SCIS 2019 2019 Symposium on Cryptography and Information Security
Shiga, Japan, Jan. 22 - 25, 2019
The Institute of Electronics, Information and Communication Engineers

Adding Authenticity into Tree-based Group Key Agreement by Public Ledger

Seongho Han* Rakyong Choi† Kwangjo Kim*†

Abstract: With the development of communication technologies and embedded devices, the importance of group communication among various parties is growing. Group key agreement (GKA) is playing an important role in sharing critical information in a group. GKA protocol is divided into four categories: tree-based, star-based, link-based, and ring-based. Tree-based GKA has greater flexibility in membership management and is easier to achieve decentralization than other GKA protocols. Tree-based GKA protocols, which are based on Diffie-Hellman Key Exchange (DHKE), become vulnerable to active attacks since DHKE does not check the identity of the user. To solve the problem, authenticated key exchange (AKE) protocols were suggested. Most AKE papers solve the authentication problem based on an existence of trusted third party (TTP). However, relying on TTP has a risk of a single point-of-failure which paralyzes the protocols. In this paper, we suggest two protocols to add authenticity into a typical tree-based GKA protocol TGDH by public ledger: PLAKE and dPLAKE. PLAKE uses Proof of Work (PoW)-based public ledger and dPLAKE uses Delegated Proof of Stake (DPoS)-based public ledger not to depend on TTP for authentication. We discuss the security of PLAKE and dPLAKE.

Keywords: Tree-based Group Key Agreement, Public Ledger, Proof of Work, Delegated Proof of Stake

1 Introduction

1.1 Motivation

Key exchange protocol over an insecure network becomes necessary to establish a secure channel that prevents leak of information after Diffie and Hellman [1] proposed the breakthrough on key exchange protocol using public key cryptosystem. Key exchange protocol between two parties shares common secret key to secure communication. If we need a communication channel between multiple parties, we need to share the common key among these multiple parties.

The importance of sharing common secret key among multiple parties is growing with the development of communication technologies. A number of multi-party key exchange protocols [2, 3, 4, 5] have been proposed to establish a common secret key among multiple parties. Group key agreement (GKA) protocol is a kind of multi-party key exchange protocols in which a shared secret is derived from group members. Each group member contributes to deriving a shared secret. Every group member has to interact in order to compute the group key and no entity can predetermine the resulting value. GKA protocol does not require the existence of secure channels between its participants since no secure transfer takes place during processing.

GKA protocol is divided into four categories: tree-based [3], star-based [2], link-based [4], and ring-based [5]. Tree-based GKA has greater flexibility in membership management and is easier to achieve decentralization than other GKA protocols. Tree-based Group Diffie-Hellman (TGDH) [3] is a typical tree-based GKA protocol.

Tree-based GKA protocols depend on Diffie-Hellman key exchange (DHKE). However, DHKE was known to be secure only for passive attackers. DHKE cannot prevent any active attack such as man-in-the-middle (MitM) attack or impersonation attack. This was caused by the fact that DHKE does not check the identity during key exchange. Researchers then proposed authenticated key exchange (AKE) which includes authentication in two-party key exchange. Authentication was also introduced in a group key agreement to prevent active attackers.

Most of the previous AKE protocols such as YAK [6], MQV [7], HMQV [8], SIG-DH [9], and NAXOS [10] solve the authentication problem relying on trusted third party (TTP). AKE protocols relying on TTP successfully perform tasks that are required for authentication. However, TTP has a risk of a single point-of-failure. If a single point-of-failure takes place, any party in AKE protocol cannot communicate with other parties at all.

As active research on public ledger has been conducted, decentralized models are attracting attention in diverse fields. Problems from dependence on a central authority can be solved by public ledger. Thus

* Graduate School of Information Security, KAIST. 291, Daehak-ro, Yuseong-gu, Daejeon, South Korea 34141. {$hansh09, kkj$}@kaist.ac.kr)
† School of Computing, KAIST. 291, Daehak-ro, Yuseong-gu, Daejeon, South Korea 34141. {$thepride, kkj$}@kaist.ac.kr

authentication can be achieved via public ledger.

In this paper, we suggest Public Ledger based Authenticated group Key Establishment (PLAKE) and its delegated variant (dPLAKE). Two protocols add authenticity into tree-based GKA TGDH using public ledger. PLAKE uses Proof of Work (PoW)-based public ledger and dPLAKE uses Delegated Proof of Stake (DPoS)-based public ledger for authentication to remove the dependency of trusted third party. We discuss the security of PLAKE and dPLAKE.

1.2 Outline of the Paper

The rest of this paper is structured as follows: First, we describe public ledger, PoW, and DPoS in Section 2. Section 3 introduces previous work on group key agreement and authenticated key exchange over public ledger. We propose authenticated TGDH protocols PLAKE and dPLAKE over public ledger in Section 4. In Section 5, we give a security analysis of PLAKE and dPLAKE. We provide a discussion on PoW-based and DPoS-based public ledger in Section 6, and make a conclusion in Section 7.

