ISSN 2616-5775 Vol. 4, Issue 5: 32-36, DOI: 10.25236/AJCIS.2021.040505

Data Secure Communication System Based on OpenABE

Yajun Zou

School of Cybersecurity, Qufu Normal University, Qufu, Jining, 272000, China

Abstract: *In the information era, data security is important. At first, it is an identity-based encryption mechanism and then it extends various branches and deformations. A policy combined with attribute based access control is known as Attribute Based Access Control came into existence. The two classes of Attribute based Encryption Scheme, such as CP-ABE (Ciphertext-Policy Attribute-based Encryption) and KP-ABE (Key Policy Attribute-based Encryption) concepts are utilized to data security communication system. In this paper, we build a data security communication system based on OpenABE library. We use the basic system call of network communication send() and recv() to send or receive message. Then, the OpenABE encrypts or decrypts for the message. In ABE system, the public key and private key are not one-to-one, and one public key can correspond to multiple private keys. No matter how many users share the data, it is only necessary to encrypt it once. When encrypting the information, the encryption party does not need to know who is decrypted, and the decryption party can decrypt it which only needs to meet the corresponding conditions, so as to achieve data access control while encrypting. Besides, this paper also briefly introduced the ABE and the OpenABE.*

Keywords: *Attribute based encryption, ciphertext policy, OpenABE, attributes.*

1. Introduction

With large-scale network data in the information world, most users choose to upload their data to the cloud server, which needs to take into account the security of information storage. Network security has been a key issue to our country. Cryptography is one of the most important tools to protect data. In 1977, the Data Encryption Standard (DES) was put forth by the US NSA. In the early days, DES was widely used in the world because of high linear complexity, easy realization, standardization and generalization. But DES has weakness revealed. Its key was short so that DES cannot resist exhaustive attacks with the help of Internet, so it cannot guarantee data security. In order to alleviate the disadvantages, a new encryption standard AES, namely Rijndael algorithm, introduced by Joan Daemen and Vincent Rijmen of Belgium, has higher security and it is impossible to carry out exhaustive attacks on it under the current situation. Secondly, AES has high computational efficiency and can be widely used in various high-speed applications. However, AES is a symmetric cryptosystem. The encrypted key is the core to ensure data security communication. Once the key is leaked, security cannot be guaranteed. Therefore, the public-key cryptography appeared. ABE is a powerful public key encryption system which allows an encryptor to share encrypted data with others according to access policies. This feature also overcomes many obstacles in encrypting user data using traditional public key encryption methods.

2. Attribute-based encryption

In an ABE system, there are two parameters we have to know. An attribute can describe the property of object and data type can be either string or number, for example “MALE”, “eighteen years old”. Attribute lists are made up of many attributes, which are essentially an array. An access control policy can be any Boolean formula over these attributes, which are comprised of OR and AND gates such as “MALE”AND “18 years old”.

Then, we briefly discuss the implementation principle of ABE. Attributes and any access control policy are embeded into private key and ciphertext by designers. This process of trying to input the private key and the ciphertext into the decryption algorithm is actually the process of matching the attribute list with control policy. If match successfully, decryption succeeds else decryption fails. Most existing public key encryption methods allow a party to encrypt data to a particular user, but are unable to efficiently handle more expressive types of encrypted access control. In ABE system, the data owner

ISSN 2616-5775 Vol. 4, Issue 5: 32-36, DOI: 10.25236/AJCIS.2021.040505

only needs to formulate an access control policy that only these N users can satisfy, and then input the public parameter, the policy and the plaintext to the ABE encryption algorithm for encryption. After obtaining the ciphertext, it is sent to these N different users.

