IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON HUMAN-CENTERED SMART SYSTEMS AND TECHNOLOGIES

Received November 30, 2017, accepted December 26, 2017, date of publication January 2, 2018, date of current version March 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2788443

A New Inter-Domain Information Sharing Smart System Based on ABSES in SDN

JINGJING XU, HANSHU HONG, GUOFENG LIN, AND ZHIXIN SUN

Laboratory of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Corresponding author: Zhixin Sun (sunzx@njupt.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61373135, Grant 61672299, Grant 61702281, and Grant 61602259.

ABSTRACT With the continuous innovation and development of agent technology, the application of smart technique is gradually changing to intelligent and human-centered technology. Also, intelligent technique is involved in many fields, and software defined network (SDN) is one of them. SDN is one of the most popular and most promising new technologies in the current network area, which has particular advantages over other traditional networks. However, most of the research work focus on the efficiency and function of SDN, while the access control and ciphertext search function in SDN have not been taken into full consideration. In this paper, we propose a new smart system based on SDN. It also equips with the functions of fine-grained access control and searchable encryption function, which combines a novel attribute-based searchable encryption scheme (ABSES) with SDN architecture. Our construction realizes inter-domain information sharing, fine-grained access control, and ciphertext searchable function. The proposed ABSES effectively ensures the security of ciphertext stored in data center, and that users cannot obtain unauthorized information or illegal network resources without any certification. It reduces network bandwidth and local resources, while improving the scalability and flexibility of SDN access control at the meantime. Finally, our ABSES is proved to be correct and secure under chosen-keyword attack. The comparison with other representative attribute-based searchable encryption schemes demonstrates that our ABSES has somewhat better performance. The proposed new smart system can be used in human life and add more convenience to our daily life.

INDEX TERMS SDN, attribute-based searchable encryption, inter-domain information sharing, human-centered, smart system.

I. INTRODUCTION

Under the tide of the rapid development of computer network technology, the simple functions of numerical calculation and problem solving of computer network cannot meet the requirement of computer function. People demand that the service function of computer network technology should be intelligent and human-oriented, but there are still a series of problems in the current computer network technique. The problems in network information security are particularly prominent. More specifically, attention should be given to the inter-domain information sharing security issue in SDN. SDN originated from the Clean Slate research project at Stanford University in 2006 [1]. In 2009, Nick McKeown formally proposed the SDN concept [2]. Since then, SDN has been one of the most influential network architectures in the IT industry, arousing widespread concern in academia and industry. SDN utilizes the layered idea of separating the data layer from the control layer and uses an open unified interface (OpenFlow [3], etc.) between the two layers to interact. As a result, SDN technology is one of the most promising network technologies that can effectively reduce equipment load and help network operators better control infrastructure while reducing overall computation costs at the same time.

SDN simplifies network management and shortens the refactoring cycle and has been researched and applied in the fields of routing decision-making, network virtualization, wireless access, cloud computing data center network, etc. However, there still remain some security issues which have not been fully tackled at the same time. For instance, SDN controllers introduce the signal peer invalid problem. And attackers may use SDN and its open interface to control applications maliciously. Also, there are some security problems

because of the open environment of PC or server and so on. If there are not suitable solutions, these issues will certainly hinder the deployment of SDN in the future.

At the beginning of the design and development of most controllers, researchers mainly focused on functions such as resource scheduling and rules delivery without regarding the controller's own security issues as the key research contents. The controller collects and stores the user's status and traffic data, including the user's location and browsed content, etc., which can be easily called by the open API. Thus, the user's individual information will be disclosed in this way. Therefore, the SDN controller should ensure that privacy information is provided only to authorized users so that network statistics will not be stolen by attackers due to remote administration.

To better solve the problems discussed above, in this paper, we propose a new smart system aiming to deal with the access control issue in SDN based on a novel attribute-based searchable encryption scheme. In our system, attribute sets are used as a condition of access control to ensure that users cannot obtain unauthorized information or illegal network resources. The main contributions are summarized as follows:

1) A novel attribute-based searchable encryption scheme is presented, and is applied to SDN. The scheme implements the sharing of inter-domain information and fine-grained access control. At the same time, it improves the scalability and flexibility of secure SDN access control.
2) In this scenario, users use their own access structures to obtain the secret key. If the ciphertext attribute set meets the user's access policy, the corresponding ciphertext is returned, otherwise it is not returned. It saves system computing time without adding authorization and verification process.
3) We indicate that the SDN controller encrypts the user's information in the domain, which is managed by itself. And then the SDN controller interacts with other controllers to ensure the confidentiality of the information during transmission.
4) The scheme helps users search ciphertext without decrypting them, which improves the efficiency of information retrieval, and saves network bandwidth and local resources.
5) The proposed system is human-oriented, in which authorized users can search for the information they want by some keywords. It is suitable and intuitive for users no matter what their abilities are. For users, they need smaller computing costs and resources.

The rest of paper is organized as follows: Section II introduces the related works. In Section III, the essential preliminaries are given. The system model is presented in Section IV. In Section V, we describe attribute-based searchable encryption scheme and present how to use it to improve the security in SDN. Comprehensive analysis on the security and performance of the proposed scheme are provided in Section VI. The paper is concluded in Section VII.

II. RELATED WORK

Software Defined Network (SDN) is one of the most popular network technologies at present. It is an emerging network architecture with the particular advantages over traditional networks. SDN can effectively simplify the management of the network and provide users with good network programmability. Because of this, it brings great convenience to people. At the same time, there are some security issues and threats that cannot be ignored. Although the SDN architecture has an advantage of efficient network and flexible reconfiguration, the security issue still hinders the development of SDN.

Ethane [4] is an early form of SDN, which is a more practical network architecture that provides centralized network security management. It turns the encrypted control header with horizontal transmission into encrypted control message with vertical transmission in the SANE network architecture. Moreover, control information is transmitted between the controller and the switch via the secure channel. Ethane provides a model for user access and management, yet user management is too simple to meet the security demand for large-scale deployments. To deal with this problem, IETF [5] clearly pointed out that users can be provided with authorization and authentication mechanisms to enable multiple organizations to access network resources.

