A Study on PGP (Pretty Good Privacy) Using Blockchain

Chang-Hyun Roh and Im-Yeong Lee(✉)

Department of Computer Software Engineering, Soonchunhyang University,
22, Soonchunhyang-ro, Sinchang-myeon, Asan-si
Chungcheongnam-do, Republic of Korea
{rohch, Imylee}@sch.ac.kr

Abstract. E-mail has developed as the Internet has evolved. Encryptions such as Secure/Multipurpose Internet Mail Extensions or Pretty Good Privacy (PGP) are used to ensure privacy and confidentiality. However, the public PGP key is stored on a server and is synchronized with the PGPs of other servers, and thus is not in real time. In addition, public key search, registration, and cancellation are difficult when a key server is down. To solve this problem, we develop a key management server integrating PGP into a blockchain; this allows all participating servers to synchronize registration and deletion in real time.

Keywords: Blockchain · Distribute ledger technology · Pretty Good Privacy

1 Introduction

E-mail has evolved along with the Internet. The use of a Web browser to access e-mail has become routine, and technical perfection has been attained. Recent developments in Social Network Services (SNSs) and instant messaging have greatly reduced personal usage, but many businesses continue to rely greatly on e-mail [1]. A great deal of effort has been devoted to e-mail confidentiality. E-mail security requires protected communication and encryption of the e-mail content [2]. Network security systems include the Secure HTTP and the Secure Socket Layer (SSL) methods. Secure/Multipurpose Internet Mail Extensions (S/MIMEs) and Pretty Good Privacy (PGP) are commonly used to encrypt e-mails. PGP affords high-level security [3] using public key infrastructure (PKI)-based algorithms to ensure e-mail confidentiality, integrity, and authentication by reference to the public key of the recipient. However, a key server manages public keys. A public key management system synchronizes the operations of servers in many organizations; synchronization is not performed in real time. If a key management server is down, key discovery, registration, and cancellation become very difficult. Thus, single-point-failure is in play. The synchronization and single-point error problems associated with PGP can be solved using blockchain techniques. Blockchain was first featured in the peer-to-peer (P2P) cryptography of Distributed Ledger Technology; integrity is assured by connecting all blocks containing the same data to their prior and subsequent blocks [4].

Here, we describe a key management server that can be implemented within a blockchain network to solve the abovementioned PGP problems.

J. J. Park et al. (Eds.): CUTE 2018/CSA 2018, LNEE 536, pp. 316–320, 2020.
https://doi.org/10.1007/978-981-13-9341-9_54

2 Related Research

2.1 PGP

PGP was created by Phil Zimmermann in 1991. PGP encrypts e-mails and associated files, ensuring both confidentiality and authentication. It has been repeatedly refined, and in 2001 was designated an encryption standard by the Internet Engineering Task Force (IETF) [5].

PGP features an encryption algorithm (Fig. 1) employing both symmetric and public key encryption. When encrypting a message, a session key is randomly generated using a symmetric key cryptosystem [6]. The public key cryptosystem encrypts the session key together with the recipient's public key, then encrypts the message, and sends the session key to the recipient. The receiver decrypts the session key using a private key, and then decrypts the message employing the newly accessible session key. Table 1 shows the relevant encryption algorithms. The session key is generated by the user, but the public key must be obtained from the recipient, either directly or via registration with a public key server. Most users access online public key servers to retrieve recipient identities and obtain the public keys required for PGP transfer.

2.2 Blockchain

Blockchain is a cryptocurrency created by Nakamoto Satoshi in 2009. Bitcoin users transfer currency and engage in transactions that are not mediated by institutions such as central banks. In a P2P network, all participants record identical transactional data; they cooperate to maintain the network. In a blockchain, all participants access identical data in block form; the blocks are associated with hash values that prevent tampering. Blockchains may be public, private, or consortium-based. Anyone can create a block by joining a public blockchain. Any person or consortium may seek to create a network, but only authorized participants or organizations can create/validate the block. The blockchain structure is shown in Fig. 1.

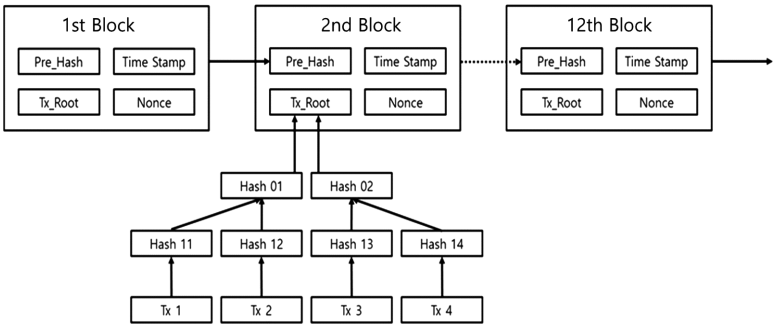

Fig. 1. Blockchain structure

3 Proposed Scheme

The proposed method can perform key registration, key stop, key signing, and key retrieval functions in a blockchain network. The overall scenario of the proposed Scheme is shown in Fig. 2.

