the username user_ID, user password user_PWD, and trusted platform authentication certificate R_{cer} by SK. R's private key signs this cryptograph, $\{nonce_{R-SK}\}NAMS_{pub}$, SID and C_{MAC}, sends this signature information to NAMS.

$$R \rightarrow NAMS : \{\{nonce_{R-SK}\}NAMS_{pub} \,||\, SID \,||\, C_{MAC} \\ ||\, \{user\_ID \,||\, user\_PWD \,||\, R_{cer}\}SK\}R^{-1} \tag{7}$$

viii) NAMS sends an acknowledgement message to R. $NAMS^{-1}$ decrypts $\{nonce_{R-SK}\}NAMS_{pub}$. NAMS receives the random number $nonce_{R-SK}$. NAMS calculates the session key SK using the two existing random numbers of session and the current session identifier SID. Then $\{user\_ID \,||\, user\_PWD \,||\, R_{cer}\}SK$ is decrypted by the session key SK. NAMS can obtains the user name and password, compares the user information with the user registration information in NAMS. If the user is legal, NAMS sends ACK' acknowledgement of administrator authentication success to R and allows the R to access the trusted network with the administrator login. The certification process is completed with the following action:

$$NAMS \rightarrow R : \{\{ACK\}SK\}NAMS^{-1} \tag{8}$$

3. TNEAAP FORMALIZATION ANALYSIS

According to BAN logic analysis method and the characteristics of TNEAAP, we analyze whether the authentication protocol is secure or not.

3.1 Analysis Procedure

The analysis procedure is as follows:

1) According to BAN logic representation, the authentication protocol is formalized;
2) The security goals of the protocol are determined. The assumptions are initialized. The security goals and initial assumptions are described by logical symbols;
3) The proof of security is to show that if the initial assumptions are applied to the messages of the protocol, and all of the security goals can be inferred from the messages;
4) In the reasoning process, the protocol defects and redundancy can be detected.

The authentication protocol is formalized as equation (1) to equation (8), and the other analysis steps are as follows.

3.2 Security Targets

This section discusses the general BAN logic forms of security objectives. The primary goal is $A \mid\equiv X$ and $B \mid\equiv X$. The ultimate goal is $A \mid\equiv B \mid\equiv X$ and $B \mid\equiv A \mid\equiv X$.

The initialization assumption is the primary goal in BAN. Thus, the security goals can be defined as the following seven goals:

$$R \mid\equiv NAMS \mid\equiv\mid \xrightarrow{R_{pub}} R \tag{9}$$

$$NAMS \mid\equiv R \mid\equiv\mid \xrightarrow{NAMS_{pub}} NAMS \tag{10}$$

$$NAMS \mid\equiv R \mid\equiv PCR \tag{11}$$

$$R \mid\equiv NAMS \mid\equiv ACK \tag{12}$$

$$R \mid\equiv NAMS \mid\equiv N_{NAMS-SK} \tag{13}$$

$$NAMS \mid\equiv R \mid\equiv N_{R-SK} \tag{14}$$

$$R \mid\equiv NAMS \mid\equiv ACK' \tag{15}$$

In words, the seven goals are:

① R believes that NAMS believes that R_{pub} is R's public key;
② NAMS believes that R believes that $NAMS_{pub}$ is NAMS's public key;
③ NAMS believes that R believes that PCR is credible;
④ R believes that NAMS believes that ACK is credible;
⑤ R believes that NAMS believes that $nonce_{NAMS\text{-}SK}$ ($N_{NAMS\text{-}SK}$) is credible;
⑥ NAMS believes that R believes that $nonce_{R\text{-}SK}$ ($N_{R\text{-}SK}$) is credible;
⑦ R believes that NAMS believes that ACK' is credible.

The fourth goal is to prove that the platform authentication process is trusted. The seventh goal is to prove that the administrator authentication process is trusted. These two goals are the most important ones.