2 Background

2.1 Public Ledger

Public ledger, which is also known as blockchain, is a linked-data structure that connects each block made up of transactions as shown in Fig. 1. Public ledger technology emerged with the introduction of Bitcoin [11]. Public ledger provides data forgery prevention and distribution of stored data.

Data Forgery Prevention: Bitcoin uses SHA-256 hash function to ensure the integrity of the previous block. Every data from the previous block is converted into a hash value which is referred to by the most recently added block. This structure makes data forgery infeasible because the link between blocks could be broken if someone changes previous data.

Distribution of Stored Data: Data stored in each node should be synchronized. If data is stored in a distributed way, the globally identical database could not be naturally maintained. The consensus algorithm is a way to share the same database among nodes in the network.

Each block contains various kinds of data. We can publicize any information in the public ledger by adding the information needed for authentication into the block.

2.2 Proof of Work

There are several consensus algorithms to achieve decentralization. Proof of Work (PoW), Proof of Stake (PoS), and Delegated Proof of Stake (DPoS) are typical examples of a consensus algorithm. PoW, which is adopted as a consensus algorithm in Bitcoin, is the most prominent algorithm. Nodes have to solve the hash puzzle to prove their work under PoW algorithm. The hash puzzle on Bitcoin is as follows:

$$H(H(r||nonce)) < target \qquad (1)$$

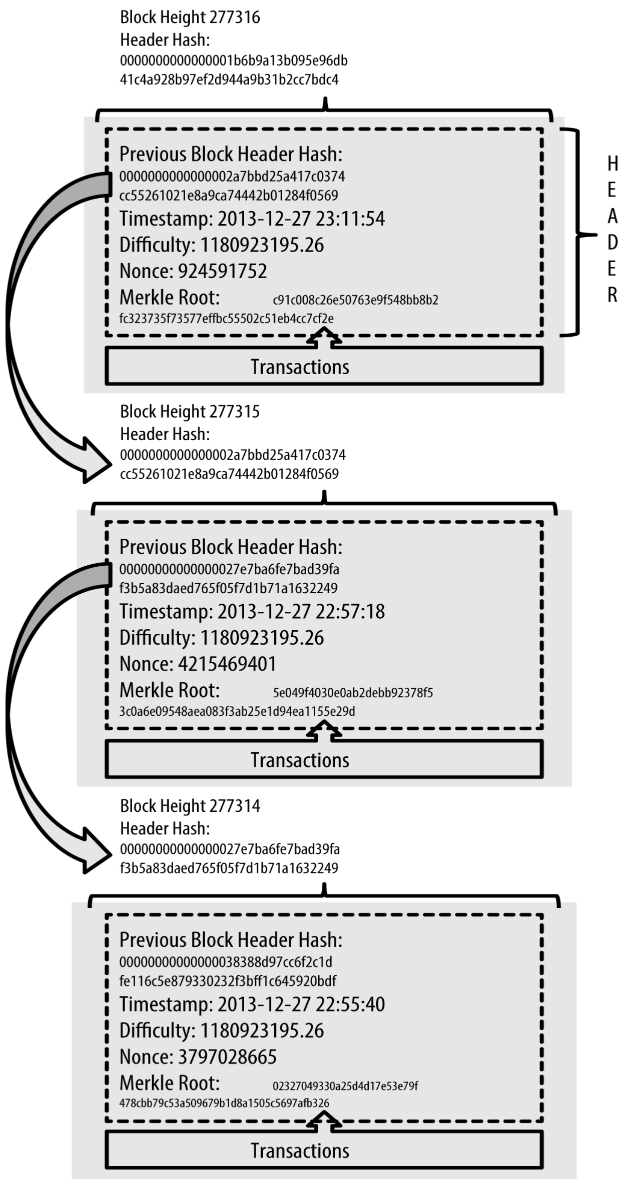

Block Height 277316
Header Hash:
0000000000000001b6b9a13b095e96db
41c4a928b97ef2d944a9b31b2cc7bdc4
Previous Block Header Hash:
0000000000000002a7bbd25a417c0374
cc55261021e8a9ca74442b01284f0569
Timestamp: 2013-12-27 23:11:54
Difficulty: 1180923195.26
Nonce: 924591752
Merkle Root: c91c008c26e50763e9f548bb8b2
fc323735f73577effbc55502c51eb4cc7cf2e
Transactions
HEADER
Block Height 277315
Header Hash:
0000000000000002a7bbd25a417c0374
cc55261021e8a9ca74442b01284f0569
Previous Block Header Hash:
00000000000000027e7ba6fe7bad39fa
f3b5a83daed765f05f7d1b71a1632249
Timestamp: 2013-12-27 22:57:18
Difficulty: 1180923195.26
Nonce: 4215469401
Merkle Root: 5e049f4030e0ab2debb92378f5
3c0a6e09548aea083f3ab25e1d94ea1155e29d
Transactions
Block Height 277314
Header Hash:
00000000000000027e7ba6fe7bad39fa
f3b5a83daed765f05f7d1b71a1632249
Previous Block Header Hash:
00000000000000038388d97cc6f2c1d
fe116c5e879330232f3bff1c645920bdf
Timestamp: 2013-12-27 22:55:40
Difficulty: 1180923195.26
Nonce: 3797028665
Merkle Root: 02327049330a25d4d17e53e79f
478cbb79c53a509679b1d8a1505c5697afb326
Transactions

