Blockchain-Envisioned Provably Secure Multivariate Identity-Based Multi-Signature Scheme for Internet of Vehicles Environment

Vikas Srivastava, Sumit Kumar Debnath, Basudeb Bera, Ashok Kumar Das, *Senior Member, IEEE*, Youngho Park, *Member, IEEE*, and Pascal Lorenz, *Senior Member, IEEE*

***Abstract*—The deployed vehicles in an Internet of Vehicles (IoV) can take intelligent decisions by means of exchanging the real-time traffic-related information between the vehicles and IoV infrastructures. This further reduces the probability of the traffic jams and accidents. However, the insecure (public) communication among the various entities in IoV makes various security threats and attacks that can be launched by passive/active adversaries present in the network. In view of this context, there is a need of an efficient cryptographic primitive which can produce single compact signature. A multi-signature scheme (MSS) empowers a collection of signers to conjointly sign a given message using a single compact signature that can be verified by any verifier. Herein, we put forward a new identity-based multivariate MSS, namely MV-MSS, which is built on top of the intractability of multivariate-quadratic (MQ) problem. The fact is that multivariate public key cryptosystem provides fast, post-quantum safe and efficient primitives, which makes it the front runner candidate among the post-quantum cryptographic candidates. MV-MSS is proven to be secure in the existential unforgeability under chosen-message and chosen identity attack model if solving the MQ problem is NP-hard. We then incorporate the designed MV-MSS in IoV application where the leader (cluster head) selected from a group of vehicles in a dynamic cluster forms the multi-signatures on the messages securely received from its member vehicles. Later, the messages along with their multi-signatures are forwarded to the nearby road-side unit (RSU) of the cluster head, which are then forwarded to a cloud server in the blockchain center maintained by a Peer-to-Peer (P2P) cloud servers network. In this way, the messages and their signatures considered as transactions are put in blocks and added into a public blockchain with the help of consensus algorithm. A comparative study among the proposed MV-MSS and other existing schemes shows that MV-MSS is efficient and secure as compared to other schemes. Finally, a blockchain implementation through simulation study has been performed to show its practical use in IoV application.**

***Index Terms*—Internet of Vehicles (IoV), identity-based cryptography, multi-signature, Blockchain, consensus, security.**

Manuscript received 29 January 2022; revised 13 April 2022; accepted 18 May 2022. Date of publication 23 May 2022; date of current version 19 September 2022. This work was supported in part by the National Research Foundation of Korea (NRF) through the Basic Science Research Program, funded by the Ministry of Education under Grant 2020R1I1A3058605, and in part by the Blockchain Project under the Ripple Centre of Excellence (CoE) Scheme, CoE in Blockchain, IIIT Hyderabad, India under Grant IIIT/R&D Office/Internal Projects/001/2019. The review of this article was coordinated by Prof. J. Ben-Othman. *(Corresponding authors: Youngho Park; Pascal Lorenz.)*

Vikas Srivastava and Sumit Kumar Debnath are with the Department of Mathematics, National Institute of Technology, Jamshedpur 831 014, India (e-mail: vikas.math123@gmail.com; sdebnath.math@nitjsr.ac.in).

Basudeb Bera is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India (e-mail: basudeb.bera@research.iiit.ac.in).

Ashok Kumar Das is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India, and also with the Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Suffolk, VA 23435 USA (e-mail: shok.das@iiit.ac.in).

Youngho Park is with the School of Electronics Engineering, Kyungpook National University, Daegu 41566, Korea (e-mail: parkyh@knu.ac.kr).

Pascal Lorenz is with the University of Haute Alsace, 68100 Mulhouse, France (e-mail: lorenz@ieee.org).

Digital Object Identifier 10.1109/TVT.2022.3176755

I. Introduction

In VEHICULAR AdHoc networks (VANETs), each deployed vehicle typically communicates with other vehicles by broadcasting the messages. On the other side, Internet of Vehicles (IoV) comprises of intelligent vehicles that are equipped with smart Internet of Things (IoT) devices, with sharpened processing capabilities, heightened communication technologies as well as easy connectivity to the Internet, or to other vehicles directly or indirectly in order to support extended services for large applications unlike VANETs. As a results, such new advancements not only allow intelligent vehicles to communicate with its other vehicles, but also to collaborate the vehicles with infrastructure and Internet by exchanging messages as well. With the increased population and boost in the number of vehicles, IoV has now become one of the most stretched incentives in today's world [1], [2].

The major issue in IoV that can lead to disaster is misleading data or various attacks on data or even on the devices in the network. In IoV, such types of issues can lead to reduce the quality of human lives. There can be several attacks on IoV, such as replay, man-in-the-middle, impersonation and privileged-insider attacks, which will not only cause harm to the vehicle drivers or consumers, but also to the whole industrial business as well. This susceptibility of the paradigm requires to consider the security aspects, such integrity, confidentiality and authentication [3], [4], [5], [6].

Multiparty transactions require multi-signatures by involved entities in a network to validate the exchange messages before being written on the public ledger, for example in the public blockchain. Recently, the blockchain technology has been applied in many domains ranging from smart grid, smart city, healthcare, Internet of Drones (IoD), intelligent transport systems and so on apart from the traditional bitcoin and crypto-currencies applications [7]–[13] due to the immutability, decentralization and transparency of the blockchain technlogy. In addition, the blockchain-based solutions for IoV have shown the significant advancements in the literature, such as in the domains of “coalition game and blockchain-based optimal data pricing” [14], “secure crowdsensing” [15], “security of Internet of Energy and electric vehicle interface” [16], and “spectrum sensing” [17].

Given the state of affairs, a multi-signature scheme (MSS) is an efficient, logical, and cost-effective primitive that addresses the issue at hand by providing a method to produce compact aggregate signature while ensuring confidentiality and secrecy. Since the blocks in Blockchain are duplicated over a distributed network, so the signature size of MSS is needed to be as condensed as possible. The first identity-based MSS (IB-MSS) scheme appeared in the work by Ramzan and Gentry [18]. The scheme is provable secure in the random oracle model (ROM) under the assumption that solving gap Diffie-Hellman (GDH) problem is computational hard. It allows a set of signers, each having their own set of secret key and public key pair, to jointly produce a short and compact single aggregate signature on a message.

In the groundbreaking work [19], Shor pointed out that classical schemes built on the intractability of the number-theoretic problems would fall under attacks by a quantum computer. Thus, once efficient quantum computers come into the picture, the classical interpretation of soundness and security of cryptographic primitives may not encapsulate the right notion of security. In an endeavor, to ensure the privacy and security of cryptographic applications and to tackle the challenge brought by quantum algorithms, efforts are being taken to find an efficient and robust alternative which can replace these classical schemes.

Multivariate public-key cryptosystem (MPKC) appears to be at the forefront among the post-quantum cryptography candidates. A system of multivariate polynomials works as a public key in an MPKC. The security of MPKC is based on the fact that finding a solution to a system of a random quadratic multivariate polynomial is NP-hard [20]. Unlike number-theoretic problems, which form the base of classical schemes, the MQ problem is conjectured to withstand quantum attacks. Upto this date, there does not exist any quantum algorithm that can break the MQ problem in polynomial time. Over the last decade, many MSS schemes [21]–[29] have been emerged. However, there is no construction of multivariate MSS in the current state of art. This indicates the requirement of designing a secure and efficient multivariate MSS.

Hou *et al.* [30] designed a multi-signature scheme in which a group of parties can collaboratively sign a message in order to create a compact united (joint) signature in vehicular networks. Their scheme helps in reducing the usage of the multiple trusted certification authorities (CAs) model for the storage of the certificates given to the vehicles and road-side unites (RSUs), and also for reducing the computational overhead for RSUs in vehicle-to-infrastructure (V2I) communication. They shown that their scheme may be practical for vehicular networks that can enhance security with low computation and storage costs.

Saha *et al.* [31] proposed a post-quantum decentralization method in the blockchain, which is based on the lattices with polynomials. It uses the “identity-based encryption (IBE)” and “aggregate signatures for the consensus” to assure suitability as well as efficiency in post-quantum blockchain applications.

Cai *et al.* [32] proposed a “quantum blockchain framework” that can improve the blockchain quantum resistance property. In their framework, there are multiple traders who can perform quantum signing as well as verification for completing a multi-party transaction. In this line, they suggested a “quantum blind multi-signature algorithm” including the four phases: a) initialization, b) signing, c) verification, and d) implementation. They also employed a blind message in a multi-party business that can help in protecting (private) secret information.

Jiao and Xiang [33] designed a “lattice-based ring signature” by combining both the lattice-based cryptography (LBC) and a ring signature in VANETs. In their proposed ring signature scheme, the unforgeability is reduced to the “small integer solution (SIS) on the lattice” which is a difficult assumption under the quantum-era attacks. Their scheme was utilized in VANETs environment which offers unconditional anonymity to the deployed vehicles.

A. Motivation

Certification by multiple parties is required to validate IoV-related exchanged information, such as data related to traffic, road conditions, road accidental information like accident detection and notification messages, before the information is written into the public blockchain. Given such circumstances, there is a need for an efficient and effective cryptographic primitive which can produce compact signatures. A multi-signatures scheme (MSS) seems to be the natural choice to address this issue because it empowers a collection of signers to conjointly sign a given message using a single compact signature that any user can verify. Almost all of the existing MSS rely upon the hardness assumption of discrete logarithm or number factorization. However, these MSS will become useless in the future due to the possibility of attack by quantum computers. In order to provide a smooth sailing into a world, where a quantum computer may become a reality, we need for transition to MSS that can offer post-quantum security. Moreover, the use of the blockchain technology provides immutability, transparency and decentralization of the IoV-related information that is put into the blockchain.

B. Research Contributions

The following are the major research contributions in this work:

- We introduce the first multivariate based MSS, MV-MSS which provides security against the threat of quantum computers since it hinges on the intractability assumption of MQ problem. We take advantage of a secure signature scheme built on MPKC together with a 3-pass identification protocol of [34] as the fundamental blocks of MV-MSS. The MV-MSS consists of five algorithms, namely (i) MV-MSS.Setup, (ii) MV-MSS.KeyGen, (iii) MV-MSS.Sign, (iv) MV-MSS.KeyAgg, (v) MV-MSS.Verify. Secret key generator (SKG) on input of security parameter η, runs MV-MSS.Setup and produces public key pk and master secret key MSK. In the next step, SKG executes $\mathsf{MV\text{-}MSS.KeyGen}$ to generate secret key $\mathsf{sk}_{\mathsf{Id}}$ of a user U with identity Id with $\mathsf{Id} \in \{0,1\}^*$. Given a message msg and a set $S = \{U_1, \ldots, U_M\}$ of signers with secret keys $D = \{\mathsf{sk}_{\mathsf{Id}1}, \ldots, \mathsf{sk}_{\mathsf{Id}M}\}$, a leader L (chosen from the set S) interacts with other members of S to produce the multi-signature σ on the message msg, where $\mathsf{sk}_{\mathsf{Id}i}$ is the secret key of user U_i. In the following, a message signature pair (msg, σ) is verified by a verifier using $\mathsf{MV\text{-}MSS.Verify}$ by making use of the aggregate public key apk which is generated by running $\mathsf{MV\text{-}MSS.KeyAgg}$ on the users' public keys and pk as inputs.

