Design of Anonymous Endorsers in Hyperledger Fabric with Linkable Threshold Ring Signature

Dharani J[1]*, Sundarakantham K[1], Kunwar Singh[2], and Mercy Shalinie S[1]
[1]Department of Computer Science & Engineering, Thiagarajar College of Engineering, Madurai, India
jdharani791@gmail.com, {kskcse, shalinie}@tce.edu
[2]Department of Computer Science & Engineering, National Institute of Technology, Trichy, India
kunwar@nitt.edu

Abstract

Blockchain technology has gained its attention from its application in bitcoin which circumvented the problem of double spending that existed in the prior digital currencies, through validation. Particularly permissioned blockchain framework became popular with organizations forming consortium that allowed only authorized entities to participate in the network. Hyperledger Fabric, a popular distributed ledger technology hosted by Linux Foundation has remarkable features because of the fact that it is open source. It stands out from other frameworks as it focuses on the privacy-preserving requirements of the enterprises. Apart from only allowing authenticated organizations to participate in the network it implements channels that allows a subset of organizations to communicate concealing the existence of such a channel to other members. Optionally fabric also provides anonymity and unlinkability of the participating clients through a cryptographic protocol suite called Idemix that operates based on zero-knowledge proofs. Fabric follows the execute-order-validate transaction flow as opposed to order-execute flow that had certain limitations in other platforms. For executing the transactions submitted by the clients, fabric has designated endorsing peers which holds the smart contract - programmable business logic. Endorsing peers or endorsers execute the transactions and attach their signatures to the results for validation purpose. But revealing the endorser identities may be a problem when there is conflict of interest among the enterprises. Hence to have an unbiased flow of work it is important to conceal the endorser identity. According to [1] anonymization of endorsing peers is still a open problem in fabric community. We propose a solution to this problem which uses linkable threshold ring signatures that conceals the identity of endorsers. Ring signatures are known for preserving the privacy of the signer in a group. Threshold ring signature allows t-out-of-n signers to collaborate on the signing procedure. Employing threshold ring signature implicitly addresses one more problem stated in [10] where the verifiers need to manually count the valid ring signature that increases the verification time. This process of separately verifying each of the signatures and checking if the number of signatures is more than the threshold value is replaced by having just one threshold signature collaboratively signed by the required endorsers.

Keywords: Permissioned blockchain, Hyperledger Fabric, Anonymization, Threshold Ring Signature, Endorsing Peer

1 Introduction

The immutable ledger technology with strong cryptographic background has come up as a reliable information sharing platform. The variants of blockchain can be conveniently realized from the fact that if authentication is needed for joining the network or not. Permissionless blockchain puts no demand on the participants where anybody can join the network any time. UTXO (Unspent Transaction Output) graph of a bitcoin network is used to validate if every input (receive 5 bitcoins from X) transaction is less

Research Briefs on Information & Communication Technology Evolution (ReBICTE), Vol. 6, Article No. 4 (October 1, 2020)

*Corresponding author: Tel: +86-13951179091 than or equal to the output (2 bitcoins to Y and 3 bitcoins to Z). But doing so reveals the financial base of every participant. Hence only pseudonyms are tied to the participants rather than the real identities. Enterprise-grade permissioned blockchains evolved with a requirement that every party be known and authenticated. Though parties are known already trust may still be an issue in a consortium. This is where blockchain plays a major role where it transforms the need of trust into generating proofs. It is ideally suited for applications where multiple entities are involved in a business and requires a global view of data being transacted. Hyperledger fabric [4], a permissioned model with its modular architecture can be integrated into the existing business ecosystem.

2 Background

2.1 Hyperledger Fabric System Architecture

Hyperledger Fabric is an open source consortium for developing business blockchain technologies. It is an enterprise-grade permissioned distributed ledger platform. Fabric provides a modular, scalable and secure platform that supports private transactions and confidential contracts. Its modular architecture accomodates the diversity of usecases through plug and play components such as consensus, privacy and membership services. All this put together guarantees trust, transparency and accountability for businesses. Blockchain (Hyperledger Fabric) is a decentralized system with multiple nodes which interact with each other. It executes programs called smart contract (chaincode), holds state and ledger data, and executes transactions. Transactions are functions invoked on chaincode which is a principal component. Transactions must be approved or endorsed because only endorsed transactions will be committed in the ledger and reflected on the state. Dedicated system chaincodes exist for managerial functions.