Klaedtke *et al.* [6] proposed an access control program for SDN controller, which is mainly divided into four parts: separation, execution, conflict and authorization. Firstly, in order to distinguish who can access the network and how to access it, the various components of the network should be separated logically. Secondly, network users have their own security requirements for accessing network components, which means it is necessary for mechanisms to support, express and enforce these requirements. Furthermore, user conflicts when sharing network components must be resolved and the authorization to access component should be permitted. Jäger *et al.* [7] concentrated their work on the environment combining cloud computing with network function virtualization (NFV) /software defined network (SDN) and proposed a multi-level access control system. The basic idea is to limit the executable instruction set of telephone components. And the access control of the application layer and control layer is mainly analyzed.

Traditional networks usually require specific operations or protocol support from the host side. Nevertheless, the capabilities of the host or device are not universal and some of them may not be under the certification process which is required to join the network. In order to solve this problem, Kamath *et al.* [8] proposed an authentication framework (SAFE) for SDN to isolate unauthenticated devices and provide access control with flexible authentication methods, which will not affect the isolation of untrusted hosts and the security of network resources. By systematically separating authentication and access control, the network can authenticate multiple terminals based on its capabilities

and enforce access control and policies on a single unified platform of SDN. SAFE combines MAC-based identity and location awareness, which is the port number and the switch in the SDN called DPID. With this approach, unauthenticated devices can be isolated and still be able to communicate with their reachable authentication servers.

For the cause of preventing eavesdropping, controller impersonation, unauthorized connection, and other attacks, a channel with authentication and secure transmission function must be established between the controller and the switch. The OpenFlow [9] specification recommends that the TLS/DTLS protocol should be used for secure encryption. Specifically, authentication between the switch and controller through certificate swapping and encryption of the interaction data through the negotiation key should be used to achieve secure connection. This is also the first time that SDN is combined with an encryption protocol to realize authentication and access control.

In the encryption scheme, initially proposed technology that is associated with the identity authentication is Identity Based Encryption (IBE) mechanism based on bilinear pairings by Shamir [10] and Boneh and Franklin [11], which directly uses the user's identity as the public key, and so that it is unnecessary for resource providers to query the user's public key certificate online. Then, Sahai and Waters [12] proposed the attribute-based encryption (ABE) mechanism based on IBE technology. In this encryption system, the encrypting party does not need to know the specific identity information of each decrypting party, but only needs to master a series of descriptions of their attributes. Subsequently, the message will be encrypted with the access structure defined by specific attributes during the encryption process, and the ciphertext can be decrypted when the user's key satisfies this access structure. Therefore, the ABE scheme is closely related to the access control policy. The basic ABE [12] can only represent the ''threshold'' operation of the attribute, and the threshold parameter is set by the authority. The access control policy cannot be determined by the sender. However, many realistic applications need to support the AND, OR, threshold and non-operation of attributes according to flexible access control policies so as to ensure that the sender specifies an access control policy when encrypting. Due to the reason that basic ABE cannot support flexible access control policies, Goyal *et al.* [13] proposed a Key policy attribute-based encryption(KP-ABE) mechanism for setting access policies by receivers to support attribute AND, OR and threshold operations. Bethencourt *et al.* [14] proposed a ciphertext policy attribute-based encryption(CP-ABE) mechanism for the sender's ciphertext access policy. In KP-ABE, when the key is generated and the ciphertext is associated with a set of attributes, the user's decryption key is associated with the access structure. The user can decrypt the ciphertext only when the attribute satisfies the access structure. In CP-ABE, data is encrypted according to the access policy, which describes the combination of required attributes. The user's secret key contains the attribute value owned by the user. If the user's key matches the access policy, he can encrypt the document.

Attribute-based encryption mechanism can flexibly represent access control policies, and it has been widely used in the field of cloud storage security. While searchable encryption (SE) [15] mechanism provides the encryption and ciphertext retrieval function as the result that the server cannot eavesdrop on the user's personal data but can return the target ciphertext file according to the query request. This will ensure the security and privacy of user data and not unduly reduce the query efficiency. Therefore, combining attribute-based encryption scheme with searchable encryption scheme, namely, attribute-based searchable encryption (ABSE), can not only achieve fine-grained access control, but also improve the retrieval efficiency, and finally achieve the purpose of efficient data sharing.

The great majority of SE schemes are limited to single-user settings or multi-user settings with coarse-grained access control. However, in the cloud environment, it is more practical to have multiple users with shared data access. To bridge this gap, Wang *et al.* [16] presented a searchable encryption for fine-grained access control under multi-user settings. The proposed scheme requires less computational cost on the client side, but provides fine-grained access control for authorized users under the new hybrid architecture. The user collusion defense scheme is implemented with the help of private cloud by the way of combining symmetric encryption and CP-ABE. The analysis demonstrated that this scheme effectively solves the common problem of key sharing. It exposes the attribute key to the public cloud without affecting the security of the system, and provides dynamic and efficient user revocation. Yang [17] introduced an ABE-based keyword search scheme (ABKS) and applied it to the electronic health system. The proposed solution supports fine-grained authorization and flexible revocation in semi-trusted cloud servers. However, this scheme generates a unique additional key pair for each user in the system and the generation of a search index also involves each user's public key. When the number of users is large, a considerable computational burden will be brought. Sun *et al.* [18] concentrated their work on a multi-user and multi-owner scenario and proposed an attribute-based keyword search scheme. In their construction, efficient user revocation (ABKS-UR) function was introduced and scalable fine-grained search for authorization was achieved (i.e. file-level). This approach allows multiple owners to independently encrypt and outsource their data to the cloud server. Their research work demonstrated that users can generate their own search capabilities without relying on trusted authorities. In addition, the heavy system update workloads during user revocation can be entrusted to an intelligent, semi-trusted cloud server by introducing proxy re-encryption and delay re-encryption technology.