3.3 Initialization assumption

According to the characteristics of TNEAAP, this paper makes the following initialization assumptions:

$$R \mid\equiv\mid \xrightarrow{NAMS_{pub}} NAMS \tag{16}$$

$$R \mid\equiv NAMS \mid\Rightarrow N_{NAMS}(nonce_{NAMS}), N_{NAMS-SK} \tag{17}$$

$$NAMS \mid\equiv R \mid\Rightarrow N_{R-SK} \tag{18}$$

$$R \mid\equiv \#(N_{NAMS}, N_{NAMS-SK}, R_{pub}) \tag{19}$$

$$NAMS \mid\equiv \#(PCR, N_{R-SK}) \tag{20}$$

$$NAMS \mid\equiv\mid \xrightarrow{R_{pub}} R \tag{21}$$

3.4 Analysis

Logical Reasoning Process 1:

From the equation (2)we can derive:

$$R \mid\equiv\mid \xrightarrow{NAMS_{pub}} NAMS, \quad R \triangleright \text{[illegible]} R_{NAMS}\}NAMS^{-1} \tag{22}$$

We can infer the following:

$$R \mid\equiv NAMS \mid \text{[illegible]}) \tag{23}$$

Then, we draw the conclusion:

$$R \mid\equiv NAMS \mid\equiv\mid \xrightarrow{R_{pub}} R \tag{24}$$

Logical Reasoning Process 2:

Similarly to reasoning process 1, from the equation (2) we can draw the conclusion.

$$NAMS \mid\equiv R \mid\equiv \xrightarrow{NAMS_{pub}} NAMS \tag{25}$$

Logical Reasoning Process 3:

From the equation (3), we can obtain

$$NAMS \triangleright \{\{\{PCR, N_{NAMS}\}R^{-1}, SML, plat\_ID\}Ks\}R^{-1}, NAMS \mid\equiv \xrightarrow{R_{pub}} R \tag{26}$$

We can then infer:

$$NAMS \triangleright \{\{PCR, N_{NAMS}\}R^{-1}, SML, plat\_ID\}NAMS_{pub}, NAMS \mid\equiv \xrightarrow{NAMS_{pub}} NAMS \tag{27}$$

We can also infer:

$$NAMS \triangleright \{\{PCR, N_{NAMS}\}R^{-1}, SML, plat\_ID\}, NAMS \mid\equiv \xrightarrow{R_{pub}} R \tag{28}$$

and

$$NAMS \mid\equiv R \mid \{PCR, N_{NAMS}\}, NAMS \mid\equiv \#(PCR) \tag{29}$$

We draw the conclusion

$$NAMS \mid\equiv R \mid\equiv PCR \tag{30}$$

Logical Reasoning Process 4:

From the equation (4), we can infer:

$$R \mid\equiv NAMS \mid \{R_{succ}\}R_{pub}, R \mid\equiv \#(R_{succ}), R \mid\equiv \xrightarrow{R_{pub}} R \tag{31}$$

And draw the conclusion

$$R \mid\equiv NAMS \mid\equiv ACK \tag{32}$$

Logical Reasoning Process 5:

From the equation (6), we can infer:

$$R \mid\equiv NAMS \mid\sim \{SID, \{N_{NAMS-SK}\}R_{pub}\} \text{ , and} \tag{33}$$

$$R \mid\equiv \#(N_{NAMS-SK}), R \mid\equiv \xrightarrow{R_{pub}} R \tag{34}$$

We draw the conclusion

$$R \mid\equiv NAMS \mid\equiv N_{NAMS-SK} \tag{35}$$

Logical Reasoning Process 6:

Similarly to reasoning 5, we can draw the conclusion from the equation (7):

$$NAMS \mid\equiv R \mid\equiv N_{R-SK} \tag{36}$$

Logical Reasoning Process 7:

From the equation (8), we can infer:

$$R \mid\equiv NAMS \mid\square \qquad R \text{, and} \tag{37}$$

$$R \mid\equiv \#(ACK'), R \mid\equiv \mid \xrightarrow{R_{pub}} R \tag{38}$$

And we draw a conclusion

$$R \mid\equiv NAMS \mid\equiv ACK' \tag{39}$$

3.5 Results

1) Although PCR, SML, platform ID, ACK, and ACK' are not responsible for the authentication procedure directly, they are not redundant information. They are important for the security of the trusted network, and they play determinative roles in TNEAAP. We do not find redundant information in TNEAAP based on aforementioned analysis, thus it is a concise protocol.

2) In this method, seven security targets can be inferred from the messages of R and NAMS. Thus, TNEAAP is secure in BAN logic reasoning.

