Privacy Preserving Records Sharing using Blockchain and Format Preserving Encryption

Sai Sandilya Konduru* and Vishal Saraswat**[0000−0001−7082−9568]

*Shiv Nadar Institute of Eminence, **Robert Bosch Engineering & Business Solutions Pvt. Ltd., Bangalore, India
{sandel2001,Vishal.Saraswat}@gmail.com

Abstract. Healthcare providers cannot share their patients' encrypted data among themselves because of interoperability issues. Many blockchain- based solutions have been proposed to allow for sharing medical data in a privacy-preserving manner, but interoperability problems persist. In this paper, we present a protocol called Blockchain-Format Preserving Encryption (B-FPE) to preserve patients' data privacy. Each patient is provided with an FPE key at the time of registration. All medical records are encrypted with the FPE key and stored in the blockchain. All the blockchain transactions are signed using group signatures. We use group signatures for signing the transactions to maintain the anonymity of healthcare providers. The new encrypted data block is concatenated to the blockchain. We present two cases: The regular phase, in which a patient is in a conscious state to share their FPE key with the healthcare provider, and the Emergency phase, in which a patient is not in a conscious state to share their key with the healthcare provider. In the latter case, the healthcare provider reconstructs the FPE key and decrypts the ciphertext. We assume this decryption happens in an oblivious manner.

Keywords: Format Preserving Encryption · Blockchain · Group Signatures · Secret Sharing · Medical Records · Privacy · Security

1 Introduction

The internet's pervasiveness has increased rapidly in the last decade, and many industries transformed their business types and stored records into digital form. Increased digitization has made data more accessible. Security is equally essential for confidential data. Security should be prioritized for industries that handle users' sensitive information. The healthcare industry is one such industry that deals with the sensitive information of patients. Healthcare data breaches can put patients' lives in danger. There are cases like the breach of the Broward health network in which around 1.3 million patients' data was exposed. A recent study [39] shows that data breaches are increasing yearly. In 2021, there were 45 million individuals who were affected by healthcare breaches [39]. Hence, it is crucial to store all medical records securely.

Patients do not visit only a single healthcare provider throughout their life. People may migrate from one city to another or even from one country to another. They visit

different healthcare providers at different places. As a result, patients are leaving their healthcare data fragmented across various healthcare providers. There might be cases where a patient suffers from a chronic illness and follows the medical prescription given by a particular healthcare provider. The patient might migrate to a new city where the patient happens to visit a different healthcare provider. Doctors at new healthcare providers cannot start the medication from scratch and hence require access to the patient's past medical history. Neither the patient can provide complete medication details (assuming they does not know the technicalities) nor the new healthcare provider can contact the previous healthcare provider that the patient visited and obtain the patient records unless both belong to the same parent organization. Since patient data is confidential, healthcare providers do not share their patients' data with other parties.

A solution to this problem is blockchain technology. Although David Chaum first introduced blockchain in 1979 [14], the popularity of blockchain increased after Sotashi Nakomato's Bitcoin in 2008 [44]. Because of its robust technology, blockchain has emerged as one of the promising peer-to-peer technology. Applications of blockchain include industries like vehicle-to-vehicle communications [50,11], healthcare [32,31,59], financial sector [60,48,1], supply chain [15,41,57], etc. In this paper, we present a new blockchain model which integrates patient's medical record using Format Preserving Encryption (FPE) without compromising the security of patient's data.

Using blockchain, healthcare providers can assemble patients' details like date of birth, social security number (SSN), medication details, etc., in the blocks. Since blockchain is immutable, malicious parties cannot delete or edit patients' data. Thus, it provides a structure for data sharing along with the security of the data. Here comes a new problem, since healthcare providers do not hold large databases, they store their data on the cloud, and they cannot store the data in plaintext format. Hence, the solution is to encrypt and store the data. Using available encryption modes like AES can destroy the formats of patient data. Consider the following case: In general, the patient details include the date of birth (DoB) and the social security number (SSN). Encrypting with AES-128 results in a ciphertext of length 128 bits each. Healthcare providers cannot distinguish between the ciphertexts of the date of birth and the social security number (SSN). Hence, it is difficult for healthcare providers to check whether all the details (which are in different formats) are collected. Note that healthcare providers need not know the actual data values but only the formats. Also, the data in the blockchain should also be in a specific data structural format. To preserve the format of patients' data, we use Format Preserving Encryption (FPE) for encrypting patients' medical records and storing them in the blockchain. Applications of Format Preserving Encryption (FPE) includes credit card numbers, social security numbers (SSN), phone numbers, postal addresses, etc. In short, FPE can be used to encrypt personally identifiable information (PII) of a person. Most healthcare providers require patients' PII to create their medical records. Hence, we store the encrypted data of users in the blockchain using FPE.

1.1 Our Contribution

We propose the B-FPE protocol[1] for sharing patients' medical records among healthcare providers in a privacy-preserving manner. The patient's medical records are encrypted using Format Preserving Encryption (FPE) and are then stored on a blockchain. This solves the interoperability problem among healthcare providers without compromising security. Each user/patient is provided a unique FPE key with which their data is encrypted and stored on the blockchain. The FPE key is distributed using a proactive secret sharing technique to all the healthcare providers in the network. Following are some of the features of our protocol.

1. Our B-FPE protocol has two variants.
   (a) **Regular Phase:** In this case, we assume that a patient visits a healthcare provider in a conscious state to share their FPE key.
   (b) **Emergency Phase:** In this case, we assume that a patient visits a healthcare provider and is not in a conscious state to share their FPE key. Hence, the healthcare provider obtains the key shares from other healthcare providers, reconstruct the key, and decrypt the encrypted medical records without knowing anything about the reconstructed FPE key.
2. We use proactive secret sharing scheme to update all the FPE key shares which are held at different healthcare providers.
3. All the blockchain transactions and key shares are signed using group signatures.

1.2 Outline of the Paper

The rest of the paper is arranged as follows: Section 2 discusses the related work to our paper. Section 3 explains the technical concepts that we have used for our framework. Section 4 discusses the threat model and security notions of our proposed framework. Section 5 discusses our B-FPE protocol in detail. Section 6 discusses the security analysis of our proposed B-FPE protocol. Finally, we conclude our discussion in Section 7.

2 Related Work

In this section, we give an overview of existing works on using blockchain in medical records.

Aritra et al. [43] proposed a new framework to store the medical records of patients using a permissioned blockchain. They use two frameworks to store the electronic medical records of patients. In general, medical records include small-size data like medical prescriptions, scans, etc., as well as large-size data like MRI scans, CT scans, high-resolution PET scans, etc. Small-sized data records are stored in a hyper-ledger fabric [34], and large-sized elements are stored in the inter-planetary file system (IPFS).