- Next, we incorporate the designed MV-MSS in a blockchain-enabled IoV environment, the multi-signature on a message is validated by the nearby RSU of a cluster head acting as a leader of a dynamically formed cluster of vehicles, and then by a cloud server in the P2P cloud servers network before taking account of a transaction in a created block. Once the generated blocks are mined using a voting-based consensus algorithm, such as "Practical Byzantine Fault Tolerance (PBFT)" [35], the messages along their multi-signatures can be verified by any verifier before considering them into account for other purposes, like Big data analytics. Furthermore, our scheme is also optimally suited for cryptocurrency exchanges where more than one signature is required to carry out transactions in the blockchain.
- The proposed MV-MSS produces single compact signature, while ensuring confidentiality and secrecy. MV-MSS is proven to be secure in the model "existential unforgeability under chosen-message and chosen identity attack" if solving the MQ problem is NP-hard. Thus, MV-MSS belongs to the family of MPKCs, and hence, it is naturally very efficient and only requires computing field multiplications and additions.
- Finally, the blockchain based implementation on the proposed scheme shows its practical application in IoV scenario.

C. Paper Outline

The paper is structured in the following manner. Two models (network and threat) related to the blockchain-based scheme in IoV environment are given in Section II. In Section III, the preliminaries are contained. We describe our proposed MV-MSS in Section IV. In Section V, we show how to incorporate the proposed MV-MSS in blockchain-based IoV applications. Security analysis as well as efficiency analysis are given in Section VI and Section VII, respectively. In Section VII-B, we discuss the simulation results for blockchain part of the proposed scheme. Finally, the conclusion is provided in Section VIII.

II. System Models

In this section, we give two models (network and threat) for discussion and analysis of the proposed scheme (MV-MSS).

A. Network Model

This section provides a network model which is shown in Fig. 1, where the network entities are considered as: a) vehicles (users), b) road-side unites $(RSUs)$ (also called aggregators), and c) cloud servers (verifiers).

In this model, we have the following hierarchy:

- *Cluster of Vehicles:* The vehicles dynamically form various clusters, where a particular vehicle can be selected as a leader (L) or cluster head (CH) from a cluster. The CH in a cluster has a message or data, where the data can be public data and it can be any type such as data related to traffic, road conditions, road accidental information, etc. and the message can be sent via public channel to the associated road-side unit (RSU). Note that a dynamic clustering mechanism can be adopted as suggested by Kakkasageri and Manvi [36] for creation of the clusters of vehicles on the fly. In their mechanism, the vehicles which are moving on the same lane segment ending at the intersection with the other lane can be considered. Now, each vehicle may find their neighbor vehicles that are moving on the same lane segment as well as towards the same direction with the nearly same speed. In this way, the vehicles become the optimum choice for being the members of a possible cluster that can be formed on that lane. A vehicle which leads among all other vehicles on the lane is considered as an initiator because it needs to begin the cluster formation process.
- *Road-side Units (RSUs):* The in-charge RUSs will interact with their respective cluster head(s) and aggregate the data from the cluster, where the cluster head gets the data from its member vehicles. Here, the messages can be sent via a public channel to the associated RSU by its associated cluster head with its multi-signature generated with the help of the other vehicles belonged to this cluster. The RSU then can verify the message or data with its received multi-signature and derive an aggregate public key based on the received public keys of the other vehicles. Next, the RSU forwards a message with the received data, multi-signature, and generated public key to the cloud server.
- *Blockchain Center:* The cloud servers form a peer-to-peer (P2P) cloud servers network in the blockchain center. After receiving the messages from the RSUs, a cloud server verifies the received data with its multi-signature. Once the multi-signature is validated, the cloud server considers the data as a transaction and puts it into a global transactions pool, which is accessible to all peer nodes. Whenever the pool reaches to a certain per-defined threshold value for block creation, a new block will be created by a newly elected leader from the P2P network. The leader then executes a voting based distributed consensus algorithm for the newly generated block to add it into the blockchain.

In this paper, we consider the consensus algorithm as "Practical Byzantine Fault Tolerance (PBFT)" consensus algorithm [35] for block verification and addition into the blockchain in blockchain center (BC).

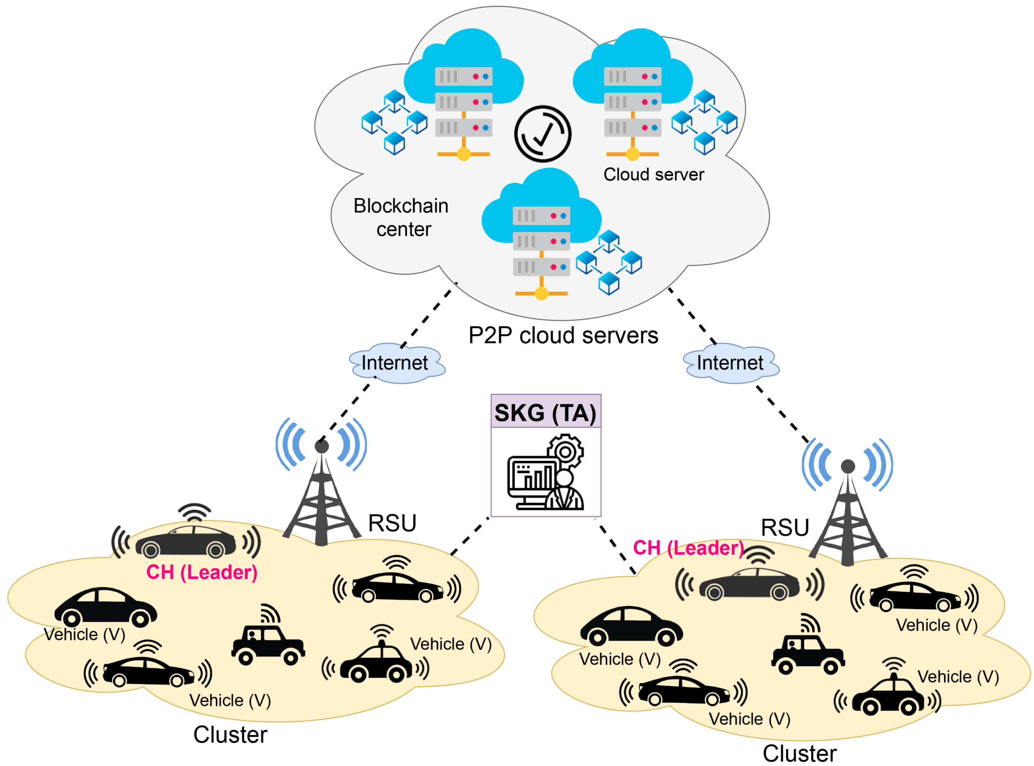


Fig. 1. Identity-based multi-signature enabled IoV architecture using blockchain.

B. Threat Model

In the proposed multivariate based multi-signature (MSS) scheme MV-MSS, the sensing information is communicated over the public channel. Since the users (vehicles) exchange the information with their leader or cluster head CH (which is also a vehicle) within a cluster, the communications between the CH and RSU, and the CH and cloud server are made over the public channel (insecure channel). Therefore, the message authenticity or legitimacy is a biggest concern.

In this paper, we consider the following two models:

- *Dolev-Yao (DY) Threat Model:* A widely-recognized security threat model, called the "Dolev-Yao (DY) threat model" [37] is considered, where an adversary, say $\mathcal{A}$, can interrupt the transmissions by executing various tasks, such as, messages modification and deletion, malicious messages injection over the communication channel.
- *CK-adversary Model:* We consider another de facto adversary threat model, called the "Canetti and Krawczyk's model (CK-adversary model)" [38], where the $\mathcal{A}$ has the ability more than the DY threat model, and $\mathcal{A}$ can compromise the session states by hijacking a session.

Finally, we assume that the secret credentials stored into the physically captured vehicles can be extracted by $\mathcal{A}$ using the "power analysis attacks" [39].

III. Mathematical Preliminaries

In this section, we first describe the computational hard multivariate-quadratic (MQ) problem which is required in designing our proposed scheme in this paper. Moreover, we also discuss the relevant multivariate identification protocol, general construction of identity-based multi-signature scheme (IB-MSS) and the existential unforgeability under chosen-message and chosen-identity attack (uf-cmia) on an IB-MSS.

Our proposed scheme attains its security on the hardness of MQ problem. In laymen terms, it says that solving a system of multivariate polynomials is NP-hard, which is formulated as follows.

Definition 1 (Multivariate-Quadratic (MQ) Problem): Given a system $\mathcal{W} = (w_{(1)}(\phi_1, \ldots, \phi_n), \ldots, w_{(m)}(\phi_1, \ldots, \phi_n))$ of m quadratic equations in variables $(\phi_1, \ldots, \phi_n)$, find a n tuple $(\bar{\phi}_1, \ldots, \bar{\phi}_n)$ such that $w_{(1)}(\bar{\phi}_1, \ldots, \bar{\phi}_n) = \cdots = w_{(m)}(\bar{\phi}_1, \ldots, \bar{\phi}_n) = 0$.

A. Multivariate Identification Protocol

Let $\mathbb{F}_q$ denote the finite field of order q. The primary idea behind the design of MPKC is to choose a system $\mathcal{F} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ of m multivariate polynomials of degree two in n variables. We stipulate that this map $\mathcal{F}$, also known as central map, is easily inverted in the sense that finding preimage of y under $\mathcal{F}$ is easy. To obfuscate the structure of $\mathcal{F}$, we pick two affine invertible transformations $\mathcal{S} : \mathbb{F}_q^m \rightarrow \mathbb{F}_q^m$ and $\mathcal{T} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$. To find the public key of the cryptosystem, we move on by taking the composed map $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$. The secret key of the MPKC is a three tuple $(\mathcal{S}, \mathcal{F}, \mathcal{T})$.

Sakumato *et al.* [40] presented the first provably secure 3-pass identification protocol with probability of impersonation being $\frac{2}{3}$. General idea of the scheme is following. Suppose we are given public key of the underlying MPKC as $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$. Prover who wish to identify himself, chooses $s \in \mathbb{F}_q^n$ as his secret key and evaluates $\mathcal{P}$ at s to derive $v = \mathcal{P}(s)$, which works as the public key of the prover. To prove his identity to a verifier, he is expected to satisfy the verifier of his knowledge of the secret s without revealing s. Polar form of $\mathcal{P}$ is formulated as

$$\mathcal{G}(\iota, \tau) = \mathcal{P}(\iota + \tau) - \mathcal{P}(\iota) - \mathcal{P}(\tau) \quad (1)$$

To construct the identification, we split the secret into various parts by using the bilinearity of $\mathcal{G}$. More details can be found in [40].

Monterio *et al.* [34] later improved the scheme [40] in order to achieve $\frac{1}{2}$ as the impersonation probability. This protocol reduces the communication cost as number of rounds required to reach a given security level is far less when compared with scheme [40]. The proposed MV-MSS in this paper uses this scheme as an important constituent in its design.

B. General Construction of Identity-Based Multi-Signature Scheme

This section describes the general construction of an IB-MSS.

