International Journal of Innovative Computing, Information and Control
Volume **14**, Number **5**, October **2018**

ICIC International ©2018 ISSN 1349-4198
pp. **1947–1957**

AUTHENTICATION METHOD IN SOFTWARE-DEFINED NETWORK BASED ON CIPHERTEXT-POLICY ATTRIBUTES ENCRYPTION

Ali Alshahrani, Khaled Suwais and Basil Alkasasbeh

Faculty of Computer Studies
Arab Open University
P.O. Box 84901, Riyadh 11681, Kingdom of Saudi Arabia
{ a.shahrani; khaled.suwais; bkasasbah }@arabou.edu.sa

Received November 2017; revised March 2018

Abstract. *The controller of Software-Defined Networks (SDN) is lacking well-established authentication controls for accessing terminals to protected networks. This gap may lead to several network attacks gaining full access to the network or revealing sensitive traffic details. In order to resolve this problem, a secure authentication method in SDN is designed and implemented. The design of the authentication method utilizes the concept of Ciphertext-Policy Attributes Based Encryption (CP-ABE). We aim to satisfy the credibility theory by allowing authenticated users with authenticated platforms to fully gain their privileged access. On the other hand, our method allows limited access for authenticated users of unauthenticated platforms. Experimental results show that our authentication method is secure with an acceptable range of access delay and CPU utilization.*

Keywords: Software-defined network, Authentication, Trusted network, Trusted network connect

1. **Introduction.** As Software-Defined Network (SDN) enables new network applications and services, several security concerns have been raised. It is shown that SDN is subjected to different security attacks from different security perspectives. The core idea of SDN is to decouple the control plan and data plan which is implemented in traditional networks. This decoupling aims to increase the level of flexibility of network implementation and control. It also enhances the network programmability and re-programmability [1].

One of the main security concerns of SDN is the absence of an effective access authentication method [1,2]. SDN is lack of user access control function to authenticate the request from network users and platforms. Such issues may allow attackers to gain control over the network and illegally obtain sensitive information.

In this paper, we aim to resolve the problem of security checkup of SDN by proposing an alternative secure access authentication method based on the concept of Attribute-Based Encryption (ABE). This method aims to provide the Trusted Network Connect (TNC) architecture of SDN, a secure authentication procedure that also manages the limited access issue that may arise. Limited access issues are observed whenever a request is required from an authenticated user but the platform that is used by that user is not authenticated.

2. **Related Works.** TNC architecture was first proposed in 2003 [3]. TNC is based on integrating trusted computing technologies with the measurement of the integrity of the platform used by the user who requests access to the SDN network. Unlike traditional

DOI: 10.24507/ijicic.14.05.1947

networks, TNC increases the efficiency of the verification process of the platform. Concerning the flexibility of TNC, it has been shown by [1] that TNC works with existing network technologies (hardware and software). This includes the use of 802.1x and PPP protocols to implement access control functions. However, this technique and many other techniques [4-7] were introduced to control user access to traditional networks based on TNC. Nevertheless, these techniques are not applicable to SDN since SDN decouples the data plan and control plan as in traditional networks.

Many researchers started to focus on the security of access control of SDN. The architecture presented in [8] is a security solution for the controller NOX. It increases the role-based authentication module. Each flow rule is signed and privileges are specified. The main concern of NOX controller is that it is not flexible to re-program or develop, since the core of NOX is implemented in C programming language.

In [9], the authors proposed an enhanced controller that is based on integrating the user registry with user roles to come up with a role-based access control. However, this controller is not capable of handling multi-dimensional access control function. In [10], the authors proposed an authentication mechanism based on host certification. It works in conjunction with the SDN controller. This mechanism reports the status of authentication to the controller, which in turn allows or denies the flow of the traffic. The drawback of this mechanism is that it is costly and complex to implement.

The current SDN architectures suffer from security vulnerabilities [1]. The security access can only authenticate user identity but not the platform used by that particular identity. To resolve the issue of performing efficient authentication in SDN, we propose combining the ABE method in SDN based on the architecture of TNC.

3. **Trusted Network Connect Architecture of SDN.** TNC is an architecture that includes a set of rules and procedures to ensure secure access to protected networks. The overall architecture of TNC is visualized in Figure 1. The architecture of TNC is composed of three main components. The first component is the Access Requester (AR), which is responsible for requesting access to a secure network. The second component is the Policy Enforcement Point (PEP), which is a trusted certification with respect to a particular AR. It also generates decisions. The third component is the Policy Decision Point (PDP), which decides whether a particular AR can access the network. In addition, PDP has its own policies for accessing the network.