Figure 1: The structure of public ledger [12]

H denotes *SHA-256* hash function. Nodes have to find *nonce*. r and *target* are fixed until Eq. (1) is solved. Nodes change *nonce* value when *nonce* does not satisfy Eq. (1). The puzzle is solved once node finds *nonce* that satisfies Eq. (1).

The hash puzzle is easy to check whether the solution is correct, but hard to find the solution. As the difficulty of puzzle increases, so does the time required to solve the puzzle. Bitcoin network automatically adjusts the difficulty so that the puzzle is solved every 10 minutes. In PLAKE, the difficulty is different for existing group members and a joining member.

2.3 Delegated Proof of Stake

Nodes select delegates through a continuous approval voting system in DPoS as we can see in Fig. 2. Any node can be a candidate for a delegate and be chosen as a delegate if a node gets a certain level of votes. The roles that delegates should perform are as follows:

1) Block Production: A delegate should produce blocks according to protocol. The number of blocks that a delegate produces depends on protocol. For example, a delegate on EOS [13] produces 6 blocks during one round. In dPLAKE, each delegate will produce 10 blocks during one round.

2) Block Validation: A delegate should validate blocks generated by other delegates. If the block has invalid data, delegates must reject the block for the security of public ledger.

3) Block Finalization: Blocks are finalized by an agreement of two-thirds of the delegates. If blocks are finalized, we cannot reverse the public ledger before the last finalized block.

The number of delegates depends on the design of protocol. EOS [13] has 21 delegates and Lisk [14] has 101 delegates. Our protocol has 3 delegates.

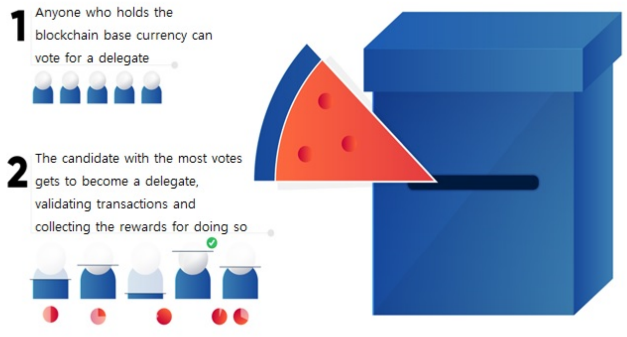

1
Anyone who holds the blockchain base currency can vote for a delegate
2
The candidate with the most votes gets to become a delegate, validating transactions and collecting the rewards for doing so

Figure 2: Delegated Proof of Stake [14]

3 Previous Work

3.1 Group Key Agreement

Group key agreement protocol is a kind of key agreement protocols that allow a group of users communicating over an insecure network to establish a shared secret key. GKA can be applicable to any secure group communications such as secure group chatting. Amir *et al.* [15] evaluate the performance of five most prominent group key agreement protocols: TGDH [3], GDH [16], CKD [17], STR [18], and BD [5]. Amir *et al.* [15] show that TGDH has better performance than other protocols in both LAN and WAN settings. Thus we consider TGDH as group key agreement protocol for PLAKE and dPLAKE protocols. Any entities cannot have higher authority in TGDH since all users are considered leaf nodes.

3.2 Authenticated Key Exchange over Public Ledger

Authenticated key exchange protocol relying on the root of trust requires TTP for authentication. Interest in blockchain and distributed ledger technology has increased enormously after Satoshi presented Bitcoin in 2008 [11]. Then a protocol using public ledger to decentralize the trust has also been considered in AKE protocols. Yao *et al.* [19, 20] propose AKE protocol between simplified payment verifiable nodes through a pre-shared secret. Pre-shared secrets are used in Bitcoin transactions to provide authentication. They use OP_RETURN code in Bitcoin script language to contain pre-shared secret in transaction. In two papers [19, 20], AKE is successfully implemented through the existing Bitcoin network. In contrast to Yao *et al.*'s protocol, we do not utilize the existing Bitcoin network. We use our own consensus algorithms for authentication.

Bui *et al.* [21] propose AKE protocols with public ledger. They construct AKE from proving the existence of event E with context and time. They use natural context including application identifier or out-of-band context which is shared through outside channels for context. In their paper, the authors prove that both a protocol based on natural context and one based on out-of-band context could defend MitM attack. They prove that it is possible to defend impersonation attack for a protocol based on out-of-band context. However, protocol based on natural context cannot prevent impersonation attack. To prevent denial-of-service (DoS) attack that intentionally generates a spam event, they propose to use private ledger for both protocols, but using private ledger contradicts their assumptions. Our protocols successfully prevent DoS attack since PLAKE uses PoW that is designed to prevent DoS attack and only delegates have the authority to authenticate participants in dPLAKE.