4. THE SECURITY TESTING AND ANALYSIS OF TNEAAP

In this section, we use an experiment to further demonstrate the protocol is secure through an attacking detection method. The experiment should find defects of TNEAAP if there exists any. The protocol authentication processes are formalized using HLPSL [24] protocol description language. Then formalization of the protocol process is plugged into an attacking model of a safety testing tool. Finally the tool gives a conclusion whether TNEAAP can prevent the attack generated by the tool.

4.1 Security targets

In this section, we choose Dolev-Yao intruder model [25] for testing our protocol. In this model, an intruder controls the whole network and can perform any operations. The intruder can intercept, analyze, and revise all of messages. The intruder can pretend to be any agent and send disguised messages to anyone in the network. The attacker has the following knowledge and abilities:

1) The attacker is familiar with encryption, decryption, hash, and other cryptographic operations. The attacker has its own public key and private key;
2) The attacker has obtained the subject's identifier and public key;
3) The attacker has the knowledge and ability of cryptanalysis;
4) The attacker can make various attacks, such as knowledge and ability of replay attacks.

This is the strongest intruder model. If TNEAAP can meet the security goals under this intruder model, it proves the protocol can prevent all possible attacks from any attacker.

In this experiment, we define 9 security goals:

1) The transportation process of PCR is confidential;
2) The transportation process of SML is confidential;
3) The transportation process of random number $\text{nonce}_{\text{NAMS}}$ is confidential;
4) The transportation process of session key random number $\text{nonce}_{\text{NAMS-SK}}$ is confidential;
5) The transportation process of session key random number $\text{nonce}_{\text{R-SK}}$ is confidential
6) The transportation process of the administrator's password is confidential;
7) The transportation process of the administrator's username is confidential;
8) The platform authentication success ACK is correct;
9) The user authentication success ACK' is correct.

According to the above security goals, we use AVISPA (Automated Validation of Internet Security Protocols and Applications) [26] network communication protocol security inspection system to test the security of our protocol.

4.2 AVISPA

AVISPA is a set of the establishment and analysis tools of security protocols [26]. It is one of the widest used protocols in the cryptography. As shown in Fig.3, it combines four types of analyses at backends: On-the-fly Model-Checker (OFMC), CL-based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC), and Tree-Automata based Protocol Analyzer (TA4SP). The user first inputs the participants' identification of protocol and then selects the running environment, goal, attacker ability variables. Finally, the user specifies the desired security properties in order to find whether there are problems in the protocol under test. The code written in HLPSL language is translated into IF (Intermediate Format) language through HLPSL2IF translation tools. Analysis of terminal AVISPA tool set can directly read IF language. It can analyze whether security goals are successful or not. If the protocol is unsafe, the analysis terminal will give us the attack track events.

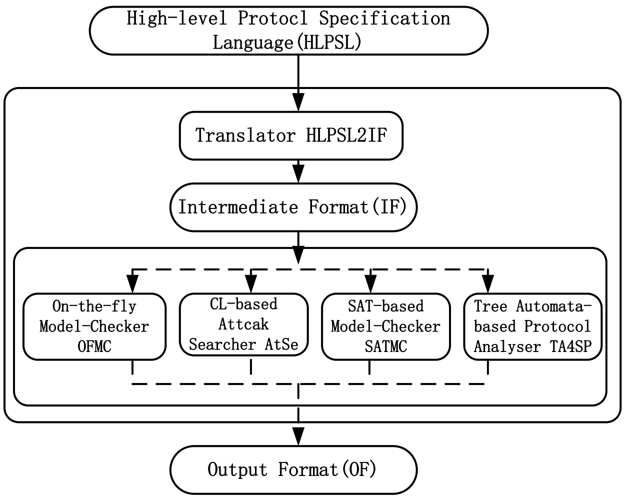

High-level Protocl Specification Language(HLPSL)
Translator HLPSL2IF
Intermediate Format(IF)
On-the-fly Model-Checker OFMC
CL-based Attcak Searcher AtSe
SAT-based Model-Checker SATMC
Tree Automata-based Protocol Analyser TA4SP
Output Format(OF)

Fig.3 Architecture of AVISPA

4.3 Experiment process

4.3.1 Basic roles

In HLPSL, each participant is defined in a module separately, called basic role, which describes its initial state and its state transitions. We define two basic roles, as shown in Table 1: a network authentication manager server NAMS (a) and a Requester (r).

