# AUTHENTICATED QUALITY OF SERVICE AWARE ROUTING IN SOFTWARE DEFINED NETWORKS

by

Samet Aytaç

B.S., Computer Engineering, Middle East Technical University, 2016

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2019

AUTHENTICATED QUALITY OF SERVICE AWARE ROUTING IN SOFTWARE DEFINED NETWORKS

APPROVED BY:

| | |
|---|---|
| Prof. Fatih Alagöz<br>(Thesis Supervisor) | . . . . . |
| Prof. Tuna Tuğcu | . . . . . |
| Assist. Prof. Şerif Bahtiyar | . . . . . |

DATE OF APPROVAL: 09.05.2019

# ACKNOWLEDGEMENTS

I would like express special appreciation and thanks to my supervisor Prof. Fatih Alagöz for all his support, kindness and patience. With his guidance, thesis process became extremely productive. I also would like to express my deepest gratitude to Prof. Ufuk Çağlayan. I have always felt his support in my academic life and my personal life.

I would like to express my sincere gratitude to members of my thesis jury, Prof. Tuna Tuğcu and Assist. Prof. Şerif Bahtiyar for their feedbacks and constructive comments.

I would like to thank Dr. Orhan Ermiş for his guidance during my masters studies. This thesis would not be possible without him. I also would like to thank Prof. Cem Ersoy for all his support. In addition, I would like to thank all members of SATLAB, NETLAB and TETAM for peaceful and friendly environment.

Last, but not the least, I would like to express my gratitude to my family for their support and love.

# ABSTRACT

## AUTHENTICATED QUALITY OF SERVICE AWARE ROUTING IN SOFTWARE DEFINED NETWORKS

Quality of Service (QoS) aware routing is an ongoing and major problem for traditional networks since they are not able to manage network traffic for an immense variety of users due to their inflexible and static architectures. Software Defined Networking (SDN) has emerged to remove these limitations by separating the control plane and the data plane to provide centralized control with the help of programmable controllers. Such improvements also make SDN more flexible than traditional networks in terms of achieving QoS-aware routing. However, providing QoS-aware routing in SDN without using any security mechanism may become a challenging issue. For instance, malicious users in the network may escalate their privileges to monopolize resource utilization. The provision of an authentication mechanism that jointly works with QoS-aware routing is expected to solve the issue. In this thesis, we propose an Authenticated QoS-Aware Routing (AQoSAR) for Software Defined Networks to determine routing paths of a single user and a group of users in an authenticated manner. AQoSAR consists of the authentication application and the routing application. In the authentication application, we employ Ciphertext Policy Attribute Based Encryption since it easily operates with a huge variety of users by defining attributes such as QoS-aware routing metrics. In the routing application, we propose a routing approach based on a metric list rather than a single metric for determining the QoS level of users. To show the applicability of AQoSAR, the security analysis and the performance analysis are presented.

# ÖZET

## KİMLİK DOĞRULAMA TABANLI HİZMET KALİTESİ FARKINDA YÖNLENDİRME

Geleneksel ağlar için hizmet kalitesi farkında yönlendirme günümüzde dahi çözülemeyen bir sorun olarak karşımıza çıkmaktadır. Bunun başlıca sebebi geleneksel ağ mimarilerinin yeterince esnek olmayışıdır. Yazılım tabanlı ağlar programlanabilir merkezi kontrol birimi kullanarak esnek bir mimari sağlamaktadırlar. Bu nedenle, yazılım tabanlı ağlar hizmet kalitesi farkında yönlendirme uygulamaları için en uygun ağ mimarisidir. Fakat ağda oluşabilecek herhangi bir güvenlik açığı kullanıcıların hizmet kalitesini olumsuz yönde etkileyebilir. Bu nedenle, hizmet kalitesi farkında yönlendirme uygulamaları kimlik doğrulama mekanizmasına ihtiyaç duyar. Bu tezde, Kimlik Doğrulama Tabanlı Hizmet Kalitesi Farkında Yönlendirme sistemini öneriyoruz. Önerilen sistem, kimlik doğrulama uygulaması ve yönlendirme uygulaması adında iki uygulamadan oluşmaktadır. Kimlik doğrulama uygulaması kullanıcıların özelliklerine bağlı olarak kullanıcıların kimliklerini doğrular. Yönlendirme uygulaması ise kimliği doğrulanmış kullanıcıların yüksek kalite yönlendirme hizmeti almasını sağlar. Bu tezde, ayrıca, önerilen sistemin uygulanabilirliğini göstermek için güvenlik ve performans analizleri de sunulmuştur.

# TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . x
LIST OF SYMBOLS . . . xi
LIST OF ACRONYMS/ABBREVIATIONS . . . xiii
1. INTRODUCTION . . . 1
2. LITERATURE OVERVIEW . . . 3
2.1. Overview of Software Defined Networks . . . 3
2.1.1. SDN Architecture . . . 3
2.1.2. QoS-Aware Routing in SDN . . . 5
2.1.3. Authentication Mechanisms for SDN . . . 7
2.2. Overview of Attribute Based Encryption . . . 8
2.2.1. Definition of Attribute Based Encryption . . . 8
2.2.2. Applications of Attribute Based Encryption . . . 10
2.2.3. Attribute Based Authentication . . . 11
3. AQoSAR: AUTHENTICATED QOS AWARE ROUTING . . . 13
3.1. Authentication Application . . . 14
3.1.1. Single User Authentication . . . 16
3.1.2. Group Authentication . . . 18
3.1.2.1. Create Operation . . . 19
3.1.2.2. Join Operation . . . 21
3.1.2.3. Leave Operation . . . 22
3.2. Routing Application . . . 23
3.3. Use Case Scenario For AQoSAR: A University Department . . . 27
4. SECURITY OF AQoSAR . . . 29
5. PERFORMANCE OF AQoSAR . . . 34
6. CONCLUSION . . . 39

REFERENCES . . . . . 40

# LIST OF FIGURES

Figure 2.1. Traditional Network Infrastructure vs SDN Infrastructure [1] . . . . . 4

Figure 2.2. Basics of KP-ABE . . . . . 9

Figure 2.3. Basics of CP-ABE . . . . . 9

Figure 3.1. Overview of AQoSAR . . . . . 14

Figure 3.2. El-Gamal Encryption Scheme . . . . . 16

Figure 3.3. Schnorr Signature Scheme . . . . . 16

Figure 3.4. CP-ABE Scheme . . . . . 17

Figure 3.5. Sequence Diagram of Single User Authentication . . . . . 17

Figure 3.6. Details of Single User Authentication . . . . . 18

Figure 3.7. Sequence Diagram of Create Operation . . . . . 19

Figure 3.8. Details of Create Operation . . . . . 20

Figure 3.9. Sequence Diagram of Join Operation . . . . . 21

Figure 3.10. Details of Join Operation . . . . . 22

Figure 3.11. Sequence Diagram of Leave Operation . . . . . 22

Figure 3.12. Details of Leave Operation . . . . . 23

Figure 3.13. Path Construction Procedure . . . . . 26

Figure 3.14. Authentication Levels of the University Department . . . . . 27

Figure 5.1. Time Required to Execute Authentication Application with Respect to Change in the Number of Attributes . . . . . 34

Figure 5.2. Time Required to Execute Authentication Application with Respect to Change in the Number of Users . . . . . 35

Figure 5.3. Average Time a User Spends in the System with respect to Arrival Rate of Users . . . . . 37

# LIST OF TABLES

Table 2.1. SDN vs Traditional Networks . . . . . 5

Table 3.1. Notations for the Authentication Application . . . . . 15

Table 5.1. Computation and Performance Costs of Authentication Protocols . . . 36

# LIST OF SYMBOLS

| | |
|---|---|
| $a$ | Availability Metric |
| $a_{max}$ | Maximum Availability Metric in the Network |
| $\overline{a}$ | Average Availability Metric in the Network |
| $b$ | Bandwidth Capacity |
| $b_{max}$ | Maximum Bandwidth Capacity in the Network |
| $\overline{b}$ | Average Bandwidth Capacity in the Network |
| $c_{ij}$ | Cost of $(i, j)$ link |
| $C$ | Controller |
| $CA$ | Certificate Authority |
| $CP$ | Ciphertext |
| $e$ | Entity |
| $E_{ABE,\gamma}$ | Attribute Based Encryption for metric list $\gamma$ |
| $E_k$ | Encryption with Key k |
| $E[S]$ | Expected Service Time |
| $f_c(p)$ | Cost Function of Path p |
| $G$ | Group |
| $h()$ | Hash Function |
| $ID_e$ | Identity of entity e |
| $k$ | Number of New Participants for Join Operation |
| $l$ | Number of Leaving Participants for Leave Operation |
| $L_s$ | Average Number of Users in System |
| $m$ | Number of Participants for Create Operation |
| $MA$ | Group Manager |
| $ME_n$ | nth Member of a Group |
| $ME_{new}$ | Joining Participant to the Group |
| $ME_{old}$ | Leaving Participant from the Group |
| $MK$ | Master Key of Ciphertext Policy Attribute Based Encryption |
| $MS_n$ | $n^{th}$ Message |

| | |
|---|---|
| $n$ | Randomly Generated Nonce |
| $p$ | Path |
| $P(s,d)$ | Set of All Paths Between Node s and Node d |
| $PK_e$ | Public Key of Entity e |
| $PP$ | Public Parameters of Ciphertext Policy Attribute Based Encryption |
| $SK_e$ | Secret Key of Entity e |
| $SK_{ABE,e}$ | Attribute Based Encryption Key of Entity e |
| $Q$ | Unvisited Neighbour Node List |
| $r$ | Reliability Metric |
| $r_{max}$ | Maximum Reliability Metric in the Network |
| $\overline{r}$ | Average Reliability Metric in the Network |
| $s$ | Switch |
| $t$ | Timestamp |
| $T_{exp}$ | Time Required to Compute Modular Exponentiation Operations |
| $U$ | User |
| $W_s$ | Average Time that Each User Spends in the System |
| $z$ | Number of Switch |
| $(i,j)$ | Link Between Node i and Node j |
| $\beta_t(i)$ | Backward variable |
| $\gamma_e$ | Metric List of entity e |
| $\Theta$ | Parameter set |
| $\lambda$ | Arrival Rate of Users |
| $\mu$ | Service Rate |
| $\rho$ | Utilization Ratio |

# LIST OF ACRONYMS/ABBREVIATIONS

| | |
|---|---|
| ABA | Attribute Based Authentication |
| ABE | Attribute Based Encryption |
| API | Application Programming Interface |
| AQoSAR | Authenticated Quality of Service Aware Routing |
| CP-ABE | Ciphertext Policy Attribute Based Encryption |
| DDos | Distributed Denial of Service |
| DNS | Domain Name System |
| ICN | Information Centric Networks |
| IND-CPA | Indistinguishable Under the Chosen-Plaintext Attack |
| IP | Internet Protocol |
| KP-ABE | Key Policy Attribute Based Encryption |
| MCSP | Multiple Constraint Shortest Path |
| NFV | Network Functions Virtualization |
| ONOS | Open Network Operating System |
| QoS | Quality of Service |
| SDN | Software Defined Networks |

# 1. INTRODUCTION

Software Defined Networks (SDN) has emerged to remove limitations of traditional networks such as inflexible and static architectures by separating the control plane and the data plane with the help of programmable devices, namely controllers and switches. Since SDN is able to serve different variety of users with different expectations, it provides better performance in terms of achieving Quality of Service (QoS) aware routing rather than traditional networks. However, providing QoS-aware routing is still an open issue for SDN. There have been many studies to solve QoS-aware routing issues in literature such as [2–5]. OpenQoS in [2] provides QoS-aware routing for multimedia streaming. In [3], OpenQoS is extended for the distributed SDN environment. Dijkstra's shortest path algorithm with switch utilization was proposed [4]. Another study on QoS-aware routing is [5], in which users are isolated by network virtualization mechanism and routing is performed by considering such users. In this thesis, we propose QoS-aware routing for SDN enhanced by enabling extended authentication mechanism for groups of users and individual users.

One of the important characteristics of QoS-aware routing systems is the ability to classify users by considering their privileges. These classifications are necessary to share resources such as bandwidth capacity, latency in the communication, reliability of communications, etc. For instance, high privileged users may request high bandwidth capacity and low latency in communications if they transfer mission critical information for an organization. On the other hand, low or medium privileged users may request high reliability for communications. However, the existence of malicious users may degrade the QoS level by using security vulnerabilities of controllers. Such users may impersonate themselves as high privileged users to monopolize resources of network even if they are low privileged users. Then, controllers become impotent of determining the correct QoS level for users. Thus, an extra layer should be added for the routing application of controllers to provide a verification mechanism for identities and corresponding privileges of users and groups. This verification mechanism can be accomplished by using an authenticated QoS-aware routing, which is our main motivation for the thesis. In order to address authenticated QoS-aware routing problem in SDN, variants of Attribute Based Encryption (ABE) schemes [6] can be used since

they easily operate with huge variety of users by defining different attributes for QoS-aware routing metrics.

ABE is a variant of public-key encryption that employs public-private key pairs together with user attributes in a form of well-defined access policy to encrypt/decrypt plaintexts/ciphertexts. Since user attributes are used rather than user identities in ABE schemes, it can provide fine-grained access control on the encrypted data. ABE schemes are classified as Key Policy Attribute Based Encryption (KP-ABE) [7] and Ciphertext Policy Attribute Based Encryption (CP-ABE) [8]. In KP-ABE, the ciphertext is labeled with a set of attributes and secret keys of entities are associated with access structures. In CP-ABE, the ciphertext is associated with an access structure and secret keys of entities. In this thesis, we prefer to use CP-ABE rather than KP-ABE to avoid the key distribution problem among entities in the network.