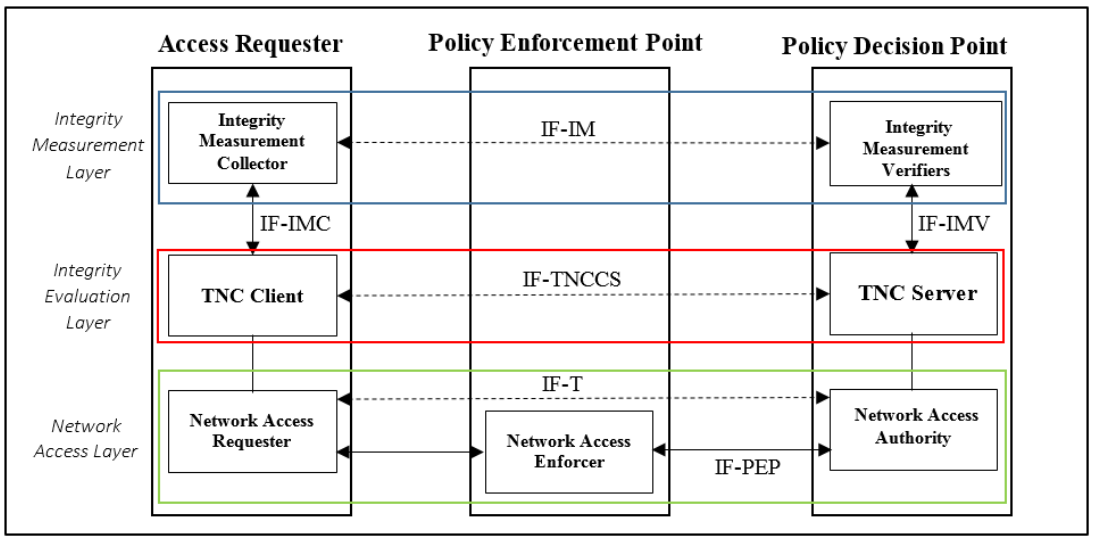

FIGURE 1. TNC architecture

Each component of a TNC includes a set of functions that facilitates secure access to protected networks. Figure 2 shows taxonomy of TNC components and the main functions that are carried out by each component. The Access Requester performs three main functions: Network Access Requester (NAR), Integrity Measurement Controller (IMC) and TNC Client (TNCC). NAR sends a network request and participates in identity authentication. IMC measures the security-based integrity of AR, while TNCC gathers the values of the integrity measurements according to its own IMC.

The Policy Enforcement Point controls the access to the protected network through the Network Access Enforcer (NAE). On the other hand, the Policy Decision Point performs three functions: Network Access Authority (NAA), TNC Server (TNCS), and Integrity Measurement Verifier (IMV). The function NAA determines whether an AR can access network and check with TNCS whether the integrity of AR satisfies the policies embedded in NAA. The TNCS exchanges data with the TNCC to collect the decision of AR. TNCS will accordingly formulate the final decision and send it to NAA. Last, IMV verifies the integrity of AR.

Generally, TNC works under three layers: the Integrity Measurement Layer (IML) that works at the level of collecting integrity values of each component; the Integrity Evaluation Layer (IEL) that assesses the integrity of AR component with respect to the adopted access policy; the Network Access Layer (NAL) that is responsible for authenticating the identities and assists for establishing proper network connections. However, Figure 3 presents the code snippets of the main flow of integrity checkup in TNC. The initialization

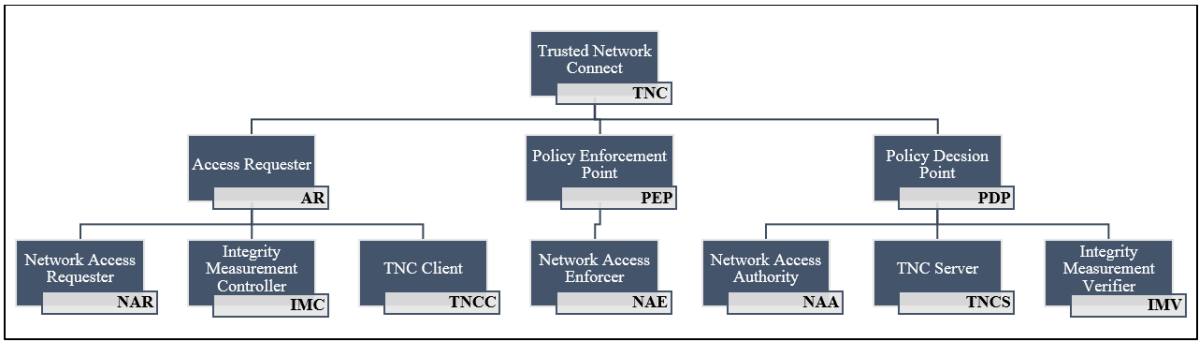

FIGURE 2. Taxonomy of TNC functions

1. **Initialization**
   1.1. *TNCC initializes IMC.*
   1.2. *TNCS initializes IMV.*
2. **NAR sends access request to PEP.**
3. **Verification**
   3.1. *PEP send a request to PDP.*
   3.2. *PDP verify identity according to AR.*
   3.3. *PDP verify platform integrity according to security policies.*
   3.4. *Make final decision (enable/disable).*
4. **Implementation**
   4.1. *PDP send decision results to PEP.*
   4.2. *PEP implement the decision.*
5. **Connection process is completed**

FIGURE 3. Code snippet of integrity checkup in TNC

of IMC by TNCC is represented by gathering values of the integrity measurements, while IMV is initialized by a set of embedded policies. However, the fundamental concept of TNC lacks effective security protocol support. The identity of the user as a requester can be verified but it lacks support of verification of users' platform as part of the theory of credibility.