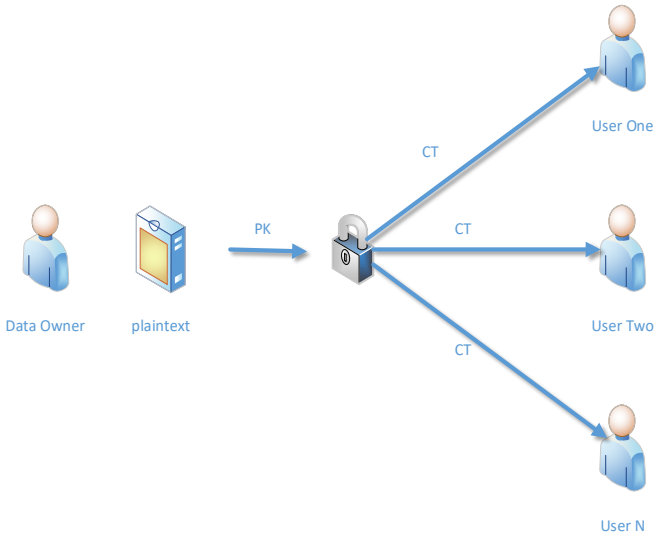

Figure 1: Example of OpenABE implementation.

Attribute based Encryption is classified into two classes. 1. Role-Based Access Controls: Ciphertext-Policy ABE. In a CP-ABE system, the attributes are associated with a user's secret key and policies are associated with ciphertexts. If the attribute list on the key satisfies the policy on the ciphertext, the user can decrypt the ciphertext. 2. Content-Based Access Controls: Key-Policy ABE. In a KP-ABE system, the attributes are associated with ciphertexts and policies are associated with a user's secret key. If the attribute list on the ciphertext satisfies the policy on the key, the user can decrypt the ciphertext.

3. Openabe

OpenABE is a C++ library that implements several attribute-based encryption schemes. It provides a crypto-box like interface for encryption and decryption.

3.1 The features of OpenABE:

- Modular: the OpenABE provides a generalized application programming interface for developers which can allow users to swap one cryptographic scheme for another without updating application logic.
- Comprehensive: it provides a neutral mathematics API to support base elliptic curve and bilinear operations. The API can be connected to and implemented via one or more specific math libraries. The generic mathematics API makes it easy to update or replace subcomponents. Besides, it enables deploying multiple versions of OpenABE that make use of specific advantage in different external mathematics libraries.
- Extensible: the OpenABE can support several additional functional encryption scheme types with relatively little effort.

3.2 CP-ABE implementation

The construction consists of four algorithms: Setup, Keygen, Encrypt, and Decrypt.

- $Setup\ (\tau,\ n) \rightarrow PK,\ MSK.$ The setup algorithm takes a security parameter τ and n in a collision-resistant hash function $H_1 : G_T \rightarrow \{0,1\}^n$ as input and outputs public parameters PK and master secret key MSK. It chooses random exponents α, $\alpha \in \mathbb{Z}_p$, and outputs PK and MSK as formulas (1) and (2), respectively:

ISSN 2616-5775 Vol. 4, Issue 5: 32-36, DOI: 10.25236/AJCIS.2021.040505

$$PK = \{g_1, g_2, g_1^a, e(g_1, g_2)^\alpha\} \tag{1}$$

$$MSK = \{\alpha, g_2^a\} \tag{2}$$

- $Keygen\ (PK,\ MSK, \gamma) \rightarrow SK$. An authority can run keygen to generate a private key for a particular user that grants them a set of attributes. The key generation algorithm takes as input the master secret key and a set of attributes γ. Chooses random number $t \in \mathbb{Z}_p$. It creates the private key SK as follows:

$$K = g_2^\alpha \bullet g_2^{\alpha \bullet t} \quad L = g_2^t \quad \forall x \in SK_x = H_2(x)^t \tag{3}$$

- $Encrypt_{KEM}(PK, T; u) \rightarrow (Key, CT)$. The encryption algorithm takes as input an access structure and outputs a symmetric key and ciphertext. The algorithm returns the following:

$$Key = H_1(e(g_1, g_2)^{\alpha s}),$$

$$\begin{aligned} CT = \{C' = g_1^s, (C_1 = g_1^{\alpha\lambda_1} H_2(\rho(1))^{-r_1}, D_1 = g_2^{r_1}) \\ ,..., (C_l = g_1^{\alpha\lambda_l} H_2(\rho(l))^{-r_n}, D_l = g_2^{r_l})\} \end{aligned} \tag{4}$$