Miao *et al.* [19] applied ABKS to modern medical systems and indicated the efficiency and security of the scheme. Wang *et al.* [20] introduced some path breaking

work in attributes and keyword vectors to optimize the decryption efficiency. Dong *et al.* [21] proposed a lightweight ABKS scheme, whose application is well suited to networks with limited computing resources (e.g. mobile networks). Li *et al.* [22] investigated the problem of search authorization in the cloud and proposed a secure ABKS scheme, which not only achieves connectivity and confidentiality of the trapdoor, but also defends collusion attacks. Yousefipoor *et al.* [23] proposed a novel ABKS scheme in attribute-based cryptography and fuzzy search tokens. Chaudhari and Das [24] proposed an anonymous attribute-based searchable encryption (A2BSE) scheme that allows users to retrieve only a portion of the document. Fan and Liu [25] proposed a verifiable attribute-based encryption scheme that supports multi-keyword search. Wang *et al.* [26] proposed a new encryption algorithm based on CP-ABE, which preserves fine-grained access control inherited from ABE system and supports both hiding strategy and fast keyword search. They also proved that these schemes are all safe under the Keyword Guessing Attack (KGA).

He *et al.* [27] first combined attribute-based encryption with SDN and proposed a hierarchical attribute-based access control scheme that combines hierarchical identity-based encryption with CP-ABE. Their efficient scheme achieves scalability and flexibility, and implements fine-grained access control. At the same time, an authentication protocol is proposed to enhance the controller in SDN and so as to flexibly manage users, devices, and data flows.

In this paper, we propose a new smart system based on a scheme that is different from the above schemes. Our research combines attribute-based searchable encryption with SDN, and authorize different users through the set of ciphertext attributes. This scheme can provide fine-grained access control while reducing the calculation time of the certification process, and eventually avoid illegal access to network resources by unauthorized users. Meanwhile, the ciphertext of user information can be searched so that the inter-domain information can be shared with high retrieval efficiency. Accordingly, the network bandwidth and local resources can be saved at the cost of encryption.

III. PRELIMINARIES

A. BILINEAR PAIRING

Let G_1, G_2 be two multiplicative cyclic groups of prime order p. The map $e : G_1 \times G_2 \rightarrow G_2$ is a bilinear map with the following properties:

1) *Bilinearity:* For arbitrary $x, y \in G_1$, and $a, b \in Z_q$, we get $e\left(x^a, y^b\right) = e\left(x^b, y^a\right) = e\left(x, y\right)^{ab}$.
2) *No-Degeneracy:* Given $x, y \in G_1$, we have $e\left(x, y\right) \neq 1$.
3) *Computability:* Given $x, y \in G_1$, there exits an efficient algorithm to compute $e\left(x, y\right)$.

In addition, there are some other features of the bilinear pairing: Given three random elements $x, y, z \in G_1$, where G_1 is a bilinear group, then $e\left(x \cdot y, z\right) = e\left(x, z\right)\left(y, z\right)$, $e\left(x, y \cdot z\right) = e\left(x, y\right)\left(x, z\right)$.

B. ACCESS STRUCTURE

Let γ be an access tree used to describe the access structure. Each non-leaf node in the tree can be used as a threshold represented by the node's children and threshold value. If num_x denotes the number of sub nodes of node x, and k_x is a threshold, then $0 < k_x < num_x$. If $k_x = 1$, the threshold is "OR" gate. If $k_x = num_x$, the threshold is "AND" gate. Each leaf node is represented as an attribute with threshold $k_x = 1$. We represent $parent(x)$ as the parent of node x, and $att(x)$ is denoted as the attribute associated with the leaf node x. For each node y, whose parent node is x, we describe its index with $index\ (y)$ by the number $\{1, 2, \cdots, num_x\}$. Therefore, for a given arbitrary key, the index is uniquely assigned to nodes in the access tree. That is to say, if $parent\ (y) = parent\left(y'\right)$, then $index\ (y) \neq index\left(y'\right)$.

Let the root node of the access tree γ be r, and γ_x be the subtree whose root node is x in the tree γ. If the attribute set A satisfies the access tree γ_x, $\gamma_x\ (A) = 1$. $\gamma_x\ (A)$ may be recursively computed by the following formulas: if x is a non-leaf node, we calculate $\gamma_{x'}\ (A)$ for all child nodes x' of node x. $\gamma_x\ (A)$ returns 1 if and only if the number of child nodes which return 1 is at least k_x. If x is a leaf node, $\gamma_x\ (A)$ returns 1 if and only if $att(x) \in \gamma$.

C. DECISIONAL BILINEAR DIFFIE-HELLMAN ASSUMPTION

Given a tuple $\left(g, g^a, g^b, g^c, r\right)$, where $a, b, c \in Z_p$ and they are unknown. Let $r \in G_2$, and g be a generator of cyclic group G_1. We should decide whether $r = e\ (g, g)^{abc}$.

IV. MODELS AND DEFINITIONS

A. SYSTEM ARCHITECTURE

The whole system consists of five components: system authority, domain authority, SDN controller, user and access equipment. The system architecture is shown in FIGURE.1.

1) *System Authority:* The system authority and domain authority are all trusted third parties. The system authority is responsible for generating public parameters and master keys, and managing each domain authority.
2) *Domain Authority:* A domain authority manages the user's attributes in its domain and distributes the private key to the users. When a new user wants to join the system, the domain authority is responsible for checking the legitimacy of the user's attributes. Each domain in SDN manages different attributes with different functions.
3) *SDN Controller:* A SDN controller is in charge of managing the user's access requests and network flows. It handles the sensitive information generated by users, devices or itself. The SDN controller has public parameters, which encrypt the user's information in the domain, and generate the index of keywords. At the same time, it is responsible for the interaction between ciphertexts among SDN controllers, receiving user's search query, and returning satisfactory ciphertext to users during the search. The function of the SDN con-

troller in the control layer is equivalent to the cloud server.

4) *User:* Data providers and search users are called users. A data provider delivers personal information to the SDN controller and encrypts it through the SDN controller. A search user is a user who sends a search query and obtains ciphertext from a SDN controller. It owns an access structure and a secret key issued from the domain authority, with which one can calculate a trapdoor of the desired keyword.
5) *Access Equipment:* An access equipment is responsible for forwarding information between the SDN controller and the user.