4. Secure Authentication Method in SDN.

4.1. Attributes Based Encryption (ABE).

The concept of encrypting access control was introduced firstly by Sahai and Waters in [11] in the form of new schema known as Attributes Based Encryption (ABE). The ABE is classified as a public-key encryption in which the secret keys of a user and the ciphertext are dependent upon attributes [12]. The decryption of the ciphertext in ABE is only possible if the set of attributes of the user's secret key matches the attributes of the ciphertext. Practically, ABE involves encrypting the attributes rather than the data [13].

In the original work of [11], the authors presented the concept of a *threshold*, and the ciphertexts are labeled with a set of attributes. On the other hand, the user's secret key is associated with a *threshold* value and a subset of attributes. Decrypting a ciphertext is possible if the attributes of ciphertext and the secret key are matched under the predefined *threshold* value. However, the model is modified later by several scholars to avoid some performance's limitations [14,15].

The access structure of ABE is defined as monotonic tree access structure. The definition of the monotonic structure assumes that the set $\{P_1, P_2, \ldots, P_n\}$ is a set of parties. The collection $\ell \subseteq 2^{\{P_1,P_2,\ldots,P_n\}}$ is monotonic if $\forall X, Y$: if $X \in \ell$ and $X \subseteq Y$, then $Y \in \ell$. A monotonic access structure is a monotonic collection ℓ of non-empty subsets of $\{P_1, P_2, \ldots, P_n\}$, i.e., $\ell \subseteq 2^{\{P_1,P_2,\ldots,P_n\}} \backslash \{\emptyset\}$. The sets in ℓ are identified as authorized sets, while the sets not in ℓ are identified as unauthorized sets.

There are two primary types of ABE: Key-Policy based ABE (KP-ABE) and Ciphertext-Policy based ABE (CP-ABE). In KP-ABE, the ciphertext (C) is associated with a given set of attributes (att) as shown in Equation (1):

$$C \rightarrow Set_{att} \in Set_{ATT \rightarrow P} \tag{1}$$

where $\mathrm{Set_{att}}$ is a subset of attributes from the general set of attributes ($\mathrm{Set_{ATT \rightarrow P}}$) which is based on the policy (P), while the secret key is associated with the access policy (P). At the encryption side, we define a set of attributes needed to allow proper ciphertext decryption process. However, it is the responsibility of the trusted authority to generate user's secret key and the attributes for which the secret key can be used.

Unlike KP-ABE, the ciphertext in CP-ABE is associated with the access policy, and the encrypter is the one responsible for specifying the policy related to the data that can be decrypted. The secret key of the user ($\mathrm{SK_U}$) in CP-ABE is associated with a specific set of attributes as presented in Equation (2). Figure 4 visualizes the general scheme of CP-ABE model.

$$SK_U \rightarrow Set_{att} \in Set_{ATT \rightarrow P} \tag{2}$$

The CP-ABE scheme involves the execution of four primary algorithms: *Setup*, *Encrypt*, *KeyGen*, and *Decrypt* algorithms. In the *Setup* algorithm, no input parameters are considered except the implicit security parameters. The output of this algorithm is the Public Key (PK) and the Master Secret Key (MSK).

Consequently, the encryption algorithm (*Encrypt*) takes three input parameters: (PK), the message (M), and the access structure ℓ over the set of the predefined universe of attributes. The *Encrypt* algorithm encrypts (M) to produce the ciphertext (C). Only a