Our contributions for the thesis are as follows:

- We propose an Authenticated Quality of Service Aware Routing (AQoSAR) to securely determine routing paths of a single user and a group of users in the network.
- To provide authentication for huge variety of users with different expectations, we employ CP-ABE as an authentication mechanism.
- We propose to use metric list rather than using a single metric for the QoS-aware routing to meet different expectations of users.
- Moreover, we provide detailed analysis for the security of the proposed approach against impersonation, collision, eavesdropping and replay attacks.
- Furthermore, we have presented numerical evaluations by using simulations to show the applicability of AQoSAR.

The rest of the thesis is organized as follows. We present related works in Chapter 2. Then, in Chapter 3, we propose AQoSAR. In Chapter 4 and 5, we present security analysis and performance analysis for the proposed approach. Finally, the thesis is concluded in Chapter 6.

# 2. LITERATURE OVERVIEW

To provide further understanding about the thesis, we present a detailed literature overview in this chapter. First, we give basics of Software Defined Networking (SDN), Quality of Service (QoS) aware routing issues in SDN and authentication mechanisms in SDN. Then, we overview Attribute Based Encryption (ABE) and its use as an authentication mechanism.

## 2.1. Overview of Software Defined Networks

In this section, we start with an overview of SDN architecture. Then, we present QoS-aware routing methods in SDN. Finally, we examine authentication mechanisms in SDN.

### 2.1.1. SDN Architecture

SDN architecture provides decoupled data plane and control plane. This architecture provides more information about the state of the entire network from controller to applications. In addition, having a centralized control plane simplifies the configuration and the management of the network. SDN architecture has three main components [9]:

- Data Plane: This plane consists of simple switches which are only capable of forwarding. Data plane elements are not involved in any decisive action.
- Controller Plane: This plane consists of an entity named controller, which is a logical entity that receives instructions or requirements from the SDN application plane and sends them to the networking components. The controller also collects network information from data plane and sends them to application plane. In this way, application plane can perform logical operations with the network information.
- Application Plane: This plane consists of applications for the basic functions in the network such as load balancing, intrusion detection, etc. SDN applications are programs that control network through the controller. Applications communicate with SDN controller by an Application Programming Interface (API). In addition, to real-

ize networking functions, applications can build an abstracted view of the network by collecting information from the controller for routing decisions.

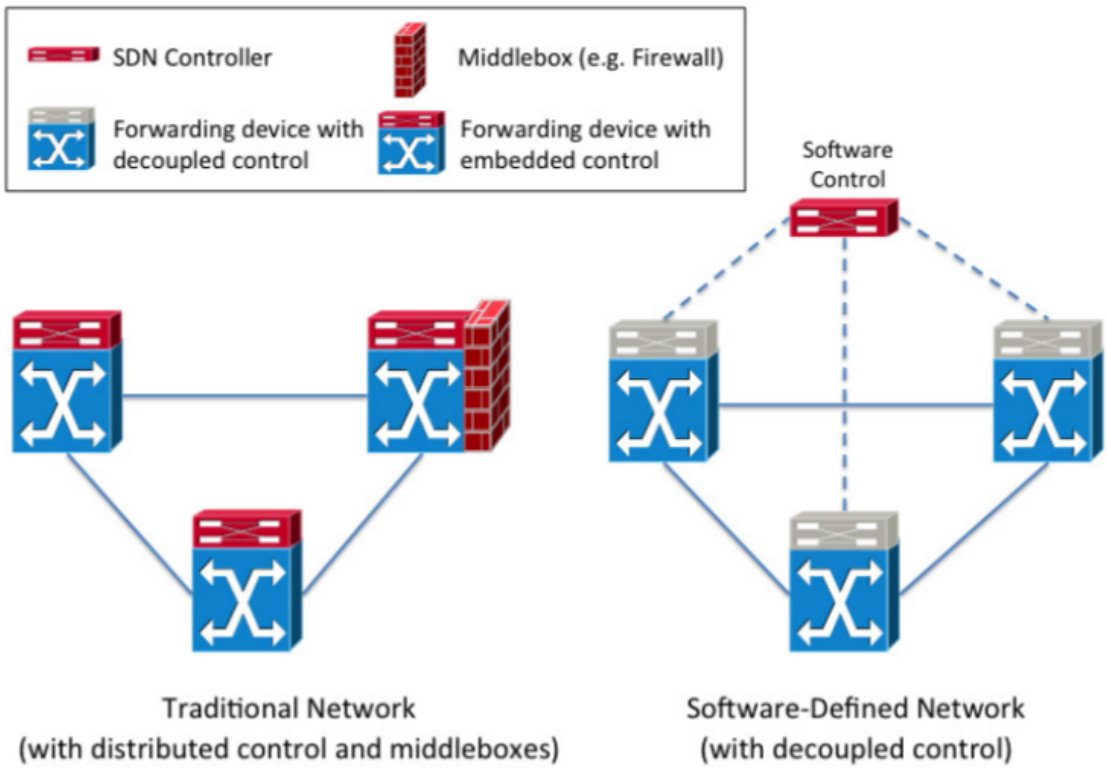



Figure 2.1. Traditional Network Infrastructure vs SDN Infrastructure [1]

In SDN architecture, communication between planes is realized by northbound and southbound interfaces. The northbound interface is defined as a connection between the controller and applications, whereas the southbound interface is the connection between the controller and switches. Because SDN is a virtualized architecture, controller and switches do not have to be physically located in the same place.

Infrastructural differences between SDN and traditional networks are as shown in Fig. 2.1. As shown in the figure, traditional networks need middleboxes to realize networking functions. Moreover, control logic is distributed to routers in traditional network architecture. Despite traditional networks, SDN architecture collects all network functionalities and control logic into programmable controller. More abstract differences between two architectures [10] are examined in Table 2.1.

Table 2.1. SDN vs Traditional Networks

| | **Software Defined Networks** | **Traditional Networks** |
|---|---|---|
| **Complexity** | Less complex architecture. | Complex architecture due to vertical integration. |
| **Configuration** | Automated configuration with centralized controller. | Manual configuration per device. |
| **Manageability** | Dynamic centralized control with abstract view of the network. | Limited network information and distributed control logic. |
| **Network Functions** | Controller applications. | Middleboxes. |
| **Scalability** | High scalability due to agile controller. | Low scalability due to static and complex architecture. |
| **Policy Enforcement** | Performed only on controller. | Needs to be performed on all devices separately. |

Routing mechanism of SDN is also different from traditional networks. SDN separates the control logic from switches and places it in centralized controller. Switches which does not have control logic are used as a forwarding device. The routing operation is performed based on a set of rules which are declared to switches by the controller. Every SDN switch has a flow table. When a new flow arrives to a switch, the switch controls if there is a matching rule in the flow table. If the switch finds a matching rule, it applies the rule to the flow. If there is no match, switch sends the first packet of the flow to the controller. Controller analyses the packet and creates a rule for the flow. Then, controller sends the rule to the switch. The switch keeps that new rule in the flow table and applies it to the flow.

### 2.1.2. QoS-Aware Routing in SDN

Routing applications in SDN can perform more powerful routing methods with the abstract view of the network and real time statistics that are collected from switches. In [2], a network virtualization algorithm to isolate tenants and perform a QoS-aware routing algorithm on these isolated tenants was proposed. Application area of the algorithm selected as cloud computing and data centers since most of the cloud computing and data center applications use multi-tenancy architectures. In [11], users specify their bandwidth capacity expectations for each service. While system provides the bandwidth capacity to the user, it also optimizes utilization of a switch. Thus, QoS-aware routing for tenants is provided.

SCOR [12] Northbound API is introduced as a platform for QoS-aware routing application. SCOR platform takes QoS constraints of routing application as an input and outputs suitable paths to the applications. In addition, several QoS-aware routing applications were implemented to show applicability of SCOR.

Statistics of switches are also important for routing in SDN. In [3], an extended version of Dijkstra's shortest path algorithm was proposed. The proposed approach uses switch utilization as the weight of a node in order to compute the shortest path. In [4], NSV-Guard was proposed to construct paths in a secure manner. NSV-Guard computes trust values of switches by using switch statistics such as number of successfully delivered packets and the probability of successfully relaying packets. Then, NSV-Guard utilizes the path with the highest trust score.

A QoS-aware routing mechanism, namely OpenQoS, was proposed in [13] for achieving QoS in multimedia streaming. OpenQoS provides QoS by only considering multimedia flows. Then, paths of multimedia flows are computed by minimizing the delay for a constant jitter. An extended version of OpenQoS was proposed in [5] to operate on the distributed SDN environment for providing QoS-aware routing in multimedia streaming.

Since QoS metrics are prone to change frequently, a QoS-aware routing system should update routes periodically. To achieve seamless routing updates, [14] uses a multicast based scheme. According to the scheme, during the update, both new route and old route is preserved until the copies of the same package arrive to a switch. Then, the old route is permanently removed. The scheme is improved in [15]. The study compares the update process for all the possible alternative routes with Virtual Network Function (NFV) and chooses an optimal new route.

Even QoS-aware routing applications exist in the literature, these applications do not operate according to QoS requirements of users. These applications perform the same routing procedure for different types of users. To address that problem, AQoSAR performs routing according to QoS metric lists of users.

### 2.1.3. Authentication Mechanisms for SDN

Use of authentication mechanism is a pervasive approach for software defined networks. For instance, in [16], a lightweight authentication mechanism that operates between controller and switches was proposed. The mechanism confirms the authenticity of switches to controller. Another example usage of authentication mechanism in SDN is the authentication between switches and users in the network. Mechanisms in [17, 18] are example for such usage of authentication in SDN. In both mechanisms, user authentication is performed by using a trusted third party to regulate authentication in the network. Once the user is authenticated, its access to resources is controlled with respect to the user policy definitions in the authentication server. In this way, access control in network is achieved.

The provision of an authentication mechanism is also provided solutions for existing attack models for traditional networks such as the DNS flood attack. Such attacks are a type of Distributed Denial of Service (DDoS) attacks that endanger the availability of DNS servers. In [19], CAuth was proposed as a countermeasure by using IP spoofing to distinguish authenticated DNS queries in DNS requests while discarding unauthenticated ones. In [20], CAuth is improved and become compatible with existing DNS application and OpenFlow protocol. In [21], the model provides automated initialization for IPSec configurations in order to authenticate OpenFlow switches. In [22], FreeSurf is presented as a wireless access application in SDN. FreeSurf allows any service provider to authenticate their clients in a public network. Also, service providers can determine an access policy for their clients through Freesurf.

Since SDN switches are simple and capable to process excessive number of packets in limited time, SDN is suitable for high speed optical networks. Nevertheless, authentication issues of SDN are open for optical networks. In [23], authentication schemes are presented to provide mutual authentication and fine grained access control for SDN over optical networks. Since proposed schemes were based on only hash function, they promise a lightweight authentication mechanism.

Authentication is also important for verifying the identities of controllers when they communicate with each other as defined in [24, 25]. In [24], an authentication model that employs computation trust for the security of controller was proposed. The model provides authentication between two controllers without a trusted third party entity. In [25], authentication handover mechanism over SDN controllers was proposed. The mechanism automatically transfers authentication information of mobile users while they are moving from one network to another.

Authentication is an open problem for SDN as far as the QoS-aware routing is concerned. The existence of programmable devices may cause security vulnerabilities such as malicious users may take control of the programmable entities to monopolize resource utilization in the network. To address that problem, AQoSAR provides single and group authentication mechanisms.

## 2.2. Overview of Attribute Based Encryption

In this section, we start the with definition of Attribute Based Encryption (ABE). Then, we present applications of ABE. Finally, we introduce Attribute Based Authentication (ABA) and its applications.

### 2.2.1. Definition of Attribute Based Encryption

ABE is based on public-key cryptography. In conventional public-key cryptography, a message is encrypted for a specific receiver by using the public-key of receiver. In ABE, a message is encrypted for a set of predefined attributes. Participants who satisfy the predefined attributes can decrypt the message. ABE schemes can be divided as Key Policy Attribute Based Encryption (KP-ABE) and Ciphertext Policy Attribute Based Encryption (CP-ABE).

In KP-ABE, the ciphertext is labeled with a set of attributes and secret keys of entities are associated with access structures. [7]. An entity who has the correct access structure can decrypt the ciphertext. The basic flow of KP-ABE is as shown in Fig. 2.2. KP-ABE scheme consists of four algorithms.

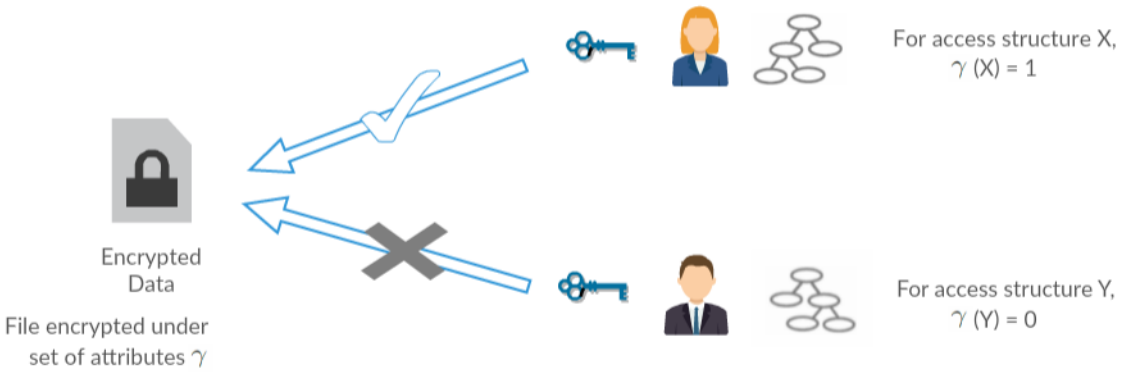



Figure 2.2. Basics of KP-ABE

- *Setup:* Master key MK and public parameters PK are generated.
- *Encrypt:* The algorithm takes a message M, a set of attributes γ and public parameters PK as input. It generates ciphertext CP.
- *Key Generation:* The algorithm takes access structure A, public parameters PK and master key MK as input. It generates private key of user SK.
- *Decrypt:* The algorithm takes ciphertext CP that was encrypted under the set of attributes γ, private key SK for access control structure A and public parameters PK. If γ satisfies A, the algorithm decrypts the ciphertext and outputs message MS.