Table 1 Definition of the basic roles

Basic role	Basic role configuration
a	role a (R, NAMS:agent, Ka, Kr:public_key, SND, RCV:channel(dy))
r	role r (R, NAMS:agent, Ka, Kr:public_key, SND, RCV:channel(dy))

4.3.2 Security goals

For evaluating the security properties of TNEAAP, we first formulize its security goals [27]. The specified equations are used to assess whether the goals are achieved or not.

AVISPA defines different macros for formalizing the security goals. In our experiment, the macros are described as follow:

1. Macro of information secrecy. T is the information produced by agent A. If T is a shared secret and is shared between agent A and a group of agents, B and C, the secrecy of information T is expressed as follows:

Secret (T, t, A, B, C)

where, t is the identification of the information T.

2. Macro of strong authentication detection. This property is formalized using two macros as follows:

Request (B, A, t, T)

Witness (A, B, t, T)

where, Request (B, A, t, T) indicates that agent B receives information T (identified by t) from agent A. Witness (A, B, t, T) indicates that agent A receives information T (identified by t) from agent B.

For evaluating the security of TNEAAP, the following security goals are defined:

1. Authentication of ACK and ACK', where their process IDs are r_ack1 and r_ack2, respectively. We model this goal in HLPSL.

Role r (R, NAMS: agent, Ka, Kr:public_key, SND, RCV:channel(dy))

Played_by R

init State:=0

```
Transition:
        State':=6/\request(R, NAMS, r_ack1')
        State':=10/\request(R, NAMS, r_ack2')
```

It means that R requests to check both ACK and ACK'. If the authentication is successful, R can obtain an acknowledge message.

2. Check of the SK shared between NAMS and R. SK is only known by NAMS and R. Similar to SK, check of the PCR, SML, $\text{nonce}_{\text{NAMS}}$, user_ID, user_PWD, $\text{nonce}_{\text{NAMS-SK}}$ and $\text{nonce}_{\text{R-SK}}$ are all needed. These goals are modeled in HLPSL as follows:

```
Role a (R, NAMS: agent, Ka, Kr:public_key, SND, RCV:channel(dy))
Played_by NAMS
init State:=0
Transition:
        State':=5/\request(PCR', a_pcr, {R, NAMS})
                /\request(SML', a_sml, {R, NAMS})
                /\request(Nonce1', a_nonce1, {R, NAMS})
        State':=9/\request(User_id', a_userid, {R, NAMS})
                /\request(User_pwd', a_userpwd, {R, NAMS})
                /\request(Nonce3', a_nonce3, {R, NAMS})
                /\request(SK',a_sk, {R, NAMS})
Role r (R, NAMS: agent, Ka, Kr:public_key, SND, RCV:channel(dy))
Played_by R
init State:=0
        State':=8/\request(Nonce2', a_nonce2, {R, NAMS})
```

In the model NAMS declares PCR, SML, $\text{nonce}_{\text{NAMS}}$, user_ID, user_PWD, $\text{nonce}_{\text{R-SK}}$ and SK as the secrets, and R declares $\text{nonce}_{\text{NAMS-SK}}$ as a secret, where a_x stands for x's process id.

4.3.3 Session scenarios

For the validation purpose, we define three different scenarios. First, we implement a single session with all the roles played by legitimate agents (Scenario 1). Then we test the situations in which the intruder would impersonate the network authentication manager server (Scenario 2) or the requester (Scenario 3). Table 2 lists the HLPSL definition of the sessions associated with each of the mentioned scenarios, where kx refers to the public key of x.

Table 2 Summary of session configurations

Scenario	Session configuration
Scenario 1	session(a, r, ka, kr)
Scenario 2	session(a, i, ka, ki)
Scenario 3	session(i, r, ki, kr)

4.3.4 Experiment results

The experiments are conducted based on the aforementioned model specification. For our verification, we have used OFMC and CL-AtSe backends to search for the attacks on the protocol. The test outputs of the experiment results are summarized in Table 3. The left column lists the output from the OFMC backend and the right column is from the CL-AtSe backend.