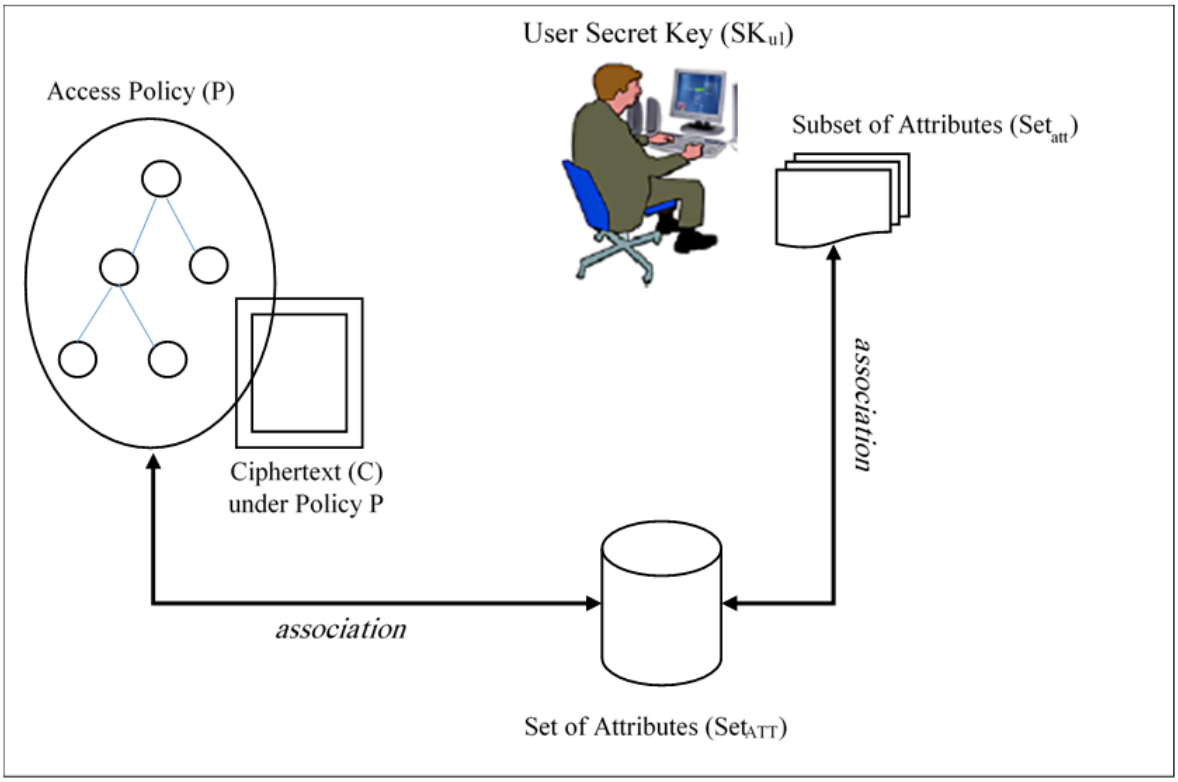

FIGURE 4. General scheme of CP-ABE model

user that possesses a set of attributes that satisfies monotonic structure will be able to decrypt the message, under the assumption that (C) contains ℓ.

The third algorithm is the key generation *(KeyGen)*, which takes the (MSK) and the set (S) of attributes that are associated with (MSK) as input parameters. This algorithm produces the user's secret key (SK_{u_i}). Lastly, the decryption algorithm (*Decrypt*) takes three input parameters: (PK), (C), and (SK_{u_i}). Note that (C) contains the access policy ℓ, while the secret key (SK_{u_i}) is associated with a set of predefined attributes. If the set of attributes satisfies the access policy in ℓ, the algorithm decrypts the ciphertext and returns the message (M).

4.2. **Secure authentication method based on CP-ABE.** The TNC architecture adopted in our secure authentication method is based on the recently published secure TNC architecture in [1] as shown in Figure 5. This architecture represents the network access requester of the original TNC as an access device. On the other hand, it represents the TNCC as access client for providing network access authentication service. At the PEP side, the NAE is represented as OpenFlow switches which are part of the data plan of SDN.

The core of TNC architecture is located at the PDP, where the TNCS works as access verifiers service for evaluating the integrity measurement of user identity and its platform. The access verifier service sends the decision to the controller, for controlling and managing the network connection, and denies the access of unauthorized users. Figure 6 presents the code snippet of the access flow of the proposed TNC.

According to the structure of TNC, we present a secure authentication method based on CP-ABE. The authentication method is integrated as a built-in function at the Access Verifier Service (AVS) as described in Figure 7. The access request received from the access device is evaluated and verified against a particular network access policy.