2.1.1 Transactions

Transactions are of two variants:

- **Deploy Transactions:** The operation is responsible for creation and installation of new chaincode on the blockchain.
- **Invoke Transactions:** An invoke transaction alludes to an already deployed chaincode and to one of its specified function which updates the ledger state and returns the desired output.

2.1.2 State

Current state of the blockchain is depicted as a versioned key-value store (KVS), where keys correspond to names that uniquely identify the set of values associated with the keys. State ’s’ can be realized as a mapping between $K \rightarrow (V \times N)$ where K is the set of keys V is the set of values and N is a set of version numbers. There are two basic operations with KVS:

- **put(k,v):** Updates the value corresponding to key k
- **get(k):** Returns the value corresponding to key k

State is maintained by peers. The database options for state are LevelDB and CouchDB. LevelDB is the default database setting and CouchDB is the other possible option when assets are modeled as JSON and hence complex rich queries can be supported. Assets here correspond to anything with monetary value that is transacted over the network. Assets can be thought of as key-value pair collection and the subsequent state changes are committed as transactions on the ledger.

2.1.3 Ledger

Ledger contains the history of ’all’ transactions. By all we mean both successful state changes and unsuccessful attempts to change state i.e., valid and invalid transactions respectively. Ledger is maintained at the peer nodes and each peer locally maintains a bitmask to distinguish between valid and invalid transactions. It is the orderer node that is responsible for constructing the ledger as an ordered hashchain of block of all transactions.

2.1.4 Nodes

Nods can be realized in three ways:

- **Clients** are the representative nodes of end-users who get the service of the blockchain network. It communicates with the blockchain network through peers. Clients are responsible for creation and subsequently invocation of transactions.
- **Peers** can be either endorser or committer or even both based on the functionality they perform. Endorsers execute the chaincode and sign on the results for verification by other peer nodes. Committer is responsible for validating the transactions and update the state information onto the ledger.
- **Orderer** nodes are responsible for packaging the ordered transactions in the blocks.

2.2 Transaction Flow in Fabric

A distinguishing factor of Hyperledger fabric from other blockchain framework is its transaction lifecycle. Transaction lifecycle in other platforms is the following:

- **Order** Transactions are first ordered based on prefixed criteria and then sent to all the peers
- **Execute** Transactions are sequentially executed by all peers

In contrast to the above sequence fabric follows a different transaction flow as briefed below:

- **Execute** Transactions are executed by selected peers (endorsers) in parallel with order being trivial
- **Order** Transactions are ordered and broadcasted to peers for validation
- **Validate** Finally all the peers validate the transactions and append the validated one to the ledger based on consensus mechanism.

The above ordering have the advantage of execution of chaincode happens only at a subset of peers which guarantees chaincode privacy and execuitng in parallel improves performance. Fabric is flexible as the chaincode can be written in general-purpose language in contrast other platforms that requires the smart contract to be written in domain specific language.

2.3 Transaction Endorsement WorkFlow

Figure 1 depicts the transaction flow in fabric network.

1. Client submits a transaction proposal to the endorsing peers.

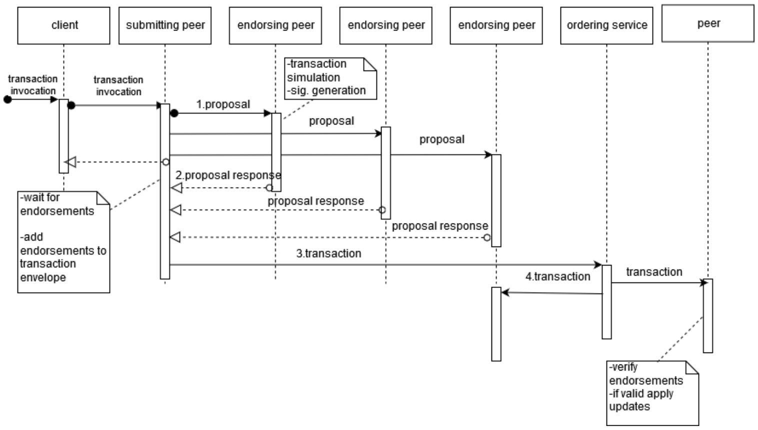

Figure 1: Transaction Flow in Fabric

2. Endorsing peers verify the client signatures and on being valid executes the specified function of the corresponding chaincode according to the chaincodeID. Then appends its signature to the proposed response and sends it back to the client.