McCorry *et al.* [22] introduce Diffie-Hellman-over-Bitcoin which is non-interactive and YAK-over-Bitcoin which is interactive for post-transaction management. YAK-over-Bitcoin is based on YAK protocol proposed by Hao *et al.* [6]. Both protocols provide Bitcoin address authentication and transaction authentication. McCorry *et al.* prove private key security and session key security of Diffie-Hellman-over-Bitcoin protocol and YAK-over-Bitcoin protocol based on the security proof of Bellare's work [23]. In addition, they prove that YAK-over-Bitcoin provides perfect forward secrecy. While McCorry *et al.* use a public key infrastructure in their protocol, our approaches use hash puzzle and one-way authentication.

4 Adding Authenticity into TGDH by Public Ledger

We design two authenticated TGDH protocols using public ledger: PLAKE and dPLAKE. We use public ledger to authenticate a new member and store the list of authorized group members that perform group key agreement. Both PLAKE and dPLAKE build an authenticated group, record authenticated group members in the public ledger, and perform group key agreement based on public ledger. One block corresponds to one session.

PLAKE uses a PoW algorithm and dPLAKE uses

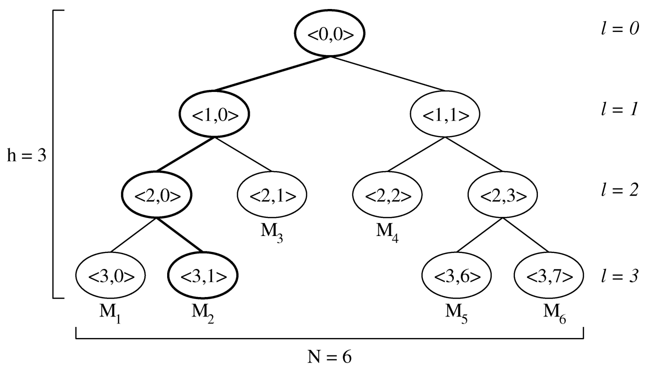


Figure 3: The structure of TGDH protocol [3]

a DPoS algorithm for authentication. Two protocols use TGDH as group key agreement. Fig. 3 shows the structure of TGDH protocol. M_i denotes a group member, N denotes the number of total group member, l denotes the level, and h denotes the height of a tree. As shown in Fig. 3, each member becomes a leaf node and contributes to establishing a group key. TGDH needs a special node called a sponsor that calculates the intermediate key that is required to establish a group key and broadcasts intermediate keys. In our protocols, a sponsor is required for the authentication process.

We describe each protocol in Section 4.1 and 4.2 in detail.

4.1 Detailed Description of PLAKE

4.1.1 Authentication over PoW-based Public Ledger

PLAKE uses PoW-based public ledger. Our PoW algorithm is different from PoW algorithm of Bitcoin. In Bitcoin, the person who solves a hash puzzle first has the authority to create a block. In PLAKE, the sponsor has the authority to generate a block. Thus block generation is not competitive in PLAKE.

PLAKE performs Join, Leave, Merge, and Key Refresh algorithms. PLAKE expels a malicious node if it exists.

The following is a detailed description of each algorithm:

Join

A joining member should be recorded on the block to participate in group key agreement. Node that was the sponsor in the last session of TGDH generates a new block. The sponsor gives a hash puzzle that can be solved with an average of 10 minutes to existing group members. Contrary to group members, the sponsor gives a joining member three puzzles that take 1 hour on average for each. It is noted that the sponsor does not give three puzzles to a new member at once. If a new member solves the first puzzle, then the sponsor gives the second puzzle. If a new member solves the second puzzle, then the sponsor gives the third puzzle to a new member. The reason for not giving the puzzle to new members at the same time is to prevent parallel operations to increase their workload.

The reason for differentiating puzzle difficulty is that each node has a different level of trust. As existing members have already agreed group keys safely, there is an expectation that they will perform group key agreement safely in the future. On the other hand, a joining member has a risk of being the attacker, so a new member should try to prove that he/she is willing to participate in group key agreement despite the penalty on solving a puzzle. The difficulty of solving puzzles prevents low-cost attacks.

After solving the puzzles, the sponsor aggregates the puzzles and answers that each member gives to the sponsor, as shown in Fig. 4. The reason the sponsor records the puzzles and answers is to leave the evidence that calculations are done well at a certain time. As shown in Fig. 5, the sponsor creates a block that contains the hash value of the block header, the hash value of the previous block header, time-stamp, nonce, a list of group members, current operation such as Join or Leave, and puzzles and nonces given to group members. Nonce is required to create a new block for forcing the sponsor to solve a puzzle that takes an average of one hour. This is to prevent malicious sponsors or external intruders from generating false blocks at low cost.