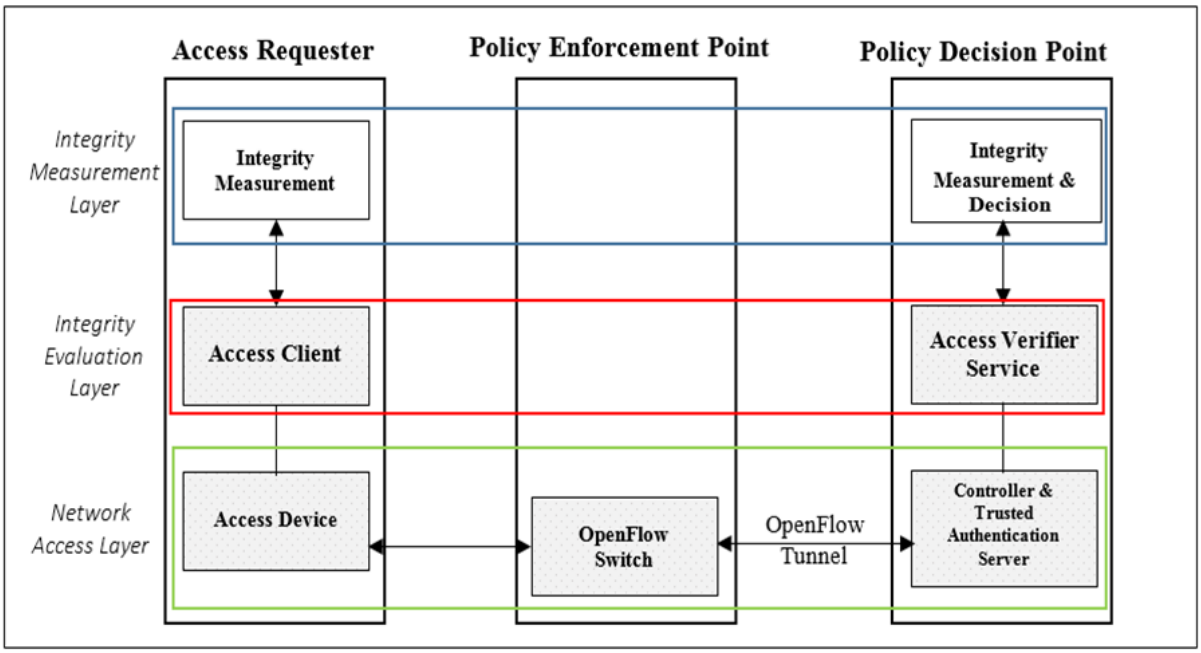

FIGURE 5. TNC architecture of [1]

1. User Login the access client

1.1 *Access client obtain user and platform authentication data*

1.2 *Formatted data structure is generated according to access protocol*

2. Access client send access data to access server

3. Evaluate the integrity measurement of user and platform

3.1 *Formulate the final decision*

3.2 *Send the decision to controller*

4. Controller generate flow table

4.1 *Send flow table to OpenFlow switches*

4.2 *OpenFlow switches forward network packets accordingly.*

FIGURE 6. Code snippet of access flow in TNC

To prevent unauthorized access to the policy, the policy is encrypted using the controller's Public Key (PK). Therefore, the attributes of the user and platform will be only verified if the user uses the proper private secret key (SK_{u_i}). The general procedure of our TNC-based SDN secure authentication methods is presented in Figure 8.

The access device first logs in the access client, which obtains the login authentication details of the user and the user's platform. These details include: user's private secret key (SK_{u_i}), the list of attributes ($_{att}$) of user U (U_{att}), and the PCR code of the user's platform. The key SK_{u_i} is derived from the Master Secret Key (MSK) of the controller (C) as shown in Equation (3).

$$SK_{u_i} \in MSK_C \quad (\forall i \in n) \tag{3}$$

where n denotes the list of authorized users of the protected network. The user's attributes are assigned by the controller according to the role of the user in a particular access policy,

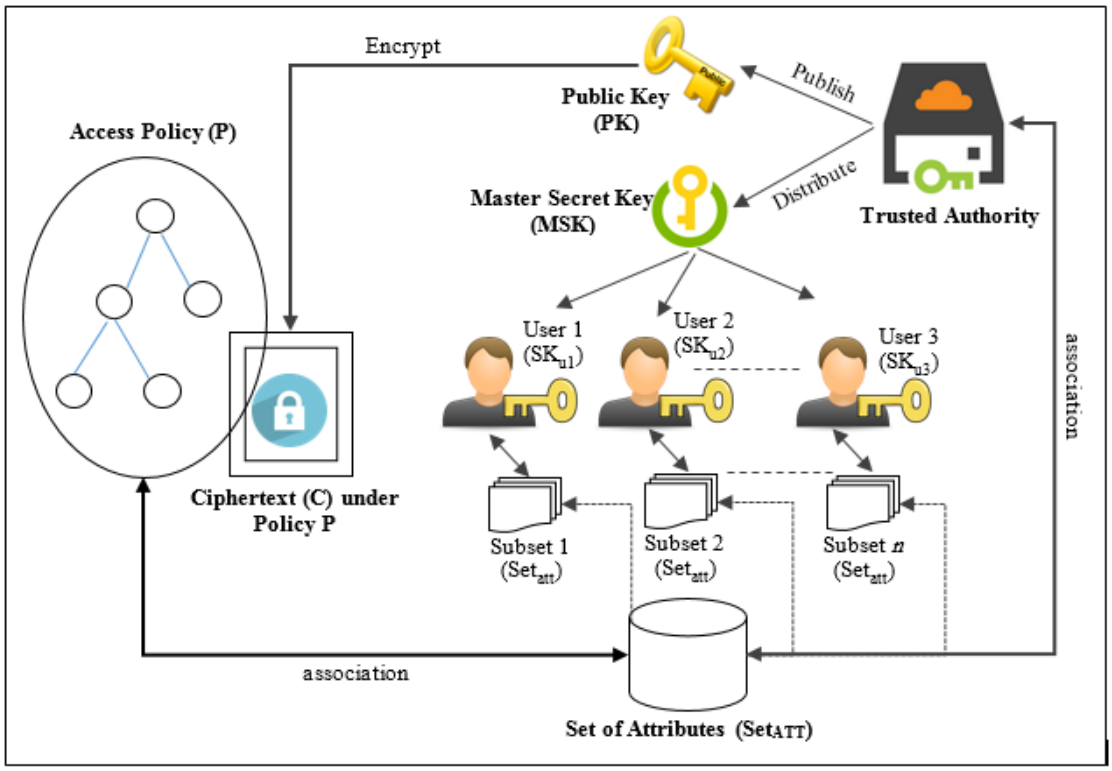

FIGURE 7. The structure of the secure authentication method of AVS in TNC-based SDN network

as shown in Equation (4).

$$U_{att_i} \in C_{att_s} \quad (\forall i \in u \text{ and } s \in P_{att}) \tag{4}$$

where (s) presents the list of attributes derived from the set of attributes specified in the access policy (P).