3. Upon receiving 'adequate' endorsements for the proposal response client then forwards it to the orderer.

4. Orderer node performs all the validation checks through consensus and then packages them into block and broadcast to the anchor peer. Anchor peer sends it to other peers within its organization through a gossip data dissemination protocol.

2.4 Endorsement Policy

Every peer holds a set of predefined endorsement policy with respect to the chaincode. Endorsement policy is one of the evaluation criteria to segregate the transactions as valid or invalid. A transaction needs to be endorsed by all the endorsers as dictated by the endorsement policy. It is the Chaincode that specifies the set of endorsers. Assume an endorser set E of the form, $E = A, B, C, D, E, F$ Fabric allows clients to frame the endorsement policy as transactions are initiated by clients they have control on the confidentiality aspect of chaincode and dictate certain peers can only (view and) execute the chaincode, that are called as endorsers. Endorsement policy preserves the chaincode privacy by eliminating the need for every peer to hold a copy of the chaincode. Some sample endorsement policies are:

- Peers $A\ AND \{B\ OR D\}$
- 4-out-of-6 peers have to endorse
- All peers have to endorse

2.5 Validated ledger (VLedger)

Every peer maintains a VLedger apart from having the ledger and state or world state of the ledger. This VLedger or Validated Ledger as its name suggests filters out the invalid transactions and holds only the hashchain of valid transactions.

3 Confidentiality and Privacy Features offered by Fabric

Fabric authenticates every entity through Membership Service Provider (MSP) a pluggable component. Each organization will have a dedicated MSP that helps to map each peer with their respective organization. It also provides roles to the clients and peers as a measure of access control. Roles from client side include administrator, emplyee or any user getting the service of the organization. Hence every entity will be authenticated and access grant will be provided based on the roles tied to them. A subset of the organizations may be willing to transact without the rest of the members knowing the connection. For this reason fabric implements channels within the network. Assume organizations org1,org2,org3,org4,org5 form a consortium and agree to implement blockchain for their business. A situation may arise where only organizations org1,org2 and org3 need to do a business deal and doesn't want those communications to be seen by org4 and org5. With fabric organizations org1,org2 and org3 can form a channel and maintain a separate ledger for that channel. Transactions pertaining to this channel is updated to this ledger alone. Hence such a channelization enhances the privacy aspect of companies within a consortium. There may be even more stringent requirements on the confidentiality of sensitive data which cannot be stored on the ledger in plain format. For this reason fabric offers a private database called SideDB that is local to every peer and only hashes of the data are recorded on the ledger. Optionally end to end communication can be encrypted with TLS protocol. One of the most striking feature is anonymization of the clients to provide a measure of privacy to the clients. This is handled by the Identity Mixer (Idemix) component of the fabric that works based on a strong cryptographic primitive called zero-knowledge proofs.

3.1 Anonymization & Unlinkability of Clients with Idemix

MSP implementation with Idemix provides granular authentication along with privacy-preserving guarantees of anonymity and unlinkability. Anonymity refers to concealing the identity of transactor and unlinkability is when a transactor can initiate multiple transactions still unable for anyone to track that these transactions were sent by a single entity. There are three entities involved in an Idemix implementation of fabric: user, issuer, and verifier. An issuer issues digital certificate called credential that encompasses the set of user attributes. The user then generates a ”zero-knowledge proof” of the credential and selectively reveals the attributes, based on a presentation policy, to the verifier without disclosing the entire set. This is depicted in the figure 2 below. Zero-knowledge proof is a strong privacy-preserving authentication protocol suite. To get a driver's license it involves submitting a government generated proof in order to prove the requester is above 18 years of age. The proof may contain several attributes such as the requester name, address, date of birth and much more. But with zero-knowledge proof it is possible for one to generate a proof hiding all the attributes of the identity and only disclosing the fact that the age is above 18 years. This is possible with the rich set of properties that the zero-knowledge proof holds:

- **Completeness:** An honest prover can convince an honest verifier if the statement is true.
- **Soundness:** The probability that a dishonest prover convinces the honest verifier is very minimal.
- **Zero-knowledge:** Ability for any verifier to verify that the statement is true without knowing any additional information. It can be thought of as a prover holding a secret and has to prove to a verifier that he holds without revealing the secret itself.