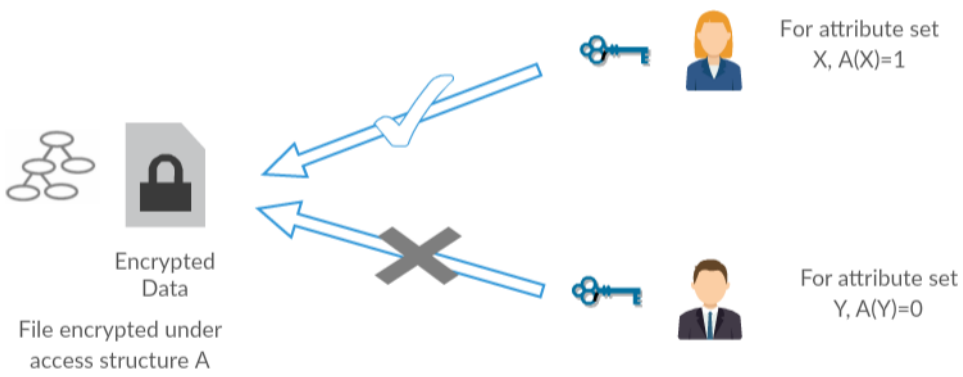



Figure 2.3. Basics of CP-ABE

The concept of CP-ABE is very similar to KP-ABE. However, in CP-ABE, the ciphertext is associated with an access structure and secret keys of entities [8]. An entity who has the correct set of attributes can decrypt the ciphertext. The basic flow of KP-ABE is as shown in Fig. 2.3. The scheme consists of four algorithms.

- *Setup:* MK and PK are generated.
- *Encrypt:* This algorithm takes a message M, an access structure A and PK as input. It generates ciphertext CP.
- *Key Generation:* This algorithm takes a set of attributes $\gamma$ and master key MK as an input. In generates private key of user SK.
- *Decrypt:* This algorithm takes ciphertext CP that contains access structure A, private key SK for set of attributes $\gamma$ and public parameters PK. If $\gamma$ satisfies A, algorithm decrypts the ciphertext and outputs message MS.

### 2.2.2. Applications of Attribute Based Encryption

ABE has many different application areas. One of the most popular application areas for ABE is Information Centric Networking (ICN). In ICN, data owner has no control after data is published. Therefore, as an access control mechanism, ABE is utilized for several ICN implementations. In [26], ABE is used to solve confidentiality issue of ICN. The protocol proposes to use attribute based routing to ensure confidentiality. Attribute based routing allows defining attributes for naming of content and performs access control based on the attributes. In [27], a modified version of ABE for ICN was proposed. The approach uses different attribute sets for content search and data access. Therefore, the content name is protected and privacy of the access policy is preserved. In [28, 29], ABE is implemented in sensor devices which have limited battery and computational power. These studies show that it is possible to use ABE for sensors who have limited battery and computational power. [30] uses ABE for both confidentiality and cache management in ICN. Attributes, which are used for defining access control of content, are also used for cache management of content.

In [31], CP-ABE is used for providing security management of personal health records. Since personal health records should be accessible for predefined users (e.g. doctors, physi-

cians), CP-ABE is suitable for the use case. In this study, personal records are encrypted according to attributes of predefined users. Thus, patient privacy is achieved with CP-ABE.

In [32], a model to provide attribute updates in CP-ABE was proposed. Normally, in CP-ABE, whenever an attribute changes, the entire private key of user should be changed. To solve the problem, [32] proposes a fading function which allows users to update each attribute separately. In [33], an approach that can recrypt data after policy change was proposed. The approach defines attribute duration for every entity. The authority generates and distributes secret keys based on attribute duration. When a new time period starts, entities switch secret keys which belongs to the new time period. In this way, an entity who does not has a valid secret key for the time period cannot decrypt data.

### 2.2.3. Attribute Based Authentication

In general, ABE schemes are used for providing the fine-grained access control on the encrypted content. Another use of ABE variant is to verify users with respect to the attributes assigned to them. Such usage is called as Attribute Based Authentication (ABA) [34–37]. The scheme in [34] is one of the example usages for ABA schemes. The proposed scheme uses group signature in order to provide group authentication. In [35], a hierarchical ABA scheme for cloud systems was proposed. The approach provides user based and attribute based hierarchical ABA for two scenarios. In [36, 37], privacy preserving ABA systems were proposed for health systems. In both frameworks, different privacy levels are used for entities in the network. All entities in the system are categorized with privacy levels. Then, these levels are used in order to authenticate users.

Another usage of ABA framework is the use for resource constrained devices as proposed in [38]. In the proposed framework, verification of identities is realized in a proxy server rather than the device itself due to the excessive computational cost of verification. Further examples for the use of ABA schemes together with proxy server are given in [39] and [40] to provide proxy signature for the privacy of secret keys of sender entities.

To preserve an access control mechanism for IoT environments, ABA is used in [41–43]. In [41], an ABA demo for IoT home environments is presented. Authentication of Things (AoT) is presented as a holistic authentication mechanism throughout the IoT device life-cycle in [42]. AoT provides access control and authentication for known IoT devices. In addition, AoT ensures interoperability for guest devices in a seamless manner. In [43], privacy of IoT devices is preserved by ABA. The approach uses attributes instead of identity of users to ensure unlinkability of device transactions.

ABA is used with other authentication mechanisms in [44, 45]. In [44] ABE and location information are used for authenticating users. In the scheme, Bluetooth beacons broadcast an encrypted message with an access policy. Users that are in the range of beacons and have correct attributes are able to decrypt the message and to login the system. In [45], biometric and attribute based encryption are jointly worked for providing a robust authentication mechanism.

Studies which are presented in the section show that CP-ABE is applicable when users have predefined attributes. In this thesis, we proposed an approach where all users in the system have predefined QoS attributes. Thus, CP-ABE is the best candidate as an authentication mechanism for our approach.

# 3. AQoSAR: AUTHENTICATED QOS AWARE ROUTING

In this chapter, we introduce Authenticated QoS Aware Routing (AQoSAR) for SDN. AQoSAR is a novel approach to achieve both QoS and authentication while determining routing paths of a user and a group of users in the network. As shown in Fig. 3.1, AQoSAR consists of two applications, namely the authentication application and the routing application. The authentication application is responsible for verifying identities of users and group of users by considering their privileges. During the authentication process, QoS requirements of users are recorded to QoS metric list. Each user has a QoS metric list which defines QoS requirements of the user. The authentication process can be realized by using the certificate authority and the authentication application in the controller. The certificate authority is responsible for the distribution of certificates to users with respect to their privileges. The authentication application is a computer program that runs on the controller to verify certificates of users. When the certificates of users are verified, the list of authenticated users is transmitted to the routing application ((1) in Fig. 3.1). Later, the routing application determines paths for authenticated users by considering the metric list of each user. As a part of the routing application, statistics of switches and links are collected by controller ((2) in Fig. 3.1). Controller transmits network statistics to the routing application ((3) in Fig. 3.1). With the metric lists of users and the network statistics, routing application constructs an appropriate path for each user or group.

QoS of users depends on limited resources of a network. If all users can access all of the resources in network, then bottlenecks occur in the network. These bottlenecks prevent the system to provide QoS to users. Therefore, to ensure QoS of users, a mechanism which controls access of users to limited resources is needed. Such access control has two main functionalities. The first functionality is to prevent system resources against the attempts of malicious users to monopolize resources. If any malicious user in network manages to reserve a high amount of resource for itself, it will directly decrease QoS of other users. The second functionality is to separate users based on their privileges. In this way, AQoSAR prioritizes to serve high privileged users.

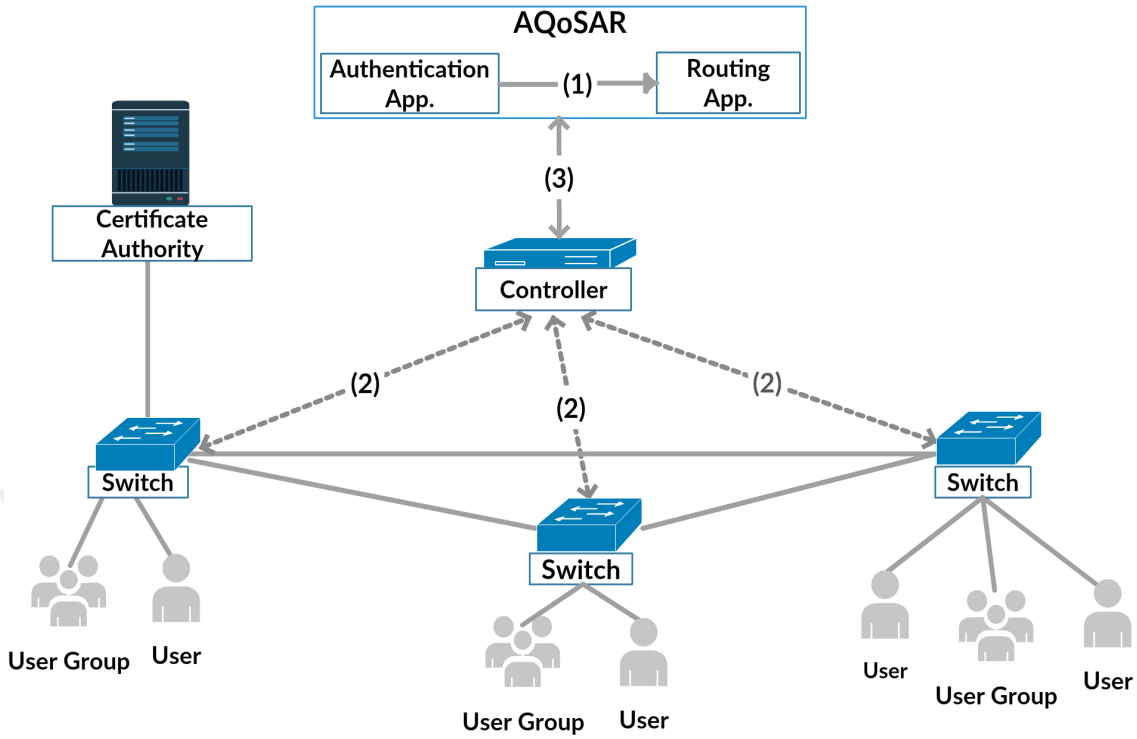



Figure 3.1. Overview of AQoSAR

Another important goal of AQoSAR is to enable each user to personalize their QoS requirements. With the pervasive use of communication technologies, device variety and number of devices have been increased. Increase of device variety results as a user diversity. As a consequence of user diversity, users become to have different QoS expectations. For instance, a health monitor sensor and a smart TV can be found in the same hospital network. Main QoS requirements of smart TV are high bandwidth and low latency. In contrast to smart TV, a health monitor sensor can operate with low bandwidth. However, high reliability is vital for a health monitor system. In addition, people can use the same type of device for different purposes. Therefore, using the same QoS metric for all users does not ensure QoS. As a solution to the problem, routing module of AQoSAR performs QoS-aware routing based QoS metric lists of users.

### 3.1. Authentication Application

Authentication application of AQoSAR is used for increasing the security of QoS-aware routing. The application prevents malicious users from accessing routing benefits of privileged users. Authentication operation is performed by considering user privileges with

respect to their QoS metric list. Authentication application consists of two protocols, namely single user authentication protocol and group authentication protocol. The application uses El-Gamal encryption algorithm [46], Schnorr signature [47] and CP-ABE algorithm [8] for confidentiality, authentication among entities and authentication for QoS-aware routing, respectively. Notations for the authentication application are as given in Table 3.1.

Table 3.1. Notations for the Authentication Application

| **Symbol** | **Definition** | **Symbol** | **Definition** |
|---|---|---|---|
| $CA$ | Certificate Authority | $ME_n$ | $n^{th}$ Member of Group |
| $C$ | Controller | $MA$ | Group Manager |
| $e$ | Entity | $MS_n$ | $n^{th}$ Message |
| $E_{ABE,\gamma}$ | Attribute Based Encryption for metric list $\gamma$ | $n$ | Randomly Generated Nonce |
| $E_k$ | Encryption with Key k | $PK_e$ | Public Key of Entity e |
| $G$ | Group | $SK_e$ | Secret Key of Entity e |
| $h()$ | Hash Function | $SK_{ABE,e}$ | Secret ABE Key of Entity e |
| $ID_e$ | ID of Entity e | $t$ | Timestamp |
| $U$ | User | $\gamma_e$ | Metric List of Entity e |

Before introducing single user authentication and group authentication protocols, first, we present definitions of ElGamal encryption algorithm, Schnorr signature and CP-ABE in Definitions 1-3.

**Definition 1.** *El-Gamal is an asymmetric key encryption scheme for public-key cryptography [46]. We use El-Gamal to provide confidentiality of messages among entities. El-Gamal scheme operates as shown in Fig. 3.2.*

**Definition 2.** *We prefer to use Schnorr signature scheme [47] as a signature scheme for messages. The scheme operates as given in Fig. 3.3.*

Let $G$ be a cyclic group of prime order $q$ and $g$ be a generator of $G$.