[1] The abbreviation is derived from the two terms *blockchain* and *format preserving encryption*

Hyper-ledger fabric [34] is an open-source project through which permissioned blockchain networks are created. A permissioned blockchain is a distributed ledger that is not publicly accessible. Only a few users with permission to participate in the network can access the ledger. This hyper-ledger fabric enables the participants in the network to know all the performed actions like transactions; deploying a smart contract on blockchain follows an endorsement established for the network. The inter planetary file system (IPFS) is a distributed file management system over a peer-to-peer network. The hash function of the required file retrieves the required file from IPFS. Since IPFS is a distributed file system, the remaining nodes can still deliver the requested file even if one or two network nodes are unresponsive. As mentioned, large data records like CT scans and MRIs are stored in IPFS by splitting the record into smaller fragments.

Gordan et al. [25] describe two types of interoperability in healthcare: institution driven, and patient driven. In institution driven interoperability, healthcare providers exchange data for their businesses, whereas in patient driven interoperability, patients' data is made available through standard mechanisms like APIs. Patient driven interoperability has more challenges than compared to institution based. These challenges include patient consent, governance, security, and privacy. They present blockchain as a solution for patient driven interoperability. They provide a high-level framework on how a patient can communicate with multiple healthcare providers and aggregate their medical records.

Zhang et al. [61] propose a new medical-sharing scheme using consortium blockchain. The network members include doctors, patients, large medical institutions, etc. In this scheme, only patients can upload their medical records on blockchain because they own them. The medical records are encrypted using attribute-based encryption. Following are the entities involved in the protocol: i) Supervision Centre: Initializes the consortium blockchain network and deploys smart contracts. ii) Record Owner (Patients): The patient encrypts their medical record and submits the encrypted medical record to the storage server. iii) Record Requester: These are the doctors who need to view the patient's medical records. Records can be accessed by initiating an access transaction. If requester attributes meet the access policy set by the owner, the requester can download the medical records.

Kiana et al. [36] published an extensive survey about the usage of blockchain in storing health records. Apart from this, there are many more works on using blockchain as a solution for storing medical records [23,56,38,2,17,40,24].

3 Preliminaries

In this section, we discuss a high-level overview of concepts that we use for building our framework.

3.1 Blockchain

Blockchain consists of three components: Data, Hash, and Hash of the previous block. A protocol similar to the blockchain was first proposed by David Chaum in 1982 in his

dissertation [14]. It was improved further by Stuart et al. in 1991 [30]. Blockchain is a decentralized peer-to-peer system in which every node has a copy of the blockchain. A block gets added to the chain if and only if it gets verified by the other nodes in the network. Each block consists of a hash of the previous block. Tampering a block is impossible because all the other blocks succeeding that block must also be changed. Since hashing can be computed within no time, blockchain uses a concept called Proof of Work (PoW) [44] to prevent tampering.

Proof of Work is a mechanism to slow down the creation of new blocks. In Bitcoin, it takes around 10 minutes to add a new block. Proof of Work (PoW) is a kind of computer puzzle given to all the participants in the network. All the participants in the network are referred to as miners. The puzzle's complexity depends on the network size and length of the chain. If more than 50% of nodes compute PoW in the network, the block gets added to the blockchain. If the adversary needs to tamper a block, he should be able to compute more than 50% of PoW (This is popularly known as 51% attack), which is not computationally possible. Computing Proof of Work requires a lot of computational resources.

To make blockchain technology more efficient, Sunny King and Scott Nadal introduced the concept of Proof of Stake in 2012 [37]. Computational resources required to compute Proof of Stake are much less than for computing Proof of Work (PoW). In Proof of Stake (PoS) all the participants are referred to validators. Each validator submits some part of their cryptocurrency as a stake. These validators are picked randomly based on the amount of stake they submit. Higher submitted stake results in higher chances of receiving validating requests. Each validator receives some incentives for validating a new block. Incorrect validation results in deducting some or all the stakes submitted by the validator.

3.2 Format Preserving Encryption

The term Format Preserving Encryption (FPE) was first coined by Voltage Securities CTO Terence Spies [5]. Prior to this, there were few works [46] where ciphertext is also in the plaintext space using DES encryption. Applications to Format Preserving Encryption (FPE) gained much attention in industries where plaintext format is to be preserved without compromising the security of plaintext data. Some applications include social security numbers (SSN), Credit Card Numbers (CCN), postal addresses, etc.

FPE aims to encrypt the plaintext with a randomized key such that the ciphertext is also of the same format. For example, if we encrypt a 16 digit Credit Card Number, the ciphertext is also 16 digit Credit Card Number (CCN). Effectively, we are just permuting the digits of plaintext to form a new ciphertext with the same format. Format Preserving Encryption (FPE) is a deterministic mode of encryption in which we give a message m from a message space M ($m \in M$), the key generated by a randomized key generation algorithm, publicly known tweak T as input. We get a ciphertext c, which is the same format as the message ($c \in M$). Since the ciphertext is obtained by permuting the digits in the plaintext, we can use block ciphers (Pseudo Random Permutation (PRP)).

FPE can be constructed by block ciphers like AES using a technique called cycle walking [7]. In this technique, we recursively encrypt the plaintext until the ciphertext reaches the format of plaintext. Bellare et al. [5] presented other methods to construct FPE schemes like Rank-then-Encipher, and Feistel network based encryption.

In rank-then-encipher mode, plaintext of size N is arbitrarily ordered as $\{s_0, s_1, \ldots, s_{N-1}\}$. This Process consists of 3 stages: 1) index i such that $s = s_i$ is found. 2) i is encrypted to an index j, using FPE algorithm. 3) Encryption of s is the message s_j.

3.3 Group Signatures

Group signatures were introduced by Chaum and Heyst [13] in 1991. The general digital signature algorithm outputs the signature and puts the signer information in the public domain. Group signatures are a type of digital signature that makes the signer anonymous. It provides anonymity property. If n parties form a group, anyone in the group can sign the message, and no other group member can identify the signer. A group consists of two entities: *group members* and *group managers*. Group members can sign the message with a group public key without revealing their identity. The group manager is a trusted party with additional trapdoor information that can reveal the signer's identity and identify malicious group members.

There are numerous applications for group signatures in various domains including vehicle safety communication [27,47,53,35], blockchain [10,20,54], military etc.

Group Signature Algorithm: A *group signature scheme* consists of four algorithms:

KeyGen(λ): This algorithm generates the group public key, the group members' secret key, and the group manager's secret key. The key generation algorithm inputs security parameters and generates a group public key (gpk), a group member's secret key sk, and a group manager's secret key ($gmsk$). If there are n members, keyGen outputs (gpk, gsk_1, gsk_2, gsk_3, ..., gsk_n, $gmsk$).