4 Limitations of the existing architecture

Hyperledger Fabric with a myraid of privacy features has some problems to be addressed. The MSP implementation with idemix currently supports only anonymous client authentication as signing is only

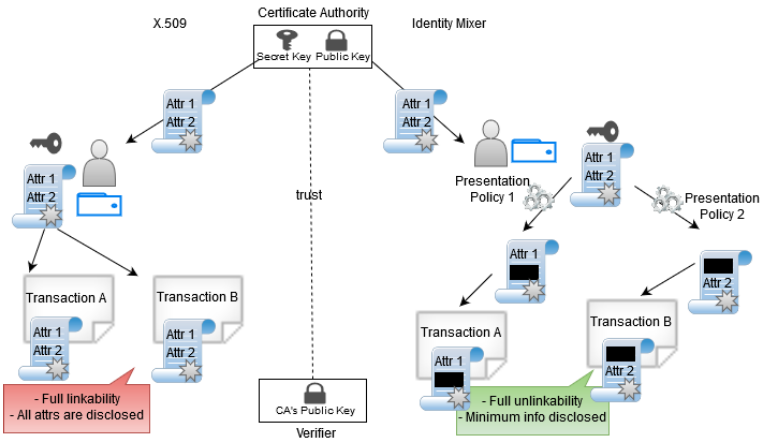

Figure 2: X.509 vs. Identity Mixer

done through Client SDK. According to [3], knowing the identity of endorsers may be prone to attack such as DoS in order to prevent a particular client from getting the desired service or to bring down network efficiency.

5 Literature Survey

Rivest et al.,[11] developed the notion of ring signatures to hide the identity of the signer. Following it several works were proposed for the ring signature scheme[2, 8]. Threshold ring signature scheme was first proposed by Bresson et al., [5]. [6], an ID-based threshold ring signature suitable for ad-hoc networks. But they lack the linkability among signatures from the same source. Also they are prone to insider attack.

Monero[13] functions based on an efficient Ring Confidential Transaction protocol called RingCT 2.0 that uses Pederesen Commitment. It lacks linkability of two signatures coming from the same signer and hence cannot be used by endorsers. Zcash [9] shields the transactions and the participants by employing zero-knowledge protocol but has a set-up phase that requires a trusted third-party. The notion of blockchain is to transform trust into proofs henceforth we choose to avoid this notion of depending on a trusted third party. Threshold signatures for blockchain [12] has summarized the application of signatures for consensus algorithms, CA (Certificate Authoritites) and to validate transactions. In security model, they have not considered the insider attack i.e., considering one of the compromised endorser participating in the construction of threshold signature. [10] has addressed the same problem of anonymizing the endorsers in Hyperledger Fabric framework. They have proposed a constant sized ring signature algorithm, Fabric's Constant-Sized Linkable Ring Signature (FCsLRS). This protocol effectively conceals the identity of the endorser and also has the signature size independent of the number of endorsers. They also offer linkablity feature where a signer cannot sign more than once. But as their protocol generates ring signatures rather than threshold ring signatures, the verifier have to individually count and check the valid ring signatures.

The proposed work involves an efficient linkable threshold ring signatures [14] that is linkable, threshold based and secure against not just random oracle model but a standard model. The signature size has a complexity of $O(\sqrt{n})$ and a constatnt linking complexity of $O(1)$ which is far better than Fujisaki[7] signature scheme that has $O(n)$ and $O(nlogn)$ signature size and linking complexity.

6 Proposed Anonymous Endorser Framework

In the proposed framework we first look into the details of the Linkable Theshold Ring Signature (LTRS) construction by Yuen et al. and subsequently get into the details of embedding the scheme into the MSP of Hyperledger Fabric.