1) Various Phases: It consists of the following phases:

- **Setup**
  $(MSK, \mathsf{pk}) \leftarrow \mathsf{Setup}(\eta)$: Given a security parameter η, Setup algorithm outputs public key pk and a master secret key MSK.
- **KeyGen**
  $(\mathsf{sk}_{\mathsf{Id}}, \mathsf{pk}_{\mathsf{Id}}) \leftarrow \mathsf{KeyGen}(MSK, \mathsf{Id})$: Given MSK and unique identifier Id of a receiver as input, KeyGen outputs secret key $\mathsf{sk}_{\mathsf{Id}}$ and public key $\mathsf{pk}_{\mathsf{Id}}$ of a user with identity Id.
- **Sign**
  $\sigma \leftarrow \mathsf{Sign}(\mathsf{pk}, S, D, \mathsf{msg})$: Given a message msg, public key pk and a set $S = \{U_1, \ldots, U_M\}$ of signers with respective secret keys $D = \{\mathsf{sk}_{\mathsf{Id}1}, \ldots, \mathsf{sk}_{\mathsf{Id}M}\}$, a leader L (chosen from the set S) interacts with other members of S to produce the multi-signature σ on the message msg using the algorithm Sign.
- **KeyAgg**
  $\mathsf{apk} \leftarrow \mathsf{KeyAgg}(E, \mathsf{pk})$: Given a set E of public keys $\{\mathsf{pk}_{\mathsf{Id}1}, \ldots \mathsf{pk}_{\mathsf{Id}M}\}$, and pk as inputs, the algorithm KeyAgg outputs a single aggregate public key apk.
- **Verify**
  $0/1 \leftarrow \mathsf{Verify}(\sigma, \mathsf{apk}, \mathsf{msg}, \mathsf{pk})$: A verifier checks the validity of a signature σ on message msg by calling the algorithm Verify which on input $(\sigma, \mathsf{apk}, \mathsf{msg}, \mathsf{pk})$ outputs 0 or 1 indicating that the signature is valid or invalid respectively.

2) Existential Unforgeability Under Chosen-Message and Chosen-Identity Attack (uf-Cmia): The notion of uf-cmia is considered as the standard security notion for an IB-MSS [41]. Let us consider an IB-MSS made up of algorithms Setup, KeyGen, Sign, KeyAgg, Verify. The experiment described in Algorithm 1 explains a chosen-message and chosen-identity attack against an IB-MSS [41]. We define the adversarial advantage of $\mathcal{A}$ against $IB-MSS =$ (Setup, KeyGen, Sign, KeyAgg, Verify) as a probability of $\mathbf{Exp}^{uf-cmia}_{\text{IB-MSS}}(\mathcal{A})$ that outputs 1. In other words, $\mathbf{Adv}^{uf-cmia}_{\text{IB-MSS}}(\mathcal{A}) = Pr|\mathbf{Adv}^{uf-cmia}_{\text{IB-MSS}}(\mathcal{A}) = 1|$.

Algorithm 1: $\mathbf{Exp}^{uf-cmia}_{\text{IB-MSS}}(\mathcal{A})$.

1: Set $(\mathsf{pk}, MSK) \leftarrow \mathsf{Setup}(\eta)$; $\mathsf{MLst} \leftarrow \emptyset$; $\mathsf{CLst} \leftarrow \emptyset$
2: Run $\mathcal{A}(\mathsf{pk})$ and control key derivation and signature queries of $\mathcal{A}$
3: During key derivation query for Id, append Id to CLst and return output $\mathsf{sk}_{\mathsf{Id}}$ of KeyGen on (MSK, Id) to $\mathcal{A}$
4: During a signing query on (msg, Id), append (msg, Id) to MLst and on behalf of Id run the algorithm Sign on msg, forward messages to and from $\mathcal{A}$
5: **if** $\mathcal{A}$ halts **then**
6:   Split its output as $(\mathsf{msg}, Set.Id, \sigma)$, where $Set.Id$ is the set of Id's used for generating signature σ.
7: **end if**
8: **if** $\mathsf{Verify}(\mathsf{pk}, \mathsf{msg}, Set.Id, \sigma) = 1 \wedge (\exists;\ \mathsf{Id}^* \in Set.Id$ such that $(\mathsf{Id}^* \notin \mathsf{CLst}) \wedge ((\mathsf{msg}, \mathsf{Id}^*) \notin \mathsf{MLst}))$ **then**
9:   **return** Success (1)
10: **else**
11:   **return** Failure (0)
12: **end if**

Definition 2: An IB-MSS is said to be $(T, \epsilon, M, Q_{Ke}, Q_{Si})$-secure if $\mathbf{Adv}^{uf-cmia}_{\text{IB-MSS}}(\mathcal{A}) \leq \epsilon$ for all such $\mathcal{A}$ which can generate a forgery on behalf of at most M participants by running in time at most t and, making at most Q_{Ke} queries for key derivation and at most Q_{Si} queries for signature generation.

In the ROM, the security notion is extended to $(T, \epsilon, M, Q_{Ke}, Q_{Si}, Q_{Ha})$-secure, where $\mathcal{A}$ is additionally bound to make at most Q_{Ha} hash queries.

IV. Proposed Multivariate Identity-Based Multi-Signature Scheme (MV-MSS)

In this section, we first give a high-level overview of our proposed multivariate identity-based multi-signature scheme (MV-MSS) before describing the details of the scheme.

A. High Level Overview of MV-MSS

The proposed MV-MSS is executed among a group of signers and a verifier. We use the identification protocol of Monteiro *et al.* [34] as the building block of our construction. To convert the identification protocol into a signature scheme, we use the technique of Hülsing *et al.* [42]. The MV-MSS consist of five algorithms: (i) MV-MSS.Setup, (ii) MV-MSS.KeyGen, (iii) MV-MSS.Sign, (iv) MV-MSS.KeyAgg, and (v) MV-MSS.Verify. On input of security parameter η, secret key generator (SKG) executes MV-MSS.Setup to generate public key pk and master secret key MSK. In the next step, SKG runs MV-MSS.KeyGen to produce secret key $\mathsf{sk}_{\mathsf{Id}i}$ for each user U_i with identity Id_i with $\mathsf{Id}_i \in \{0,1\}^*$. During MV-MSS.Sign, a leader L, selected from a set $S = \{U_1, \ldots, U_M\}$ of signers with respective secret keys $\{\mathsf{sk}_{\mathsf{Id}1}, \ldots, \mathsf{sk}_{\mathsf{Id}M}\}$, interacts with other members of S to generate the multi-signature σ on the message msg. The algorithm MV-MSS.KeyAgg is executed to produce the aggregate public key apk. Finally, a verifier runs MV-MSS.Verify on input $(\sigma, \mathsf{apk}, \mathsf{msg}, \mathsf{pk})$ to verify the validity of the message-signature pair (σ, msg).

B. Detailed Description of MV-MSS

We now discuss the proposed MV-MSS in more detail, which consists of the following algorithms.

1) MV-MSS.Setup Algorithm $[(\mathsf{pk}, MSK) \leftarrow$ MV−MSS.Setup$(\eta)]$: Given a security parameter η, the SKG generates public key $\mathcal{P}: \mathbb{F}_q^n \to \mathbb{F}_q^m$ and secret key $MSK = (\mathcal{S}, \mathcal{F}, \mathcal{T})$ for the underlying multivariate signature scheme. It sets MSK as the master secret key, and publishes $\mathsf{pk} = \mathcal{P}$ as the public key. Here,

- $\mathcal{S}: \mathbb{F}_q^m \to \mathbb{F}_q^m$ and $\mathcal{T}: \mathbb{F}_q^n \to \mathbb{F}_q^n$ are affine invertible map of the form $\mathcal{S}(y_1, \ldots, y_m) = (\mathcal{S}_1(y_1, \ldots, y_m), \mathcal{S}_2(y_1, \ldots, y_m), \ldots, \mathcal{S}_m(y_1, \ldots, y_m))$ and $\mathcal{T}(y_1, \ldots, y_n) = (\mathcal{T}_1(y_1, \ldots, y_n), \mathcal{T}_2(y_1, \ldots, y_n), \ldots, \mathcal{T}_n(y_1, \ldots, y_n))$.
- $\mathcal{F}: \mathbb{F}_q^n \to \mathbb{F}_q^m$ is a system of m multivariate polynomials $(\mathcal{F}_1, \ldots, \mathcal{F}_m)$ of the following form

$$\mathcal{F}_\kappa = \sum_{i=1}^{n}\sum_{j=1}^{n} b_{\kappa_{ij}} y_i y_j + \sum_{i=1}^{n} c_{\kappa_i} y_i,$$

  where $b_{\kappa_{ij}}, c_{\kappa_i} \in \mathbb{F}_q$ are constants.
- $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} = \begin{pmatrix} \mathcal{P}_1(y_1, \ldots, y_n) \\ \mathcal{P}_2(y_1, \ldots, y_n) \\ \cdots\cdots \\ \mathcal{P}_m(y_1, \ldots, y_n) \end{pmatrix}$ such that $\mathcal{P}_k$ can be written as

$$\mathcal{P}_\kappa = \sum_{i=1}^{n}\sum_{j=1}^{n} \bar{b}_{\kappa_{ij}} y_i y_j + \sum_{i=1}^{n} \bar{c}_{\kappa_i} y_i,$$

  where $\bar{b}_{\kappa_{ij}}, \bar{c}_{\kappa_i} \in \mathbb{F}_q$ are constants.

2) MV-MSS.KeyGen Algorithm $[(\mathsf{Sk}_{\mathsf{Id}}, \mathsf{pk}_{\mathsf{Id}}) \leftarrow$ MV−MSS.KeyGen $(MSK, \mathsf{Id})]$: The SKG, on input $MSK = (\mathcal{S}, \mathcal{F}, \mathcal{T})$ and an identity $\mathsf{Id} \in \{0,1\}^*$ of a user U, the SKG executes the following steps:

- Derives $\mathsf{k}_{\mathsf{Id}} \in \mathbb{F}_q^m$ by computing $\mathsf{Hash}_1(\mathsf{Id}) = \mathsf{k}_{\mathsf{Id}}$ for some cryptographically secure collision-resistant hash function $\mathsf{Hash}_1: \{0,1\}^* \to \mathbb{F}_q^m$, and sets $\mathsf{pk}_{\mathsf{Id}} = k_{Id}$.
- Evaluates $\mathsf{u}_{\mathsf{Id}} = \mathcal{P}^{-1}(\mathsf{k}_{\mathsf{Id}}) \in \mathbb{F}_q^n$ using $MSK = (\mathcal{S}, \mathcal{F}, \mathcal{T})$.
- Sends $\mathsf{sk}_{\mathsf{Id}} = \mathsf{u}_{\mathsf{Id}}$ as the secret key to the user U with the identity Id and outputs $\mathsf{pk}_{\mathsf{Id}}$ as the associated public key.