Once the block is created, Join protocol of TGDH group key agreement is performed among the group members recorded on the block.

Leave

Leave is the process of leaving an existing group member. The remaining group members must be recorded on the block to perform group key agreement. Block creation is performed by the sponsor node in the last TGDH group key agreement. When the sponsor leaves, a block creator is designated for a random node. The sponsor gives existing group members puzzles that are solved within 10 minutes on average. The sponsor records the solved puzzles and answers, solves the given puzzle which takes an hour on average, and then creates a block containing the information in Fig. 5. The recorded members perform Leave protocol of TGDH group key agreement after the block is created.

Partition

Partition is the process of leaving multiple group members at once. That is, Partition is a repeated Leave operation. The rest of the group members are recorded on the block for group key agreement. Block creation is performed by the sponsor node in the previous TGDH group key agreement. If the sponsor leaves, a randomly chosen node is designated as the block creator as Leave. The subsequent process is the same as Leave.

Key Refresh

If group members do not change for a long time, a new session is not processed and so a group key remains unchanged. The longer the same group key is maintained, the probability of group key leaking increases. Thus Key Refresh is performed to periodically change the group key. The group key is refreshed when

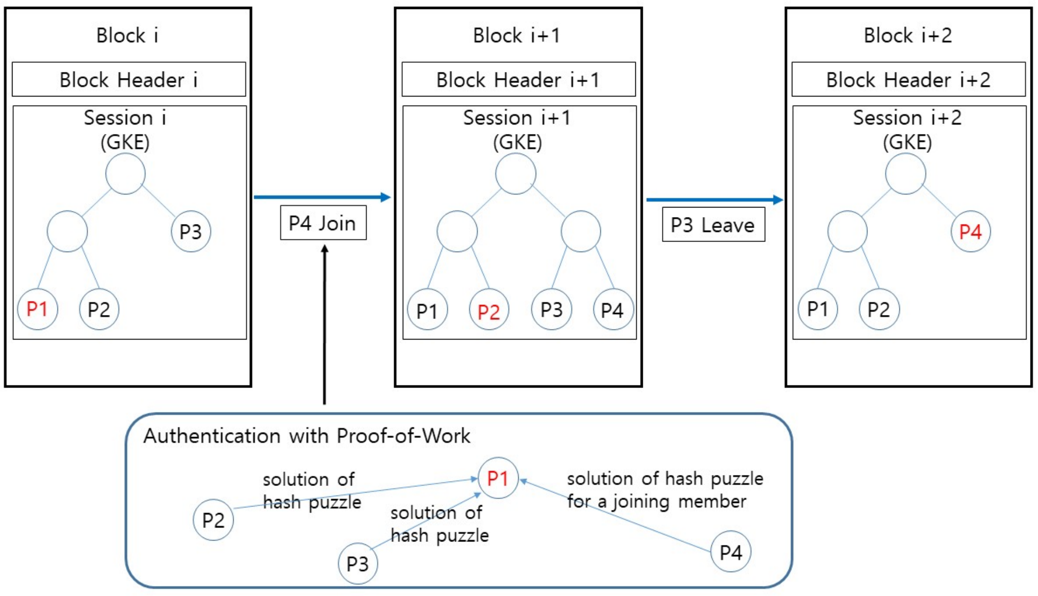


Figure 4: PLAKE protocol. Members with red color indicate delegates who play the role of sponsor in TGDH

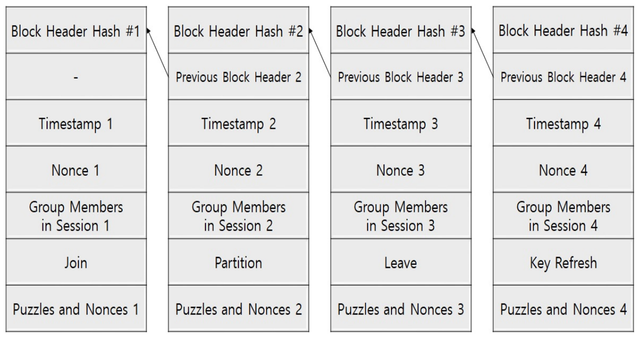


Figure 5: Block structure of public ledger in PLAKE

time-stamp of the latest block indicates one day before. If a new block is created due to Join or Leave, the deadline for key refreshment is reset. As no group member changes for Key Refresh, the block is created based on existing group members. The block generator is randomly chosen except for the latest block creator.

Expulsion of Malicious Node

If a malicious node exists in an authenticated group, the malicious node should be removed from the group for secure communication. The sponsor node will not give a hash puzzle to a malicious node for the next block generation. If the assigned sponsor turns out to be a malicious node, then nodes will select other nodes as a sponsor. Then a new sponsor will give hash puzzles to group members except a malicious node. Thus a malicious node is expelled by being omitted on the block records.