The last item of the authentication details is the PCR code of the user's platform. This code is awarded for authorized platforms by the controller. The PCR codes are also stored in the encrypted access policy as part of the set of attributes. The importance of this code arises from the needs of flexible limited access policy for authorized users of unauthorized platforms.

According to the authorization information obtained from the access device, the access client sends the authentication information to the AVS through the OpenFlow switches. The AVS evaluates the integrity of the received information against the access policy. Note that access policy (P) is encrypted (using encryption algorithm E) by the public key of the controller ($E_{PK}(P)$). If the secret key (SK_{u_i}) is correctly used, the access policy will be decrypted and then compared to the user and platform's attributes.

Upon completing the verification process of the access request, the final decision (R) is passed to the controller for generating the corresponding flow table. Note that, limited access is granted for users who could satisfy the user's authentication requirements but failed to authenticate its platform. This feature is very critical for many network services as it grants users different levels of flexibility in accessing protected network using different platforms. Figure 9 presents the code snippet of our secure authentication procedure in TNC-based SDN network.

5. **Results and Discussions.** Our authentication method is built over the TNC architecture of SDN network proposed in [1]. Our contribution is mainly observed in integrating

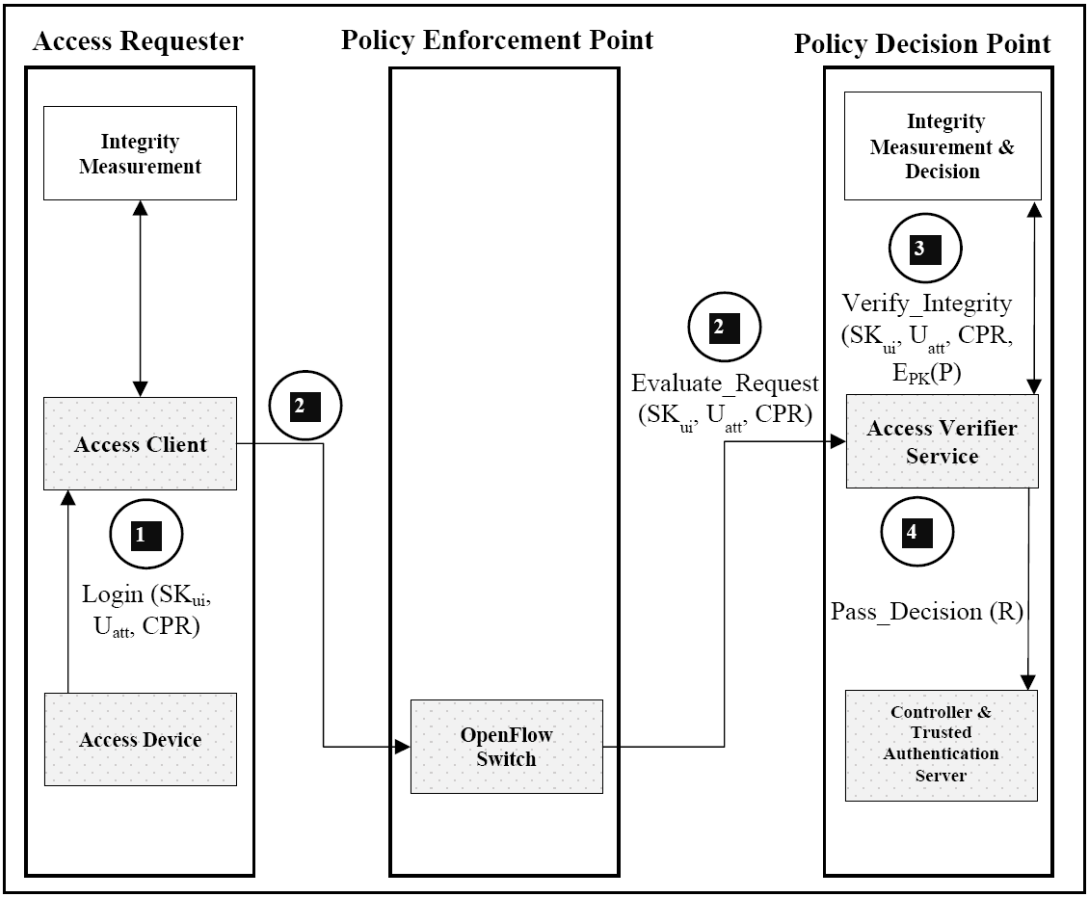

FIGURE 8. Secure authentication procedure in TNC-based SDN network

1. Initialization

1.1 *Controller generates PK, MSK*

1.2 *Encrypt policy* $E_{PK}(P)$

1.3 Distribute SK_U for users

1.4 Distribute PCR codes for platforms

2. Request Access

2.1 *Login*(SK_U, U_{att}, *CPR*)

2.2 *Evaluate_Integrity*(SK_U, U_{att}, *CPR*)

2.3 *Forward request to AVS*

3. Evaluate_Request (SK_U, U_{att}, CPR)

4. Verify_Integrity (SK_U, U_{att}, CPR, $E_{PK}(P)$)

4.1 *Decrypt policy* $D_{SK_U}(E_{PK}(P))$

4.2 *R= Compare* ((U_{att}, *CPR*), *P*)

5. Pass_Decision (R)

FIGURE 9. Code snippet of our authentication procedure in TNC-based SDN network

CP-ABE with TNC to achieve reliable authentication procedures in TNC-based SDN networks. Several researchers have also proved the security of CP-ABE against cryptanalysis attacks [16-18].