3) MV-MSS.Sign Algorithm $[\sigma \leftarrow$ MV-MSS.Sign $(\mathsf{pk}, S, D, \mathsf{msg})]$: A leader L is chosen from a set $S = \{U_1, \ldots, U_M\}$ of users with respective secret keys $\{\mathsf{sk}_{\mathsf{Id}1}, \ldots, \mathsf{sk}_{\mathsf{Id}M}\} = \{\mathsf{u}_{\mathsf{Id}1}, \ldots, \mathsf{u}_{\mathsf{Id}M}\}$. To sign the message $\mathsf{msg} \in \{0,1\}^*$, following steps are executed:

- Each of $U_i \in \{U_1, \ldots, U_M\}$ having secret key $\mathsf{sk}_{\mathsf{Id}i} = \mathsf{u}_{\mathsf{Id}i}$ excutes the following steps:
  – Choose randomly $r_0^{(i)}, t_0^{(i)}, d_0^{(i)} \in \mathbb{F}_q^n$ and $e_0^{(i)}, u_0^{(i)} \in \mathbb{F}_q^m$
  – Compute $r_1^{(i)} = \mathsf{u}_{\mathsf{Id}i} - r_0^{(i)}$, $t_1^{(i)} = r_0^{(i)} - t_0^{(i)}$, $d_1^{(i)} = r_1^{(i)} - d_0^{(i)}$, $e_1^{(i)} = \mathcal{P}(r_0^{(i)}) - e_0^{(i)}$, $u_1^{(i)} = \mathcal{P}(r_1^{(i)}) - u_0^{(i)}$.
  – Evaluate $a_0^{(i)} = \mathsf{Commit}(r_0^{(i)}, \mathcal{G}(r_0^{(i)}, d_1^{(i)}) + u_1^{(i)})$, $a_1^{(i)} = \mathsf{Commit}(r_1^{(i)}, \mathcal{G}(t_0^{(i)}, r_1^{(i)}) + e_0^{(i)})$, $a_2^{(i)} = \mathsf{Commit}(t_0^{(i)}, e_0^{(i)})$, $a_3^{(i)} = \mathsf{Commit}(t_1^{(i)}, e_1^{(i)})$, $a_4^{(i)} = \mathsf{Commit}(d_0^{(i)}, u_0^{(i)})$, $a_5^{(i)} = \mathsf{Commit}(d_1^{(i)}, u_1^{(i)})$.
- Each $U_i \in \{U_1, \ldots, U_M\} \setminus L$ sends $\{a_0^{(i)}, a_1^{(i)}, a_2^{(i)}, a_3^{(i)}, a_4^{(i)}, a_5^{(i)}\}$ to the leader L.
- Upon receiving the individual commitments from the users $\{U_1, \ldots, U_M\} \setminus L$, the leader L does the following steps:
  – Compute $A_0 = \mathsf{Commit}(a_0^{(1)}, \ldots, a_0^{(M)})$, $A_1 = \mathsf{Commit}(a_1^{(1)}, \ldots, a_1^{(M)})$, $A_2 = \mathsf{Commit}(a_2^{(1)}, \ldots, a_2^{(M)})$, $A_3 = \mathsf{Commit}(a_3^{(1)}, \ldots, a_3^{(M)})$, $A_4 = \mathsf{Commit}(a_4^{(1)}, \ldots, a_4^{(M)})$, $A_5 = \mathsf{Commit}(a_5^{(1)}, \ldots, a_5^{(M)})$;
  – Determine the master commitment as $COM = \mathsf{Commit}(A_0||A_1||\ldots,||A_5)$;
  – Evaluate the challenge $cl = \mathsf{Hash}_2(\mathsf{msg}||COM) \in \{0,1,2,3\}$ using cryptographically secure collision resistant hash function $\mathsf{Hash}_2: \{0,1\}^* \to \{0,1,2,3\}$;
  – Send cl to each cosigner $U_i \in \{U_1, \ldots, U_M\} \setminus L$.
- Each co-signer $U_i \in S$ computes its response Rsp_i in the following manner:
  – If $cl = 0$, then $Rsp_i = (r_1^{(i)}, t_1^{(i)}, e_1^{(i)}, d_0^{(i)}, u_0^{(i)})$.
  – If $cl = 1$, then $Rsp_i = (r_1^{(i)}, t_0^{(i)}, e_0^{(i)}, d_1^{(i)}, u_1^{(i)})$.
  – If $cl = 2$, then $Rsp_i = (r_0^{(i)}, t_0^{(i)}, e_0^{(i)}, d_1^{(i)}, u_1^{(i)})$.
  – If $cl = 3$, then $Rsp_i = (r_0^{(i)}, t_1^{(i)}, e_1^{(i)}, d_0^{(i)}, u_0^{(i)})$.

In the following, each $U_i \in \{U_1, \ldots, U_M\} \setminus L$ send its Rsp_i to L.

- The leader L, on receiving the individual responses Rsp_i from the co-signers, computes the master response RSP in the following way:
  – If $cl = 0$, then $RSP = (Rsp_1, \ldots, Rsp_M, A_0, A_2, A_5)$
  – If $cl = 1$, then $RSP = (Rsp_1, \ldots, Rsp_M, A_0, A_3)$
  – If $cl = 2$, then $RSP = (Rsp_1, \ldots, Rsp_M, A_1, A_4)$
  – If $cl = 3$, then $RSP = (Rsp_1, \ldots, Rsp_M, A_1, A_2, A_5)$

Finally, L publishes $\sigma = (COM||RSP)$ as the signature on the message msg.

4) MV-MSS.KeyAgg Algorithm $[\mathsf{apk} \leftarrow$ MV-MSS.KeyAgg $(E, \mathsf{pk})]$: Given a set E of public keys $\{\mathsf{pk}_{\mathsf{Id}1}, \ldots \mathsf{pk}_{\mathsf{Id}M}\} = \{\mathsf{k}_{\mathsf{Id}1}, \ldots \mathsf{k}_{\mathsf{Id}M}\}$, and pk as inputs, a single aggregate public key is computed as $\mathsf{apk} = \{\mathcal{P}||\mathsf{k}_{\mathsf{Id}1}|| \ldots ||\mathsf{k}_{\mathsf{Id}M}\}$.

5) MV-MSS.Verify Algorithm $[0/1 \leftarrow$ MV -MSS.Verify $(\sigma, \mathsf{apk}, \mathsf{msg}, \mathsf{pk})]$: On receiving the signature $\sigma = (COM||RSP)$, a verifier performs the following steps to verify the validity of the signature.

- Parses σ as COM, RSP.
- Derives the challenge cl by computing $\mathsf{Hash}_2(\mathsf{msg}||COM)$.
- Checks the correctness of COM in the following manner:
  * If $cl = 0$, he parses RSP into $r_1^{(1)}, t_1^{(1)}, e_1^{(1)}, d_0^{(1)}, u_0^{(1)}, \ldots, r_1^{(M)}, t_1^{(M)}, e_1^{(M)}, d_0^{(M)}, u_0^{(M)}$. For each $i = 1, \ldots, M$, he computes $\widetilde{a}_1^{(i)} = \mathsf{Commit}(r_1^{(i)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(r_1^{(i)}) - \mathcal{G}(t_1^{(i)}, r_1^{(i)}) - e_1^{(i)})$, $\widetilde{a}_3^{(i)} = \mathsf{Commit}(t_1^{(i)}, e_1^{(i)})$ and $\widetilde{a}_4^{(i)} = \mathsf{Commit}(d_0^{(i)}, u_0^{(i)})$. Next, he evaluates $\widetilde{A}_1 = \mathsf{Commit}(\widetilde{a}_1^{(1)}, \ldots, \widetilde{a}_1^{(M)})$, $\widetilde{A}_3 = \mathsf{Commit}(\widetilde{a}_3^{(1)}, \ldots,$

$\widetilde{a}_3^{(M)})$, $\widetilde{A}_4 = \mathsf{Commit}(\widetilde{a}_4^{(1)}, \ldots, \widetilde{a}_4^{(M)})$, and checks whether $COM \stackrel{?}{=} \mathsf{Commit}(A_0, \widetilde{A}_1, A_2, \widetilde{A}_3, \widetilde{A}_4, A_5)$.

* If $cl = 1$, he parses RSP into $r_1^{(1)}, t_0^{(1)}, e_0^{(1)}, d_1^{(1)}, u_1^{(1)}, \ldots, r_1^{(M)}, t_0^{(M)}, e_0^{(M)}, d_1^{(M)}, u_1^{(M)}$. For each $i = 1, \ldots, M$, he derives $\widetilde{a}_1^{(i)} = \mathsf{Commit}(r_1^{(i)}, \mathcal{G}(t_0^{(i)}, r_1^{(i)}) + e_0^{(i)})$, $\widetilde{a}_2^{(i)} = \mathsf{Commit}(t_0^{(i)}, e_0^{(i)})$, $\widetilde{a}_4^{(i)} = \mathsf{Commit}\ (r_1^{(i)} - d_1^{(i)}, \mathcal{P}(r_1^{(i)}) - u_1^{(i)})$, $\widetilde{a}_5^{(i)} = \mathsf{Commit}(d_1^{(i)}, u_1^{(i)})$. In the following, he computes $\widetilde{A}_1 = \mathsf{Commit}(\widetilde{a}_1^{(1)}, \ldots, \widetilde{a}_1^{(M)})$, $\widetilde{A}_2 = \mathsf{Commit}(\widetilde{a}_2^{(1)}, \ldots, \widetilde{a}_2^{(M)})$, $\widetilde{A}_4 = \mathsf{Commit}(\widetilde{a}_4^{(1)}, \ldots, \widetilde{a}_4^{(M)})$, $\widetilde{A}_5 = \mathsf{Commit}(\widetilde{a}_5^{(1)}, \ldots, \widetilde{a}_5^{(M)})$, and checks whether $COM \stackrel{?}{=} \mathsf{Commit}(A_0, \widetilde{A}_1, \widetilde{A}_2, A_3, \widetilde{A}_4, \widetilde{A}_5)$.

* If $cl = 2$, he parses RSP into $r_0^{(1)}, t_0^{(1)}, e_0^{(1)}, d_1^{(1)}, u_1^{(1)}, \ldots, r_0^{(M)}, t_0^{(M)}, e_0^{(M)}, d_1^{(M)}, u_1^{(M)}$. For each $i = 1, \ldots, M$, he evaluates $\widetilde{a}_0^{(i)} = \mathsf{Commit}(r_0^{(i)}, \mathcal{G}(r_0^{(i)}, d_1^{(i)}) + u_1^{(i)})$, $\widetilde{a}_2^{(i)} = \mathsf{Commit}(t_0^{(i)}, e_0^{(i)})$, $\widetilde{a}_3^{(i)} = \mathsf{Commit}(r_0^{(i)} - t_0^{(i)}, \mathcal{P}(r_0^{(i)}) - e_0^{(i)})$, $\widetilde{a}_5^{(i)} = \mathsf{Commit}(d_1^{(i)}, u_1^{(i)})$. He then derives $\widetilde{A}_0 = \mathsf{Commit}(\widetilde{a}_0^{(1)}, \ldots, \widetilde{a}_0^{(M)})$, $\widetilde{A}_2 = \mathsf{Commit}(\widetilde{a}_2^{(1)}, \ldots, \widetilde{a}_2^{(M)})$, $\widetilde{A}_3 = \mathsf{Commit}(\widetilde{a}_3^{(1)}, \ldots, \widetilde{a}_3^{(M)})$, $\widetilde{A}_5 = \mathsf{Commit}(\widetilde{a}_5^{(1)}, \ldots, \widetilde{a}_5^{(M)})$, and checks whether $COM \stackrel{?}{=} \mathsf{Commit}(\widetilde{A}_0, A_1, \widetilde{A}_2, \widetilde{A}_3, A_4, \widetilde{A}_5)$.

* If $cl = 3$, he parses RSP into $r_0^{(1)}, t_1^{(1)}, e_1^{(1)}, d_0^{(1)}, u_0^{(1)}, \ldots, r_0^{(M)}, t_1^{(M)}, e_1^{(M)}, d_0^{(M)}, u_0^{(M)}$. For each $i = 1, \ldots, M$, he computes $\widetilde{a}_0^{(i)} = \mathsf{Commit}(r_0^{(i)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(r_0^{(i)}) - \mathcal{G}(r_0^{(i)}, d_0^{(i)}) - u_0^{(i)})$, $\widetilde{a}_3^{(i)} = \mathsf{Commit}(t_1^{(i)}, e_1^{(i)})$ and $\widetilde{a}_4^{(i)} = \mathsf{Commit}(d_0^{(i)}, u_0^{(i)})$. In the following, he executes $\widetilde{A}_0 = \mathsf{Commit}(\widetilde{a}_0^{(1)}, \ldots, \widetilde{a}_0^{(M)})$, $\widetilde{A}_3 = \mathsf{Commit}(\widetilde{a}_3^{(1)}, \ldots, \widetilde{a}_3^{(M)})$, $\widetilde{A}_4 = \mathsf{Commit}(\widetilde{a}_4^{(1)}, \ldots, \widetilde{a}_4^{(M)})$, and checks whether $COM \stackrel{?}{=} \mathsf{Commit}(\widetilde{A}_0, A_1, A_2, \widetilde{A}_3, \widetilde{A}_4, A_5)$.