4.1.2 Group Key Agreement

While TGDH protocol includes Join, Leave, Merge and Partition protocol, Our GKA protocol only performs Join, Leave, and Partition. In the case of Join operation, if the new member is authenticated and added to a list of group members, the same process as Join algorithm of TGDH is performed. In the case of Leave and Partition, the same process as Leave and Partition algorithms of TGDH is performed. Key Refresh process is also identical to Key Refresh of TGDH. After Key Refresh, the node who is responsible for block producing leaves the role of the sponsor to arbitrarily selected nodes except itself.

4.2 Detailed Description of dPLAKE

4.2.1 Authentication over DPoS-based Public Ledger

We use DPoS-based public ledger in dPLAKE. Only delegates can generate a new block and process a new session under DPoS algorithm. As a delegate has strong authority, a delegate can serve as TTP. Thus a delegate easily and efficiently performs one-way authentication with a joining member. Fig. 6 shows that a delegate performs one-way authentication with a new member. However, we cannot find an adequate one-way authentication protocol, so it remains an open question.

We elect three delegates for each round via the voting process. Three most voted candidates are delegates. Delegates do not change until the end of the round. In case of a delegate having abnormal behaviors such as network failure, we have candidate delegates who have positions from the 4th to the 9th place in voting.

One round consists of 30 sessions if dPLAKE works well without erroneous situations. If a malicious node corrupt sessions, a new round will start. Voting is executed whenever a new round begins. Three delegates generate blocks in turn. We call a delegate who is responsible for block production as a block producer. The order of block production is randomly assigned.

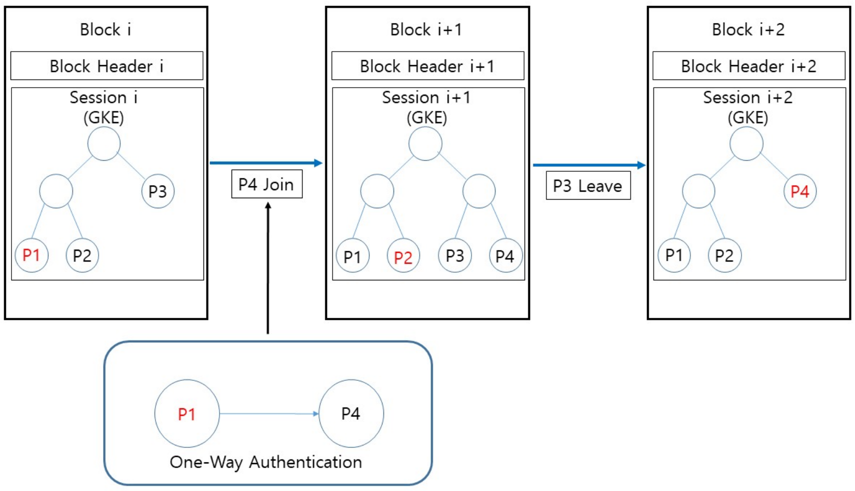
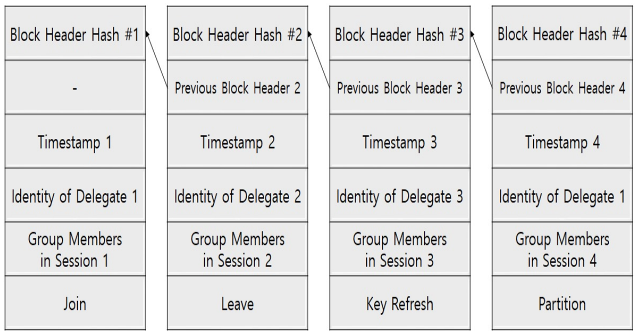


Figure 6: dPLAKE protocol. Members with red color indicate delegates who play the role of sponsor in TGDH



Block Header Hash #1	Block Header Hash #2	Block Header Hash #3	Block Header Hash #4
-	Previous Block Header 2	Previous Block Header 3	Previous Block Header 4
Timestamp 1	Timestamp 2	Timestamp 3	Timestamp 4
Identity of Delegate 1	Identity of Delegate 2	Identity of Delegate 3	Identity of Delegate 1
Group Members in Session 1	Group Members in Session 2	Group Members in Session 3	Group Members in Session 4
Join	Leave	Key Refresh	Partition

Figure 7: Block structure of public ledger in dPLAKE

Each delegate generates 10 blocks during one round, so a total of 30 blocks are generated during one round.

A block generated by a delegate contains the hash of the block header, the hash value of the previous block header, time-stamp, the identity of the delegate that creates the block, a list of group members who share a common secret, and current operation as shown in Fig. 7. A new block is created when new Join, Leave, or Partition occurs. Group key agreement occurs whenever a block is created and perform an operation according to group member information stored on the block. Also a block is generated when Key Refresh is performed. dPLAKE can expel a malicious node as PLAKE.

The following is a detailed description of each algorithm:

Join Protocol

A joining member should execute one-way authentication with the block producer for block production. If one-way authentication protocol runs correctly, the block producer adds a joining member to the list of group members and records the group member information on the block. Then a joining member participates in group key agreement protocol to share a common secret.