Sign(gpk, gsk_i, M): The signature algorithm is used to sign the message. Sign algorithm takes group public key (gpk), secret key of i^{th} group member (gsk_i) and message M as input and outputs a signature σ.

Verify(gpk, m, σ): The verification algorithm is used to verify the signature σ. Any group member can verify the signature. The algorithm takes group public key (gpk), message M, signature σ, and outputs a Boolean digit. If the output is 1, the signature σ is valid. Else, the signature σ is not a valid signature.

Open(gpk, $gmsk$, σ, m): This open algorithm is used to identify the signer of the signature σ. Only the group manager is authorized to use the open algorithm. The algorithm takes the group public key (gpk), the group manager secret key ($gmsk$) is assigned only to the group manager, message m, signature σ as input, and outputs the group member index i.

Properties of Group Signatures The following properties given by Bellare et al. [4] must be satisfied by group signature scheme.

1. **Correctness:** Honestly signed messages must pass the verification.

2. **Anonymity:** Other group members should be unable to identify the signer.
3. **Traceability:** Given a valid signature, the group manager should be able to trace which user issued the signature.

3.4 Secret Sharing Schemes

Secret sharing schemes are one of the essential mechanisms for safeguarding secret information [42] and have found many applications in modern cryptography's protocols such as distributed computing [6,3], secure multiparty computations [12,16], threshold cryptography [18], attribute-based encryption [26], access control [45], generalized oblivious transfer [52,55] and Byzantine agreement [49].

Secret sharing schemes were independently proposed by Blakley [8] and Shamir [51] in 1979. Both of these were *threshold secret sharing schemes* which allowed a *secret* s to be split into n *shares* which could be distributed among n *members* (or *participants*) $\mathcal{P} = \{P_1, \cdots, P_n\}$, in such a way that for some threshold t with $1 \leq t \leq n$, any group of t or more members could pool in their shares to *reconstruct* the secret. However, if the number of members in a group is less than the *threshold* t, then that group does not get any extra information about the secret. While the Shamir secret sharing scheme [51] is based on linear algebra and the standard Lagrange's interpolation, the scheme proposed by Blakley [8] is built upon the idea of finite geometries, particularly on the concept of intersection of hyper-planes.

3.5 Proactive Secret Sharing Schemes

Proactive Secret Sharing (PSS) is a type of secret sharing mechanism in which the shares of the shareholders are updated at regular time intervals to achieve long-term confidentiality of the secret and the shares. General secret sharing schemes do not provide long-term confidentiality for the shares that shareholders hold. Long-term confidentiality is essential because a mobile adversary who can breach shareholders can breach the threshold number of shareholders and obtain their shares and consequently reconstruct the secret. With PSS, it is not possible for a mobile adversary to reconstruct the secret because the shares are constantly updated after specific time, and it is impossible for an adversary to reconstruct the secret by breaching the threshold number of parties in a certain time window.

Proactive Secret Sharing (PSS) schemes were introduced by Herzberg et al. in 1995 [33], and further improvements were made by Desmedt et al [19], Wong et al. [58], Gupta et al. [28,29]. All the schemes mentioned above use private channels to communicate among parties during secret sharing, redistributing (updating shares), and reconstruction. All the private channels must be secure, and should be difficult for the adversary to eavesdrop. These private channels are Information Theoretic secure (IT-secure) and made up of OTP. Communication through IT secure channels is commercially infeasible and also leads to a large amount of traffic because private channels are established between every two parties.

In 2017, Brendel et al. [9] introduced an Efficient Proactive Secret Sharing Schemes (EPSS) in which all parties who hold secret shares are formed into clusters. In every cluster, one of the parties acts like a root node, and the remaining parties are the child

nodes of that respective root node. Private channels are established only between a cluster's root nodes and child nodes and between the root nodes of two clusters. Root nodes act like a bridge between two clusters for communicating across clusters. Child nodes only store the secret shares. Apart from private channels, they also have broadcasting channels for communicating non-private information, like sending acknowledgments after receiving valid shares.

We first describe an overview of an efficient, proactive secret sharing the scheme as presented by Brendel et al. [9]. Let $[n, m, N]$ be the set of existing shareholders where n is the total number of parties participating in the protocol, m- is the threshold number of parties required to reconstruct the secret, and N is the total number of root nodes. Initially, a client or a dealer initiates the protocol by distributing the secret k to all the participating parties using a general secret sharing scheme like threshold secret sharing scheme. Now assume $[n', m', N]$ be the new set of shareholders to which the redistribution needs to be done. For share redistribution, we categorize all the clusters into two types: i) Sender nodes and ii) Receiving nodes. The sender nodes are the nodes that redistribute the new shares to receiving nodes. Note that communication between the sender and receiver clusters happens only through root nodes.

Sender nodes apply the secret sharing scheme to their secret shares (These shares are obtained from the client or dealer in the initial step) to obtain the sub-shares. All the child nodes in the sender cluster send their sub-shares to their respective root nodes. Note that the sender cluster comprises many clusters, each with its own root node and corresponding child nodes.

After receiving the sub-shares, the root nodes of the sender cluster compute the summation of the sub-shares and sends it to the root nodes of intended receivers in the receiver cluster. The receiver root nodes compute the summation of the sub-shares received from different root nodes of the sender cluster. The receiver root nodes forward the results to their respective child nodes. All the child nodes store these new values as their new secret shares.

To reconstruct the secret, the client or dealer can simply pick the threshold number of parties who hold valid shares and reconstruct the original secret k.

4 Threat Model and Security notions

In this section, we discuss the threat model and security notions of our proposed blockchain based Format Preserving Encryption (B-FPE). Our B-FPE protocol uses group signatures to sign the transactions and proactive secret sharing techniques for updating the key shares. Following are the security notions of our protocol.

We briefly discuss about the general security notions of FPE schemes which we have used to preserve patients privacy [5]. The proof follows as the proof given in [5], relevant excerpts of which are reproduced verbatim below for easy reference.