**KeyGeneration** : Key generation algorithm chooses $PK \leftarrow_{\$} \{-1,0,...,q-1\}$ and computes $SK = g^{PK}$. The private key is $SK$ and the public key is $PK$.

**Encrypt**$(\mathbf{m},\mathbf{PK})$ : Encrypt algorithm takes message m and public key $PK$ as input. The algorithm chooses random $a \leftarrow_{\$} \{-1,0,...,q-1\}$, compute $c_1 = g^a$, $s = PK^a$ and maps message $m$ on to $m' \leftarrow_{\$} G$. Then, computes $c_2 = m'.s$ . The ciphertext is $(c_1,c_2)$.

**Decrypt**$(\mathbf{SK},(\mathbf{c_1},\mathbf{c_2}))$ : Decrypt algorithm takes ciphertext $(c_1,c_2)$ and private key $SK$ as input. The algorithm computes $s = c_1^{SK}$ and $m' = c_2.s^{-1}$. Then, the algorithm converts $m'$ to $m$.

Figure 3.2. El-Gamal Encryption Scheme

Let $G$ be a cyclic group of prime order $q$ and $g$ be a generator of $G$. Let $H : \{0,1\}* \times G \rightarrow Z_q$ be a hash function.

**KeyGeneration** : Key generation algorithm chooses $PK \leftarrow_{\$} \mathbb{Z}\backslash\{0\}$, and computes $PK = g^{SK}$. The private key is $PK$ and the public key is $SK$.

**Sign**$(\mathbf{m},\mathbf{SK})$ : Sign algorithm takes message m and private key $SK$ as input. The algorithm chooses random $a \leftarrow_{\$} \mathbb{Z}$, compute $r = g^a$, $c = H(m,r)$ and $s = a + c \cdot SK \mod q$. The signature is $(s,c)$.

**Verify**$(\mathbf{m},\mathbf{PK},(\mathbf{s},\mathbf{c}))$ : Verify algorithm takes message m, public key $PK$ and signature $(s,c)$ as input. The algorithm computes $r = g^s \cdot PK^{-c}$ and verifies $c = H(m,r)$.

Figure 3.3. Schnorr Signature Scheme

**Definition 3.** *CP-ABE scheme is a variant of ABE which provides fine-grained access control for the encrypted text by using attributes of users [8]. In CP-ABE, the ciphertext is associated with an access structure and secret keys of entities are labeled with the set of attributes. Only users, who have the correct set of attributes are able to decrypt the encrypted text. We use CP-ABE for authentication of messages while defining QoS-aware routing privileges. CP-ABE algorithm for AQoSAR operates as defined in Fig. 3.4.*

### 3.1.1. Single User Authentication

As the first functionality of the authentication module, AQoSAR provides single user authentication to users. A legitimate user who wants to use privileges of itself can send an authentication request to certificate authority. Certificate authority produces a certificate

Let $U$ be user, $\gamma_U$ is metric list of user, $CA$ is certificate authority and $C$ is controller.

**Setup** : CA generates public parameters PP and a master key MK.

**KeyGeneration**$(\mathbf{MK}, \gamma_\mathbf{U})$ : CA generates secret key $SK_{ABE,U}$ for user U by using master key MK and metric list of user $U$ $\gamma_U$.

**Encrypt**$(\mathbf{PP}, \mathbf{n}, \gamma_\mathbf{U})$ : C encrypts nonce $n$ for user $U$ by using public parameters $PK$ and metric list of user $U$ $\gamma_U$ and outputs ciphertext CT.

**Decrypt**$(\mathbf{PP}, \mathbf{CT}, \mathbf{SK_{ABE,U}})$ : U decrypts ciphertext CT by using public parameters $PP$ and it's secret key $SK_{ABE,U}$. The algorithm outputs nonce $n$.

Figure 3.4. CP-ABE Scheme

for a user if certificate authority confirms that the user has an authority to perform single user authentication. User can use the certificate for authenticating itself to authentication application. Sequence diagram of the single user authentication protocol is as shown in Fig. 3.5. Authentication of a user is realized in two steps. In the first step, a certificate that is produced by certificate authority is used. In the second step, a nonce which is encrypted with CP-ABE is used.

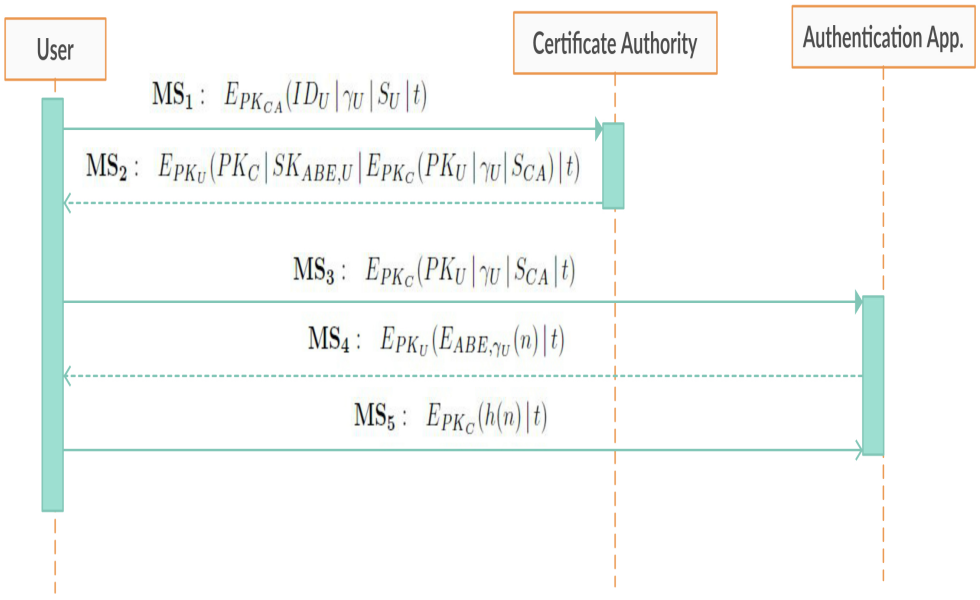



Figure 3.5. Sequence Diagram of Single User Authentication

Let $U$ be the user, $SK_U$ is secret key of user $U$, $PK_U$ is public key of user $U$, $\gamma_U$ is metric list of user $U$ and $SK_{ABE,U}$ is CP-ABE key of user $U$. In addition, $CA$ denotes Certificate Authority and $C$ denotes controller. Single user authentication protocol for a user $U$ operates as given in Fig. 3.6.

1.1 : $U$ signs $\gamma_U$, $ID_U$ and $t$ with $SK_U$, encrypts them with $PK_{CA}$ and sends it to $CA$.

$$\mathbf{MS_1}: \quad E_{PK_{CA}}(ID_U \,|\, \gamma_U \,|\, S_U \,|\, t)$$

2.1 : $CA$ decrypts $MS_1$ with $SK_{CA}$, then verifies $S_U$ with $PK_U$.
2.2 : $CA$ generates $SK_{ABE,U}$ based on $\gamma_U$.
2.3 : $CA$ signs $PK_U$ and $\gamma_U$ with $SK_{CA}$, then encrypts them with $PK_C$.
2.4 : $CA$ encrypts $(PK_C \,|\, SK_{ABE,U} \,|\, E_{PK_C}(ID_U \,|\, \gamma_U \,|\, S_{CA})$ and t with $PK_U$, then sends it to U.

$$\mathbf{MS_2}: \quad E_{PK_U}(PK_C \,|\, SK_{ABE,U} \,|\, E_{PK_C}(PK_U \,|\, \gamma_U \,|\, S_{CA}) \,|\, t)$$

3.1 : $U$ decrypts $MS_2$ with $SK_U$ and redirects $E_{PK_C}(E_{SK_{CA}}(PK_U|\gamma_U)|t)$ to $C$.

$$\mathbf{MS_3}: \quad E_{PK_C}(PK_U \,|\, \gamma_U \,|\, S_{CA} \,|\, t)$$

4.1 : $C$ decrypts $MS_3$ with $SK_C$, then verifies $S_{CA}$ with $PK_{CA}$.
4.2 : $C$ generates a nonce and encypts it with $E_{ABE,\gamma_U}$
4.3 : $C$ encrypts $E_{ABE,\gamma_U}(n)$ and $t$ with $E_{PK_U}$ and sends it to U.

$$\mathbf{MS_4}: \quad E_{PK_U}(E_{ABE,\gamma_U}(n)|t)$$

5.1 : $U$ first decrypts $MS_4$ with $SK_U$, then decrypts $E_{ABE,\gamma_U}(n)$ with $SK_{ABE,U}$.
5.2 : $U$ take hash of $n$ with $h()$, encrypts $h(n)$ and t with $PK_C$ and sends it to $C$.

$$\mathbf{MS_5}: \quad E_{PK_C}(h(n)|t)$$

6.1 : $C$ decrypts $MS_5$ with $SK_C$.
6.2 : **if** $h(n)$ is correct **then**
6.2.1 : Authentication of U is completed.
6.3 : **else**
6.3.1 : Authentication of U is failed.
6.3.1 : **end if**

Figure 3.6. Details of Single User Authentication

### 3.1.2. Group Authentication

To provide authentication for a group of users, AQoSAR offers a dynamic group authentication mechanism. A predefined group of users can use group authentication to have the same QoS privileges. Group authentication protocol consists of three operations, namely group creation, join and leave. Presence of join and leave operation is increasing the applicability of AQoSAR into real world scenarios. If any modification on the group is needed, group manager can perform these operations instead of forming a new group.

In group authentication, we use the extended version of the single user authentication protocol. First, users, who form the group, are delegated as group managers. Group man-

agers are responsible for determining the metric list for the QoS-aware routing. Moreover, group manager also needs to specify a member list. After the metric list and member list are determined, group manager transmits them to certificate authority. Then, certificate authority distributes certificates to group members. Users can authenticate themselves to controller with the certificates. After a group is formed, modification on group can be performed with dynamic group operations. The dynamic group operations are realized by group manager.

In addition to use single user authentication and group authentication protocols apart, these protocols can be used together for determining multi-level QoS-aware routing for user. For instance, some users may have high privileges when they are communicating as a group member and they may have low privileges when communicating as a single user. As a consequence, it is possible to assign different QoS levels for users by considering their memberships for group in the network.

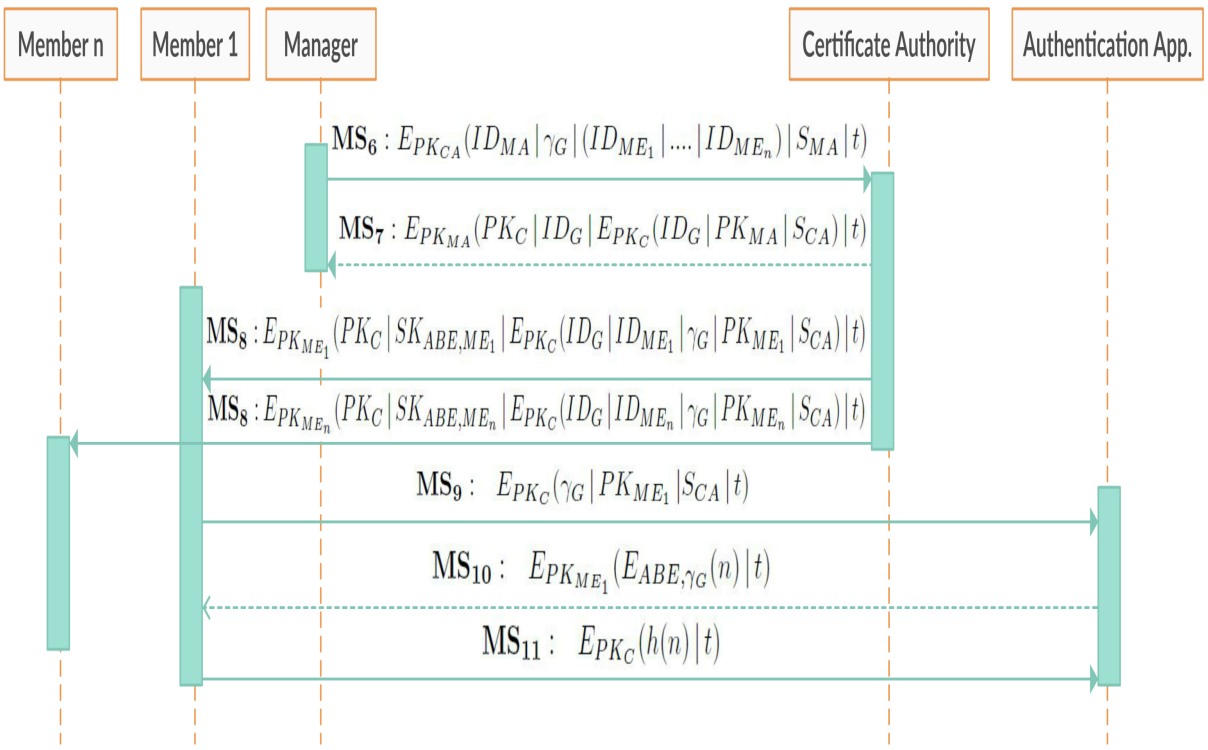



Figure 3.7. Sequence Diagram of Create Operation

3.1.2.1. Create Operation. Create operation is the first operation of the group authentication. To perform create authentication, a user who is authorized as a group manager is needed. As a first step of the create operation, the manager sends a group creation request

1.1 : $MA$ signs $ID_{MA}$, $\gamma_G$, $ID_{ME_1}|....|ID_{ME_n}$ and t with $SK_{MA}$.
1.2 : $MA$ encrypts $(ID_{MA}\,|\,\gamma_G\,|\,(ID_{ME_1}\,|\,....\,|\,ID_{ME_n})\,|\,S_{MA}\,|\,t)$ with $PK_{CA}$ and send it to $CA$.