Table 3 Summary of test outputs from OFMC backend, and from CL-AtSe backend

SUMMARY	SUMMARY
SAFE	SAFE
DETAILS	DETAILS
BOUNDED_NUMBER_OF_SESSIONS	BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL	TYPED_MODEL
D:\NPLAB\temp\130401203137024905.if	PROTOCOL
GOAL	D:\NPLAB\temp\130401203858035709.if
As_specified	GOAL
BACKEND	As Specified
OFMC	BACKEND
COMMENTS	CL-AtSe
STATISTICS	STATISTICS
parseTime: 0.00s	Analysed : 4 states
searchTime: 0.03s	Reachable : 4 states
visitedNodes: 4 nodes	Translation: 0.01 seconds
depth: 2 plies	Computation: 0.00 seconds

According to the summary results in Table 3, TNEAAP is safe in both OFMC and AtSe backends (Summary: SAFE), and no vulnerabilities in the proposed protocol. If some defects are detected, Summary will be UNSAFE. DETAILS section provides the information that an attack is found in the protocol specification. The IF form of the protocol resides in the path given under the PROTOCOL section of the output, with the file name, 130401203137024905.if. The GOAL section of output describes the result of the goal, which is written in the specitication for the verificaiton process. The backend that verifies the protocol specification is OFMC or CL-AtSe, which is given under the BACKEND. The STATISTICS gives us the time required to execute the protocol specification by the tool and the number of the visited nodes or states during the execution.

For comparison, we have used OFMC and CL-AtSe backends to search for the attacks on the CHAP protocol. The test outputs of the experiment results are summarized in Table 4. The left column lists the output from the OFMC backend and the right column is from the CL-AtSe backend.

Table 4 Summary of CHAP from OFMC backend, and from CL-AtSe backend

SUMMARY	SUMMARY
UNSAFE	UNSAFE
DETAILS	
ATTACK_FOUND	DETAILS
PROTOCOL	ATTACK_FOUND
D:\NPLAB\temp\130570765157676751.if	TYPED_MODEL
GOAL	PROTOCOL
secrecy_of_sec_kab1	D:\NPLAB\temp\130570766808871194.if
BACKEND	
OFMC	GOAL
COMMENTS	Secrecy attack on (kab)
STATISTICS	
parseTime: 0.00s	BACKEND
searchTime: 0.01s	CL-AtSe
visitedNodes: 1 nodes	
depth: 1 plies	STATISTICS
ATTACK TRACE	
i -> (a,3): start	Analysed : 7 states
(a,3) -> i: a	Reachable : 7 states
i -> (a,3): x238	Translation: 0.01 seconds
(a,3) -> i: Na(2).h(kab.Na(2).x238.a)	Computation: 0.00 seconds
i -> (i,17): kab	
i -> (i,17): kab	
	ATTACK TRACE
% Reached State:	i -> (a,3): start
% % secret(kab,sec_kab1,set_61)	(a,3) -> i: a

% contains(a,set_61) % contains(b,set_61) % state_chap_Init(b,i,kbi,h,0,dummy_nonce,dummy_nonce,set_77,9) % state_chap_Init(a,i,kai,h,0,dummy_nonce,dummy_nonce,set_74,6) % state_chap_Init(a,b,kab,h,2,Na(2),x238,set_61,3) % state_chap_Resp(b,a,kab,h,0,dummy_nonce,dummy_nonce,set_69,3) % witness(a,b,na,Na(2))	i -> (a,3): Nb(2) (a,3) -> i: n2(Na).{kab.n2(Na).Nb(2).a}_h & Secret(kab,set_61); Witness(a,b,na,n2(Na)); Add a to set_61; & Add b to set_61;

According to the summary results in Table 4, CHAP is unsafe in both OFMC and AtSe backends (Summary: UNSAFE), DETAILS section provides the information that a Man-in-the-middle attack is found in the protocol specification.

5. NETWORK EQUIPMENT PERFORMANCE EVALUATION

Like all security protocols, TNEAAP take extra steps to ensure the security of the network and its devices. This section evaluates the overhead of implementing TNEAAP from network equipment's point of view. We take both storage and computation overheads into consideration.

5.1 Storage overhead

In TNEAAP, each requester has to store two keys permanently: the secret key SK and the NAMS's public key $NAMS_{pub}$.