1. **PRP Security [5]:** (based on algorithm 4:) The standard notion of PRP (Pseudorandom Permutation) security is extended to FPE schemes via a game FPE_E (E is the encryption scheme) and the corresponding adversary advantage is as follows:

$$Adv_E^{FPE}(\mathcal{A}) = 2 \cdot Pr[FPE_E^{\mathcal{A}} \Longrightarrow true] - 1$$

2. **SPI Security** [5] (based on the algorithm 5): Single Point Indistinguishably (SPI) requires that the adversary be unable to distinguish between the encryption of a single chosen message or a random range point, even when given adaptive access to a true encryption oracle. The advantage is as follows:

$$Adv_E^{SPI}(\mathcal{A}) = 2.Pr[SPI_E^{\mathcal{A}} \Longrightarrow true] - 1$$

3. **Message Privacy** [5] (based on the algorithm 6): In message privacy we are trying to measure the ability of an adversary with an encryption challenge plaintext X^* from its encryption C^*. If the encryption is randomized, we would require that the challenge ciphertext C^* is of no use in such an attack. For deterministic encryption, the intuition we aim to capture is that the adversary should do no better than it could if the encryption were ideal. In this case, the encryption oracle provides no more than the capability of testing whether a message of the adversary's choice equals the challenge message. The advantage of $\mathcal{A}$ is defined as follows:

$$Adv_E^{mp}(\mathcal{A}) = Pr[MP_E^{\mathcal{A}} \Longrightarrow true] - P_{\mathcal{A}}$$

4. **Message Recovery** [5] (based on the algorithm 7): An FPE scheme secure against message recovery is one of which an adversary is unable to recover plaintexts from ciphertext, even given an encryption oracle and a favorable distribution of plaintext, tweak (T), formats (N).

$$Adv_E^{mr}(\mathcal{A}) = Pr[MR_E^{\mathcal{A}} \Longrightarrow true] - P_{\mathcal{A}}$$

   Where $p_{\mathcal{A}} = max_s.Pr[MR_E^{\mathcal{A}} \Longrightarrow true]$ with maximum over all simulators.

Indistinguishability games for above security notions are described in A.2.

Following are the assumptions of our B-FPE protocol:

1. We assume that adversary can access the blockchain but cannot tamper the data because of its inherent property of immutability.
2. Adversary can only have access to ciphertext of the medical records and cannot deduce anything about the plaintext except their format.
3. A malicious healthcare provider can sign incorrect transactions, might not participate.
4. A malicious healthcare provider can misuse the emergency phase variant and obtain the decrypted medical records.
5. Adversary is mobile. That means, adversary can move from one node to other and can compromise the nodes. All the compromised nodes are said to be malicious nodes.
6. Malicious nodes can alter their key shares.
7. All the malicious nodes can collude and reconstruct the secret key. However, there should be at least threshold number of malicious nodes.

5 B-FPE Protocol

All healthcare providers in a city/state form a group. Each group has a group manager, which is a trusted entity. We can extend the same concept to include multiple groups

and form a supergroup. Participants in the super group are all the healthcare providers in the country. Each user/patient will be given a unique FPE key at registration to encrypt or decrypt their data. After allotting an FPE key, the healthcare provider uses a proactive secret sharing technique [9] and distributes the key to the remaining healthcare providers in the group. Patients hold their FPE keys and enter their respective keys at the healthcare provider after their visit. Healthcare providers decrypt and access patient's medical history if needed.

After the patient's diagnosis, the healthcare provider encrypts the patient's details like Name, contact number, SSN, medical prescription etc., using FPE and broadcasting it to the entire healthcare network in the city/state. The patient needs to enter their FPE key to encrypt their details. We refer to this process as a transaction. Note that patients need not visit the same healthcare provider always. The network size must be decided wisely because healthcare providers do not have high computation power to compute/solve the blockchain puzzle as part of consensus protocol (ex: Proof of Work (PoW)). Note that any efficient consensus protocol can be used.

Healthcare providers sign their transactions using group signatures. Section 6 discusses the reason for using group signatures. Healthcare providers sign the transaction using a secret key obtained from the key generation center. The network's remaining nodes (healthcare providers) can verify the signature using a group public key. Many transactions happen during a certain interval of time. We concatenate all the transactions into a block that occurred during a certain time window, say 10 minutes. This block is broadcasted to all the nodes in the network, that is, healthcare providers. The nodes (healthcare providers) in the network/group compute a consensus protocol for the new block. The block gets added to the blockchain.

There might be cases in which a patient may visit a healthcare provider in an emergency condition. In such a situation, the patient may not be able to provide their FPE key to the healthcare provider.

The healthcare provider can reconstruct the key and decrypt the block to obtain past medical records in such a way that healthcare provider remains oblivious about the Format Preserving Encryption (FPE) key. Later, Group Manager can audit the emergency cases to check the genuineness of the healthcare provider who requested the reconstruction of the FPE key. Figure 1 shows an overview of reconstruction process.

Our protocol has three stages:

1. Registration Phase.
2. Regular Phase or Emergency Phase.
3. Verification Phase.

Registration Phase: We assume that there are n healthcare providers (H_0, H_1, H_2, ..., H_{n-1}) in the city and form a group. Every group consists of two entities: i) Group Members and ii) Group Manager. All the healthcare providers are the group members and the group manager is a trusted party. In this stage, let us assume that a patient registers at some healthcare provider, say H_0 where they go for diagnosis. After consultation, the patient provides their details like name, date of birth etc. and obtains an FPE key (k). Healthcare provider H_0 uses Format Preserving Encryption (FPE) to encrypt patient details. Since the patient's data is encrypted, there is no

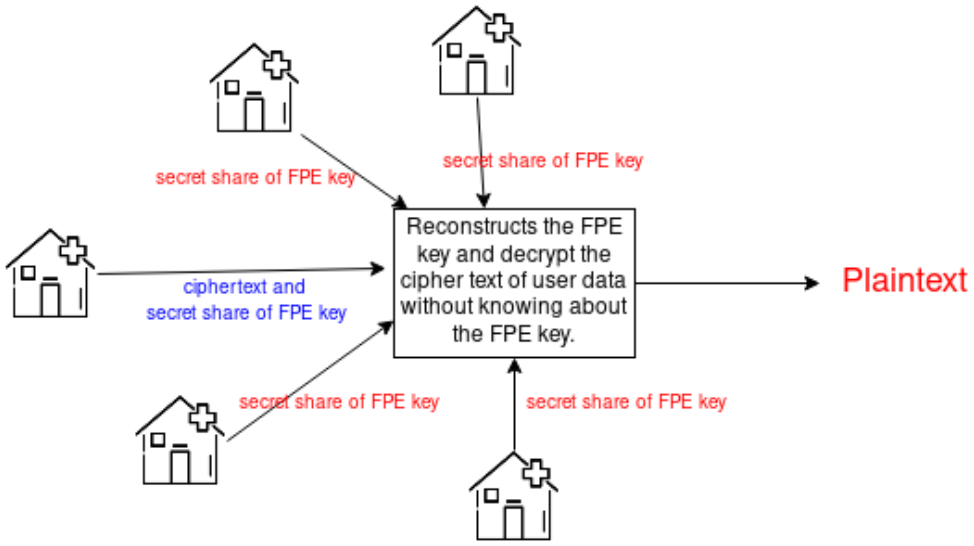

Fig. 1. Reconstruction of FPE key for decrypting ciphertext

need to anonymize the patient's data. A healthcare provider will be able to decrypt if and only if it requires. Also, the decryption happens only if a patient enters their key at the healthcare provider. For emergency phase cases, Group Manager can audit the healthcare provider who requests key reconstruction. H_0 divides FPE key k into n secret shares $(k_0, k_1, \ldots, k_{n-1})$ where n represents the total number of nodes and distributes it to all the parties in the group using a proactive secret sharing scheme as discussed in section 3.5. Since we are using proactive secret sharing, all the n secret shares $(k_0, k_1, \ldots, k_{n-1})$ are updated at frequent intervals of time. Updating the secret shares prevents a mobile adversary from stealing the secrets and reconstructing the FPE key. Algorithm 1 shows the process of the registration phase.