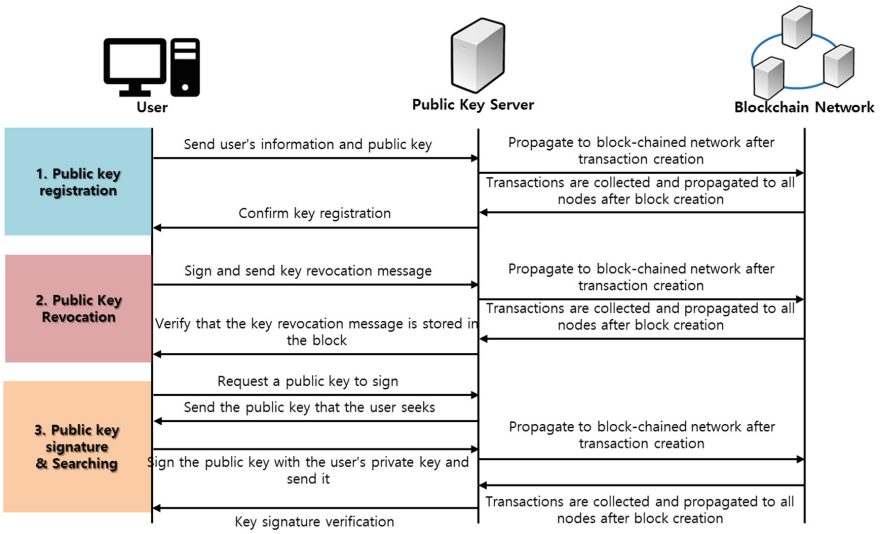

Fig. 2. Overall scenarios of the proposed scheme

3.1 System Parameters

Table 1. System parameters

Parameters	Define	Parameters	Define
PK_U	User U's public key	SK_U	User U's private key
$Info_U$	User U's information	$Block_1$	i - th block
Sig_{SK_U}	Signed with private key	TS	Transaction
$R \cdot Cert$	Public key revocation certificate		

*: participate object (U: User, PKS: Public Key Server)

3.2 Public Key Registration

The user generates a public/private key pair, sends the user information and public key to the public key server, and the public key server creates and propagates the transaction. The blockchain network completes the registration of the public key by

generating a block through the consensus algorithm and connecting it to the blockchain through validation and block propagation.

1. U generates $PK_U \cdot SK_U$
2. U sends PK_U and $Info_U$ to PKS
3. PKS generates PK_U and $Info_U$ as TS
4. PKS deploys TS as a blockchain network
5. Blockchain network collects TS every minute
6. Collect the collected TS and create a $Block_1$
7. $Block_1$ identification and propagation to blockchain networks
8. PKS sends the header of the generated block to U

3.3 Public Key Revocation

Public keying sends a revocation certificate to the public key server to prevent the user from using the stored public key. public key server creates a transaction with the revoked certificate and forwards it to the blockchain network. blockchain network then creates a block and links it to the block to disable the key.

1. $R \cdot Cert_U$ is created and signed with private key: $Sig_{SK_U}(R \cdot Cert_U)$
2. $R \cdot Cert_U$, $Sig_{SK_U}(R \cdot Cert_U)$ are send to PKS
3. PKS validates $Sig_{SK_U}(R \cdot Cert_U)$ as PK_U
4. Generate TS by inserting $R \cdot Cert_U$ after validation
5. Blockchain network collects TS every minute
6. Collect the collected TS and create a $Block_1$
7. $Block_1$ identification and propagation to blockchain networks
8. PKS sends the header of the generated block to the U
9. Confirm that the public key has been revoked

3.4 Public Key Signature

Public key signatures occur when you trust another user's public key in the public key ring. Users sign their public key for other trusted users to indicate their trust.

1. U determines the PK_1 to sign and generates $Sig_{SK_U}(PK_i)$
2. PK_1, $Sig_{SK_U}(PK_i)$ are send to PKS
3. PKS validates $Sig_{SK_U}(PK_i)$ as PK_1
4. PKS checks for duplicate $Sig_{SK_U}(PK_i)$ in the *Block*.
5. If duplication does not occur, insert PK_1 and $Sig_{SK_U}(PK_i)$ into TS, generate and propagate to blockchain network
6. Blockchain network collects TS every minute
7. Collect the collected TS and create a $Block_1$
8. $Block_1$ identification and propagation to blockchain networks

3.5 Public Key Search

Public key search is used when a user adds another user to the public key ring. The user uses the data of another user (user ID, key ID, e-mail address, etc.) to request a search to the public key server.

1. U learns $Info_1$ of another user
2. U to search the PKS for another user's information $Info_1$
3. PKS retrieves $Info_1$ from $Block$
4. Reply PK_1 and $Sig_{SK_U}(PK_i)$ to U if successful

4 Conclusion and Future Research

In this paper, we propose a public key server composed of blockchain network in PGP public key server. The public key server operated by the agency had a single point of failure that could not provide service when the server was stopped due to an attack. In addition, public key servers operated by each organization do not provide real-time synchronization, so you cannot store keys securely.

The disadvantage of existing public key servers is solved by configuring a blockchain network to solve a single point of failure. We also proposed a distributed system that provides data integrity because all servers are synchronized in real time. In the future, the proposed technique should be studied in such a way that users' public keychain management can be automatically added to smart contracts.

References

1. Radicati Group: Eamil Statistics Report, 2017–2021. Radicati Group, 6 February 2017
2. Park, D.-U., Park, J.-H., Kim, J.-S., Kim, I.-M.: A web based secure e-mail system using the PGP algorithm. KIPS Trans. Part C **8C**(1), 16–22 (2001)
3. Rhee, M.-Y., Kim, J.-H., Ryou, J.-C., Song, Y.-J., Youm, H.-Y., Lee, I.-Y.: E-Commerce Security Technology. Saengneung Publisher (1999)
4. Nakamoto. S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008). http://bitcoin.org
5. IETF: MIME Security with OpenPGP. RFC3156, August 2001
6. Stallings W.: Cryptography and Network Security: Principles and Practice, Global Edition, Pearson Education Limited (2016)