Remark 1: The impersonation probability of our MV-MSS is $\frac{1}{2}$ due to the use of one round of the identification protocol of [34] to generate the signature. This can be reduced by using multiple rounds of the associated identification protocol during the generation of the signature. In particular, γ rounds of the identification protocol has the impersonation probability $(\frac{1}{2})^\gamma$. The correctness of MV-MSS is provided with a detailed proof in Appendix A.

V. Incorporating MV-MSS in Blockchain-Based IoV Applications

In this section, we incorporate the blockchain technology in the proposed MV-MSS based on the network model provided in Fig. 1.

In this work, we consider a dynamic cluster having M members as its vehicles and a cluster head, called the leader, as L. This scenario is considered for creating different clusters of vehicles on the fly, [43]. A dynamic clustering can be contructed as follows. The vehicles, which move on the same lane segment and end at the intersection with the other lane, can be included in a cluster. As a result, a vehicle requires to find its neighboring other vehicles that are moving on the same lane segment towards the same direction and also with the same speed. We assume that $V_1, V_2, \ldots, V_M$ are the members of the L^{th} cluster in an IoV environment. Each vehicle V_i treated as a user will have sensing information, such as vehicle accident related data, traffice related data, etc.

The following steps are executed:

1) Given a security parameter η as input, the secret key generator (SKG) first runs MV-MSS.Setup to generate the public key pk and the master secret key MSK.
2) Next, the SKG runs MV-MSS.KeyGen in order to generate the secret key $\mathsf{sk}_{\mathsf{Id}i}$ for each vehicle being a user V_i $(i = 1, 2, \ldots, M)$, where M is the number of vehicles in the L^{th} cluster with an identity Id_i with $\mathsf{Id}_i \in \{0, 1\}^*$.
3) Condiser L^{th} cluster with its cluster head CH as leader L, which has been picked from a set $S = \{V_1, \ldots, V_M\}$ of signers with their respective secret keys $\{\mathsf{sk}_{\mathsf{Id}1}, \ldots, \mathsf{sk}_{\mathsf{Id}M}\}$, has a message msg that is the sensing information of a vehicle in this cluster. The leader L will then interact with other members of S to create a multi-signature σ on the message msg with the help of the MV-MSS.Sign algorithm using the public key $\mathsf{pk} = \mathcal{P}, S$, the set of secret keys $D = \{\mathsf{sk}_{\mathsf{Id}1}, \ldots, \mathsf{sk}_{\mathsf{Id}M}\}$ and msg. After that L creates a message $Msg_1 = \{\sigma, E, \mathsf{msg}\}$ and send it to the nearby RSU via public channel, where E is the set of public keys $\{\mathsf{pk}_{\mathsf{Id}1}, \ldots \mathsf{pk}_{\mathsf{Id}M}\}$.
4) After receiving the message $Msg_1 = \{\sigma, E, \mathsf{msg}\}$ from the leader L, the in-charge RSU needs to apply the MV-MSS.KeyAgg algorithm to produce the aggregate public key apk as $\mathsf{apk} = \{\mathcal{P}||\mathsf{k}_{\mathsf{Id}1}|| \ldots ||\mathsf{k}_{\mathsf{Id}M}\}$. The RSU then can (optionally) verify the signed message $\{\sigma, \mathsf{msg}\}$ on the received msg. For this purpose, the RSU can run the MV-MSS.Verify algorithm on inputs $(\sigma, \mathsf{apk}, \mathsf{msg}, \mathsf{pk})$ to verify the validity of the message-signature pair (σ, msg). If the signature verification is successful, the RSU constructs another message $Msg_2 = \{\sigma, \mathsf{apk}, \mathsf{msg}\}$ and sends it to a cloud server CS_j residing in the blockchain center containing the cloud servers of a peer-to-peer (P2P) network.
5) Once the Msg_2 is received by the CS_j, it verifies the validity of the multi-signature σ on msg by running the the MV-MSS.Verify algorithm on inputs $(\sigma, \mathsf{apk}, \mathsf{msg}, \mathsf{pk})$. If the verification is successful, CS_j treats the multi-signature σ on the data (msg) as valid and the data will be then considered as a transaction, say Tx_l, as $Tx_l = (\sigma_l, apk, msg_l)$.
   Since the transaction is verified by the cloud server, the transaction can be injected into a global transactions pool in the P2P network. Whenever the transactions pool reaches to a certain threshold value (say, t_n), a leader from the P2P cloud servers network, say CS_l, is elected by the round-robin fashion from the P2P network or by applying a leader selection algorithm as suggested in [44]. Assume that the leader CS_l constructs a new block with the t_n number of transactions, Tx_l $(l = 1, 2, \ldots, t_n)$ as shown in Fig. 2. The constructed block has the following components:

Block Header	
Block Version	$BVer$
Previous Block Hash	PBH
Merkle Tree Root	MTR
Timestamp	TS
Block Creator	Cloud Server CS_j
Block Payload	
List of Transactions	$\{Tx_l = (\sigma_l, apk, msg_l) \| l = 1, \ldots, t_n\}$
Current Block Hash	$CBHash$

Fig. 2. Block structure using multivariate multi-signatures on data in IoV.

- *Block Header:* It contains various fields like block version ($BVer$) which is a unique serial number, previous block hash (PBH) which is the hash value of the previous block in the chain (here, the hash function is applied as Secure Hash Algorithm (SHA-256) [45] that produces 256-bit hash output or message digest), Merkle tree root (MTR) which contains the hash value of the transactions computed under the Merkle tree construction, timestamp (TS) which presents the timestamp when the block was created and block creator which is the cloud server (CS_j).
- *Block Payload:* It contains a list of t_n transactions of the form: $\{Tx_l = (\sigma_l, apk, msg_l) | l = 1, \ldots, t_n\}$.
- *Current Block Hash:* It is the hash of all the fields in the block where $CBHash = H(\text{Block Header}||\text{Block Payload})$, where $H(\cdot)$ is a "collision-resistant one-way cryptographic hash function" (for example, $H(\cdot)$ can be SHA-256).

Once a block, say $Block_m$ is constructed, a consensus algorithm will be executed by CS_l for the newly proposed block $Block_m$'s verification and addition into the blockchain. In this case, we may consider a voting-based consensus algorithm, called "Practical Byzantine Fault Tolerance (PBFT)" consensus algorithm [35]).

Finally, the overall process in the proposed MV-MSS under blockchain context is provided in Fig. 3.

Remark 2 (Applications to Bitcoin): We will now discuss how our proposed design MV-MSS can be helpful within the confines of Bitcoin transactions. Validation by multiple parties are required to validate the exchange of Bitcoins before they are written into the public Blockchain. Keeping in mind that blocks in Blockchain are duplicated over a distributed network, we need size of the signature to be as condensed as possible. Designing any system needs to take these issues seriously. At present, Bitcoin makes use of the secp256k1 curve along with the "elliptic curve digital signature algorithm (ECDSA)" to validate and accredit currency exchanges. As we already discussed in the Introduction section, these classical schemes face a big threat in a world where big quantum computers become a reality. MV-MSS is built on top of the hardness of MQ problem, and to date, there has been no algorithm that can break the problem in polynomial time [20]. Hence, MV-MSS provides a post-quantum safe alternative to existing classical schemes.

Our proposed MV-MSS can help to reduce the storage as well as bandwidth costs as it outputs a single condensed signature with shrunk size. Thus it is ideally suited for transactions where multiple keys are required to validate each transaction. In multiparty cryptocurrency transactions, we have α-of-β addresses with β private keys for $\alpha \leq \beta$. It means to validate the transaction, α signatures are required. This system offers two huge advantages. Firstly, it is an arduous task for an adversary to thieve Bitcoins because to set an attack in motion, he needs to take control over all α machines. For example, given a 3-of-3 address, three keys could have been stored on three separate machines, and the attacker would have to compromise them all to initiate an attack. Secondly, multi-signatures also ensure that there is no single point of failure. For example, in a 3-of-4 address, currency exchange can be carried out even if a key is lost. In financial transactions protocols, the value of α is usually less than or equal to 5. MV-MSS is optimally suited for all such scenarios. Table III testifies the competitive capacity of MV-MSS in these practical scenarios where M is equal to 5.

We discuss the practicality of our proposed scheme by taking a real-life example. Let Alice be a seller with identity $\mathsf{ID_A}$, Bob be a buyer with identity $\mathsf{ID_B}$, and Oscar is an arbitrator with identity $\mathsf{ID_O}$. They submit their respective ID's to SKG which by running MV-MSS.KeyGen generates public key-secret key pair for each of Alice, Bob, and Oscar. We denote by $(\mathsf{sk}_{\mathsf{Id}A}, \mathsf{pk}_{\mathsf{Id}A})$, $(\mathsf{sk}_{\mathsf{Id}B}, \mathsf{pk}_{\mathsf{Id}B})$, and $(\mathsf{sk}_{\mathsf{Id}O}, \mathsf{pk}_{\mathsf{Id}O})$, the secret key-public key pair of Alice, Bob, and Oscar respectively. They establish a multi-signature crypto wallet where each one of them holds a single key, and two out of three keys are required to carry out the transaction. In technical terms, this is also known as 2-of-3 seller-buyer with trustless escrow. Bob, the buyer sends funds to this address. Alice who is seller ships the product. If things go smoothly and the product arrives, Alice and Bob sign the transaction with their respective secret keys $\mathsf{sk}_{\mathsf{Id}A}$ and $\mathsf{sk}_{\mathsf{Id}B}$ by running MV-MSS.Sign among themselves, and money is transferred to Alice (see Fig. 4). If the product doesn't arrive, Oscar verifies this fact, and Bob and Oscar sign a transaction with their secret keys $\mathsf{sk}_{\mathsf{Id}B}$ and $\mathsf{sk}_{\mathsf{Id}O}$ by running MV-MSS.Sign among themselves, and send the funds to Bob. If the product arrives but Bob refuses to pay, Oscar verifies this fact, and Alice and Oscar sign the transaction sending funds to Alice. To sum up, MV-MSS not only provides compact signatures for Bitcoin transactions but also presents an economical and computational friendly solution. In addition, unlike classical schemes, MV-MSS is secure against attacks by quantum computers. Thus, it provides a long term secure solution for critical financial applications like Bitcoin.

VI. Security Analysis

In this section, we prove that the proposed MV-MSS attains existential unforgeability against a chosen message and chosen identity adversary.