Additionally, each requester also needs to store two nonce values: $nonce_{NAMS}$ and $nonce_{NAMS-SK}$. Considering the requesters that attempt to access a given node in a completely random basis at mean rate 1/T, the second step of authentication messages in equation (2) received by NAMS can be modelled as a Poisson processwith mean 1/T. For each authentication message received, NAMS stores a $nonce_{NAMS}$, until it receives the fourth step of authentication message in equation (4) from NAMS or until a timer set to $T_{Lifetime}$ expires. Suppose that the $T_{Lifetime}$ has an upper bound and is reached whenever a packet is lost, assuming a packet loss probability of the network is P, the average number of $nonce_{NAMS}$ that the requester must store is given by equation (40), where T_{ACK} denotes the average time elapsed between the reception of an equation (2) message and its corresponding equation (4) message. We assume that the storage overhead of $nonce_{NAMS}$ and $nonce_{NAMS-SK}$ is the same. Therefore, the requester storage overhead is

$$\text{Storage overhead} = N \times \left[\frac{1}{T} T_{Lifetime} P + \frac{1}{T} T_{ACK} (1-P) \right] \quad (40)$$

As an example, we set T_{ACK}= 10s, $T_{Lifetime}$= $2T_{ACK}$= 20s, and T=5min. Fig. 4 shows overheads incurred from 5, 10, and 50 requesters, with an increasing packet loss probability. As can be seen in the figure, the average number of $nonce_{NAMS}$ and $nonce_{NAMS-SK}$ that NAMS stores is small. It implies that NAMS does not need to have a large buffer to maintain the nonce values.

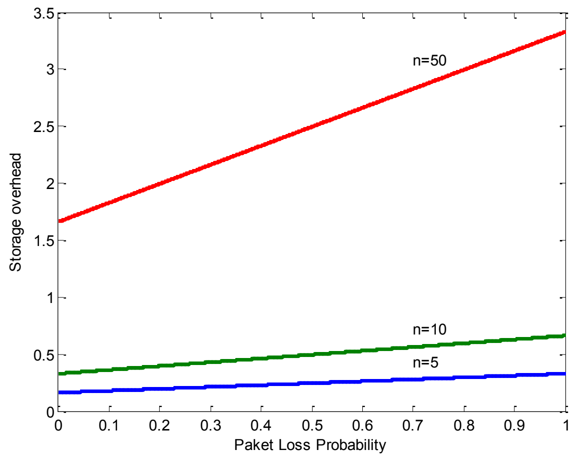


Fig.4 Storage overhead

5.2 Computation overhead

The overall computational overhead incurred by TNEAAP is a fixed number. The individual overheads are from equation (3) and equation (7). Table 5 summarizes the individual overheads in NEAAP for each type of requesters.

Table 5 Computational overhead of cryptographic operations in equation (3) and equation (7)

Type of operation	equation (3) (calculation times)	equation (7) (calculation times)
Symmetric key encryption	0	1
Symmetric key decryption	0	0
Asymmetric key encryption	1	1
Asymmetric key decryption	2	1
MAC	0	1

5.3 Comparison with other authentication protocols

Now we compare the capacity and overhead of TNEAAP with that of the other security protocols. The results are summarized in Table 6. As shown in Fig. 5 and Fig. 6, for 50 requesters, our method does not need to have a large buffer to maintain nonce values, and computational overhead is similar to the other protocols. However, TNEAAP supports more security functions, including both authentication capability and authorization capability, which ensure the credibility of the network system is strong.

Table 6 Comparison with other protocols

	Kerberos [29]	LEAP [30]	SPINS-based Protocol [31]	TNEAAP
authentication capability	yes	no	yes	yes
authorization capability	no	two levels: legitimate member of thesensor network or attacker	no	yes
storage cost	1 symmetric key	1 symmetric key + identitiesof neighboring nodes	1 symmetric key	1 public key +1symmetric key+2 nonce
computation overhead	1 encryption + 2 decryption	1 MAC+ 1 pseudo-random function	1 decryption + 2 MAC	2 encryption+1 decryption+1 MAC
Support trusted module	no	no	no	yes