All entities involved are deployed in the hierarchy structure shown in FIGURE 1. We assume that SDN controllers are not completely honest and some of them may maliciously steal user privacy. In addition, devices or access users will not be trusted with the possibility that they may want to bypass access control to illegally achieve honest user data or unauthorized use of additional network resources. In our paper, we focus on the access control and ciphertext searchable capabilities in SDN without detailed description of the interaction between the SDN controller and other components.

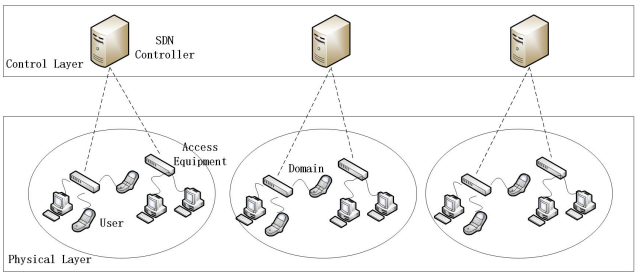


FIGURE 1. System architecture.

B. SYSTEM MODEL

For the sake of simplicity, we combine access control in SDN with attribute-based searchable encryption to form the system model shown in FIGURE 2. Firstly, the system authority initializes the system to obtain the public parameters and system master keys. The system authority here is not shown in FIGURE 2. Secondly, the SDN controller performs attribute-based encryption with a set of attributes of user information in the domain for all users in its domain. Ciphertext and index can be generated to make ciphertext searchable. In fact, it is necessary for the SDN controllers in the control layer to interact with each other in ciphertext so as to obtain the information of other controllers, and finally get the ciphertext of the controllers in the entire network. If a user wants to access certain information in other domains, he will first send the set of attributes owned by himself to the domain authority where the user is located. Once getting the set of attributes, the domain authority will check whether the attribute of the user is legitimate. If the user is authorized, the domain authority will assign a specific access structure to him, which specifies the type of information it can access and generates the user's private key. Next, by taking advantage of his own private key and keywords that he wants to search for information, the user computes a trapdoor and sends it to the SDN controller. Eventually, the SDN controller determines whether to return the corresponding ciphertext by computing the index and trapdoor.

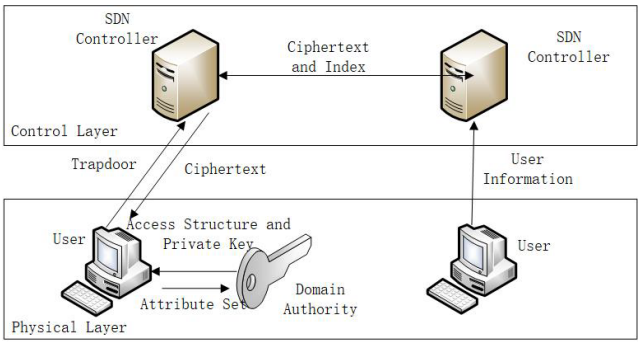


FIGURE 2. System model.

C. FORMULIZED DEFINITION OF ABSES

In this section, we will show the formulized definitions of the algorithms among the entities shown in Fig.2. The proposed scheme contains five algorithms, and they are described as follows:

1) *Setup:* This algorithm is run by the system authority. It takes a safe number as input and output public parameters and system master keys.
2) *KeyGen:* This algorithm is run by the domain authorities. With an access tree and system master keys as input, it outputs a private key for the user.
3) *IndexGen:* This algorithm is run by the SDN controllers. It takes public parameters, a ciphertext attribute set and keywords as input and outputs the index of ciphertext.
4) *Trapdoor:* This algorithm is run by the users. It takes user's private key as well as keywords as input, and output a trapdoor.
5) *Test:* This algorithm is run by the SDN controllers. It takes user's trapdoor as input and output the corresponding ciphertext.

D. SECURITY REQUIREMENT

The security goal of the proposed attribute-based searchable encryption scheme is to achieve keyword semantic security under chosen keywords attack. Our security model aims at the concept that adversaries cannot deduce ciphertext from their indexes without an effective trapdoor. In this paper, the requirement for keyword semantic security can be demonstrated by the game described as below:

1) *Setup:* The challenger runs the *Setup* algorithm, gets the relevant parameters, and sends the public parameters to the adversary.

2) *The First Phase:* The adversary selects some access structures A_j, in which for all j requires $\gamma \notin A_j$. An adversary can choose the keyword w to ask the challenger for trapdoor T_w.
3) *Challenge:* The adversary selects two keywords w_0 and w_1 that have not been previously queried. The challenger randomly chooses w_σ, where $\sigma \in \{0, 1\}$. It runs the *IndexGen* algorithm to calculate ciphertext index IN_{w_σ} and sends it to the adversary.
4) *The Second Phase:* The operation of this phase is the same as the first phase that the adversary asks for the trapdoor of more keywords $w(w \neq w_0, w_1)$.
5) *Guess:* The adversary outputs $\sigma' \in \{0, 1\}$ as the guess of σ. If $\sigma' = \sigma$, then the adversary wins the game. The advantage of the adversary is defined as $Adv = \left|Pr\left[\sigma' = \sigma\right] - \frac{1}{2}\right|$.