Theorem 1: The proposed MV-MSS attains existential unforgeability under chosen message and chosen identity attack in the random oracle model under the assumption that MQ problem is NP-hard if

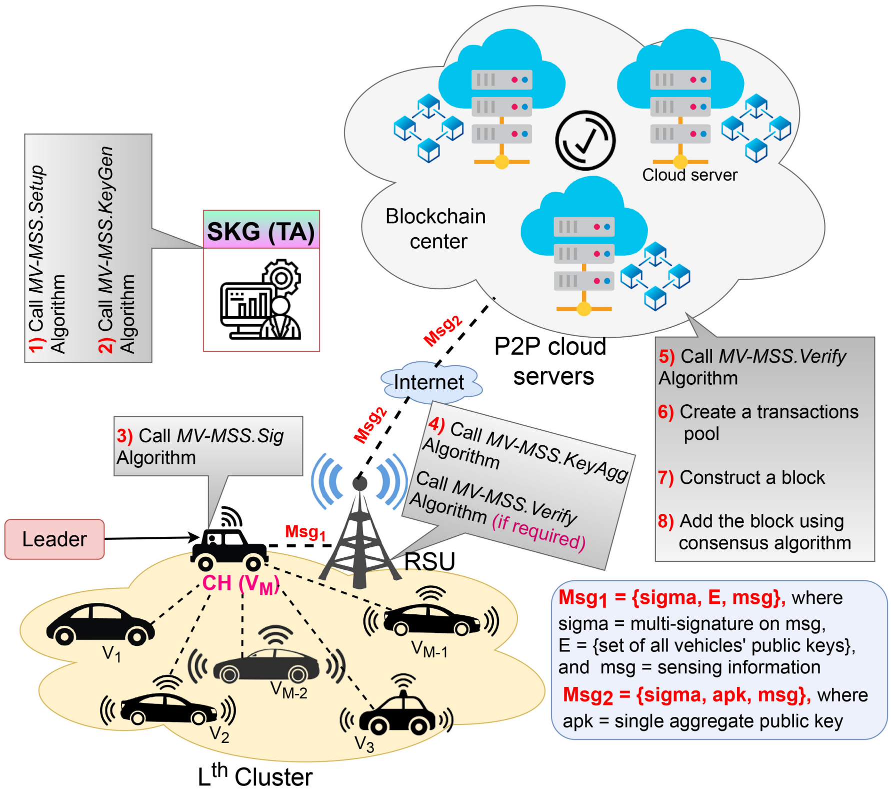


Fig. 3. Overall process in the proposed MV-MSS under blockchain context.

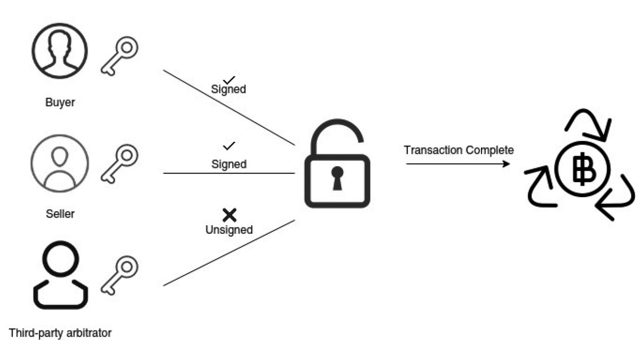


Fig. 4. Successful transaction under Bitcoin context.

1) the associated commitment scheme Commit is computationally binding and perfectly hiding,
2) the hash functions $\mathsf{Hash}_1, \mathsf{Hash}_2$ are designed as random oracles.

Proof: We show that MV-MSS is existential unforgeable under the chosen-message and chosen-identity attack (uf-cmia) under the hardness of MQ problem. We will prove the result by the method of contradiction. Let $\mathcal{Z}$ be a forger who possess, a non-negligible probability of success in the $uf - cmia$ game of MV-MSS. Then we will demonstrate the possibility of designing an oracle machine $\mathcal{B}$ for solving the MQ problem by executing $\mathcal{Z}$ and having a control over the outputs of the random oracles $\mathsf{Hash}_1, \mathsf{Hash}_2$ in a sequence of games $\mathsf{GA}_0, \ldots, \mathsf{GA}_4$. Here GA_i slightly modifies GA_{i-1} for $i \in \{1, 2, 3, 4\}$. Let $Pr[\mathsf{GA}_i]$ denote the probability of success $\mathcal{Z}$ has in the game GA_i.

GA_0: GA_0 is completely same as $uf - cmia$ game for MV-MSS. Therefore, $\mathsf{Adv}_{\mathcal{Z}}^{\mathsf{Ex}_{\text{uf-cmia}}^{\text{MV-MSS}}} = Pr[\mathsf{Ex}_{\text{uf-cmia}}^{\text{MV-MSS}} = 1] = Pr[\mathsf{GA}_0]$

GA_1: GA_1 is exactly same as GA_0 except that during extract query the oracle $\mathcal{B}$ substitutes the output of the random oracle Hash_1 query of $\mathsf{Id} \in \{0, 1\}^*$ by $\mathsf{k} = \mathcal{P}(\mathsf{u})$ and the corresponding secret key by u for randomly chosen $\mathsf{u} \in \mathbb{F}_q^n$. Note that $|Pr[\mathsf{GA}_1] - Pr[\mathsf{GA}_0]|$ is non negligible means it is possible to employ $\mathcal{Z}$ for distinguishing the random oracle Hash_1's output distributions, which is impossible. Therefore, $|Pr[\mathsf{GA}_1] - Pr[\mathsf{GA}_0]| = \epsilon_1(\eta)$, for some negligible function $\epsilon_1(\eta)$.

GA_2: This game is similar to GA_1 except that during Sign-query, the oracle $\mathcal{B}$ replaces output of random oracle Hash_1 of $\mathsf{Id} \in \{0, 1\}^*$ by $\mathsf{k} = \mathcal{P}(\mathsf{u}) \in \mathbb{F}_q^m$ for randomly chosen $\mathsf{u} \in \mathbf{F}_q^n$ and the signature by σ which is generated using secret key u for the system $\mathsf{k} = \mathcal{P}(\mathsf{u})$. Now if $|Pr[\mathsf{GA}_2] - Pr[\mathsf{GA}_1]|$ is non-negligible then we may use $\mathcal{Z}$ to distinguish the output distributions of the random oracle Hash_1, which is impossible. Therefore, there exists a negligible function $\epsilon_2(\eta)$ such that $|Pr[\mathsf{GA}_2] - Pr[\mathsf{GA}_1]| = \epsilon_2(\eta)$.

GA_3: GA_3 runs in a similar fashion as GA_2 except that the oracle $\mathcal{B}$ substitutes the output of Hash_2 by random element from $\{0, 1, 2, 3\}$. Note that $|Pr[\mathsf{GA}_3] - Pr[\mathsf{GA}_2]|$ is non-negligible means it is possible to employ $\mathcal{Z}$ for distinguishing random oracle Hash_2's output distributions. It is impossible. Thus, $|Pr[\mathsf{GA}_3] - Pr[\mathsf{GA}_2]| = \epsilon_3(\eta)$, for some negligible function $\epsilon_3(\eta)$.

GA_4: GA_4 is same as GA_3 except that $\mathcal{B}$ substitutes the output of the random oracle Hash_1 query of Id^* by randomly chosen $\mathsf{k}^* \in \mathbb{F}_q^m$, the output of Hash_2 query by a random element of $\{0,1,2,3\}$. Using the similar arguments as of GA_2 and GA_3, we can argue that there exist some negligible function $\epsilon_4(\eta)$ such that $|Pr[\mathsf{GA}_4] - Pr[\mathsf{GA}_3]| = \epsilon_4(\eta)$.

Note that $|Pr[\mathsf{GA}_4] - Pr[\mathsf{Ex}^{\mathsf{MV\text{-}MSS}}_{\mathsf{uf\text{-}cmia}} = 1]| = |Pr[\mathsf{GA}_4] - Pr[\mathsf{GA}_0]| \leq |Pr[\mathsf{GA}_4] - Pr[\mathsf{GA}_3]| + |Pr[\mathsf{GA}_3] - Pr[\mathsf{GA}_2]| + |Pr[\mathsf{GA}_2] - Pr[\mathsf{GA}_1]| + |Pr[\mathsf{GA}_1] - Pr[\mathsf{GA}_0]| = \epsilon_4(\eta) + \epsilon_3(\eta) + \epsilon_2(\eta) + \epsilon_1(\eta) = \rho(\eta)$, a negligible function. Thus the success probability of $\mathcal{Z}$ in GA_4 is same as the success probability of $\mathsf{Adv}_{\mathcal{Z}}^{\mathsf{Ex}^{\mathsf{MV\text{-}MSS}}_{\mathsf{uf\text{-}cmia}}} = Pr[\mathsf{Ex}^{\mathsf{MV\text{-}MSS}}_{\mathsf{uf\text{-}cmia}} = 1]$ of $\mathcal{Z}$ in the game $uf - cmia$. This implies $Pr[\mathsf{GA}_4]$ is non-negligible since $\mathsf{Adv}_{\mathcal{Z}}^{\mathsf{Ex}^{\mathsf{MV\text{-}MSS}}_{\mathsf{uf\text{-}cmia}}}$ is so by our assumption.

We demonstrate below that $\mathcal{B}$ can solve MQ problem by finding a solution u^* of the system $\mathsf{k}^* = \mathcal{P}(y)$ with the assistance of $\mathcal{Z}$ such that $\mathsf{k}^* = \mathcal{P}(\mathsf{u}^*)$.

1) $\mathcal{B}$ produces valid transcripts $(COM, cl^i, RSP^i)_{i=1,2,3}$ with the help of $\mathcal{Z}$ and controlling the output of random oracles $\mathsf{Hash}_1, \mathsf{Hash}_2$, where $COM = \mathsf{Commit}(A_0|| A_1||\ldots||A_5)$, $cl^1 = 1, cl^2 = 2, cl^3 = 3$, $RSP^1 = (Rsp_1, \ldots, Rsp_M, A_0, A_3)$, $RSP^2 = (Rsp_1, \ldots, Rsp_M, A_1, A_4)$, $RSP^3 = (Rsp_1, \ldots, Rsp_M, A_1, A_2, A_5)$.

2) Let $\widetilde{a}_k^{(i,j)}$ denote the value of $\widetilde{a}_k^{(i)}$ as computed in $\mathsf{MV\text{-}MSS.Verify}$ for a signer U_i corresponding to challenge j. Then by the binding property of the commitment scheme, we have

$$\widetilde{a}_0^{(1,2)} = \widetilde{a}_0^{(1,3)}, \ldots, \widetilde{a}_0^{(M,2)} = \widetilde{a}_0^{(M,3)} \tag{2}$$

$$\widetilde{a}_4^{(1,1)} = < /list - item > \widetilde{a}_4^{(1,3)}, \ldots, \widetilde{a}_4^{(M,1)} = \widetilde{a}_4^{(M,3)} \tag{3}$$

$$\widetilde{a}_5^{(1,1)} = \widetilde{a}_5^{(1,2)}, \ldots, \widetilde{a}_5^{(M,1)} = \widetilde{a}_5^{(M,2)} \tag{4}$$

3) From Eqs. (2), (3) and (4), we can write the following for $i = 1, \ldots M$:

$\mathsf{Commit}(r_0^{(i,2)}, \mathcal{G}(r_0^{(i,2)}, d_1^{(i,2)}) + u_1^{(i,2)}) = \mathsf{Commit}(r_0^{(i,3)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(r_0^{(i,3)}) - \mathcal{G}(r_0^{(i,3)}, d_0^{(i,3)}) - u_0^{(i,3)})$, $\mathsf{Commit}(r_1^{(i,1)} - d_1^{(i,1)}, \mathcal{P}(r_1^{(i,1)}) - u_1^{(i,1)}) = \mathsf{Commit}(d_0^{(i,3)}, u_0^{(i,3)})$, $\mathsf{Commit}(d_1^{(i,1)}, u_1^{(i,1)}) = \mathsf{Commit}(d_1^{(i,2)}, u_1^{(i,2)})$.