- $Decrypt_{KEM}(CT, SK) = Key$. The decryption algorithm is used by an authorized user to decrypt a ciphertext. It takes as inputs a ciphertext CT for access structure T and a private key for a set of attributes γ. For each such attribute $i \in S$ and the corresponding coefficient w_i, the decryption algorithm first computes:

$$\begin{aligned} &\frac{e(C', K)}{\prod_{i \in S}(e(C_i, L) \bullet e(K_{\rho(i)}, D_i))^{w_i}} = \\ &\frac{e(g_1, g_2)^{\alpha s} \bullet e(g_1, g_2)^{\alpha s t}}{\prod_{i \in S}(e(g_1, g_2)^{t\alpha\lambda_i w_i})} = e(g_1, g_2)^{\alpha s} \end{aligned} \tag{5}$$

The algorithm can then compute

$$Key = H_1(e(g_1, g_2)^{\alpha s}). \tag{6}$$

4. Data Security Communication System

CryptHook is a modular implementation that uses symmetric block cipher encryption to protect existing applications. It works by connecting the basic system call of network communication send()/sendto() and recv() / recvfrom(). As shown in the Fig2:

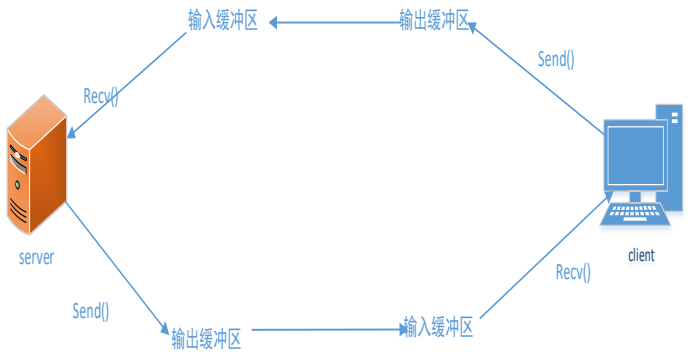

Figure 2: Example of send and recv

ISSN 2616-5775 Vol. 4, Issue 5: 32-36, DOI: 10.25236/AJCIS.2021.040505

4.1 Encryption process

First, we need to initialize the OpenABE before encrypting plaintext. In order to integrate the OpenABE into our C++ application, we will need to include a single header file and use the oabe namespace. To initialize the OpenABE library, call the OpenABE init function in the beginning of our application. Then we have to construct an ABE scheme context such as CP-ABE and load an existing set of ABE parameters back into a context. According to the CP-ABE context constructed above, we use CP-ABE encryption function to encrypt a message. If encryption is successful, then the ciphertext is stored in ct. Copy ciphertext to output buffer. Finally, add header information and pack full packet length. This is the encryption process.

4.2 Decryption process

The following is a brief introduction to the decryption process. CryptHook calls the send function through the system to send the ciphertext to the receiver. The receiver receives the ciphertext through the recv function and copies it to the input buffer. Then, call the decrypt_data function to decrypt message. Similarly, we initialize OpenABE, build the CP-ABE context. And load an existing set of ABE parameters back into the context. Generating or loading ABE master public parameters is required to perform encrypt and decrypt operations. Next, generating a new key is used to decrypt the ciphertext finally, simply call the shutdown method prior to exiting our application.

4.3 Implementation process

Netcat is a read and write data command through TCP/UDP, which is called 'Swiss Army Knife'. Using this tool, the data can be completely sent to another host terminal for display or storage.

Suppose there are two hosts A and B, A wants to communicate with B simply, then we can specify one host as a server, bind one of its own ports as a communication port (for example, host A as a server, 5000 as its own communication port) and then host B as a client to connect 5000 ports of host A (the premise is to know the IP address of host A).