Leave Protocol

Leave protocol indicates the removal of an authenticated group member as PLAKE. The block producer records the remaining group members except the leaving member on the block. If the block producer leaves at this session, one of the other delegates performs the task of a delegate instead of a leaving delegate.

Partition Protocol

Partition protocol indicates that more than one group member leaves dPLAKE protocol as PLAKE. The block producer records the remaining group members except those leaving members on the block. If multiple delegates leave the protocol, candidate delegates will replace the existing delegates. If all delegates including candidate delegates leave the protocol, a new round begins with a voting process and the appointed delegate will continue the protocol by creating a new block.

Key Refresh

Key Refresh process of dPLAKE is identical to that of PLAKE. The only difference is that we choose other delegates to produce block in dPLAKE and choose random nodes for block production in PLAKE.

Expulsion of Malicious Node

If a malicious node exists in an authenticated group, the malicious node should be expelled from the group as PLAKE. If the deported node is not a delegate, the block producer considers the remaining group members except for the deported node as a group member, and then the block producer creates the block based on the remaining group members. If the deported node is a delegate, then a new round gets started and the voting is processed. We can expect that the remaining group members will not vote for a malicious node. After the voting process, it is possible to exclude a malicious node from an authenticated group by creating a block from the new delegate.

4.2.2 Group Key Agreement

Group key agreement process is almost identical to PLAKE protocol as we use TGDH for both PLAKE and dPLAKE. In dPLAKE protocol, the block producer plays the role of the sponsor. In the case of Join operation, if the new member is authenticated and added to the group during the one-way authentication protocol, the same process as the Join algorithm of TGDH is performed. In the case of Leave, Partition, and Key Refresh, the same process is performed as that of TGDH. After Key Refresh, the block producer leaves the role of the sponsor to an arbitrarily selected one of the remaining two delegates.

5 Security Analysis of (d)PLAKE

We analyze the security of PLAKE and dPLAKE. We divide two parts for security analysis. The first part corresponds to authentication over public ledger. The second part corresponds to group key agreement.

5.1 Security Analysis of Authentication over Public Ledger

PoW-based Public Ledger

If the session is Leave, Partiton, or Key Fresh, the total time for authentication takes 1 hour and 10 minutes on average since nodes should solve a hash puzzle in PLAKE protocol. If an attacker with a low-cost does not make an enough effort to join the group, he/she cannot participate in group key agreement. Thus authentication is guaranteed for low-cost attackers.

However, three hours to solve a puzzle given to a joining member may not be enough to prevent high-cost attackers. Therefore, it cannot prevent a high-cost attacker from authenticating with group members and corrupting the protocol. If an attacker succeeds in joining an authenticated group, he/she may leak information or intentionally create a fork to disturb group key agreement. In case of leakage of information, as TGDH provides session key security [3], we can minimize damage and exclude a malicious node. If an attacker creates a fork, nodes in the group may get confused for a while, but nodes will not follow the blocks that an attacker creates.

DPoS-based Public Ledger

The security of Join algorithm of dPLAKE depends on the one-way authentication protocol. Thus the security of Join algorithm can be discussed when actual one-way authentication protocol is determined.

The security of Leave and Partition Protocols is proved as group members leave the protocol voluntarily and TGDH provides key independence. However, if a public ledger is compromised by a malicious node, confidential information such as shared group key is leaked. To expel malicious group member and to maintain the authenticity of information, DPoS-based public ledger is used to finalize the block. That is, group members other than a malicious node can re-establish the session by reversing the public ledger to the finalized block. Therefore, we can prevent disturbance by malicious nodes after the finalized block. It remains an open question if a malicious node leaks the shared information before the finalized block.

5.2 Security Analysis of TGDH Group Key Agreement

PLAKE and dPLAKE follow TGDH GKA protocol as group key agreement. The security of TGDH depends on the difficulty of Decisional Tree group Diffie-Hellman Problem (DTGDH) which can be reduced to a two-party Decisional DH problem. Thus the security of PLAKE and dPLAKE is guaranteed as long as a two-party Decisional DH problem cannot be solved in a feasible time.

6 Discussion

Rigorous Security Proof

TGDH has proven to be secure under DTGDH assumption. Yet we do not know new problems caused by merging public ledger and TGDH protocol. Further studies should prove the formal security of authenticated TGDH protocol using public ledger.

Problems on PoW-based Public Ledger

The hash puzzle used in Bitcoin has a different speed of finding solutions depending on the hardware performance. Therefore, hash puzzles using memory-hard PoW such as Scrypt have been proposed so that all nodes solve puzzles in the same environment [24, 25, 26].

Another problem is that PLAKE cannot be used in emergency situations because it takes at least one hour to perform group key agreement and four hours to accept new members. Further discussion is needed to solve this problem.