**MS$_6$** :
$E_{PK_{CA}}(ID_{MA}\,|\,\gamma_G\,|\,(ID_{ME_1}\,|\,....\,|\,ID_{ME_n})\,|\,S_{MA}\,|\,t)$

2.1 : $CA$ decrypts $MS_6$ with $PK_{CA}$, then verifies $S_{MA}$ with $PK_{MA}$.
2.2 : $CA$ signs $(ID_G\,|\,PK_{MA})$ with $SK_{CA}$ and encrypts it with $PK_C$.
2.3 : $CA$ encrypts $E_{PK_C}(ID_G\,|\,PK_{MA}\,|\,S_{CA})$,$PK_C$, $ID_G$ and $t$ with $PK_{MA}$ and sends it to $MA$.

**MS$_7$** :
$E_{PK_{MA}}(PK_C\,|\,ID_G\,|\,E_{PK_C}(ID_G\,|\,PK_{MA}\,|\,S_{CA})\,|\,t)$

3.1 : $CA$ generates $SK_{ABE,ME}$ based on $\gamma_G$.
3.2 : **for** each $ME$ in $(ID_{ME_1}|....|ID_{ME_n})$ **do**
3.2.1 : $CA$ signs $PK_{ME}$ and $\gamma_G$ with $SK_{CA}$.
3.2.2 : $CA$ encrypts $(\gamma_G|PK_{ME}|S_{CA})$ with $K_C$.
3.2.3 : $CA$ encrypts $E_{PK_C}(\gamma_G|PK_{ME}|S_{CA})$ , $PK_C$, $SK_{ABE,ME}$ and $t$ with $PK_{ME}$ and sends it to $ME$.
3.2.4 : **end for**

**MS$_8$** :$E_{PK_{ME_1}}(PK_C|SK_{ABE,ME_1}|$
$E_{PK_C}(\gamma_G|PK_{ME_1}|S_{CA})|t)$

4.1 : $ME_1$ decrypts $MS_8$ with $SK_{ME_1}$ and redirects $E_{PK_C}(\gamma_G\,|\,PK_{ME_1}\,|\,S_{CA}|t)$ to $C$.

**MS$_9$** :$E_{PK_C}(\gamma_G\,|\,PK_{ME_1}\,|S_{CA}\,|\,t)$

5.1 : $C$ first decrypts $MS_9$ with $SK_C$, then verifies $S_{CA}$ with $PK_{CA}$.
5.2 : $C$ generates a nonce and encypts it with $E_{ABE,\gamma_G}$.
5.3 : $C$ encrypts $E_{ABE,\gamma_G}(n)$ and $t$ with $E_{PK_{ME_1}}$ and sends it to $ME_1$.

**MS$_{10}$** : $E_{PK_{ME_1}}(E_{ABE,\gamma_G}(n)|t)$

6.1 : $ME_1$ first decrypts $MS_{10}$ with $SK_{ME_1}$, then decrypts $E_{ABE,\gamma_G}(n)$ with $SK_{ABE,ME_1}$.
6.2 : $ME_1$ take hash of $n$ with $h()$, encrypts $h(n)$ and t with $PK_C$ and sends it to $C$.

**MS$_{11}$** : $E_{PK_C}(h(n)|t)$

7.1 : $C$ decrypts $MS_{11}$ with $SK_C$.
7.2 : **if** $h(n)$ is correct **then**
7.2.1 : Authentication of $ME_1$ is completed.
7.3 : **else**
7.3.1 : Authentication of $ME_1$ is failed.
7.3.1 : **end if**

Figure 3.8. Details of Create Operation

to the certificate authority. The request includes metric list and member list. If the request of the manager is approved by the certificate authority, group certificates are distributed to group members. Then, group members can proceed authentication with their certificates. Sequence diagram of create operation is shown in Fig. 3.7.

Let $G$ be the privileged group, $MA$ be manager of the group $G$ and $ME_n$ denotes $n^{th}$ member of group. $SK_{MA}$ is secret key of manager $MA$, $PK_{MA}$ is public key of manager $MA$, $S_{MA}$ is signature of manager $MA$. $SK_{ME_n}$ is secret key of member $ME_n$, $PK_{ME_n}$ is public key of member $ME_n$, $SK_{ABE,ME_n}$ is CP-ABE key of member $ME_n$. $\gamma_G$ is metric list of group $G$. In addition, $CA$ denotes certificate authority and $C$ denotes controller. Create operation for group $G$ operates as shown in Fig. 3.8.

3.1.2.2. Join Operation. When a new member wants to join a group, join operation is performed. To perform the join operation, consent of the group manager is needed. As the first step of join operation, group manager sends a request which includes list of new members and group ID to certificate authority. If certificate authority approves the request, it produces and distributes certificates to new members. Then, new members can proceed authentication with their certificates. Sequence diagram of join operation is shown in Fig. 3.9.

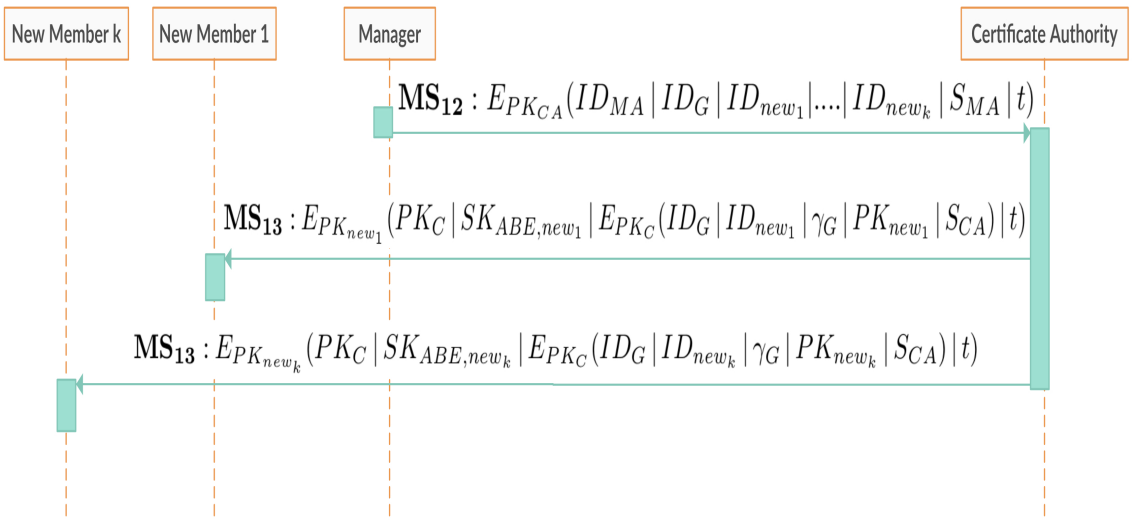



Figure 3.9. Sequence Diagram of Join Operation

Let $G$ be the privileged group, $MA$ be manager of the group $G$ and $new_n$ denotes $n^{th}$ new member of group. $SK_{MA}$ is secret key of manager $MA$, $PK_{MA}$ is public key of manager $MA$, $S_{MA}$ is signature of manager $MA$. $SK_{new_n}$ is secret key of new member $new_n$, $PK_{new_n}$ is public key of new member $new_n$, $SK_{ABE,new_n}$ is CP-ABE key of member $new_n$. $\gamma_G$ is metric list of group $G$. In addition, $CA$ denotes certificate authority and $C$ denotes controller. Join operation for group $G$ operates as shown in Fig. 3.10.

1.1 : $MA$ signs $ID_{MA}|ID_G|(ID_{new_1}|....|ID_{new_k})$ with $SK_{MA}$.
1.2 : $MA$ encrypts $(ID_{MA}\,|\,ID_G\,|\,ID_{new_1}|....|\,ID_{new_k}\,|\,S_{MA}\,|\,t)$ with $E_{PK_{CA}}$ and sends it to $CA$.

**MS$_{12}$** :
$E_{PK_{CA}}(ID_{MA}\,|\,ID_G\,|\,ID_{new_1}|....|\,ID_{new_k}\,|\,S_{MA}\,|\,t)$

2.1 : $CA$ decrypts $MS_{12}$ with $PK_{CA}$, then verifies $S_{MA}$ with $PK_{MA}$.

2.2 : $CA$ generates $SK_{ABE,new}$ based on $\gamma_G$.
2.3 : **for** each $ME$ in $(ID_{new_1}|....|ID_{new_n})$ **do**
2.3.1 : $CA$ signs $PK_{new}$ and $\gamma_G$ with $SK_{CA}$.
2.3.2 : $CA$ encrypts $(\gamma_G|PK_{new}|S_{CA})$ with $K_C$.
2.3.3 : $CA$ encrypts $E_{PK_C}(\gamma_G|PK_{new}|S_{CA})$ , $PK_C$, $SK_{ABE,new}$ and $t$ with $PK_{new}$ and sends it to $ME$.
2.3.4 : **end for**

**MS$_{13}$** :$E_{PK_{new_1}}(PK_C|SK_{ABE,new_1}|$
$E_{PK_C}(ID_G|ID_{new_1}|\gamma_G|PK_{new_1}|S_{CA})|t)$

Figure 3.10. Details of Join Operation

3.1.2.3. Leave Operation. When a member should be expelled from group, leave operation is performed. As the first step of leave operation, group manager sends a request which includes list of expelled members and manager certificate to the authentication application directly. If the authentication application approves the request, it puts expelled members to the blacklist of the group. As expelled members are in the blacklist, they cannot use routing privileges of the group. Sequence diagram of leave operation is shown in Fig. 3.11.

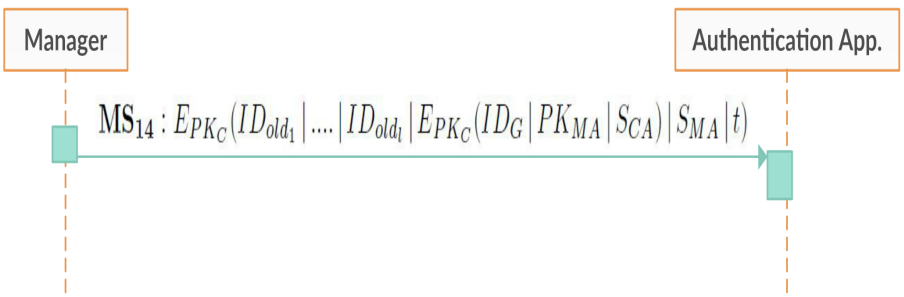



Figure 3.11. Sequence Diagram of Leave Operation

Let $G$ be the privileged group, $MA$ be manager of the group $G$ and $old_n$ denotes $n^{th}$ old member who left the group. $SK_{MA}$ is secret key of manager $MA$, $PK_{MA}$ is public key of manager $MA$, $S_{MA}$ is signature of manager $MA$. $CA$ denotes certificate authority and $C$ denotes controller. Leave operation for group $G$ operates as shown in Fig. 3.12.

| | |
|---|---|
| 1.1 : $MA$ signs $(ID_{old_1} \mid .... \mid ID_{old_l})$ and $E_{PK_C}(ID_G \mid PK_{MA} \mid S_{CA})$ with $SK_{MA}$, encrypts it with $PK_C$ and sends it to $C$.<br><br>$\mathbf{MS_{14}}$ :$E_{PK_C}(ID_{old_1} \mid .... \mid ID_{old_l} \mid E_{PK_C}(ID_G \mid PK_{MA} \mid S_{CA}) \mid S_{MA} \mid t)$ | 2.1 : $C$ decrypts $MS_9$ with $SK_C$, then verifies $S_{MA}$ with $PK_{MA}$.<br>2.2 : $C$ decrypts $E_{PK_C}(ID_G \mid PK_{MA} \mid S_{CA})$ with $SK_C$, then verifies $S_{CA}$ with $PK_{CA}$.<br>2.3 : $C$ adds $(ID_{old_1} \mid .... \mid ID_{old_l})$ to the blacklist of group $G$. |

Figure 3.12. Details of Leave Operation

## 3.2. Routing Application

The routing application performs routing operations based on QoS metric list of users and groups. The application analyzes incoming packets with respect to privileges of their owners. Then, the application determines the most suitable path for the packets by considering the network statistics collected from switches. QoS level for each user and group can be determined by using the combination of the following metrics:

- *Availability:* This metric defines availability expectation of a user. Availability of a switch is determined by using the packet drop ratio of a switch.
- *Reliability:* This metric defines reliability expectation of a user. Reliability of a switch is determined by the up-time of a switch.
- *Cost:* This metric defines the node weight for calculating the shortest path. Cost has nominal values, where $\mathcal{C} = \{utilization, delay, min\text{-}hop\}$.
- *Bandwidth Capacity:* This metric defines bandwidth capacity expectation of a user.
- *Criticality:* The metric is used for defining criticality of packets to be transmitted for each user. Criticality can only have $+1$ or $-1$ values.

As an example use case scenario, to meet the reliability and the availability expectations of users, the routing application defines reliability and availability levels for each switch by considering the packet drop ratio and the up-time of a switch. Then, routing application detects and isolates switches with low reliability and low availability for high privileged users that request more secure communication.