6.1 MSP Implementation with LTRS scheme

Transactions are initiated by clients which is not a part of the blockchain network. Submitting peer which serves as a proxy for the client within the blockchain network verifies the client signature on the transaction and wraps up the transaction into a proposal message as shown in figure 3 and forwards it to the endorsing peer. Endorser peers are chosen from an Endorser Set E according to the endorsement policy that is predefined for every transaction chaincode. The endorsers on receiving the proposal message verify the signature of the client; executes the transaction and sends back a *signed proposal response message* depicted in figure 4 back to the submitting peer. It is at this point our proposed work differs. Instead of each endorser (as dictated by the endorsement policy) signing the transaction result separately, endorsers come into consensus on the result of the executed transaction and run Yuen's Linkable Threshold Ring Signature scheme to produce a single threshold ring signature and sends it back to the submitting peer. The submitting peer at this point needs to just collect *one endorsement message*, as depicted in the figure 5, signed collectively by the designated endorsers and forward it to the orderer. Orderer node broadcasts the transaction to the anchor peers, of the participating organizations, which in turn broadcasts to every node in network of their respective enterprise. Committing Peers validate the transaction and add it to the existing blockchain if transaction passes all the prescribed validity checks. The proposed transaction flow is depicted in figure 6.

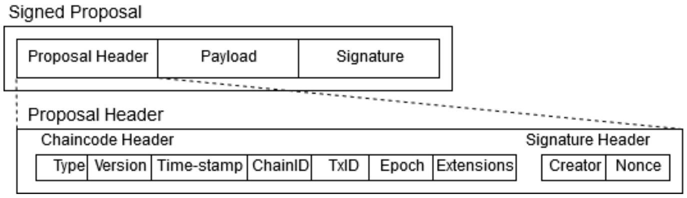

Figure 3: Structure of Proposal Message

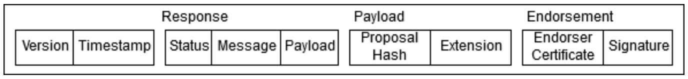

Figure 4: Structure of Proposal Response Message

6.2 Linkable Theshold Ring Signature Construction To Anonymize the Endorsers

LTRS scheme is a tuple with five algorithms $(Setup, KeyGen, VerifyandLink)$

- $param \leftarrow Setup(\lambda)$ is a Probabilistioc Polynomial Time (PPT) algorithm that takes as input λ which is a security parameter and outputs a set of parameters including λ. Domains of event-id, messages and signatures are represented as $EID, M\ and\ \Sigma$ respectively.
- $(s_i, p_i) \leftarrow KeyGen(param)$ Each Endorser E takes the *param* parameter generated in the set-up phase as input and generates the secret/public key pair. $SK\ and\ PK$ are used to denote the domains of possible secret keys and public keys respectively.

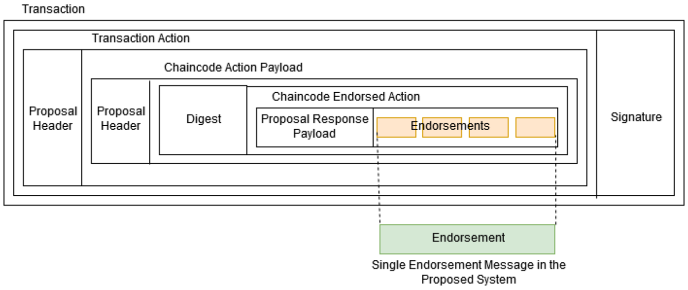

Figure 5: Transaction Envelope with Multiple vs Single Endorsement Message in current vs proposed scheme

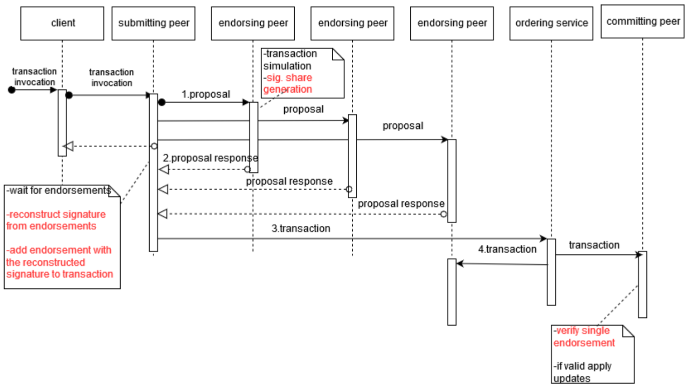

Figure 6: Proposed Transaction Flow in Fabric

- $\sigma \leftarrow Sign(e,n,d,\Upsilon,X,M)$ takes as input event id e, Endorser Set n, threshold value $d \in 1,\ldots,n$, set Υ of n public keys in the set PK, a set X of d private keys, and a message M, produces a signature σ.