Algorithm 1 Registration Phase:

All the healthcare providers $H_0, H_1, \ldots, H_{n-1}$ **form a group.**

1. user/patient $\xrightarrow{\text{details}}$ H_0 ▷ Details = Name, DOB, SSN etc.
2. $k \leftarrow \{1^\lambda\}$ ▷ H_0 generates a FPE key to user.
3. $Enc_k(m) \rightarrow C$. ▷ Encrypts patient's data (m) with FPE key k
4. H_0 uses proactive secret sharing scheme (PSS) and distributes all the shares of FPE key k to all members of the group $H_0, H_1, \ldots, H_{n-1}$.

Regular Phase: In this phase, we assume that the patient visits a healthcare provider in a conscious state such that they can enter their FPE key at the healthcare

provider H_1, and the patient enters their FPE key (k). If the healthcare provider (H_1) needs to access patient's medical history, H_1 decrypts the block to check the medical history if needed, adds new medication details, and signs the transaction with its group secret key (gsk_1) and group public key (gpk). After a certain time window, say 10 minutes, all the transactions are concatenated into a block, and the remaining nodes compute the chosen consensus protocol (ex: Proof of Work) for new blocks and verify the blocks. Block gets appended to the blockchain after successful verification. Algorithm 2 shows the detailed protocol of the regular phase.

Emergency Phase: In this case, we assume that a patient visits a healthcare provider in a state where they cannot share their FPE key directly with the healthcare provider, that is, a patient visits a healthcare provider in an emergency condition. Since the patient cannot share their FPE key (k), healthcare provider H_2 reconstructs the FPE key k by obtaining secret shares from all the healthcare providers $(k_0, k_1, \ldots, k_{n-1})$. Group signatures are used to sign and verify the the authenticity of the secret shares.

H_2 uses this FPE key to decrypt the patient's medical records (m) ($Dec_k(C)$) where C is the encrypted medical record. We assume that we have a decryption scheme where H_2 decrypts the block in such a way that H_2 gets the decrypted medical records of the patient without knowing anything about FPE key (k) (refer to figure 1). After diagnosis, we assume that the patient will be in a state to share their FPE key (K). H_2 adds new medication details and signs the transaction with its group secret key (gsk_2) and group public key (gpk). After a certain time window, all the transactions are concatenated into a block, and the remaining nodes compute the chosen consensus protocol (ex: Proof of Work) for the new block and append it to the blockchain. Algorithm 3 shows the process of the emergency phase.

Algorithm 2 Regular Phase

if user/Patient does not hold FPE key: **then**
1. Registration Phase()

else if User/Patient holds the key **then**
1. Patient $\xrightarrow{k}$ H_1.
2. $H_1 : Dec_k(C)$ ▷ H_1 decrypts and retrieves patient's medical records.
3. H_1 issues a new transaction t using group secret key gsk_1 and group public key g_{pk}. sign$(t, gsk_1, g_{pk}) \rightarrow \sigma$.
4. $H - \{H_1\}$ verify$(t, \sigma) \rightarrow 1/0$. ▷ other group members verify the signature
5. Concatenates $t \parallel t_1 \parallel \cdots \parallel t_n$. ▷ Concatenates all the transactions
6. $H - \{H_1\}$ computes consensus protocol and adds the block to the blockchain.

end if

Verification phase: Any group member can verify the transaction using group public key (gpk). Algorithm takes group public key (gpk), transaction (t), signature (σ), and outputs a boolean digit. Since group signatures have anonymity property,

Algorithm 3 Emergency Phase

1. $H_2 \leftarrow \{k_0, k_1, \ldots, k_{n-1}\}$ ▷ obtains all the shares from remaining nodes.
2. $Dec_k(C)$ without learning k.
3. $Enc_k(m) \rightarrow C$. ▷ Encrypts patient's data (m) with FPE key k
5. H_2 issues a new transaction t using group secret key gsk_2. sign$(t, gsk_2) \rightarrow \sigma$.
6. $H - \{H_2\}$ verify$(t, \sigma) \rightarrow 1/0$. ▷ other group members verify the signature
7. Concatenates $t \parallel t_1 \parallel \cdots \parallel t_n$.
8. $H - \{H_2\}$ computes consensus protocol and adds the block to the blockchain.

the remaining group members (healthcare providers) cannot identify the signed party. Thus, other healthcare providers cannot identify previous healthcare providers that the patient visited. It is only the group manager who can identify the signed party. Malicious parties who sign incorrect transactions and share incorrect secret shares of FPE key (k) can be identified by the group manager using the open algorithm as discussed in section 3.3

6 Security

In this section, we discuss the security of our B-FPE protocol.

We use Format Preserving Encryption to encrypt all the details of the patient without changing the format of their details. Since the medical records of patients are sensitive information, we encrypt them with FPE and store them in the blockchain. Since patients hold their FPE key, the patients can decrypt their medical records.

A malicious healthcare provider can request the other nodes for their key shares and obtain the plaintext of the medical records. Although the malicious healthcare provider cannot know the FPE key, it can access decrypted medical records. This can be mitigated by observing the number of key reconstruction requests the healthcare provider raises. If the number of requests is high, the group manager can identify the malicious healthcare provider by opening the signed request using the open algorithm as mentioned in section 3.3.

All healthcare providers use group signatures to sign their transactions. Recall that group signature have the properties of anonymity, traceability, and correctness (ref. Section 4). Any healthcare provider who signs the transactions will remain anonymous to all the group members. Through this anonymity property, other healthcare providers in the network cannot deduce the number of transactions/number of patients visiting that particular healthcare provider. The remaining group members might not cooperate in computing the consensus protocol. To mitigate such problems, we use group signatures. Since group signatures include a group manager who is a trusted organization can help find malicious parties if any group member behaves maliciously.