QoS-aware routing problem in SDN is defined as a constraint shortest path problem in [13]. Since we have multiple metrics for QoS-aware routing, we define our routing problem as a multi-constraint shortest path (MCSP) problem. We prefer to use the cost metric of a user as a weight of MCSP problem. For other constraints of MCSP problem, we use availability, reliability and bandwidth capacity metrics of users. In our formulation, $(i, j)$ pair represents link between node $i$ and node $j$. Let $P(source, destination)$ be the set of all paths from source node to destination node, For any path $p \in P(source, destination)$, we define cost function $f_c(p)$ as:

$$f_c(p) = \sum_{(i,j) \in p} c_{ij} \tag{3.1}$$

where $c_{ij}$ is cost of $(i, j)$ link. If cost metric of user is utilization, $c_{ij}$ denotes utilization of $(i, j)$, if the metric is delay, $c_{ij}$ denotes delay on $(i, j)$ and if the metric is min-hop, $c_{ij}$ is 1. We define our availability constraint for path $p$ as follows:

$$f_a(p) = \begin{cases} 1, & \forall s \in p, a_s \geq a_u \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

where $s$ denotes a switch, $a_s$ denotes the availability level of the switch $s$ and $a_u$ denotes availability metric of user $u$. We define our reliability constraint for path $p$ as follows:

$$f_r(p) = \begin{cases} 1, & \forall s \in p, r_s \geq r_u \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

where $r_s$ denotes reliability level of a switch $s$ and $r_u$ denotes reliability metric of user $u$. We define our bandwidth capacity constraint for path $p$ as follows:

$$f_b(p) = \begin{cases} 1, & \forall (i,j) \in p, b_{(i,j)} \geq b_u \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

where $b_{i,j}$ denotes bandwidth capacity of $(i,j)$ link and $b_u$ denotes bandwidth metric of user $u$. Then, we can formalize our MCSP problem as:

$$\begin{aligned} p^* = \arg\min_p \{ f_c(p) \mid p \in P(s,d), \\ f_a(p) = 1, f_r(p) = 1, f_b(p) = 1 \} \end{aligned} \tag{3.5}$$

When a new flow arrives to a switch, it forwards the first packet of the flow to the controller. If source node or destination node of the flow is authenticated user, path of the flow is constructed based on the path construction procedure. If both source and destination nodes are unauthenticated users, default routing procedure on the controller is performed.

The path construction procedure of routing application is as shown in Fig. 3.13. The procedure is a modified version of Dijkstra's shortest path algorithm which calculates the shortest path based on the cost metric of the authenticated user. In addition, the procedure eliminates nodes and links which do not satisfy availability, reliability and bandwidth capacity constraints of a user. If there is no path which satisfies QoS constraints of an authenticated user, criticality metric of the user is used for determining the next step. If the criticality metric of a user is $+1$, packets of the user are identified as critical packets. Therefore, all packets that come from the user are dropped. If the metric is $-1$, QoS constraints of the user are discarded and the path which offers lower cost is used.

QoS metrics, which are used in the routing application, are prone to change frequently. Therefore, paths should be recalculated periodically. As a solution, the routing application

assigns a lifetime to each path. After lifetime of a path expires, the path will be deleted from flow tables of switches. Hence, AQoSAR ensures that QoS expectations of users are fulfilled.

**Input:**
**Graph**: Network Topology
**source**: Source Node
**destination**: Destination Node
**U**: Authenticated User
$\{\mathbf{r_u}, \mathbf{a_u}, \mathbf{c_u}, \mathbf{b_u}\}$: reliability, availability, cost and metrics of user U

```
1:  procedure PATHCONSTRUCTOR
2:      dist[source] ← 0
3:      prev[source] ← Undefined
4:      Q ← Empty
5:      for each node n in Graph do
6:          if n ≠ source then
7:              dist[n] ← ∞
8:              prev[n] ← undefined
9:              add n to Q
10:         end if
11:     end for
12:     while Q is not empty do
13:         p ← node in Q with min dist[p]
14:         remove p from Q
15:         if p == destination then
16:             return dist[p], prev[p]
17:         end if
18:         for each neighbor r of p do
19:             if ( r_r <r_u || a_r <a_u || b_(p,r) <b_u ) then
20:                 remove r from Q
21:             else
22:                 calculate c_(p,r) based on c_u
23:                 if dist[p] + c_(p,r) <dist[r] then
24:                     dist[r] ← dist[p] + c_(p,r)
25:                     prev[r] ← p
26:                 end if
27:             end if
28:         end for
29:     end while
30:     return dist[p], prev[p]
31: end procedure
```

Figure 3.13. Path Construction Procedure

### 3.3. Use Case Scenario For AQoSAR: A University Department

To have a further understanding about applicability of AQoSAR, we provide a use case scenario. Due to the nature of a university department, there are different types of users in the department. For our use case scenario, we divide these users into four categories. These categories are professors, heads of research labs, research lab members and undergrad students. Each of these categories corresponds to an authentication level. We explain behaviors of AQoSAR for four different authorization levels.

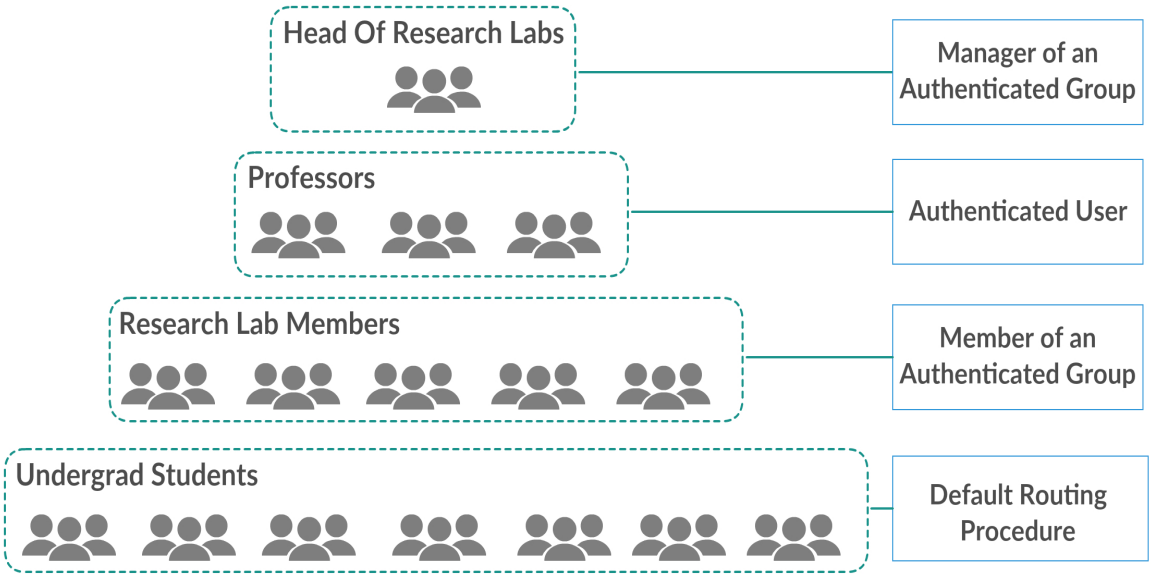



Figure 3.14. Authentication Levels of the University Department

As a first authentication level, we examine authorities of professors. Since QoS of a professor can be considered as more important than other users in network, AQoSAR allows professors to choose their QoS parameters. In other words, professors are allowed to perform single user authentication in the system. In this way, AQoSAR ensures that highly prioritized users - in our case, professors - have high QoS.

Heads of research labs are our second authentication level. In a department, there are different research labs each of which has different research interests. Therefore, expectation of users differs. For instance, the lab that studies network security may need high reliability and the lab that studies computer vision may need high bandwidth capacity due to the amount of data transferred at a time. In our scenario, each lab has a group manager in order to set

necessary metrics for the group authentication. Head of each research lab can select suitable QoS parameters for their lab members. Moreover, the head of a lab can select lab members. When a new member joins the lab or a member leaves the lab, head of the lab can use join/leave functionalities of group authentication to arrange lab members dynamically.

Research lab members are the third authentication level in our use case scenario. Lab members are people who spend lots of time in departments and needs a certain level of QoS. Therefore, the head of each lab can form own group metrics accordingly and execute group authentication protocol. If a lab member is declared as a group member by the head of the lab, he/she can perform group authentication and use QoS parameters of the group. Nevertheless, lab members are not authorized to perform single user authentication.

In the department scenario, there are also users who do not belong to any authentication level such as undergraduate students or visitors. This type of users cannot perform any authentication method of AQoSAR. The department either has a default QoS policy for this type of users or directly permit these users if network resources are very limited.

In this section, we provide a use scenario for exploring possible usage areas of AQoSAR. Our use case scenario includes users and group of users who have different priority level. The use case scenario demonstrates that AQoSAR is able to serve users who have different priority level and different QoS requirements.

# 4. SECURITY OF AQoSAR

In this chapter, we analyze the security of AQoSAR. Impersonation attack, replay attack and eavesdropping are the most popular attacks which are performed to an authentication mechanism [48]. An authentication mechanism should provide absolute security against these types of attacks. In addition, collision attacks are one of the most important security threats for ABE variants [49]. Since the authentication mechanism of AQoSAR relies on CP-ABE, the security of AQoSAR should be examined against collision attacks. Thus, we analyze the security of AQoSAR against four different attack types, namely impersonation attack, replay attack, collision attack and eavesdropping.

Impersonation attack is a type of active attacks that an adversary impersonate identity of one of the legitimate entities. AQoSAR should be secured against impersonation attack in order to prevent access of malicious user to network resources. Theorem 4.1, 4.2, 4.3 and 4.4 indicates that both single authentication and group authentication of AQoSAR are secure against impersonation attack.

**Theorem 4.1.** *Under the difficulty of discrete logarithm problem, single user authentication is secure against impersonation attacks.*

*Proof.* To access routing privileges of legitimate users in the single user authentication, adversary should impersonate messages $MS_1$, $MS_3$, $MS_5$ of legitimate users.

$$\mathbf{MS_1}: \quad E_{PK_{CA}}(ID_U \,|\, \gamma_U \,|\, S_U \,|\, t)$$

In $MS_1$, an adversary tries to impersonate itself as a legitimate user. As defined in $MS_1$, user signs the message by using Schnorr signature. Since Schnorr signature is secure against impersonation attacks as proposed in [47], adversary should have $SK_U$ to impersonate $MS_1$.

Without obtaining $SK_U$, the adversary cannot impersonate itself as a legitimate user.

$$\mathbf{MS_3}: \quad E_{PK_C}(PK_U \,|\, \gamma_U \,|\, S_{CA} \,|\, t)$$

In $MS_3$, an adversary can impersonate itself as a legitimate user and tries to generate $MS_3$ which is signed by CA. Since Schnorr signature is secure against impersonation attacks, the adversary is not able to obtain $SK_{CA}$ from $MS_3$.

$$\mathbf{MS_5}: \quad E_{PK_C}(h(n)|t)$$

To impersonate $MS_5$, adversary need to have the nonce in $MS_4$. $MS_4$ is encrypted with both $PK_U$ and $E_{ABE,\gamma_U}$. Even adversary obtains $MS_4$, he/she should have $SK_U$ and $SK_{ABE,U}$ to decrypt the message, which is as hard as the discrete logarithm problem. Therefore, it is not possible for the adversary to impersonate a legitimate user. Thus, by considering the security of messages $MS_1$, $MS_3$, $MS_5$, the single authentication protocol is secure against the impersonation attacks. □

**Theorem 4.2.** *Under the difficulty of discrete logarithm problem, create operation of group authentication provides resistance against impersonation attacks.*

*Proof.* To access routing privileges of legitimate users in the group authentication, adversary should impersonate messages $MS_6$, $MS_9$, $MS_{11}$ from legitimate users.

$$\mathbf{MS_6}: \quad E_{PK_{CA}}(ID_{MA} \,|\, \gamma_G \,|\, (ID_{ME_1} \,|\, .... \,|\, ID_{ME_n}) \,|\, S_{MA} \,|\, t)$$

In $MS_6$, adversary can impersonate itself as a group manager. As defined in $MS_6$, manager signs $\gamma_G$ and member list. Since Schnorr signature is secure against impersonation attacks as proposed in [47], adversary should have $SK_{MA}$ to impersonate $MS_6$. As adversary does not have $SK_{MA}$, he/she cannot impersonate $MS_6$.

$$\mathbf{MS_9}: \quad E_{PK_C}(\gamma_G \,|\, PK_{ME} \,|\, S_{CA} \,|\, t)$$

In $MS_9$, adversary can impersonate itself as a group member. As defined in $MS_9$, CA signs $(\gamma_G, ID_{ME})$. Since Schnorr signature is secure against impersonation attacks as proposed in [47], adversary should have $SK_{CA}$ to impersonate $MS_9$. As adversary does not have $SK_{CA}$, he/she cannot impersonate $MS_9$.

$$\mathbf{MS_{11}}: \quad E_{PK_C}(h(n)|t)$$

To impersonate $MS_{11}$, adversary needs to have the nonce in $MS_{10}$. $MS_{10}$ is encrypted with both $PK_{ME}$ and $E_{ABE,\gamma_G}$. Even adversary obtains $MS_{10}$, he/she should have $SK_{ME}$ and $SK_{ABE,G}$ to decrypt the message, which is as hard as the discrete logarithm problem. Therefore, it is not possible for the adversary to impersonate a legitimate user. Thus, by considering the security of messages $MS_6$, $MS_9$, $MS_{11}$, create operation is secure against impersonation attacks. □

**Theorem 4.3.** *Under the difficulty of discrete logarithm problem, join operation of group authentication provides resistance against impersonation attacks.*

*Proof.* To perform impersonation attack to join operation, adversary should impersonate the message $MS_{12}$ from group manager.

$$\mathbf{MS_{12}}: \quad E_{PK_{CA}}(ID_{MA} \,|\, ID_G \,|\, ID_{new_1} | .... | ID_{new_k} \,|\, S_{MA} \,|\, t)$$