- $accept/reject \leftarrow Verify(e,n,d,\Upsilon,M,\sigma)$ with input event id e, Endorser Set n, threshold value $d \in 1,\ldots,n$, set Υ of n public keys in the set PK, a message-signature pair (M,σ) returns accept/reject if the signatures are valid/invalid respectively. This algorithm is executed at the submitting peer.

- linked/unlinked $\leftarrow Link(e,d,n_1,n_2,M_1,M_2,\sigma_1,\sigma_2)$ with input event id e, Endorser Set n_1,n_2 (where $n_1 \leq n_2$), threshold $d \in 1,\ldots,n$, two sets Υ_1,Υ_2 of n_1,n_2 public keys respectively, two valid signature and message pairs $(M_1,M_2,\sigma_1,\sigma_2)$ outputs linked/unlinked. It should satisfy *Linking Correctness* and *Non-Slanderability* meaning any two signatures signed by two different signers should be unlinkable. This check is made at the submitting peer.

Yuen's LTRS scheme should satisfy the Correctness property:

- Verification Correctness: Valid signatures must be accepted during verification.

- Linking Correctness: The property ensures that two signatures are linked if and only if two signatures share at least one common signer for the same event.

7 Implementation & Results

Experiments were carried out in `Intel Core i5-4200U` CPU, `dual core processor, 8GB` RAM, OS: `Ubuntu-16.04` LTS (64 bit). `Go 1.12` programming language was used. A new Membership Ser-

Table 1: Performance Analysis

Performance Metrics	FCsLRS [10]	LTRS Scheme
Signature Size	O(1)	$O(d\sqrt{n})$
Model	ROM	Standard
Linking Complexity	$2E$	$O(dlogd)$
Sign Computation	$11E+5M$	$(8d+4d\sqrt{n})E+(4d+2d\sqrt{n})M+dOTS$
Verify Computation	$6E+10M$	$2dE+8d(1+\sqrt{n})P+dOTV$

vice Provider (MSP) is implemented in order to integrate the linkable threshold ring signature in the endorsement process. This new MSP implements a `ThresholdRingSigner` interface which provides a signing identity to the endorsers. When the endorser calls the sign function, signature share is created with the user's private key instead of a regular signature. Next, $Verify$ and $Link$ functions are implemented at the submitting peer. $Verify$ interface allows the submitting peer to verify the threshold ring signature whereas the $Link$ function checks for linkability within the signers. We have compared our performance with the FCsLRS scheme [10] which is given in the table 1. Our scheme provides a better security as it works for standard model as opposed to FCsLRS scheme [10] which is only secure under the random oracle model. The existing system has a constant signature size but the proposed scheme is dependent on the number of endorsers. E represents exponentiation of the form x^a and M represents multibase exponentiation of the form $x^a.y^b$.

8 Conclusion

In this paper we have performed anonymization of endorser peers in hyperledger fabric framework to preserve the privacy of the approvals made in consortiums. Anonymization is achieved by employing Yuen's Linkable Threshold Ring Signature (LTRS) scheme. LTRS scheme is secure under the standard model and guarantees linkable-anonymity under full key exposure and insider attacks and hence the same secure assumption can be made for the proposed work as well. The proposed methodology has also reduced the number of endorsement message to just one instead of having separate endorsement message for each endorser in the the transaction broadcast.

References

[1] Current limitations of idemix, March 2020. `https://hyperledger-fabric.readthedocs.io/en/release-2.0/idemix.html` [Online; Accessed on September 15, 2020].

[2] M. Abe, M. Ohkubo, and K. Suzuki. 1-out-of-n signatures from a variety of keys. In *Proc. of the 8th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'02), Queenstown, New Zealand*, volume 2501 of *Lecture Notes in Computer Science*, pages 415–432. Springer, Berlin, Heidelberg, December 2002.

[3] N. Andola, M. Gogoi, S. Venkatesan, S. Verma, et al. Vulnerabilities on hyperledger fabric. *Pervasive and Mobile Computing*, 59:101050, 2019.