Let t be the index in $\{1, \ldots, M\}$ corresponding to the user with identity Id^*. Then we may write the following equations:

$$\begin{aligned}&\mathsf{Commit}(r_0^{(t,2)}, \mathcal{G}(r_0^{(t,2)}, d_1^{(t,2)}) + u_1^{(t,2)})\\ &\quad = \mathsf{Commit}(r_0^{(t,3)}, \mathsf{k}^* - \mathcal{P}(r_0^{(t,3)}) - \mathcal{G}(r_0^{(t,3)}, d_0^{(t,3)}) - u_0^{(t,3)})\end{aligned} \tag{5}$$

$$\begin{aligned}&\mathsf{Commit}(r_1^{(t,1)} - d_1^{(t,1)}, \mathcal{P}(r_1^{(t,1)}) - u_1^{(t,1)})\\ &\quad = \mathsf{Commit}(d_0^{(t,3)}, u_0^{(t,3)})\end{aligned} \tag{6}$$

$$\mathsf{Commit}(d_1^{(t,1)}, u_1^{(t,1)}) = \mathsf{Commit}(d_1^{(t,2)}, u_1^{(t,2)}) \tag{7}$$

4) Using the binding property of the Commit, we derive the following relations:

$$r_0^{(t,2)} = r_0^{(t,3)} \ \text{ using Eq. (5)} \tag{8}$$

$$\begin{aligned}\mathcal{G}(r_0^{(t,2)}, d_1^{(t,2)}) + u_1^{(t,2)} &= \mathsf{k}^* - \mathcal{P}(r_0^{(t,3)})\\ &- \mathcal{G}(r_0^{(t,3)}, d_0^{(t,3)}) - u_0^{(t,3)} \ \text{ using Eq. (5)}\end{aligned} \tag{9}$$

$$r_1^{(t,1)} - d_1^{(t,1)} = d_0^{(t,3)} \ \text{ using Eq. (6)} \tag{10}$$

$$\mathcal{P}(r_1^{(t,1)}) - u_1^{(t,1)} = u_0^{(t,3)} \ \text{ from Eq. (6)} \tag{11}$$

$$d_1^{(t,1)} = d_1^{(t,2)} \ \text{ from Eq. (7)} \tag{12}$$

$$u_1^{(t,1)} = u_1^{(t,2)} \ \text{ from Eq. (7)} \tag{13}$$

5) From Eq. (9), we have: $\mathsf{k}^* - \mathcal{P}(r_0^{(t,3)}) - \mathcal{G}(r_0^{(t,3)}, d_0^{(t,3)}) - u_0^{(t,3)} = \mathcal{G}(r_0^{(t,2)}, d_1^{(t,2)}) + u_1^{(t,2)}$, that is,

$$\begin{aligned}\mathsf{k}^* &= \mathcal{P}(r_0^{(t,3)}) + \mathcal{G}(r_0^{(t,3)}, d_0^{(t,3)}) + u_0^{(t,3)}\\ &\quad + \mathcal{G}(r_0^{(t,2)}, d_1^{(t,2)}) + u_1^{(t,2)}\end{aligned} \tag{14}$$

6) From Eqs. (8) and (14), and using the bilinearity of $\mathcal{G}$, we get: $\mathsf{k}^* = \mathcal{P}(r_0^{(t,2)}) + \mathcal{G}(r_0^{(t,2)}, d_0^{(t,3)} + d_1^{(t,2)}) + u_0^{(t,3)} + u_1^{(t,2)}$. Then, using Eqs. (12) and (13), we finally obtain $\mathsf{k}^* = \mathcal{P}(r_0^{(t,2)} + r_1^{(t,1)})$.

7) Thus the oracle machine $\mathcal{B}$ extracts a solution $r_0^{(t,2)} + r_1^{(t,1)}$ of $\mathsf{k}^* = \mathcal{P}(y)$ i.e., $\mathbf{k}^* = \mathcal{P}(r_0^{(t,2)} + r_1^{(t,1)})$.

Thus, $Pr[\mathbf{Ga}_4]$ is non-negligible implies $\mathcal{B}$ is able to determine a solution of the MQ problem $\mathsf{k}^* = \mathcal{P}(\mathsf{y})$. It contradicts the assumption that MQ problem is NP-hard. Consequently, $Pr[\mathbf{Ga}_4]$ is negligible which ensures that $\mathsf{Adv}_{\mathcal{Z}}^{\mathsf{Ex}^{\mathsf{MV\text{-}MSS}}_{\mathsf{uf\text{-}cmia}}} = Pr[\mathsf{Ex}^{\mathsf{MV\text{-}MSS}}_{\mathsf{uf\text{-}cmia}} = 1]$ is negligible. Therefore, we may conclude that the proposed MV-MSS attains existential unforgeability against a chosen message and chosen identity adversary. ■

VII. Performance Analysis

In this section, we provide a detailed comparative study on overheads including computation, communication and storage complexity of the proposed scheme (MV-MSS) and other relevant schemes. Next, through the blockchain simulation study we show the effectiveness of the proposed MV-MSS.

A. Overheads Comparison

We now discuss the communication and storage complexity of our proposed MV-MSS. Note that the size of public key pk is $\frac{m(n+2)(n+1)}{2}$ field ($\mathbb{F}_q$) elements. Size of master secret key is $n^2 + m^2 + C$ field ($\mathbb{F}_q$) elements. Here m denote the number of equations, n denote the number of variables, and C denote the size of the central map of the underlying MPKC. The public key of an user, that is k_{Id} is an element of $\mathbb{F}_q^m$, and the secret key computed by $\mathsf{sk}_{\mathsf{Id}} = \mathcal{P}^{-1}(\mathsf{k}_{\mathsf{Id}})$ is an element of $\mathbb{F}_q^n$. Therefore their sizes are m and n field ($\mathbb{F}_q$) elements respectively. We now have a look at the signature size. The signature size is $4\gamma|\mathsf{Commit}| + (3n + 2m)M\gamma$ field ($\mathbb{F}_q$) elements, where $|\mathsf{Commit}|$ denote the size of the commitment scheme, and γ denote the number of rounds of the underlying identification scheme. We refer to Table I for a summary of communication and storage complexity of our MV-MSS.

TABLE I
SUMMARY OF COMMUNICATION AND STORAGE OVERHEADS OF MV-MSS

pk size	$\frac{m(n+2)(n+1)}{2}$ field ($\mathbb{F}_q$) elements
MSK size	n^2+m^2+C field ($\mathbb{F}_q$) elements
User public key size	m field ($\mathbb{F}_q$) elements
Secret key size	n field ($\mathbb{F}_q$) elements
Signature size	$4\gamma\lvert\mathsf{Commit}\rvert + (3n+2m)M\gamma$ field ($\mathbb{F}_q$) elements

TABLE II
COMPARISON FOR 128-BIT SECURITY LEVEL OVER $GF(256)$ WITH $M = 5$

Scheme	Public key size (kB)	Signature size (kB)	User secret key size (kB)	$uf\text{-}cmia$ security
IB-UOV [46] $(256, 45, 90)$	409.4	714.4	942.2	×
IBS-Rainbow [47] $(256, 40, 24, 24)$	187.7	395.7	431.7	✓
ID-Rainbow [48] $(256, 28, 20, 20, 8)$	46694.4	0.1	70	×
MV-IBS [49]	136.2	1400.12	8.01	✓
MV-MSS	136.2	217.9	0.1	✓

During the signature generation, the map $\mathcal{P}$ is executed 6γ times by for the computation of $\mathcal{G}$ in the evaluation of a_0 and a_1, and 2γ times for the computation of u_1 and e_1 by each user U_i. In the verification phase, the verifier calculates the system $\mathcal{P}$ $\frac{(4+4+4+4)\gamma M}{4}$ i.e., $4\gamma M$ times. Hence, number of times the system $\mathcal{P}$ is computed is $12\gamma M$. Note that to execute $\mathcal{P}$ once, one needs to carry out $m(n^4+n)$ field multiplications. Thereby, we require total $12m(n^4+n)\gamma M$ modulo multiplications.

We now compare our scheme with secure multivariate identity-based signature schemes [46]–[49]. We analyze the performance for 128-bit level security in terms of sizes of public key, user secret key and signature. Results are summarized in Table II. We take Rainbow with parameters $(q = 256, v = 36, o_1 = 28, o_2 = 15)$ [50] as the secure MQ signature scheme for MV-MSS. The value of M is considered as 5 since in financial transaction it is less than or equal to 5 [28]. For example, most common Bitcoin multi-signature addresses are 2-of-3, 2-of-2, 3-of-3, 3-of-4. The impersonation probability of our MV-MSS is $\frac{1}{2}$ due to the use of one round of identification protocol of [34] to generate the signature. Thereby, to attain the needed security level γ, i.e., $2^{-\gamma}$, we need to repeat the identification protocol for γ rounds. Therefore, we take number of rounds to be 128 for 128-bit security level over $GF(256)$. The output length of the commitment scheme Commit is 256 bytes if we instantiate it with SHA3-256.

The analysis in Table II also shows that the size of multi-signature outputted by MV-MSS is smaller than the size of a single signature for the scheme that achieves $uf-cmia$ security. To support our claim, we analyze the size of signature required in Bitcoin transactions where $M = 5$ parties are required to validate the transaction. We now analyze the communication aspects of MV-MSS by using the results presented in Table III. Note that the minimum size of an individual signature among the scheme that achieves uf-cmia security is 395.7 Kb. Since the signatures of five parties are required to validate the transactions, the total communication overhead, if we don't use the multi-signature, would be 1978.5 Kb. By employing MV-MSS, the five parties can conjointly sign the transaction using a signature of length only 217.9 Kb. Therefore, we see that employing MV-MSS in multi-party Bitcoin transactions greatly reduces the communication overhead (total signature size).

TABLE III
PERCENTAGE REDUCTION IN TOTAL SIGNATURE COST

Minimum size of an individual signature (Kb) (among the scheme that achieves $uf\text{-}cmia$ security)	Total size of signature when $M = 5$ parties are required to validate the transaction (Kb)	Size of signature outputted by MV-MSS when $M = 5$ parties are required to validate the transaction (Kb)	Percentage reduction in total signature cost
395.7	$395.7 \times 5 = 1978.5$	217.9	88.98%

TABLE IV
COMPARISON OF MV-MSS WITH OTHER NON-MULTIVARIATE BASED MSS

	Public key size	Multi-signature size	User secret key size	Security assumption
Boneh *et al.* [51]	$\lvert\mathbb{G}_2\rvert$	$\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{Z}_q\rvert$	co-DH
Drijvers *et al.* [52]	$\lvert\mathbb{G}\rvert+2\lvert\mathbb{Z}_q\rvert$	$2\lvert\mathbb{G}\rvert+3\lvert\mathbb{Z}_q\rvert$	$\lvert\mathbb{Z}_q\rvert$	DL
Drijvers *et al.* [54]	$\lvert\mathbb{G}\rvert+2\lvert\mathbb{Z}_q\rvert$	$2\lvert\mathbb{G}\rvert$	$\mathcal{O}(T^2)$	$l-wBDHI_3^*$
Maxwell *et al.* [53]	$\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}\rvert+\lvert\mathbb{Z}_q\rvert$	$\lvert\mathbb{Z}_q\rvert$	DL
Bansarkhani and Sturm [55]	$\mathcal{O}(b)$	$\mathcal{O}(b)$	$\mathcal{O}(b)$	Ring-SIS
MV-MSS	m	$4\gamma\lvert\mathsf{Commit}\rvert+(3n+2m)M\gamma$	n	MPKC

Note: co-DH: computational Diffie-Hellman, DL: discrete logarithm, $l-wBDHI_3^*$: weak bilinear Diffie-Hellman inversion problem for type-3 pairings, SIS: short integer solution, MPKC: multivariate public key cryptography. $\mathbb{G}, \mathbb{G}_1, \mathbb{G}_2$ are groups of prime order q, $\lvert\mathbb{G}\rvert$: bit size of an element of the group $\mathbb{G}$, T: max number of time periods in forward secrecy, λ: security parameter, $b = \mathcal{O}(\lambda)$, γ: number of rounds the underlying identification scheme, $\lvert\mathsf{Commit}\rvert$: the size of the underlying commitment scheme, m: number of variables in the MQ system, n: number of equations in the MQ system.