In $MS_{12}$, adversary can impersonate itself as a group manager. As defined in $MS_{12}$, manager signs $(ID_{MA} \,|\, ID_G \,|\, ID_{new_1} | .... | ID_{new_k})$. Since Schnorr signature is secure against impersonation attacks as proposed in [47], adversary should have $SK_{MA}$ to impersonate $MS_{12}$. As

adversary does not have $SK_{MA}$, he/she cannot impersonate $MS_{12}$. Thus, join operation is secure against impersonation attacks. □

**Theorem 4.4.** *Under the difficulty of discrete logarithm problem, leave operation of group authentication provides resistance against impersonation attacks.*

*Proof.* To perform impersonation attack to leave operation, adversary should impersonate the message $MS_{14}$ from group manager.

$$\mathbf{MS_{14}}: \quad E_{PK_C}(ID_{old_1}|....|ID_{old_l}|E_{PK_C}(ID_G|PK_{MA}|S_{CA})|S_{MA}|t)$$

In $MS_{14}$, adversary can impersonate itself as a group manager. As defined in $MS_{14}$, manager signs $(ID_{old_1}|....|ID_{old_l}|E_{PK_C}(ID_G|PK_{MA}|S_{CA}))$. Since Schnorr signature is secure against impersonation attacks as proposed in [47], adversary should have $SK_{MA}$ to impersonate $MS_{12}$. In addition, $MS_{14}$ includes a certificate $E_{PK_C}(ID_G|PK_{MA}|S_{CA})$ which is signed by CA. As adversary does not have $SK_{CA}$, he/she cannot impersonate the certificate. Thus, leave operation is secure against impersonation attacks. □

Collision attacks are one of the most important security threats for ABE variants. In this attack model, a group of adversaries try to use the aggregated set of their attributes to obtain the plaintext from ciphertext that is encrypted by using CP-ABE. Let $a_1$ and $a_2$ be adversaries that try to collude their $SK_{ABE}$ to escalate their routing privileges. For instance, let $a_1$ has $\gamma_1 = \{m_1, m_4\}$ and $a_2$ has $\gamma_2 = \{m_2, m_3\}$. $a_1$ and $a_2$ can perform collision attack to attain $\gamma_c = \{m_1, m_2, m_3, m_4\}$. A nonce is encrypted with metric list in both protocols ($MS_4$, $MS_8$) by secret CP-ABE key. As defined in [8], secret keys are randomized and cannot be combined in CP-ABE. Therefore, AQoSAR is secure against collision attacks.

Replay attack is a type of active attacks that messages of entities are repeated maliciously. One of the most efficient countermeasures of replay attack is to attach a timestamp to all messages [50]. Thus, AQoSAR attaches a timestamp value in each messages between $MS_1$ and $MS_{14}$. Therefore, AQoSAR is secure against replay attack.

Eavesdropping is a passive attack that adversary captures messages between entities and extracts information from captured messages. One of the most efficient countermeasures of eavesdropping attack is to encrypt messages between entities [51]. Thus, messages of both authentication protocols ($MS_1 - MS_{14}$) are encrypted with the public key of the receiver entity. Since El-Gamal encryption scheme is indistinguishable under the chosen-plaintext attack (IND-CPA) [52], it is not possible to extract secret key of receiver entity from ciphertext. Therefore, AQoSAR is secure against eavesdropping attack.

Security of AQoSAR against impersonation attack, replay attack, collision attack and eavesdropping is examined in the chapter. AQoSAR uses Schnorr signature scheme to provide security against impersonation attacks. Moreover, as defined in [8], secret keys are randomized and cannot be combined in CP-ABE. Thus, AQoSAR is secure against collision attacks. As a countermeasure to replay attacks, AQoSAR attaches a timestamp to messages. Finally, AQoSAR uses El-Gamal encryption scheme to avoid eavesdropping.

# 5. PERFORMANCE OF AQoSAR

In this chapter, we present the performance analysis of AQoSAR. First, we give numerical evaluations for the performance of AQoSAR by using simulations. Then, we analyze the performance of AQoSAR by using asymptotic analysis. Finally, we investigate the performance of the routing application by using queueing theory. All simulations were carried out by using FloodLight [53] controller, Openv [54] switch and Mininet [55] environment on Intel i7-6700 HQ processor and 12 GB RAM memory. We run our simulations 100 times for each simulation.

We investigate the time required to execute authentication application with respect to changes in the number of attributes used for CP-ABE for a group of five users as shown in Fig. 5.1. Since CP-ABE execution time depends on the number of attributes used for generating secret key, the time required to establish authentication proportionally increases with the number of attributes.

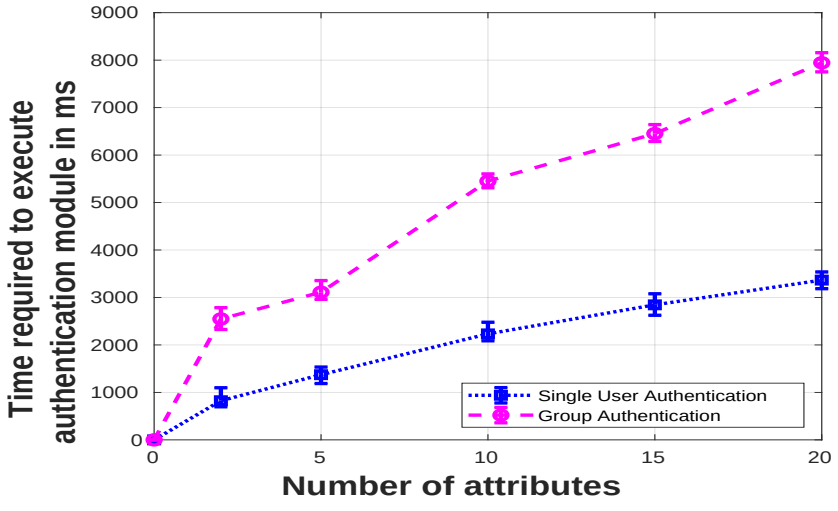



Figure 5.1. Time Required to Execute Authentication Application with Respect to Change in the Number of Attributes

We also investigate effects of changes in the number of users on the time required to execute authentication application as shown in Fig. 5.2. For the experiment, we select five as a number of attributes. Simulation results show that the total time required to execute the authentication application for a group of ten users and ten single users take the same amount

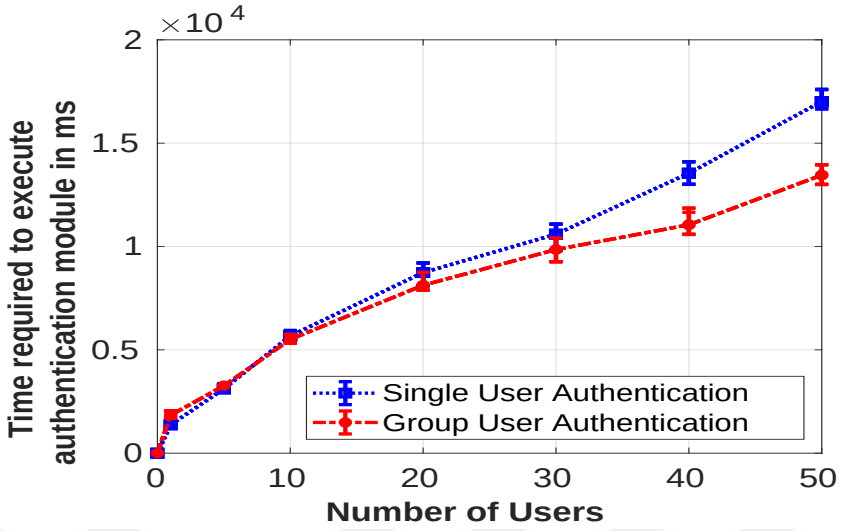



Figure 5.2. Time Required to Execute Authentication Application with Respect to Change in the Number of Users

of time. On the other hand, when the number of users is greater than ten, group authentication performs better than single user authentication as shown in Fig. 5.2. Moreover, numerical evaluations also show that the total execution time for the group authentication application for 50 users is approximately 13 seconds, which shows the applicability of AQoSAR for medium-sized groups.

In addition, we present analysis for the authentication application with respect to the computation cost and the communication cost. Let $m$ be the number of participants, $k$ be the number of joining participants, $l$ be the number of leaving participants and $T_{EXP}$ be the time required to compute modular exponentiation operations (since it is the most time consuming operations for executing CP-ABE, Schnorr Signature and El Gamal), the communication cost and computation cost analysis results are as shown in Table 5.1. Communication cost is used for representing the total number of messages exchanged to perform authentication. Single user authentication is performed with five messages per user as given in Table 5.1. Communication costs of create operation and join operation depend on the number of participants in the group. Since leave operation is performed with one message, it is independent of the number of leaving participants. For the computation cost, single user authentication and leave operation are realized in constant time. However, create operation and join operation depend on the number of participants.

Table 5.1. Computation and Performance Costs of Authentication Protocols

| **Operation** | **Communication Cost** | **Communication Complexity** | **Computation Cost** | **Computation Complexity** |
|---|---|---|---|---|
| Single User Authentication | 5 | $O(1)$ | $17 \cdot T_{EXP}$ | $O(1)$ |
| Create Operation | $2+4 \cdot m$ | $O(m)$ | $(8+13m) \cdot T_{EXP}$ | $O(m)$ |
| Join Operation | $1+4 \cdot k$ | $O(k)$ | $(4+13k) \cdot T_{EXP}$ | $O(k)$ |
| Leave Operation | 1 | $O(1)$ | $4 \cdot T_{EXP}$ | $O(1)$ |

For the performance analysis of routing application, we assume that the controller is a single server in the network, arrival rate of users determined by a Poisson process and job service times have an exponential distribution. Therefore, $M/M/1$ queue is used. Expected service time ($E[S]$) of routing application can be formalized as follows:

$$E[S] = \frac{z^2 \cdot \overline{b}}{b_{max}} + z^2 \cdot \overline{a} + z^2 \cdot \overline{r} \tag{5.1}$$

where z denotes number of switch in the network, $b_{max}$ denotes maximum bandwidth capacity in the network, $\overline{b}$ , $\overline{a}$ , $\overline{r}$ denotes average bandwidth capacity metric, average availability metric and average reliability metric of users in system, respectively. Then, service rate ($\mu$) is:

$$\mu = \frac{1}{E[S]} = \frac{b_{max}}{z^2 \cdot (\overline{b} + b_{max} \cdot (\overline{a} + \overline{r}))} \tag{5.2}$$

The average number of users in system, denoted as $L_s$, is calculated by using the following equation:

$$\begin{aligned} L_s &= \frac{\rho}{1-\rho} \text{ where } \rho = \frac{\mu}{\lambda}. \text{ Then,} \\ L_s &= \frac{\lambda}{\mu-\lambda}. \text{ For our system,} \\ L_s &= \frac{\lambda \cdot z^2 \cdot (\overline{b} + b_{max} \cdot (\overline{a}+\overline{r}))}{b_{max} - \lambda \cdot z^2 \cdot (\overline{b} + b_{max} \cdot (\overline{a}+\overline{r}))} \end{aligned} \tag{5.3}$$

where $\lambda$ denotes arrival rate of users and $\rho$ denotes utilization ratio of system. We calculate the average time that each user spends in the system by using the following equation:

$$W_s = \frac{L_s}{\lambda} = \frac{z^2 \cdot (\overline{b} + b_{max} \cdot (\overline{a}+\overline{r}))}{b_{max} - \lambda \cdot z^2 \cdot (\overline{b} + b_{max} \cdot (\overline{a}+\overline{r}))} \tag{5.4}$$

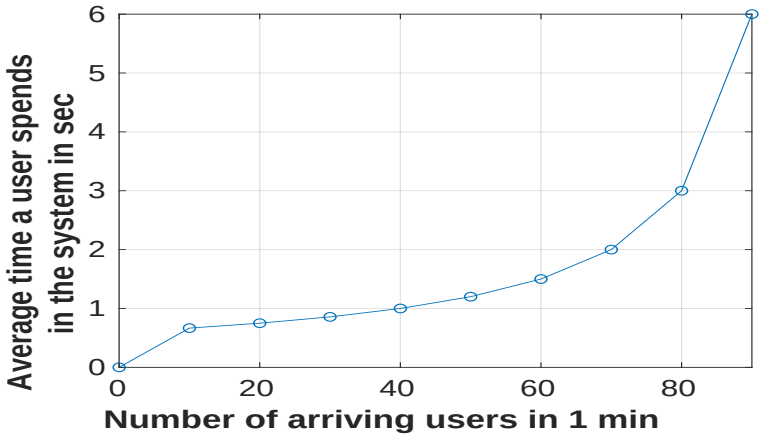



Figure 5.3. Average Time a User Spends in the System with respect to Arrival Rate of Users

We also present simulations for the average time a user spends in the system with respect to the arrival rate of users as shown in Fig. 5.3. For these simulations, we assume that the network consists of 20 switches and $\overline{b}/b_{max}$, $\overline{a}$, $\overline{r}$ values are equal to 0.5. As a result, the routing application operates efficiently for the arrival rate of users up to 80/min.

In this chapter, we have presented the performance analysis of AQoSAR. As a first step, we use simulations for investigating authentication module. Results of simulations

show that group authentication method performs better than single user authentication when the number of users is more than 10. Simulations also show that both single user authentication and group authentication execution times are highly correlated with the number of attributes. Moreover, we analyze the performance of AQoSAR by using asymptotic analysis. The analysis shows that while single user authentication and leave operation take constant time, execution time of join operation and leave operation proportion to the number of users. Finally, we use queuing theory to analyze performance of the routing application. Analysis show that execution time of routing application depends on QoS metrics of the user, the number of users in system and the number of switches in the network.