From the performance point of view, the proposed method was tested in an environment almost similar to the environment identified in [1]. The experiment environment is described in Table 1. The hardware included in the environment includes one access device, two access servers, and OpenFlow switches and controller.

Table 1. Experiment specification

Item	Specification
Access Device	Core i7 quad-core CPU, 6GB memory, Ubuntu 14.04, TSS, OpenSSL
Server	Core i7 quad-core, 6GB memory, Ubuntu 14.04
Controller	Floodlight 1.0

The CP-ABE is implemented using the set of tools called cpabe package developed by [19], which is available online under the open source GPL license. The package provides four command lines as follows:

- **cpabe-setup**: generates (PK) and (MSK).
- **cpabe-keygen**: generates (SK_{u_i}) for attributes.
- **cpabe-enc**: encrypt a file under the policy (P).
- **cpabe-dec**: decrypt a file using (SK_{u_i}).

To examine the performance impact of applying our secure authentication method on TNC-based SDN network, we compare our results with the results published by [1]. The performance tests focus on two issues: delay time and CPU utilization. We measure the delay time caused by the access device to connect to the server. The CPU load and utilization of our method also measure at the server side which carry out the CP-ABE authentication activities. The results presented in Figures 10 and 11, represent the latency measurements and CPU utilization of our method, respectively. Note that applying the CP-ABE authentication method played a role in increasing the delay time and CPU utilization. Nevertheless, this falls within the acceptable range.