Following are the security proofs of our group signature schemes.

Theorem 1. *(Anonymity property [4]) Using group signatures for transactions makes other parties in the network oblivious about the signer.*

Proof. In group signatures, participants use a group public key to sign their transactions. Every participant in the group uses the same public key to sign their transactions. Hence, neither the outside party nor the group member can identify the signer.

Theorem 2. *(Traceability [4]) Any malicious party in the network validates an incorrect transaction can be identified and penalized.*

Proof. Although group signatures hold anonymity, the group manager can identify malicious parties in the group. The group manager contains trapdoor information, also known as the group master secret key, to identify the signer.

Theorem 3. *(Correctness [4]) Transactions of all the honest parties must be validated correctly.*

Proof. Every transaction of honest parties must pass the verification algorithm. The output of the verify algorithm must be 1.

Theorem 4. *(Accountability) A malicious healthcare provider cannot misuse the emergency phase and decrypt the medical records.*

Proof. Since the request raised by the healthcare provider is signed using group signatures. The remaining nodes in the network might not identify the healthcare provider, but the group manager can identify the signer of the request and penalize the malicious node.

The proactive Secret Sharing technique is used in the emergency phase of our protocol. This proactive secret sharing updates the secret shares of FPE keys held at other healthcare providers in the group at constant intervals. This updating prevents a mobile adversary from reconstructing the key by breaching the threshold number of parties.

The ciphertext of a patient's medical records is decrypted only if requesting party receives a threshold number of shares. This scheme also prevents malicious parties in the group from reconstructing the FPE key. Interoperability between healthcare providers is achieved only if a threshold number of parties are honest.

Theorem 5. *(Integrity and Availability [9]) The Efficient Proactive Secret Sharing (EPSS) Redistribution protocol assures the integrity and availability of the secret k in the presence of fewer than threshold dishonest child nodes in [n, m, N] and [n',m',N'], respectively.*

Proof. Protocol terminates successfully if shares are stored at threshold (m'). An adversary cannot store more than m'-1 invalid shares. Such attempts will be detected, and those invalid shares get rejected. In between share redistributions the upper bound on the number of compromised current shareholders assures that there remain a threshold number of valid shares for reconstruction and redistribution. This assures the integrity and availability of secret k.

Theorem 6. *(Long-term confidentiality [9]) Let at most $m-1$ child nodes in B_u and up to $m'-2$ child nodes $[n', m', N']$ be dishonest. Furthermore, assume that cluster with two honest receiver nodes is under control of an honest receiver root node. The long-term confidentiality of secret k is assured during redistribution.*

Proof. Let us assume that an adversary has compromised $m' - 1$ nodes in $[n', m', N']$ and holds the unencrypted shares of all $m' - 1$ nodes. To reconstruct the secret, the adversary needs only one extra share. The the goal is to reveal the encrypted share of an honest receiver by executing complaint resolution as discussed in [9].

Adversarial nodes distribute invalid shares to the honest parties. Since fewer than $2m' - 1$ nodes were able to finish the protocol, complaint resolution is initiated, and one of the complaints, say node $\tilde{j}$ is chosen for resolution.

Node $\tilde{j}$ will henceforth reveal its received share pairs in encrypted form along with proof of correct decryption. The jury finds the complaint valid. The encrypted partial share revealed by root node $\tilde{J}$ shows that $\tilde{J}$ has computed the correct share pair, and therefore mistake must have happened earlier. All the sender nodes reveal their sub-shares on the broadcast channel and identify the malicious participant.

The adversary has gained knowledge of the encrypted sub-shares of honest $i \in B_u$, which it was missing to compute the encrypted share $s'_{\tilde{j}}$ of $\tilde{j}$. The adversary now holds $m' - 1$ unencrypted share plus one valid encrypted share. With this , adversary can easily break the long-term confidentiality. of secret k.

Now assume that there exists a dishonest receiver root $\bar{J}$ with at least two child node $\tilde{j_1}$ and $\tilde{j_2}$ in its cluster. Then the long-term confidentiality of secret k is broken in the presence of $m' - 2$ dishonest receiver nodes. It is assumed that all the dishonest nodes act in collusion. All nodes belong to $j \in [n', m', N']$ can receive valid shares and redistribute process will terminate successfully. Since $\bar{j}$ computes the encrypted valid shares of $\tilde{j_1}$ and $\tilde{j_2}$, it can decrypt these two values once the computational assumption is broken. Combined with the $m' - 2$ invalid shares it already has at its disposal, the secret can be reconstructed.

Theorem 7. *(Computational Confidentiality [9]). Assume that at most $m - 1$ child nodes in $[n, m, N]$ and at most $m'-2$ in $[n', m', N']$ are dishonest. Then the computational confidentiality of secret k*

Proof. Under the assumptions, the adversary knows $m - 1$ and $m' - 2$ shares in plaintext. Since all data going through the root node is encrypted for the respective receiver node in $[n', m', N']$ this information is useless to a computationally bounded adversary. Therefore, the adversary cannot gain the missing information to reconstruct secret k.

7 Conclusion and Future work

In this paper, we have designed a new blockchain protocol (B-FPE) for maintaining patients' medical records. We encrypt patients' medical records in the blockchain using Format Preserving Encryption (FPE). All the transactions of the blockchain are signed using group signatures. We considered two scenarios: i) A patient directly walks into the healthcare provider, enters their FPE key, and consult the doctor. Medical prescription is concatenated in the blockchain. ii) A patient is not in a position to share their FPE key with the healthcare provider. In this case, the healthcare provider reconstructs the FPE key by receiving the threshold number of secret shares and decrypts the medical records without knowing anything about the FPE key.

Our future work would be to build a crypto scheme that addresses the assumption mentioned above.