# 6. CONCLUSION

In this thesis, we have investigated QoS-aware routing problem in SDN. The flexible architecture of SDN is more suitable than traditional networks in terms of achieving QoS. However, malicious users who try to monopolize resource utilization are serious threats for QoS. Thus, a security mechanism which identifies malicious users is needed to achieve QoS. In order to address the problem, we use an authentication mechanism that jointly works with QoS-aware routing protocol.

Our contributions for the thesis are as follows:

- We have proposed an Authenticated Quality of Service Aware Routing (AQoSAR) to securely determine routing paths of a single user and a group of users in the network.
- To provide authentication for single users and group of users, we have employed CP-ABE as an authentication mechanism.
- We have proposed to use metric list rather than using a single metric for the QoS-aware routing to meet different expectations of users.
- Moreover, our security analysis shows that AQoSAR is secure against impersonation attack, collision attack, replay attack and eavesdropping.
- Furthermore, we have presented a performance analysis of AQoSAR that includes results of simulations, asymptotic analysis and results of queuing theory methods.

We developed AQoSAR in a simulation environment for the thesis. As future work, we want to implement AQoSAR in a real network system. Thus, we want to observe the performance of AQoSAR for real-time network traffic. In addition, we used five different QoS metrics for the routing application. We are planning to increase number of QoS metrics in the future. Moreover, adapting AQoSAR to P4 architecture [56] is left as future work.

# REFERENCES


1. Nunes, B. A. A., M. Mendonca, X.-N. Nguyen, K. Obraczka and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks", *IEEE Communications Surveys  Tutorials*, 2014.

2. Porxas, A., S. Liny, and M. Luoz, "QoS-Aware Virtualization-Enabled Routing in Software-Defined Networks", *Next Generation Networking Symposium*, IEEE, 2015.

3. Jiang, J., H. Huang, J. Liao and S. Chen, "Extending Dijkstra's Shortest Path Algorithm for Software Defined Networking", *Network Operations and Management Symposium*, IEEE, 2014.

4. Wang, M., J. Liu, J. Mao, H. Cheng and J. Chen, "NSV-GUARD: Constructing Secure Routing Paths in Software Defined Networking", *International Conferences on Big Data and Cloud Computing, Social Computing and Networking, Sustainable Computing and Communications*, IEEE, 2016.

5. Egilmez, H. and A. Tekinalp, "Distributed QoS Architectures for Multimedia Streaming Over Software Defined Networks", *Transactions on Multimedia*, IEEE, 2014.

6. Sahai, A. and B. Waters, "Fuzzy Identity-Based Encryption", *Advances in Cryptology – EUROCRYPT 2005*, pp. 457–473, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

7. Goyal, V., O. Pandev, A. Sahai and B. Waters, "Attribute-based Encryption for Fine-grained Access Control of Encrypted Data", *Conference on Computer and Communications Security*, ACM, 2006.

8. Bethencourt, J., A. Sahai and B. Waters, "Ciphertext-Policy Attribute-Based Encryption", *Symposium on Security and Privacy*, IEEE, 2007.

9. Kreutz, D., F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky and

S. Uhlig, “Software-Defined Networking: A Comprehensive Survey”, *Proceedings of the IEEE*, Vol. 103, 2015.

10. Benzekki, K., A. El Fergougui and A. El Belrhiti El Alaoui, “Software-defined networking (SDN): A survey”, *Security and Communication Networks*, 02 2017.

11. Dutra, D., M. Bagaa, T. Taleb and K. Samdanis, “Ensuring end-to-end QoS based on multi-paths routing using SDN technology”, *Global Communications Conference*, IEEE, 2017.

12. Layeghy, S., F. Pakzad and M. Portmann, “SCOR: Software-defined Constrained Optimal Routing Platform for SDN”, *CoRR*, Vol. abs/1607.03243, 2016, `http://arxiv.org/abs/1607.03243`.

13. Egilmez, H., T. Dane, T. Bagci and M. Tekinalp, “OpenQoS: An OpenFlow Controller Design for Multimedia Delivery with End-to-End Quality of Service over Software-Defined Networks”, *Signal & Information Processing Association Annual Summit and Conference*, IEEE, 2012.

14. Delaet, S., S. Dolev, D. Khankin, S. Tzur-David and T. Godinger, “Seamless SDN Route Updates”, *2015 IEEE 14th International Symposium on Network Computing and Applications*, pp. 120–125, Sept 2015.

15. Frenkel, S., D. Khankin and A. Kutsyy, “Predicting and choosing alternatives of route updates per QoS VNF in SDN”, *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, pp. 1–6, Oct 2017.

16. Won, K., S. Park and J. You, “Mynah: Enabling Lightweight Data Plane Authentication for SDN Controllers”, *Computer Communication and Networks*, IEEE, 2015.

17. Kuliesius, F. and V. Dangovas, “SDN-Driven Authentication and Access Control System”, *The International Conference on Digital Information, Networking, and Wireless Communications*, SDIWC, 2014.

18. Kuliesius, F. and V. Dangovas, “SDN Enhanced Campus Network Authentication and Access Control System”, *International Conference on Ubiquitous and Future Networks*, IEEE, 2016.

19. Sahri, N. and J. Mao, “Collaborative Spoofing Detection and Mitigation - SDN based looping authentication for DNS services”, *Computer Software and Applications Conference*, IEEE, 2016.

20. Sahri, N. and K. Okamura, “Protecting DNS Services from IP Spoofing: SDN Collaborative Authentication Approach”, *Proceedings of the 11th International Conference on Future Internet Technologies*, CFI ’16, pp. 83–89, ACM, New York, NY, USA, 2016, http://doi.acm.org/10.1145/2935663.2935666.

21. Y.Li and J.Mao, “SDN based Access Authentication and Automatic Configuration for IPSec”, *International Conference on Computer Science and Network Technology*, IEEE, 2015.

22. Cao, Z., J. Fitschen and P. Papadimitriou, “FreeSurf: Application-Centric Wireless Access with SDN”, *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM ’15, pp. 357–358, ACM, New York, NY, USA, 2015, http://doi.acm.org/10.1145/2785956.2790000.

23. Cho, J. Y. and T. Szyrkowiec, “Practical Authentication and Access Control for Software-Defined Networking over Optical Networks”, *Proceedings of the 2018 Workshop on Security in Softwarized Networks: Prospects and Challenges*, SecSoN ’18, pp. 8–13, ACM, New York, NY, USA, 2018, http://doi.acm.org/10.1145/3229616.3229619.

24. Zhou, R., Z. Liu, Y. Lai and J. Liu, “Study on authentication protocol of SDN trusted domain”, *International Symposium on Autonomous Decentralized Systems*, IEEE, 2015.

25. Ding, K., X. Wang, G. Zhang, Z. Wang and M. Chen, “A flow-based authentication handover mechanism for multi-domain SDN mobility environment”, *China Communi-*

*cations*, IEEE, 2017.

26. Ion, M., J. Zhang and E. M. Schooler, "Toward Content-Centric Privacy in ICN: Attribute-based Encryption and Routing", .

27. Li, B., A. P. Verleker, D. Huang, Z. Wang and Y. Zhu, "Attribute-Based Access Control for ICN Naming Scheme", *2014 IEEE Conference on Communications and Network Security (CNS)*, pp. 391–399, IEEE, 2014.

28. Borgh, J., E. Ngai, B. Ohlman and A. M. Malik, "Employing attribute-based encryption in systems with resource constrained devices in an information-centric networking context", *Global Internet of Things Summit (GIoTS), 2017*, pp. 1–6, IEEE, 2017.

29. Malik, A. M., J. Borgh and B. Ohlman, "Attribute-Based Encryption on a Resource Constrained Sensor in an Information-Centric Network", *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, pp. 217–218, ACM, 2016.

30. Sertbaş, N., S. Aytaç, O. Ermiş, F. Alagöz and G. Gür, "Attribute Based Content Security and Caching in Information Centric IoT", *Proceedings of the 13th International Conference on Availability, Reliability and Security*, ARES 2018, pp. 34:1–34:8, ACM, New York, NY, USA, 2018, `http://doi.acm.org/10.1145/3230833.3233273`.

31. Nzanywayingoma, F. and H. Qiming, "Securable Personal Health Records using Ciphertext Policy Attribute Based Encryption", *2012 IEEE 14th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2012.

32. Chen, N., M. Gerla, D. Huang and X. Hong, "Secure, Selective Group Broadcast in Vehicular Networks using Dynamic Attribute Based Encryption", *Ad Hoc Networking Workshop (Med-Hoc-Net), 2010 The 9th IFIP Annual Mediterranean*, 2010.

33. Touati, L. and Y. Challal, "Efficient CP-ABE Attribute/Key Management for IoT Applications", *IEEE International Conference on Computer and Information Technology*, 2015.

34. Khader, D., “Attribute-Based Authentication Scheme”, *Ph.D. dissertation*, University of Bath, 2009.

35. Yang, H. and V. Oleshchuk, “Traceable Hierarchical Attribute-based Authentication for the Cloud”, *Workshop on Security and Privacy in the Cloud*, IEEE, 2015.

36. Guo, L., C. Zhang, J. Sun and Y. Fang, “A Privacy-Preserving Attribute-Based Authentication System for Mobile Health Networks”, *Transactions on Mobile Computing*, IEEE, 2014.

37. Guo, L., C. Zhang, J. Sun and Y. Fang, “PAAS: A Privacy-Preserving Attribute-based Authentication System for eHealth Networks”, *International Conference on Distributed Computing Systems*, IEEE, 2012.

38. Aghapour, S., M. Ameri and J. Mohajeri, “A Multi Sender Attribute-Based Broadcast Authentication Scheme”, *International Symposium on Telecommunications*, IEEE, 2016.

39. Bin, W. and R. Yan, “An Attribute-Based Anonymous Authentication Scheme”, *International Conference on Emerging Intelligent Data and Web Technologies*, IEEE, 2013.

40. Hong, H., Z. Sun and Y. Xia, “Achieving secure and fine-grained data authentication in cloud computing using attribute based proxy signature”, *International Conference on Information Science and Control Engineering*, IEEE, 2017.

41. Neto, A. L. M., Y. L. Pereira, A. L. F. Souza, I. Cunha and L. B. Oliveira, “Attributed-based Authentication and Access Control for IoT Home Devices: Demo Abstract”, *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN ’18, pp. 112–113, IEEE Press, Piscataway, NJ, USA, 2018, https://doi.org/10.1109/IPSN.2018.00019.

42. Neto, A. L. M., A. L. F. Souza, I. Cunha, M. Nogueira, I. O. Nunes, L. Cotta, N. Gentille, A. A. F. Loureiro, D. F. Aranha, H. K. Patil and L. B. Oliveira, “AoT: Authentication

and Access Control for the Entire IoT Device Life-Cycle", *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems*, SenSys '16, pp. 1–15, ACM, New York, NY, USA, 2016, http://doi.acm.org/10.1145/2994551.2994555.

43. Alpár, G., L. Batina, L. Batten, V. Moonsamy, A. Krasnova, A. Guellier and I. Natgunanathan, "New Directions in IoT Privacy Using Attribute-based Authentication", *Proceedings of the ACM International Conference on Computing Frontiers*, CF '16, pp. 461–466, ACM, New York, NY, USA, 2016, http://doi.acm.org/10.1145/2903150.2911710.

44. Portnoi, M. and C. Shen, "Loc-Auth: Location-enabled authentication through attribute-based encryption", *International Conference on Computing, Networking and Communications, Communications and Information Security*, IEEE, 2015.

45. Park, H., D. Lee and J.Zhan, "Attribute-Based Access Control using Combined Authentication Technologies", *International Conference on Granular Computing*, IEEE, 2018.

46. El Gamal, T., "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", *Proceedings of CRYPTO 84 on Advances in Cryptology*, Springer, 1985.

47. Schnorr, C. P., "Efficient Identification and Signatures for Smart Cards", G. Brassard (Editor), *Advances in Cryptology — CRYPTO' 89 Proceedings*, Springer, 1990.

48. Jesudoss, A. and N. Subramaniam, "A Survey on Authentication Attacks and Countermeasures in a Distributed Environment", *International Journal of Computer Sciences and Engineering*, 2014.

49. Yu, S., K. Ren, W. Lou and J. Li, "Defending against Key Abuse Attacks in KP-ABE Enabled Broadcast Systems", Y. Chen, T. D. Dimitriou and J. Zhou (Editors), *Security and Privacy in Communication Networks*, pp. 311–329, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

50. Syverson, P., "A taxonomy of replay attacks [cryptographic protocols]", *Proceedings*

*The Computer Security Foundations Workshop VII*, pp. 187–191, June 1994.

51. Karlof, C. and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures", *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003.*, pp. 113–127, May 2003.

52. Tsiounis, Y. and M. Yung, "On the Security of ElGamal Based Encryption", *Proceedings of the First International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography*, PKC '98, Springer, 1998.

53. Big Switch Networks, *Project FloodLight*, `http://www.projectfloodlight.org`, accessed at March 2019.

54. Linux Foundation, *Open vSwitch*, `https://www.openvswitch.org/`, accessed at March 2019.

55. Bob Lantz, N. H. V. J., Brandon Heller, *Mininet*, `http://mininet.org/`, accessed at March 2019.

56. Bosshart, P., D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese and D. Walker, "P4: Programming Protocol-independent Packet Processors", *SIGCOMM Comput. Commun. Rev.*, Vol. 44, No. 3, pp. 87–95, Jul. 2014, `http://doi.acm.org/10.1145/2656877.2656890`.