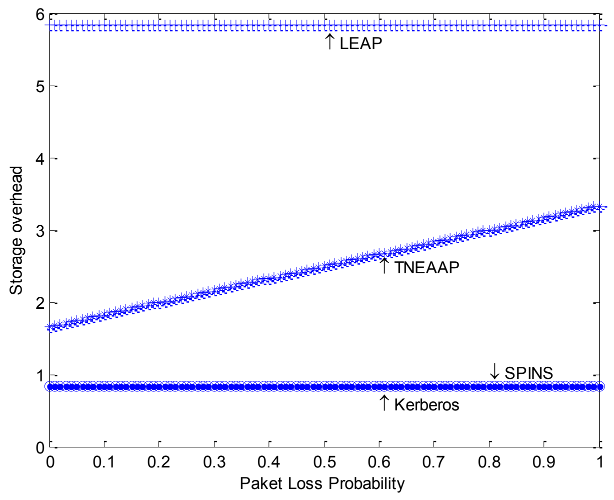


Fig.5 Comparison curve of storage overhead at n=50

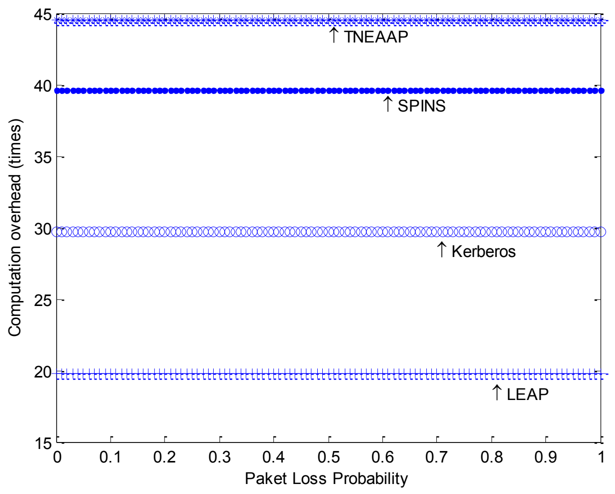


Fig.6 Comparison curve of computation overhead at n=50

6. CONCLUSIONS

Current trusted network access authentication research pays major attentions to the terminals, and thus the TNC (Trusted Network Connection) does not have an authentication protocol for network devices accessing. In this paper, we developed the TNEAAP protocol, which is more suitable for network equipment, more secure, and more reliable, due to the development of additional mechanism for the equipment's platform authentication.

In the study of TNEAAP, we theoretically and experimentally verified three issues: (1) TNEAAP does not contain any unnecessary information. We analyzed the redundant information by BAN logical system. The results show that the protocol is a concise protocol. (2) The protocol is security in theory. The BAN logic safety analysis has proved TNEAAP is safe and reliable. (3) The protocol is secure in experiment. We tested the protocol under the strongest attack model in our experiments. The nine goals which decided the protocol's security have been achieved in the attack model.

Acknowledgments

This research is partly supported by Funding Project for Academic Human Resources Development in Institutions of Higher Learning under the Jurisdiction of Beijing Municipality (PHR201108016), Open Research Fund of Beijing Key Laboratory of Trusted Computing.

References

[1] Chen Y., Tsai W.T., Service-Oriented Computing and Web Software Integration, 4th edition, Kendall Hunt Publishing, 2014

[2] Ioannis A. Moschakis, Karatza H.D.: Evaluation of gang scheduling performance and cost in a cloud computing system. The Journal of Supercomputing 59(2): 975-992 (2012)

[3] TCG Trusted Network Connect Work Group. TNC Architecture for Interoperability (April 2008), http://www.trutedcomputinggroup.org/resources/tnc_architecture_for_interoperability_version_13 (Specification Version 1.3 Revision 6)

[4] Trusted Computing Group, TPM main specification, Version 1.2 rev. 85, http://www.trustedcomputinggroup.com, 2011.4.

[5] Trusted Computing Group, TCG Storage Architecture Core Specification, Version 1.2, http://www.trustedcomputinggroup.com, 2007.

[6] Trusted Computing Group, TCG Trusted Network Connect TNC Architecture for Interoperability Specification, Version 1.2, http://www.trustedcomputinggroup.com, 2007.

[7] Luo A., Lin C., Chen Z., Peng X., et al, TNC-compatible NAC System implemented on Network Processor, The 32nd IEEE Conference on Local Computer Networks, 2007, Dublin, Ireland. pp:1069-1075.

[8] Latze C., Ultes-Nitsche U., Baumgartner F., Strong mutual authentication in a user-friendly way in EAP-TLS: Software, Telecommunications and Computer Networks, 15th International Conference on SoftCOM, 2007, Dubrovnik, Croatia. pp:27-29.