[4] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proc. of the 13th EuroSys Conference (EuroSys'18), Porto, Portugal*, pages 30:1–30:15. ACM, April 2018.

[5] E. Bresson, J. Stern, and M. Szydlo. Threshold ring signatures and applications to ad-hoc groups. In *Proc. of the 22nd Annual International Cryptology Conference (CRYPTO'02), Santa Barbara, California, USA*, volume 2442 of *Lecture Notes in Computer Science*, pages 465–480. Springer, Berlin, Heidelberg, August 2002.

[6] S. S. Chow, L. C. Hui, and S.-M. Yiu. Identity based threshold ring signature. In *Proc. of the 7th International Conference on Information Security and Cryptology (ICISC'04), Seoul, Korea*, volume 3506 of *Lecture Notes in Computer Science*, pages 218–232. Springer, Berlin, Heidelberg, December 2004.

[7] E. Fujisaki. Sub-linear size traceable ring signatures without random oracles. In *Proc. of the Cryptographers' Track at the RSA Conference 2011 (CT-RSA'11), San Francisco, CA, USA*, volume 6558 of *Lecture Notes in Computer Science*, pages 393–415. Springer, Berlin, Heidelberg, February 2011.

[8] C.-z. Gao, Z.-a. Yao, and L. Li. A ring signature scheme based on the nyberg-rueppel signature scheme. In *Proc. of the 1st International Conference on Applied Cryptography and Network Security (ACN'03), Kunming, China*, volume 2846 of *Lecture Notes in Computer Science*, pages 169–175. Springer, Berlin, Heidelberg, October 2003.

[9] D. Hopwood, S. Bowe, T. Hornby, and N. Wilcox. Zcash protocol specification. GitHub: San Francisco, CA, USA, 2016. `https://raw.githubusercontent.com/zcash/zips/master/protocol/protocol.pdf` [Online; Accessed on September 15, 2020].

[10] S. Mazumdar and S. Ruj. Design of anonymous endorsement system in hyperledger fabric. *IEEE Transactions on Emerging Topics in Computing*, 2019.

[11] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *Proc. of the 2001 International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'01), Gold Coast, Australia*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, Berlin, Heidelberg, November 2001.

[12] C. Stathakopoulous and C. Cachin. Threshold signatures for blockchain systems. Technical Report RZ3910, IBM Research - Zurich, April 2017.

[13] S.-F. Sun, M. H. Au, J. K. Liu, and T. H. Yuen. Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In *Proc. of the 22nd European Symposium on Research in Computer Security (ESORICS'17), Part II, Oslo, Norway*, volume 10493 of *Lecture Notes in Computer Science*, pages 456–474. Springer International Publishing, September 2017.

[14] T. H. Yuen, J. K. Liu, M. H. Au, W. Susilo, and J. Zhou. Efficient linkable and/or threshold ring signature without random oracles. *The Computer Journal*, 56(4):407–421, 2013.

Author Biography

Dharani J is a Full-Time Research Scholar in the CSE department, Thiagarajar College of Engineering, Madurai, India. She received a B.Tech. degree from Noorul Islam College of Engineering and Technology, an M.E. degree from National Engineering College. Her Masters Project was in Privacy-Preserving Biometrics. Her research interests include Cryptographic Protocols and Blockchain.

Sundarakantham K is currently working as a Professor in the Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai, India. She has obtained her PhD from Anna University. Her area of interest is in Artificial Intelligence, Language Technology and Network Security. She has published many papers in peer reviewed journals and conferences.

Kunwar Singh is working as an Assistant Professor at NIT-Trichy since 2006. He obtained his Ph.D. from IIT Madras. His area of research is in cryptology, Blockchain Technology. He has published many papers in peer reviewed journals and conferences in the area of Cryptology.

Mercy Shalinie S is a Professor in the CSE department, Thiagarajar College of Engineering, Madurai, India. She received a B.E. degree from Annamalai University, an M.E. degree from Coimbatore Institute of Technology and a Ph.D. degree from Madurai Kamaraj University. She was a post-doctoral fellow at the University of California, Irvine, USA. She has published over 200 research papers. Her areas of research interests include machine learning and security systems. Prof. Shalinie was the Principal Investigator for various sponsored research projects and reviewer in various journals.