V. ATTRIBUTE-BASED SEARCHABLE ENCRYPTION SCHEME

A. CONCRETE CONSTRUCTIONS

In this section, we will provide the interaction among the entities shown in Fig.2 and concrete expansions of the formulized definitions of the proposed scheme. The attribute-based searchable encryption scheme used in this solution includes five polynomial time random algorithms: *Setup*, *KeyGen*, *IndexGen*, *Trapdoor*, and *Test*, which are shown as follows:

1) *Setup:* Define G_1, G_2, which are two multiplicative cyclic groups with prime order p and q respectively. g_1 is a generator of G_1, g_2 is a generator of G_2. Define hash functions: $H : \{0, 1\}^* \rightarrow Z_p$. The system authority randomly selects $y \in Z_p$ and computes $Y = e(g_1, g_2)^y$. It selects random different elements $t_1, t_2, \cdots, t_{|U|}$ from Z_p, where $|U|$ is the number of global attribute set U. Finally, the public parameters *Pub* and system master key *MSK* can be calculated as follows:

$$Pub = \left\{G_1, G_2, e, g_1, g_2, p, q, H, Y, g_1^{t_1}, g_1^{t_2} \cdots, g_1^{t_{|U|}}, g_2^{t_1}, g_2^{t_2} \cdots, g_2^{t_{|U|}}\right\} \quad (1)$$

$$MSK = \left\{t_1, t_2, \cdots, t_{|U|}, y\right\} \quad (2)$$

2) *KeyGen(MSK,*Γ*):* If a user wants to access information in other domains, then the user must send its attribute set to the domain authority. The domain authority first checks the legitimacy of the user's attributes, and, if valid, it will assign the user a specific access structure with which the user's private key can be computed. Consequently, the access policies vary from each other if each user has different attributes. Furthermore, the access policy is only known to the user who is associated with it. This algorithm takes the access tree Γ and the system master key *MSK* as input and outputs the private key D for the user.

The algorithm is executed as follows: It selects a polynomial q_x for each node in the access tree Γ in a top-down manner, that is, it starts from the root node r. For each node x, let the degree d_x of the polynomial q_x be: $d_x = k_x - 1$, where k_x is the threshold of the node x. We start from the root node r, set $q_r(0) = y$, and then select randomly d_r other points of the polynomial q_r to completely define it. For other nodes, let $q_x(0) = q_{parent(x)}(index(x))$ and choose random d_x other points to fully define the polynomial q_x. Among them, the function $parent(x)$ represents the parent node of node x in tree Γ.

After the above operations, all the polynomials have been determined. Then, a key D_x is generated for each node x in the access tree, such as formula (3):

$$D_x = g_1^{\frac{q_x(0)}{t_i}}, \quad i = att(x) \quad (3)$$

The user's private key D is a set of all node keys D_x, as shown below:

$$D = \left\{D_x = g_1^{\frac{q_x(0)}{t_i}}, \quad i \in \Gamma\right\} \quad (4)$$

3) *IndexGen(Pub,*γ*,w):* The SDN controller arbitrarily takes a number $s \in Z_q$, and then uses the public parameter *Pub*, the ciphertext attribute set γ and the keyword w to encrypt the information. Moreover, it generates the index IN_w of the ciphertext, and interacts with other SDN controllers for ciphertext information. Index IN_w consists of N_1, N_2, N_3. They are calculated as follows:

$$N_1 = g_1^{t_i s} \quad (5)$$

$$N_2 = Y^{sH(w)} \quad (6)$$

$$N_3 = g_2^{t_i sH(w)} \quad (7)$$

After obtaining N_1, N_2, N_3, IN_w will be computed as follows:

$$IN_w = (N_1, N_2, N_3) = \left(g_1^{t_i s}, Y^{sH(w)}, g_2^{t_i sH(w)}\right), \quad i \in \gamma \quad (8)$$

4) *Trapdoor(D, w):* The user calculates the trapdoor T_w based on his private key D and the keyword w of the desired information. After that, it sends it to the SDN controller of the control plane. T_1 and T_2 are involved in the trapdoor T_w. The user picks a random number $x \in Z_p$ and computes T_1 and T_2:

$$T_1 = \left(D_x \cdot g_1^{-x}\right)^{H(w)} = \left(g_1^{\frac{q_x(0)}{t_i}} \cdot g_1^{-x}\right)^{H(w)} = g_1^{\left(\frac{q_x(0)}{t_i} - x\right)H(w)} \quad (9)$$

$$T_2 = g_2^x \quad (10)$$

Then T_w will be calculated:

$$T_w = (T_1, T_2) = \left(g_1^{\left(\frac{q_x(0)}{t_i} - x\right)H(w)}, g_2^x\right), \quad i \in \Gamma \quad (11)$$

5) *Test(T_w):* The SDN controller knows the trapdoor T_w of the keyword w, as well as the ciphertext and index of the user information. If formula (12) is satisfied, the corresponding ciphertext is returned to the user. That is to say, it is judged whether the access tree Γ and ciphertext attribute set γ satisfy $\Gamma(\gamma) = 1$:

$$e(T_1, N_1) \cdot e(T_2, N_3) = N_2 \tag{12}$$

B. CONCRETE ANALYSIS

Define a recursive algorithm $DecryNode(T_w, N_1, N_3, x)$, which takes the trapdoor T_w the two components N_1 and N_3 of the ciphertext index IN_w, as well as a node x in the tree as input, and it outputs the remaining component of the index IN_w or "$\perp$." The specific operation is as follows:

Let $i = att(x)$, if x is a leaf node, then:

$$DecryNode(T_w, N_1, N_3, x) = \begin{cases} e(T_1, N_1) \cdot e(T_2, N_3), & i \in \gamma \\ \perp, & others \end{cases}$$

Among them,

$$\begin{aligned} & e(T_1, N_1) \cdot e(T_2, N_3) \\ &= e\left(g_1^{\left(\frac{q_x(0)}{t_i} - x\right)H(w)}, g_2^{t_i s}\right) \cdot e\left(g_2^x, g_1^{t_i s H(w)}\right) \\ &= e(g_1, g_2)^{sq_x(0)H(w) - xt_i sH(w)} \cdot e(g_1, g_2)^{xt_i sH(w)} \\ &= e(g_1, g_2)^{sq_x(0)H(w)} \end{aligned}$$

If x is a non-leaf node, we will consider the following recursion. The specific operation of the algorithm $DecryNode(T_w, N_1, N_3, x)$ is as follows: For all child nodes z of the node x, set $F_z = DecryNode(T_w, N_1, N_3, z)$. We assume that there is a random set of nodes S_x with the size of k_x, and the nodes in the set are children of x. otherwise $F_z = \perp$.

After that, set $i = index(x)$, $S'_x = \{index(z) : z \in S_x\}$.