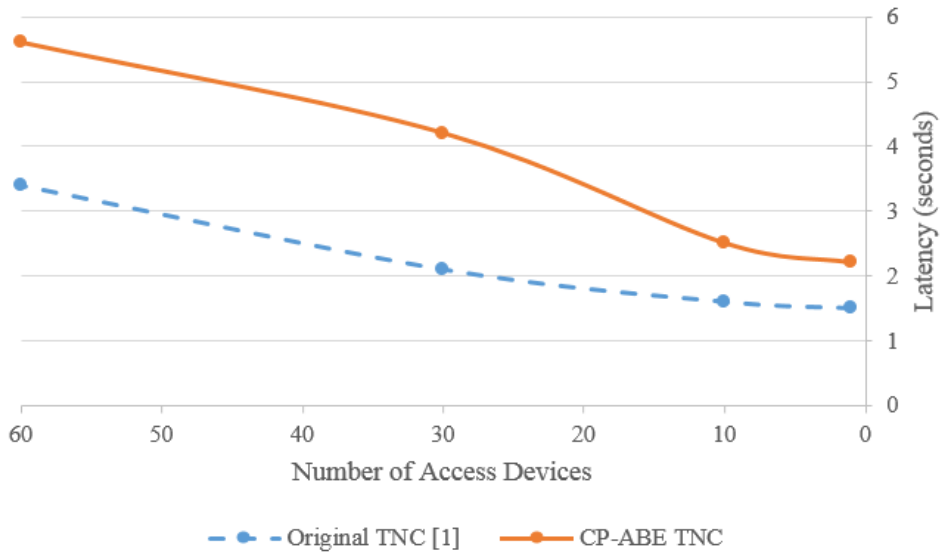

Figure 10. Variation in processing latency

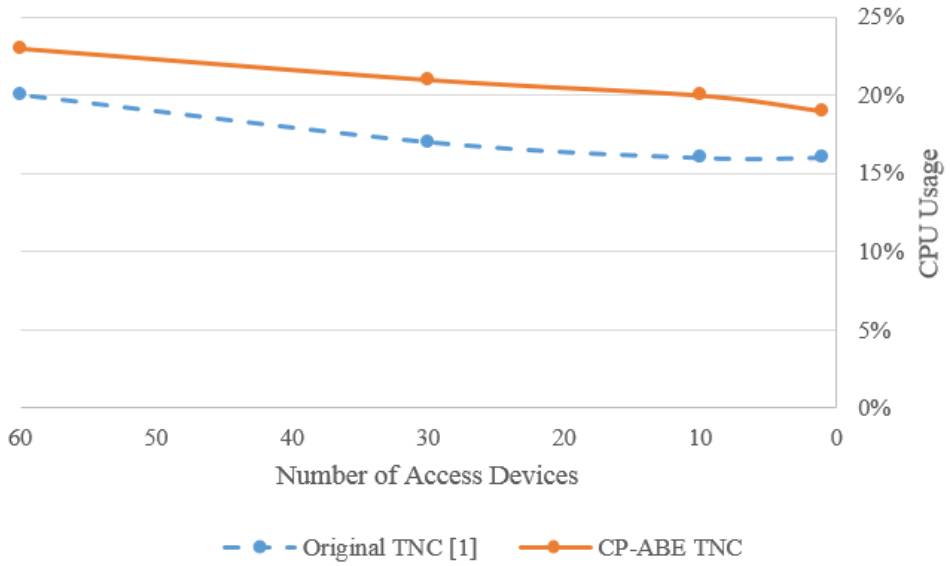

FIGURE 11. Variation in CPU usage

6. **Conclusion.** In this paper, we designed a secure authentication access method in TNC-based SDN architecture. The presented method involves the use of ciphertext-policy attributes based encryption for authenticating access requests to protected SDN networks, securely. The experimental results showed that our method could enhance two levels of authentication service: full access for authenticated users of authenticated platforms, and limited access for authenticated users of unauthenticated platforms, according to a particular predefined access policy. From the performance perspective, the results showed that our method could increase the access latency and CPU utilization compared to the adopted TNC-based SDN architecture in [1]. Nevertheless, the results are found reasonable and fall within the acceptable range. As future direction, parallel computing model might be constructed and implemented to enhance the CPU utilization and minimize access latency.

Acknowledgment. This work is supported and funded by the Arab Open University, Saudi Arabia.

REFERENCES

[1] J. Liu, Y. Lai and Y. Chen, A trusted access method in a software-defined network, *Simulation Modelling Practice and Theory*, vol.74, pp.28-45, 2017.
[2] I. Alsmadi and D. Xu, Security of software defined networks: A survey, *Computer and Security*, vol.53, pp.79-108, 2015.
[3] TCG Group, TCG specification architecture overview, *TCG Specification Revision 1.4*, 2007.
[4] Y. Lin, *The Research and Implementation of Network Access Control System*, University of Electronic Science and Technology of China, 2014.
[5] A. Luo, C. Lin, Y. Wang et al., Security quantifying method and enhanced mechanisms of TNC, *Chin. J. Comput.*, vol.5, pp.887-898, 2009.
[6] F. Yan, J. Ren, K. Dai et al., Design and implementation of secure authenticated protocol based on TNC, *Comput. Eng.*, vol.12, pp.160-165, 2007.
[7] E. Liu, *Research on the Identity in Heterogeneous Wireless Integrated Networks*, PLA Information Engineering University, 2009.
[8] P. Porras, S. Shin, V. Yegneswaran et al., A security enforcement kernel for OpenFlow networks, *Proc. of the 1st Workshop on Hot Topics in Software Defined Networks*, Helsinki, pp.121-126, 2012.

[9] T. Sasaki, Y. Hatano, K. Sonoda et al., Load distribution of an OpenFlow controller for role-based network access control, *Proc. of Network Operations and Management Symposium*, Hiroshima, Japan, pp.1-6, 2013.

[10] D. Mattos and O. Duarte, Authflow: Authentication and access control mechanism for software defined networking, *Ann. Telecommun.*, pp.1-9, 2014.

[11] A. Sahai and B. Waters, Fuzzy identity-based encryption, *Advances in Cryptology – Eurocrypt. Lecture Notes in Computer Science*, vol.3494, pp.457-473, 2005.

[12] M. Surya and N. Anithadevi, Single sign-on mechanism using attribute-based encryption in distributed computer networks, *Proc. of International Conference on Graph Algorithms, High Performance Implementations and Its Applications*, Coimbatore, India, vol.47, pp.441-451, 2015.

[13] S. Kumar, R. Lakshmi and B. Balamurugan, Enhanced attribute based encryption for cloud computing, *Proc. of International Conference on Information and Communication Technologies*, Kerala, India, vol.46, pp.689-696, 2015.

[14] M. Pirretti, P. Traynor, P. Mcdaniel and B. Waters, Secure attribute-based systems, *Proc. of the 13th ACM Conference on Computer and Communications Security*, Alexandria, USA, 2006.

[15] A. Shamir, How to share a secret, *Commun. ACM*, vol.22, no.11, pp.612-613, 1979.

[16] J. Bethencourt, A. Sahai and B. Waters, Ciphertext-policy attribute-based encryption, *Proc. of IEEE Symposium on Security and Privacy*, Berkeley, USA, 2007.

[17] H. Chung, P. Wang, T. Ho et al., A secure authorization system in PHR based on CP-ABE, *Proc. of the 5th IEEE International Conference on E-Health and Bioengineering*, Iari, Romania, 2015.

[18] H. Zhou and Q. Wen, A new solution of data security accessing for hadoop based on CP-ABE, *Proc. of the 5th IEEE International Conference on Software Engineering and Service Science*, Beijing, China, 2014.

[19] J. Bethencourt, A. Sahai and B. Waters, *The CPABE Toolkit*, http://acsc.csl.sri.com/cpabe/.