References

1. Ali, O., Ally, M., Dwivedi, Y., et al.: The state of play of blockchain technology in the financial services sector: A systematic literature review. International Journal of Information Management **54**, 102199 (2020)
2. Azaria, A., Ekblaw, A., Vieira, T., Lippman, A.: Medrec: Using blockchain for medical data access and permission management. In: 2016 2nd international conference on open and big data (OBD). pp. 25–30. IEEE (2016)
3. Beimel, A.: Secret-sharing schemes: A survey. In: IWCC. LNCS, vol. 6639, pp. 11–46. Springer (2011)
4. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In: International conference on the theory and applications of cryptographic techniques. pp. 614–629. Springer (2003)
5. Bellare, M., Ristenpart, T., Rogaway, P., Stegers, T.: Format-preserving encryption. In: International workshop on selected areas in cryptography. pp. 295–312. Springer (2009)
6. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: STOC. pp. 1–10. ACM (1988)
7. Black, J., Rogaway, P.: Ciphers with arbitrary finite domains. In: Cryptographers' Track at the RSA conference. pp. 114–130. Springer (2002)
8. Blakley, G.R.: Safeguarding cryptographic keys. In: AFIPS. pp. 313–317 (1979)
9. Brendel, J., Demirel, D.: Efficient proactive secret sharing. In: 2016 14th Annual Conference on Privacy, Security and Trust (PST). pp. 543–550. IEEE (2016)
10. Cao, Y., Li, Y., Sun, Y., Wang, S.: Decentralized group signature scheme based on blockchain. In: 2019 International Conference on Communications, Information System and Computer Engineering (CISCE). pp. 566–569. IEEE (2019)
11. Castelló Ferrer, E.: The blockchain: a new framework for robotic swarm systems. In: Proceedings of the future technologies conference. pp. 1037–1058. Springer (2018)
12. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (extended abstract). In: STOC. pp. 11–19. ACM (1988)
13. Chaum, D., Heyst, E.v.: Group signatures. In: Workshop on the Theory and Application of of Cryptographic Techniques. pp. 257–265. Springer (1991)
14. Chaum, D.L.: Computer Systems established, maintained and trusted by mutually suspicious groups. Electronics Research Laboratory, University of California (1979)
15. Cole, R., Stevenson, M., Aitken, J.: Blockchain technology: implications for operations and supply chain management. Supply Chain Management: An International Journal (2019)
16. Cramer, R., Damgård, I., Maurer, U.: General secure multi-party computation from any linear secret-sharing scheme. In: EUROCRYPT. LNCS, vol. 1807, pp. 316–334. Springer (2000)
17. Dagher, G.G., Mohler, J., Milojkovic, M., Marella, P.B.: Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustainable cities and society **39**, 283–297 (2018)
18. Desmedt, Y., Frankel, Y.: Shared generation of authenticators and signatures (extended abstract). In: CRYPTO. LNCS, vol. 576, pp. 457–469. Springer (1991)

19. Desmedt, Y., Jajodia, S.: Redistributing secret shares to new access structures and its applications. Tech. rep., Citeseer (1997)
20. Devidas, S., Rao YV, S., Rekha, N.R.: A decentralized group signature scheme for privacy protection in a blockchain. International Journal of Applied Mathematics and Computer Science **31**(2) (2021)
21. Dworkin, M., et al.: Recommendation for block cipher modes of operation: methods for format-preserving encryption. NIST Special Publication **800**, 38G (2016)
22. Dworkin, M., et al.: Recommendation for block cipher modes of operation: methods for format-preserving encryption. NIST Special Publication Revision 1 **800**, 38G (2019)
23. Eberhardt, J., Tai, S.: On or off the blockchain? insights on off-chaining computation and data. In: European Conference on Service-Oriented and Cloud Computing. pp. 3–15. Springer (2017)
24. Fan, K., Wang, S., Ren, Y., Li, H., Yang, Y.: Medblock: Efficient and secure medical data sharing via blockchain. Journal of medical systems **42**(8), 1–11 (2018)
25. Gordon, W.J., Catalini, C.: Blockchain technology for healthcare: facilitating the transition to patient-driven interoperability. Computational and structural biotechnology journal **16**, 224–230 (2018)
26. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security. pp. 89–98. ACM (2006)
27. Group, I.P.W., et al.: Vsc project. Dedicated short range communications (DSRC) (2003)
28. Gupta, V., Gopinath, K.: An extended verifiable secret redistribution protocol for archival systems. In: First International Conference on Availability, Reliability and Security (ARES'06). pp. 8–pp. IEEE (2006)
29. Gupta, V., Gopinath, K.: G_ {its}ˆ 2 vsr: An information theoretical secure verifiable secret redistribution protocol for long-term archival storage. In: Fourth International IEEE Security in Storage Workshop. pp. 22–33. IEEE (2007)
30. Haber, S., Stornetta, W.S.: How to time-stamp a digital document. In: Conference on the Theory and Application of Cryptography. pp. 437–455. Springer (1990)
31. Hang, L., Kim, B., Kim, K., Kim, D.: A permissioned blockchain-based clinical trial service platform to improve trial data transparency. BioMed Research International **2021** (2021)
32. Hardin, T., Kotz, D.: Amanuensis: Information provenance for health-data systems. Information Processing & Management **58**(2), 102460 (2021)
33. Herzberg, A., Jarecki, S., Krawczyk, H., Yung, M.: Proactive secret sharing or: How to cope with perpetual leakage. In: annual international cryptology conference. pp. 339–352. Springer (1995)
34. Hyperledger: Hyperledger Fabric. https://www.hyperledger.org/ (2016), last accessed: 2022-09-09
35. Islam, S.H., Obaidat, M.S., Vijayakumar, P., Abdulhay, E., Li, F., Reddy, M.K.C.: A robust and efficient password-based conditional privacy preserving authentication and group-key agreement protocol for vanets. Future Generation Computer Systems **84**, 216–227 (2018)
36. Kiania, K., Jameii, S.M., Rahmani, A.M.: Blockchain-based privacy and security preserving in electronic health: a systematic review. Multimedia Tools and Applications pp. 1–27 (2023)
37. King, S., Nadal, S.: PPCoin: Peer-to-peer crypto-currency with proof-of-stake. https://cryptorating.eu/whitepapers/Peercoin/peercoin-paper.pdf (2012), accessed: 2022-09-09