Problems on DPoS-based Public Ledger

DPoS-based public ledger is by definition classified as a decentralized public ledger. Every node in DPoS-based public ledger elects delegates who produce blocks via the voting process. However, DPoS-based public ledger contains the risk of being centralized since the authority is concentrated on a small number of delegates. In dPLAKE protocol, as only three delegates are elected, the risk of centralization is higher than other DPoS-based public ledgers. Thus, we need to investigate how well decentralization is achieved via dPLAKE.

7 Conclusion and Future Work

This paper combines TGDH group key agreement protocol and authentication protocol over public ledger. We suggest PLAKE and dPLAKE which are authenticated TGDH protocols using public ledger based on PoW and DPoS, respectively. We use public ledger to remove the dependency of TTP for the authentication process. We analyzed the security of PLAKE and dPLAKE. To the best of our knowledge, this is the first work that combines authentication protocol over public ledger and group key agreement.

Due to some limitations about public ledger, we should deal with the problems on PoW-based and DPoS-based public ledger more and find an adequate one-way authentication protocol for dPLAKE.

On the other hand, TGDH are known as efficient protocols. Yet how good the efficiency is when TGDH and public ledger combined have not been dealt with. We will measure the performance of PLAKE and dPLAKE in the near future.

Acknowledgement

This work was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00555, Towards Provable-secure Multi-party Authenticated Key Exchange Protocol based on Lattices in a Quantum World).

References

[1] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[2] W. Lang, M. Zhou, and K. She, "Key agreement protocol in ad-hoc networks," in *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on*, vol. 1, pp. 296–301, IEEE, 2003.

[3] Y. Kim, A. Perrig, and G. Tsudik, "Tree-based group key agreement," *ACM Transactions on Information and System Security (TISSEC)*, vol. 7, no. 1, pp. 60–96, 2004.

[4] E. Bresson, O. Chevassut, and D. Pointcheval, "Provably secure authenticated group diffie-hellman key exchange," *ACM Transactions on Information and System Security (TISSEC)*, vol. 10, no. 3, p. 10, 2007.

[5] M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system," in *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 275–286, Springer, 1994.

[6] F. Hao, "On robust key agreement based on public key authentication," in *International Conference on Financial Cryptography and Data Security*, pp. 383–390, Springer, 2010.

[7] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone, "An efficient protocol for authenticated key agreement," *Designs, Codes and Cryptography*, vol. 28, no. 2, pp. 119–134, 2003.

[8] H. Krawczyk, "Hmqv: A high-performance secure diffie-hellman protocol," in *Annual International Cryptology Conference*, pp. 546–566, Springer, 2005.

[9] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 453–474, Springer, 2001.

[10] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *International conference on provable security*, pp. 1–16, Springer, 2007.

[11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[12] A. M. Antonopoulos, *Mastering Bitcoin: Programming the open blockchain.* " O'Reilly Media, Inc.", 2017.

[13] "Eos." https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md.

[14] "Lisk." https://lisk.io.

[15] Y. Amir, Y. Kim, C. Nita-Rotaru, and G. Tsudik, "On the performance of group key agreement protocols," *ACM Transactions on Information and System Security (TISSEC)*, vol. 7, no. 3, pp. 457–488, 2004.

[16] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 8, pp. 769–780, 2000.

[17] Y. Amir, Y. Kim, C. Nita-Rotaru, J. L. Schultz, J. Stanton, and G. Tsudik, "Secure group communication using robust contributory key agreement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 5, pp. 468–480, 2004.

[18] Y. Kim, A. Perrig, and G. Tsudik, "Group key agreement efficient in communication," *IEEE transactions on computers*, vol. 53, no. 7, pp. 905–921, 2004.

[19] H. Yao and C. Wang, "A novel blockchain-based authenticated key exchange protocol and its applications," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 609–614, IEEE, 2018.

[20] H. Yao, C. Wang, B. Hai, and S. Zhu, "Homomorphic hash and blockchain based authentication key exchange protocol for strangers," in *2018 Sixth International Conference on Advanced Cloud and Big Data (CBD)*, pp. 243–248, IEEE, 2018.

[21] T. Bui and T. Aura, "Key exchange with the help of a public ledger," in *Cambridge International Workshop on Security Protocols*, pp. 123–136, Springer, 2017.

[22] P. McCorry, S. F. Shahandashti, D. Clarke, and F. Hao, "Authenticated key exchange over bitcoin," in *International Conference on Research in Security Standardisation*, pp. 3–20, Springer, 2015.

[23] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *International conference on the theory and applications of cryptographic techniques*, pp. 139–155, Springer, 2000.

[24] A. Biryukov and D. Khovratovich, "Equihash: Asymmetric proof-of-work based on the generalized birthday problem," *Ledger*, vol. 2, pp. 1–30, 2017.

[25] C. Percival and S. Josefsson, "The scrypt password-based key derivation function," tech. rep., 2016.

[26] J. Tromp, "Cuckoo cycle: a memory-hard proof-of-work system.," *IACR Cryptology ePrint Archive*, vol. 2014, p. 59, 2014.