Then compute as follows:

$$\begin{aligned} F_x &= DecryNode(T_w, N_1, N_3, x) \\ &= \prod_{z \in S_x} F_z^{\Delta_{i,S'_x}(0)} \\ &= \prod_{z \in S_x} \left(e(g_1, g_2)^{s \cdot q_z(0) \cdot H(w)}\right)^{\Delta_{i,S'_x}(0)} \\ &= \prod_{z \in S_x} \left(e(g_1, g_2)^{s \cdot q_{parent(z)}(index(z)) \cdot H(w)}\right)^{\Delta_{i,S'_x}(0)} \\ &= \prod_{z \in S_x} \left(e(g_1, g_2)^{s \cdot q_x(i) \cdot H(w)}\right)^{\Delta_{i,S'_x}(0)} \\ &= e(g_1, g_2)^{sq_x(0)H(w)} \end{aligned}$$

In summary, now we have a complete definition of the function *DecryNode*, thus the *Test* algorithm can simply call the value of it at the root node. As a result, $\Gamma_r(\gamma) = 1$ if and only if ciphertext attributes satisfy the access tree. Then we can compute:

$$\begin{aligned} DecryNode(T_w, N_1, N_3, r) &= e(g_1, g_2)^{ysH(w)} \\ &= e(g_1, g_2)^{sYH(w)} = N_2, \end{aligned}$$

that is $e(T_1, N_1) \cdot e(T_2, N_3) = N_2$, and the proof completed.

VI. DISCUSSION

A. KEYWORD SEMANTIC SECURITY

Theorem: Our scheme is keyword semantic security if the DBDH hardness assumption holds.

Proof: Assuming that there is an adversary A is capable of breaking this scheme with the advantage ε, we can construct a simulator B to break the DBDH hardness with an advantage of at least $\frac{\varepsilon}{2}$.

1) *Setup:* Let G_1 and G_2 be cyclic groups with prime of p and q respectively. Denote g_1 and g_2 as the generators of G_1 and G_2 respectively, and set $g_2 = g_1^l$. Let $e : G_1 \times G_2 \to G_2$ be a bilinear pairing. Define hash function $H : \{0,1\}^* \to Z_p$. The challenger randomly picks $\sigma \in \{0,1\}$, $a, b, c \in Z_p$ and outputs $\left(g_1^a, g_1^b, g_1^c, Z\right)$. If $\sigma = 0$, $Z = e(g_1, g_2)^{abc} = e(g_1, g_1)^{abcl}$. If $\sigma = 1$, $z \in Z_p$ is selected randomly such that $Z = e(g_1, g_2)^z = e(g_1, g_1)^{zl}$.
2) *The First Phase:* The adversary A inquires about some access structures with which the attribute set γ is not satisfied. We assume that A will ask for a key whose access structure satisfies $\Gamma(\gamma) = 0$. In order to generate the key, B must determine a polynomial of d_x degree for each non-leaf node in the access tree.

PolyUnsat $\left(\Gamma, \gamma, g_1^{\lambda_x}\right)$ is a process of determining the polynomial of the node x in the access tree (The attribute set γ does not satisfy the access tree with the root node x, that is, $\Gamma_x(\gamma) = 0$). The access tree Γ_x with root node x, attribute γ and $g_1^{\lambda_x} \in G_1$ where $\lambda_x \in Z_p$ are viewed as input. Define the d_x polynomial $q_x(0) = \lambda_x$ of the root node with x. There are less than d_x child nodes of x satisfying the attribute set because of $\Gamma_x(\gamma) = 0$, so we suppose that $h_x < d_x$ is the number of sub nodes of x that meets the attribute set. For each child node x' of x meeting the attribute set γ, we pick $\lambda_{x'} \in Z_p$ at random, and set $q_x\left(index\left(x'\right)\right) = \lambda_{x'}$. Then the remaining $d_x - h_x$ points of q_x will be randomly selected to determine q_x completely. After that, we define the points in the access tree that do not satisfy the attribute set γ in such a recursive way. It is worth noting that if $g_1^{q_x(0)}$ is given, $g_1^{q_x(index(x'))}$ will be known by interpolation and that $q_x(0) = q_x\left(index\left(x'\right)\right)$.

The simulator B invokes *PolyUnsat* (Γ, γ, A) to define the polynomial q_x of each node x in the access tree, which meets $q_r(0) = ab$. For each leaf node accessing tree Γ, q_x is fully known if x satisfies the attribute set γ, and we at least know $g_1^{q_x(0)}$ if x does not satisfy the attribute set γ. Therefore, no matter whether i is in the attribute set γ, the corresponding key of each node x

can be calculated by the following formula:

$$D_x = g_1^{\frac{q_x(0)}{t_i}}, \quad i = att(x)$$

The user's private key D is the set of all nodes' key D_x, as shown below:

$$D = \left\{ D_x = g_1^{\frac{q_x(0)}{t_i}}, i \in \Gamma \right\}$$

Next, the adversary A asks the random oracle H for the trapdoor of the keyword w:

$$T_w = (T_1, T_2) = \left(g_1^{\left(\frac{q_x(0)}{t_i} - x\right)H(w)}, g_2^x \right), \quad i \in \Gamma$$

3) *Challenge:* The adversary A selects two keywords w_0 and w_1, which have not been queried before. The simulator B randomly picks w_σ where $\sigma \in \{0, 1\}$, and chooses $s \in Z_p$, then it computes the following contents:
If $\sigma = 0$, set $N_1 = g_1^{t_i s}$, $N_{2,\sigma} = e(g_1, g_2)^{absH(w_\sigma)} = e(g_1, g_1)^{abslH(w_\sigma)}$, $N_3 = g_2^{t_i sH(w)} = g_1^{t_i slH(w)}$.
If $\sigma = 1$, set $N_1 = g_1^{t_i s}$, $N_{2,\sigma} = e(g_1, g_2)^{zH(w_\sigma)} = e(g_1, g_1)^{zlH(w_\sigma)}$, $N_3 = g_2^{t_i sH(w)} = g_1^{t_i slH(w)}$.
The simulator B sends the index $N_{2,\sigma}$ to the adversary A, and sets $g_1^s = g_1^c$, then:
If $\sigma = 0$, we will have $N_{2,\sigma} = e(g_1, g_1)^{abclH(w_\sigma)}$.
If $\sigma = 1$, we will have $N_{2,\sigma} = e(g_1, g_1)^{zlH(w_\sigma)}$.
4) *The Second Phase:* The operation of this phase is the same as the first phase that the adversary asks for the trapdoor of more keywords $w (w \neq w_0, w_1)$.
5) *Guess:* The adversary outputs $\sigma' \in \{0, 1\}$ as the guess of σ. If $\sigma' = \sigma$, then the adversary wins the game.