[9] Latze C., Ultes-Nitsche U., A Proof-of-Concept Implementation of EAP-TLS with TPM support, Proc. International Conference on Citeseer, 2008, pp:1-12.

[10] Latze C., Ultes-Nitsche U., Baumgartner F., Towards a Zero Configuration Authentication Scheme for 802.11 Based Networks, Proc. International Conference on IEEE, 2008, pp:367-373.

[11] Wang Y., Li Z., Yu P., et al. Improved trusted network connect protocol to prevent middle attack, Application Research of Computers, 2010, 27:4309-4311.

[12] Yu A., Chu X., Feng D., et al. Research of Platform Anonymous Identity Management Based on Trusted Chip, Chinese Journal of Computers, 2010, 33:1703-1712.

[13] Jungbauer M., Pohlmann N. Integrity Check of Remote Computer Systems Trusted Network Connect. ISSE/SECURE 2007 - Securing Electronic Business Processes: Highlights of the Information Security Solutions Europe/SECURE 2007 Conference,2007, pp:228-237.

[14] Rehbock S., Hunt R.Trustworthy clients: Extending TNC to web-based environments. Computer Communications, 2009, 32:1006-1013.

[15] Bente I., von Helden J. Towards trusted network access control. In: Proceedings of the First International Conference Future of Trust in Computing, 2008, pp:157-167.

[16] vonHelden J., Bente I. Towards real interoperable, real trusted network access control. In:

ISSE 2008 Securing Electronic Business Processes, 2008, pp:152-162.

[17] Bente I., ViewegJ., vonHeldenJ. Privacy Enhanced Trusted Network Connect INTRUST 2009, LNCS 6163, 2010, pp:129-145.

[18] TangJ., SongS., ZhaoL., et al. Trusted Network Model Based on Trusted Platform Module, Computer Engineering, 2011, vol. 37, no.11, pp:117-119.

[19] Trusted Network Universally Composable, Scientia Sinica (Informations), 2010, vol.2, no.2, pp:200-215.

[20] Simpson W. PPP Challenge Handshake Authentication Protocol (CHAP), RFC1994, 1996.

[21] Liu Y., Zhu F., Shi Q. An Improved Scheme of Challenge Handshake Authentication Protocol, Computer Engineer, 2005, vol.31, no.5, pp:168-169.

[22] Wang Y., Wang J., Wang M., et al. Security analysis of routing protocol for MANET based on BAN logic, Journal of China Institute of Communications, 2010, vol.25, no.4, pp:125-129.

[23] Dadeau F., Héam P.C., Kheddam, R. Mutation-Based Test Generation from Security Protocols in HLPSL, Verification and Validation (ICST), 2011 IEEE Fourth International Conference on Software Testing, 2011, pp:240-248.

[24] Paulson, L. C. Inductive analysis of the Internet protocol TLS, ACM Transactions on Information and System Security (TISSEC), 1999, vol.2, no.3, pp:332-351.

[25] Chen Q. Automatic Verification of Web Service Protocols for Epistemic Specifications under Dolev-Yao Model, 2010 International Conference on Service Sciences (ICSS), 2010, pp:49-54.

[26] http://www.avispa-project.org/.

[27] Toledo N., Higuero M., Astorga J., Aguado M. Design and formal security evaluation of NeMHIP: A new secure and efficient network mobility management protocol based on the Host Identity Protocol. Computers & Security, 2013, vol.32, pp:1-18.

[28] Astorga J., Jacob E., Huarte M., Higuero M. Ladon 1: end-to-end authorization support for resource-deprived environments. Information Security, 2012, vol.6, no.2, pp: 93-101.

[29] Kohl J., Neuman C. The Kerberos network authentication service (v5). RFC 1510, September, 1993.

[30] Zhu S., Setia S., Jajodia S. LEAP+: Efficient security mechanisms for large-scale distributed sensor networks. ACM Transactions on Sensor Networks (TOSN), 2006, vol.2, no.4, pp:500-528.

[31] Perrig A., Szewczyk R., Tygar J.D., Wen V., Culler D. SPINS: Security protocols for sensor networks. Wireless networks, 2002, vol.8, no.5, pp:521-534.