38. Kuo, T., Kim, H., Ohno-Machado, L.: Blockchain distributed ledger technologies for biomedical and health care applications. J. Am. Medical Informatics Assoc. **24**(6), 1211–1220 (2017). https://doi.org/10.1093/jamia/ocx068, https://doi.org/10.1093/jamia/ocx068
39. Landi, H.: Healthcare data breaches hit all-time high in 2021, impacting 45m people. https://www.fiercehealthcare.com/health-tech/healthcare-data-breaches-hit-all-time-high-2021-impacting-45m-people (2022), accessed: 2022-09-09
40. Li, H., Zhu, L., Shen, M., Gao, F., Tao, X., Liu, S.: Blockchain-based data preservation system for medical data. Journal of medical systems **42**(8), 1–13 (2018)
41. Madhwal, Y., Panfilov, P.B.: Blockchain and supply chain management: Aircrafts' parts' business case. Annals of DAAAM & Proceedings **28**, 1051–1056 (2017)
42. Mehta, S., Saraswat, V., Sen, S.: Secret sharing using near-MDS codes. In: Codes, Cryptology, and Information Security (C2SI 2019). LNCS, vol. 11445, pp. 195–214. Springer (2019)
43. Mukherji, A., Ganguli, N.: Efficient and scalable electronic health record management using permissioned blockchain technology. In: 2020 4th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech). pp. 1–6. IEEE (2020)
44. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review p. 21260 (2008)
45. Naor, M., Wool, A.: Access control and signatures via quorum secret sharing. In: ACM Conference on Computer and Communications Security. pp. 157–168. ACM (1996)
46. National Bureau of Standards: Implementing and using the NBS data encryption standard. Tech. rep., Federal Information Processing Standards Publication (FIPS PUB) 46 (1981)
47. Park, M.H., Gwon, G.P., Seo, S.W., Jeong, H.Y.: Rsu-based distributed key management (rdkm) for secure vehicular multicast communications. IEEE journal on selected areas in communications **29**(3), 644–658 (2011)
48. Polyviou, A., Velanas, P., Soldatos, J.: Blockchain technology: financial sector applications beyond cryptocurrencies. Multidisciplinary Digital Publishing Institute Proceedings **28**(1), 7 (2019)
49. Rabin, M.: Randomized byzantine generals. In: FOCS. pp. 403–409. IEEE Computer Society (1983)
50. Rowan, S., Clear, M., Gerla, M., Huggard, M., Goldrick, C.M.: Securing vehicle to vehicle communications using blockchain through visible light and acoustic side-channels. arXiv preprint arXiv:1704.02553 (2017)
51. Shamir, A.: How to share a secret. Communications of the ACM **22**(11), 612–613 (1979)
52. Shankar, B., Srinathan, K., Rangan, C.P.: Alternative protocols for generalized oblivious transfer. In: ICDCN. LNCS, vol. 4904, pp. 304–309. Springer (2008)
53. Sun, Y., Feng, Z., Hu, Q., Su, J.: An efficient distributed key management scheme for group-signature based anonymous authentication in vanet. Security and Communication Networks **5**(1), 79–86 (2012)
54. Tang, F., Feng, Z., Gong, Q., Huang, Y., Huang, D.: Privacy-preserving scheme in the blockchain based on group signature with multiple managers. Security and Communication Networks **2021** (2021)
55. Tassa, T.: Generalized oblivious transfer by secret sharing. Design, Codes and Cryptography **58**(1), 11–21 (2011)
56. Vujičić, D., Jagodić, D., Ranđić, S.: Blockchain technology, bitcoin, and ethereum: A brief overview. In: 2018 17th international symposium infoteh-jahorina (infoteh). pp. 1–6. IEEE (2018)

57. Wamba, S.F., Queiroz, M.M.: Blockchain in the operations and supply chain management: Benefits, challenges and future research opportunities (2020)
58. Wong, T.M., Wang, C., Wing, J.M.: Verifiable secret redistribution for archive systems. In: First International IEEE Security in Storage Workshop, 2002. Proceedings. pp. 94–105. IEEE (2002)
59. Xiao, Y., Xu, B., Jiang, W., Wu, Y., et al.: The healthchain blockchain for electronic health records: development study. Journal of Medical Internet Research **23**(1), e13556 (2021)
60. Yoo, S.: Blockchain based financial case analysis and its implications. Asia Pacific Journal of Innovation and Entrepreneurship (2017)
61. Zhang, D., Wang, S., Zhang, Y., Zhang, Q., Zhang, Y.: A secure and privacy-preserving medical data sharing via consortium blockchain. Security and Communication Networks **2022** (2022)

A Appendix

A.1 NIST approved FPE schemes

Initially, three modes of FPE: FF1, FF2, FF3 were submitted to NIST. These modes are referred as FFX mode of Format Preserving Encryption (FPE). NIST approved FF-1 and FF-3 as standards in 2016 [21]. Later, it was found that FF-3 is also vulnerable to cryptanalysis and thus the FF-3 mode was revoked. NIST published a revised version of FF3 as FF3-1 2019 [22].

FF3-1 uses Feistel network-based method to encrypt the plaintext. Initially, plaintext is divided into two parts. That is, left part and right part. Tweak space is also divided into two parts T_L and T_R. As per NIST report [22] FF-1 performs 8 rounds and FF3-1 performs 10 rounds in Feistel network. For more details, please refer NIST guidelines [22].

A.2 FPE security notions

This section contains the indistinguishable game algorithms that are used to explain the security notions as discussed in section 4.

Algorithm 4 Game PRP_E [5]

Initialize
$b \xleftarrow{\$} \{0,1\}; K \xleftarrow{\$} \mathcal{K}$
for $(N,T) \in \mathcal{N} \times \mathcal{T}$ ▷ N is the format and T is the tweak
do $\pi_{N,T} \xleftarrow{\$} \text{Perm}(\mathcal{X}_N)$
Enc(N,T,X)
if b=1 then return $E_K^{N,T}(X)$
if b=0 then return $\pi_{N,T}(X)$
Finalize(b')
return (b = b')

Algorithm 5 Game SPI_E[5]

Initialize
$b \xleftarrow{\$} \{0,1\}; K \xleftarrow{\$} \mathcal{K}$
Enc(N,T,X)
if $(N,T,X) \in S$ then return $\perp$
$S \xleftarrow{\cup} (N,T,X)$
return $E_K^{N,T}(X)$
TEST(N^*,T^*,X^*)
if $(N^*,T^*,X^*) \in S$ then return $\perp$
$S \xleftarrow{\cup} (N^*,T^*,X^*)$
if b=1 then:
$Y^* \leftarrow E_K^{N^*,T^*}(X^*)$
else $Y^* \xleftarrow{\$} \mathcal{X}_{N^*}$
return Y^*
Finalize(b')
return (b = b')

Algorithm 6 Game MP_E [5]

Initialize
$K \xleftarrow{\$} \mathcal{K}$
$(N^*,T^*,X^*) \xleftarrow{\$} \mathcal{A}(dist)$
$Y^* \leftarrow E_K^{N^*,T^*}(X^*)$
return (N^*,T^*)
Enc(N,T,X)
return $E_K^{N,T}(X)$
Eq(X)
return $(X = X^*)$
Test
return Y^*
Finalize(Z)
return $(Z = \mathcal{A}(func, X^*))$

Algorithm 7 Game MR_E[5]

Initialize
$K \xleftarrow{\$} \mathcal{K}$
$(N^*,T^*,X^*) \xleftarrow{\$} \mathcal{A}(dist)$
$Y^* \leftarrow E_K^{N^*,T^*}(X^*)$
return (N^*,T^*)
Enc(N,T,X)
return $E_K^{N,T}(X)$
Eq(X)
return $(X = X^*)$
Test
return Y^*
Finalize(X)
return $(X = X^*)$