If $\sigma = 1$, $N_{2,\sigma}$ is an invalid search index and the adversary A can only guess the value of σ randomly after receiving it. Accordingly, the adversary A has an advantage $\Pr(\sigma' = \sigma \,|\sigma = 1) = \frac{1}{2}$. If $\sigma = 0$, $N_{2,\sigma}$ is a valid index, and the advantage of the adversary A is $\Pr(\sigma' = \sigma \,|\sigma = 0) = \frac{1}{2} + \varepsilon$.

Consequently, the simulator B's advantage can be denoted by:

$$\begin{aligned} &\frac{1}{2}\Pr(\sigma' = \sigma \,|\sigma = 0) + \frac{1}{2}\Pr(\sigma' = \sigma \,|\sigma = 1) - \frac{1}{2} \\ &= \frac{1}{2}\left(\frac{1}{2} + \varepsilon\right) + \frac{1}{2} \times \frac{1}{2} - \frac{1}{2} \\ &= \frac{\varepsilon}{2} \end{aligned}$$

However, from the third definition demonstrated in Section III, it is hard to distinguish the tuple in DBDH assumption. According to the reduction to absurdity, the assumption that an adversary can launch an effective attack against the scheme is not valid, which proves that the proposed scheme is secure.

B. PERFORMANCE ANALYSIS

1) CIPHERTEXT CONFIDENTIALITY

In our scheme, the SDN controller in the system performs attribute-based encryption on the users' personal information in the domain and generates ciphertext index. The interaction between controllers is essentially the interaction of ciphertext, which ensures the security of user information during transmission.

2) USERS' PRIVACY AND SECURITY

The SDN controller uses the given threshold T_w as the input of the *Test* algorithm to decide whether to return the corresponding ciphertext. In this course, the sensitivity of the query information can be hidden without knowing the plaintext of the query keyword w.

3) TRAPDOOR UNLINKABILITY

Due to the reason that x is randomly chosen by different users, it is impossible for SDN controllers to distinguish between different trapdoors that contain the same keywords, which satisfies the security requirements of trapdoor unlinkability.

C. PERFORMANCE EVALUATION

We compare our scheme with the schemes in [18], [19], and [23], which are all attribute-based searchable encryption schemes. We calculate the computational costs for each phase of each algorithm separately. "Pair" means bilinear pairings, "Exp" means exponentiation, and "n" means the number of attributes involved. The comparison results are illustrated in TABLE 1.

TABLE 1. Comparison results with other schemes.

Scheme	Setup	KeyGen	IndexGen	Trapdoor	Test
[19]	(3n+1)Exp+1Pair	(2n+3)Exp	(n+2)Exp	(2n+1)Exp	(n+1)Pair+1Exp
[20]	3Exp	(2n+2)Exp	(2n+3)Exp	(2n+2)Exp	(2n+2)Pair
[24]	3Exp	(2n+1)Exp	(2n+6)Exp	(2n+6)Exp	(2n+2)Pair
Our ABSES	(2n+1)Exp+1Pair	nExp	(2n+1)Exp	(n+1)Exp+1Pair	2nPair

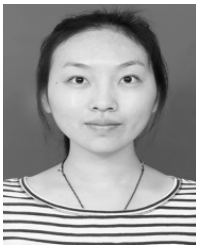
It can be seen from the comparison results that this scheme has obvious advantages over other schemes in the *KeyGen* and *Trapdoor* algorithm. In the *Setup* algorithm, our scheme requires more exponentiation, which is carried out by the system authority. Therefore, it does not increase the computational burden of the user. Similarly, the *Test* algorithm requires more bilinear pairing operations, which are run by the SDN controller. In the scheme of the reference [19], the access structure only supports "AND" gates, while our scheme provides a more flexible access structure that supports both "OR" and "AND" gates, so the *IndexGen* algorithm in our scheme needs more exponentiation. To sum up, from the overall point of view, we propose an efficient scheme with a better performance.

VII. CONCLUSION

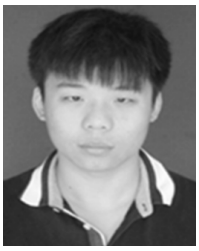
Human-centered smart system has become one of the most potential technologies. People's requirements and feelings are the problems that need to be concerned during the development of intelligent technique. In this paper, we apply smart technology to SDN and propose a new inter-domain information sharing smart system based on the novel attribute-based searchable encryption scheme. For users, the proposed scheme ensures that they cannot achieve unauthorized information or illegal network resources, which saves the system's computing time, communication overhead and local resources. More importantly, it not only meets the needs of users, but also provides a very good user experience. The proof of the scheme's security was given by the assumption that if the DBDH is indecipherable, then our scheme meets the demand of keyword semantic security. In this scheme, two functions of access control and ciphertext search are realized, so as to ensure the secure transmission of information and the sharing of inter-domain information. By analysis and calculation, our scheme is of high performance and is superior to other attribute-based searchable encryption algorithms. Our new smart system can be used in human life so that our life will become much more convenient. In future work, we will focus our work on a smarter system that is more flexible for users in SDN.