A:

LD_PRELOAD=./crypthook.so ncat -l -p 5000

B:

LD_PRELOAD=./crypthook.so ncat 127.0.0.1 5000

Then, we can input alice for encryption in host A. In host B, we can decrypt the ciphertext to get plaintext.

The code operation results are shown in the Fig3 and Fig4:

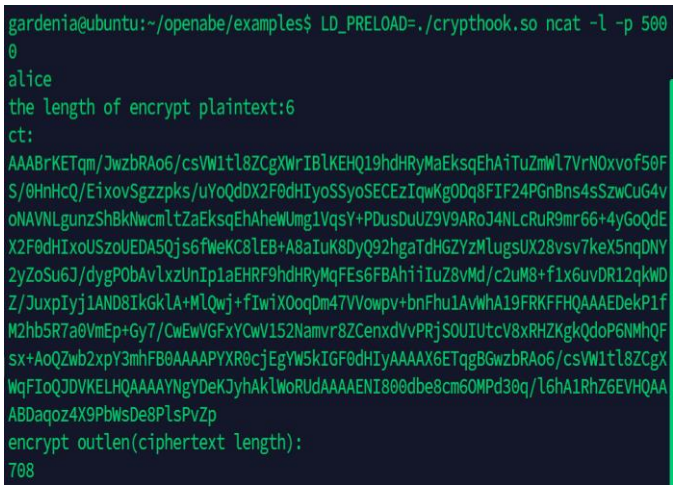

Figure 3: Command of Host A

ISSN 2616-5775 Vol. 4, Issue 5: 32-36, DOI: 10.25236/AJCIS.2021.040505

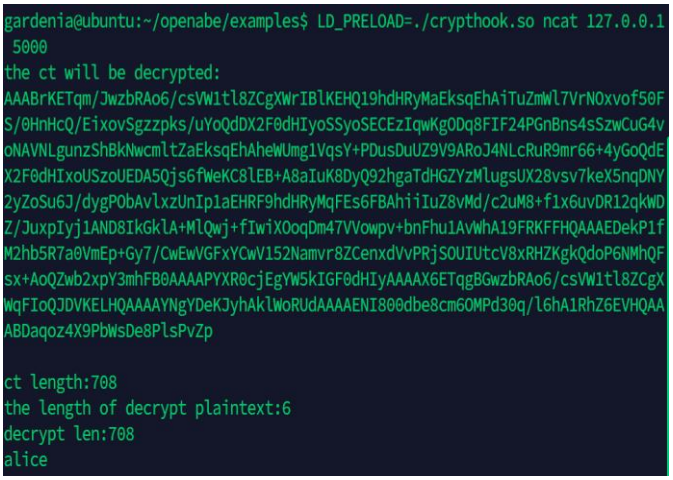

Figure 1: Command of Host B

5. Conclusion

Cryptography become vital aspect for transmitting data through network. Attribute-based encryption algorithm has been widely concerned not only because of the complexity of the theoretical design of attribute-based encryption algorithm, but also the great practical value of attribute-based cryptography and related research. Attribute-based encryption algorithm extends the control and authentication of the identity to the authentication of the attribute set and it also provides a wealth of control means. Through the threshold, the OR gate and AND gate, the access control structure can adapt to many situations. In this paper, we designed a data communication system based on OpenABE which is actually an ABE library. Our system mainly used CP-ABE for encryption and decryption. User's private keys are specified by a set of attributes and ciphertext are specified by access control policies. It contains four algorithms which are Setup, Keygen, Encryption and Decryption. In the end, we provided pictures shown the effect of our system.

References

[1] Waters B. Ciphertext-policy attribute-based encryption. 2011.
[2] Xingting Dong, Yanhua Zhang, Baocang Wang, Jiangshan Chen, "Server-Aided Revocable Attribute-Based Encryption from Lattices", Security and Communication Networks, vol. 2020, Article ID 1460531, 13 pages, 2020.
[3] Kumar, Praveen, P, et al. Attribute based encryption in cloud computing: A survey, gap analysis, and future directions [J]. Journal of Network & Computer Applications, 2018.