REFERENCES

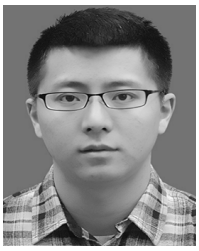
[1] Stanford University. (2006). *Clean Slate Program*. [Online]. Available: http://cleanslate.stanford.edu/
[2] N. McKeown, "Software-defined networking," *INFOCOM Key Note*, vol. 17, no. 2, pp. 30–32, 2009. [Online]. Available: http://infocom2009.ieee-infocom.org/technicalProgram.htm
[3] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM CCR*, vol. 38, no. 2, pp. 69–74, 2008, doi: 10.1145/1355734.1355746.
[4] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. Mckeown, and S. Shenker, "Ethane: Taking control of the enterprise," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*, 2007, pp. 1–12.
[5] S. Hartman and M. Wasserman. (2012). *Security Requirements in the Software Defined Networking Model*. [Online]. Available: http://tools.ietf.org/html/draft-hartmansdnsec-requirements-00
[6] F. Klaedtke, G. O. Karame, R. Bifulco, and H. Cui, "Access control for SDN controllers," in *Proc. ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, 2014, pp. 1325–1335.
[7] B. Jäger, C. Röpke, I. Adam, and T. Holz, "Multi-layer access control for SDN-based telco clouds," in *Secure IT Systems*. Cham, Switzerland: Springer, 2015, pp. 197–204.
[8] A. V. Kamath, S. Sudarshan, K. Kataoka, N. Vijayvergiya, G. B. Reddy, and S. Phatale, "SAFE: Software-defined authentication framework," in *Proc. Asian Internet Eng. Conf. ACM*, 2016, pp. 57–63.
[9] OpenFlow switch Consortium. (2013). *OpenFlow Specification V1.0*. [Online]. Available: http://www.openflow.org/
[10] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, G. R. Blakley and D. Chaum, Eds. Berlin, Germany: Springer-Verlag, 1984, pp. 47–53.
[11] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 213–229, 2001.
[12] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology*, R. Cramer, Ed. Berlin, Germany: Springer-Verlag, 2005, pp. 457–473.
[13] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, 2006, pp. 89–98, doi: 10.1145/1180405.1180418.
[14] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, Washington, DC, USA, May 2007, pp. 321–334, doi: 10.1109/SP.2007.11.
[15] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology—EUROCRYPT*. Berlin, Germany: Springer, 2004, pp. 506–522.
[16] Q. Wang, Y. Zhu, and X. Luo, "Multi-user searchable encryption with fine-grained access control without key sharing," in *Proc. 3rd Int. Conf. IEEE Adv. Comput. Sci. Appl. Technol. (ACSAT)*, Dec. 2014, pp. 145–150.
[17] Y. Yang, "Attribute-based data retrieval with semantic keyword search for e-health cloud," *J. Cloud Comput.*, vol. 4, p. 10, Dec. 201.
[18] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: Verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 4, pp. 1187–1198, Apr. 2016.
[19] Y. Miao, J. Ma, X. Liu, F. Wei, Z. Liu, and X. A. Wang, "m^2-ABKS: Attribute-based multi-keyword search over encrypted personal health records in multi-owner setting," *J. Med. Syst.*, vol. 40, p. 246, Nov. 2016.
[20] H. Wang, J. Li, Y. Yang, and Z. Ming, "Attribute-based and keywords vector searchable public key encryption," in *Proc. Smart Comput. Commun., SmartCom*, Shenzhen, China, Dec. 2016, pp. 317–326.
[21] Q. Dong, Z. Guan, and Z. Chen, "Attribute-based keyword search efficiency enhancement via an online/offline approach," in *Proc. IEEE 21st Int. Conf. Parallel Distrib. Syst.*, Melbourne, VIC, Australia, Dec. 2015, pp. 298–305.
[22] H. Li, D. Liu, K. Jia, and X. Lin, "Achieving authorized and ranked multi-keyword search over encrypted cloud data," in *Proc. IEEE Int. Conf. Commun.*, London, U.K., Jun. 2015, pp. 7450–7455.
[23] V. Yousefipoor, M. H. Ameri, J. Mohajeri, and T. Eghlidos, "A secure attribute based keyword search scheme against keyword guessing attack," in *Proc. IEEE Commun. Inf. Syst. Secur. Symp.*, Tehran, Iran, Sep. 2016, pp. 124–128.
[24] P. Chaudhari and M. L. Das, "A^2BSE: Anonymous attribute based searchable encryption," in *Proc. Asia Secur. Privacy*, Jan./Feb. 2017, pp. 1–10.
[25] Y. Fan and Z. Liu, "Verifiable attribute-based multi-keyword search over encrypted cloud data in multi-owner setting," in *Proc. IEEE 2nd Int. Conf. Data Sci. Cyberspace*, Jun. 2017, pp. 441–449.
[26] H. Wang, X. Dong, and Z. Cao, "Multi-value-independent ciphertext-policy attribute based encryption with fast keyword search," *IEEE Trans. Serv. Comput.*, to be published.
[27] H. Shuangyu, L. Jianwei, M. Jian, and C. Jie, "Hierarchical solution for access control and authentication in software defined networks," in *Proc. Int. Conf. Netw. Syst. Secur.*, 2014, pp. 70–81.



JINGJING XU received the B.Eng. degree in network engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2016, where she is currently pursuing the M.D. degree in information network. Her research interests include smart technology, software defined network, and cryptography.



HANSHU HONG received the B.Eng. degree in network engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2013, where he is currently pursuing the M.D.–Ph.D. degree in information network. His research interests include information security and cryptography.



GUOFENG LIN received the B.Eng. degree in electronic information engineering from Nantong University, Nantong, China, in 2013. He is currently pursuing the M.D.-Ph.D. degree in information network with the Nanjing University of Posts and Telecommunications. His research interests include cloud computing, network security, and cryptography.

ZHIXIN SUN was born in Xuancheng, China, in 1964. He received the Ph.D. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1998. From 2001 to 2002, he held a post-doctoral position with the School of Engineering, Seoul National University, South Korea. He is currently a Professor and the Dean of the School of Modern Posts, Nanjing University of Posts and Telecommunications. His research interests include cloud computing, cryptography, and traffic identification.