TABLE V
TIME COMPLEXITY OF MV-MSS FOR 80-BIT SECURITY LEVEL OVER $GF(31)$

Algorithm	Time (in seconds)
Setup	1.91
Key Generation	0.056
Signature	0.047
Verification	0.035

Now, we also compare our proposed design with other existing non-multivariate based MSS schemes. The results are documented in Table IV. This table shows that the schemes of Bohen *et al.* [51], Drijvers *et al.* [52], and Maxwell *et al.* [21] are based on computational hard problems, like discrete logarithm (DL) and computational Diffie-Hellman (co-DH) problems, which can be broken in quantum computing. Thus, these schemes may not be secure against quantum attacks. On the other hand, the proposed MV-MSS scheme depends on the MPKC, which resists various quantum attacks.

To assess the running time complexity of MV-MSS, we implemented our proposed design in free and open source mathematical software SageMath (version 9.2) with workstation comprising an Intel Core i5 1.60 GHz processor with 64-bit Linux Lite (v 5.2) operating system. We utilize Rainbow signature scheme with $(q = 256, v_1 = 18, o_1 = 17, o_2 = 9)$ [50] as the underlying secure multivariate signature scheme. Results are documented in Table V.

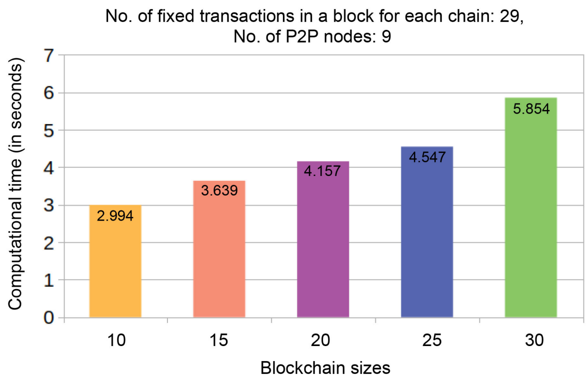


Fig. 5. Blockchain simulation for case 1 in the proposed MV-MSS.

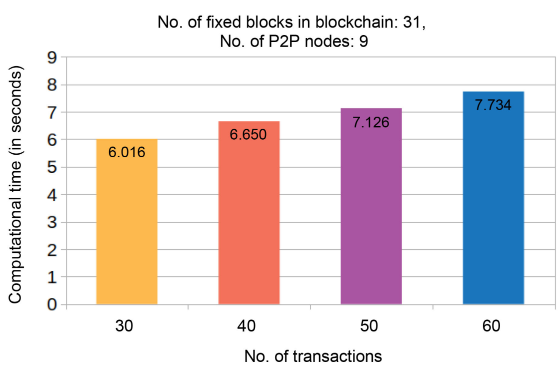


Fig. 6. Blockchain simulation for case 2 in the proposed MV-MSS.

B. Blockchain Simulation Study: Practical Perspective

In this section, we perform the blockchain simulation over a decentralized P2P distributed system, where the participant nodes are considered as the servers in a distributed P2P network. The total number of the nodes is taken as 9. Here, the considered servers run under a configuration setting: "Ubuntu 18.04.3 LTS, Intel Core i5-8400 CPU @ 2.80 GHz× 6, Memory 7.6 GiB, OS type 64-bit, disk 152.6 GB" and the script was written in node.js with VS CODE 2019. Since the block addition into the blockchain requires a distributed consensus algorithm, we consider a voting based consensus mechanism, called PBFT consensus algorithm for block addition into the blockchain center.

The simulation is performed under three scenarios: 1) Scenario 1, 2) Scenario 2, and 3) Scenario 3. The details of these scenarios are discussed below.

Scenario 1: Under this scenario, each block contains a fixed number of transactions as 29, and the created blockchain holds a varied number of such blocks, that is, we vary the blockchain sizes. The simulation results provided in Fig. 5 demonstrate the computational time (in seconds) that indicates that the total time for creating the blockchain having various blocks with a fixed number of transactions. It is observed that the computational time increases linearly with an increased number of blocks.

Scenario 2: In this scenario, the constructed blockchain comprises a fixed number of blocks as 31, where each block is capable to hold a varied number of transactions. The simulation outcomes shown in Fig. 6 illustrate that the computational time (which is the time required to construct a complete blockchain) increases linearly for constructing the blockchain when a varied number of transactions are present in the blocks.

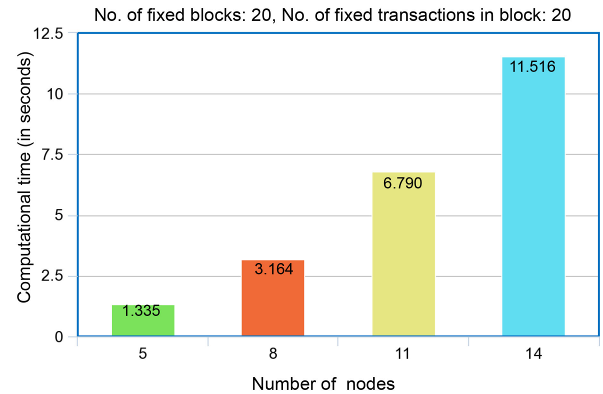


Fig. 7. Blockchain simulation for case 3 in the proposed MV-MSS.

Scenario 3: In this case, we fix the number of blocks in each chain at 20 and the number of transactions in each block is also fixed at 20. However, we vary the number of peer nodes in the P2P network. The simulation results for this scenario are provided in Fig. 7. The results indicate that whenever the peer nodes are increased in the network, the total computational time (in seconds) also increases linearly.

VIII. Conclusion

This work presented the design and analysis of a provably secure multivariate identity-based multi-signature scheme, namely MV-MSS. The proposed design achieves existential unforgeability under chosen message and chosen identity attack in the ROM. To the extent of our knowledge, MV-MSS is the *first* multivariate based MSS. MV-MSS is fast, inexpensive, efficient, and requires only modest computational resources for working owing to the fact that it is a MPKC based scheme. The proposed scheme is then incorporated in an IoV environment with the help of the blockchain technology.

Some future works can be as follows.

- The proposed MV-MSS is suitable for Bitcoin transactions where more than one key is required to certify the currency exchange. Since MV-MSS is secure against attacks by quantum computers, it would be interesting to provide a long term secure solution for critical financial applications like Bitcoin. We pointed out that the proposed application of MV-MSS to Bitcoin transactions is not only limited to Bitcoins, but it can be also extended to Ethereum and other cryptocurrencies. In other words, a multi-signature wallet can be set up for any cryptocurrency transactions with the MV-MSS as the fundamental building block.
- In an Internet of Drones (IoD) environment, multiple drones are connected in order to collect the information from a certain flying zone. Therefore, to resolve the security and privacy concerns in IoD, the proposed MV-MSS can be also applied. Thus, we would like to explore this direction in future for IoD environment.

- We have also future plan to include mobility issue of vehicles and communication link reliability in the proposed scheme for the blockchain-based IoV applications.
- By making use of post-quantum safe MV-MSS as a cryptographic building block, we can build robust and cost friendly system which logically addresses the challenges up front.
- The proposed design is proven to be secure in the random oracle model. Thus, another future direction of research would be to prove the security under the standard model.

APPENDIX
CORRECTNESS OF MV-MSS

In order to prove the correctness of the scheme, we have to check the existence of the equalities:

$COM \stackrel{?}{=} \mathsf{Commit}(A_0, \widetilde{A}_1, A_2, \widetilde{A}_3, \widetilde{A}_4, A_5)$ when $cl = 0$,
$COM \stackrel{?}{=} \mathsf{Commit}(A_0, \widetilde{A}_1, \widetilde{A}_2, A_3, \widetilde{A}_4, \widetilde{A}_5)$ when $cl = 1$,
$COM \stackrel{?}{=} \mathsf{Commit}(\widetilde{A}_0, A_1, \widetilde{A}_2, \widetilde{A}_3, A_4, \widetilde{A}_5)$ when $cl = 2$,
$COM \stackrel{?}{=} \mathsf{Commit}(\widetilde{A}_0, A_1, A_2, \widetilde{A}_3, \widetilde{A}_4, A_5)$ when $cl = 3$.

Case I $(cl = 0)$: In order to verify $COM \stackrel{?}{=} \mathsf{Commit}$ $(A_0, \widetilde{A}_1, A_2, \widetilde{A}_3, \widetilde{A}_4, A_5)$, the following equalities are needed to be validated:

$\widetilde{A}_1 = \mathsf{Commit}(\widetilde{a}_1^{(1)}, \ldots, \widetilde{a}_1^{(M)}) = A_1$,
$\widetilde{A}_3 = \mathsf{Commit}(\widetilde{a}_3^{(1)}, \ldots, \widetilde{a}_3^{(M)}) = A_3$,
$\widetilde{A}_4 = \mathsf{Commit}(\widetilde{a}_4^{(1)}, \ldots, \widetilde{a}_4^{(M)}) = A_4$.

In particular, it is enough to prove that for $i \in \{1, \ldots, M\}$ $\widetilde{a}_1^{(i)} = a_1^{(i)}$, $\widetilde{a}_3^{(i)} = a_3^{(i)}$, and $\widetilde{a}_4^{(i)} = a_4^{(i)}$.

We have $\widetilde{a}_1^{(i)} = \mathsf{Commit}(r_1^{(i)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(r_1^{(i)}) - \mathcal{G}(t_1^{(i)}, r_1^{(i)}) - e_1^{(i)}) = \mathsf{Commit}(r_1^{(i)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(r_1^{(i)}) - \mathcal{G}(r_0^{(i)} - t_0^{(i)}, r_1^{(i)}) - e_1^{(i)})$ [using $t_1^{(i)} = r_0^{(i)} - t_0^{(i)}$] $= \mathsf{Commit}(r_1^{(i)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(r_1^{(i)}) - \mathcal{G}(r_0^{(i)}, r_1^{(i)}) + \mathcal{G}(t_0^{(i)}, r_1^{(i)}) - e_1^{(i)})$ [using the bilinearity of $\mathcal{G}$] $= \mathsf{Commit}(r_1^{(i)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(r_1^{(i)}) - \mathcal{G}(r_0^{(i)}, r_1^{(i)}) + \mathcal{G}(t_0^{(i)}, r_1^{(i)}) - \mathcal{P}(r_0^{(i)}) + e_0^{(i)})$ [using $e_1^{(i)} = \mathcal{P}(r_0^{(i)}) - e_0^{(i)}$] $= \mathsf{Commit}(r_1^{(i)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(r_0^{(i)} + r_1^{(i)}) + \mathcal{G}(t_0^{(i)}, r_1^{(i)}) + e_0^{(i)})$ [expanding $\mathcal{G}(r_0^{(i)}, r_1^{(i)})$] $= \mathsf{Commit}(r_1^{(i)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(\mathsf{u}_{\mathsf{Id}i}) + \mathcal{G}(t_0^{(i)}, r_1^{(i)}) + e_0^{(i)})$ [since $r_1^{(i)} = \mathsf{u}_{\mathsf{Id}i} - r_0^{(i)}$]$= \mathsf{Commit}(r_1^{(i)}, \mathcal{G}(t_0^{(i)}, r_1^{(i)}) + e_0^{(i)})$ [as $\mathsf{k}_{\mathsf{Id}i} = \mathcal{P}(\mathsf{u}_{\mathsf{Id}i})$]$= a_1^{(i)}$.

It then follows from the definition that $\widetilde{a}_3^{(i)} = \mathsf{Commit}(t_1^{(i)}, e_1^{(i)}) = a_3^{(i)}$, $\widetilde{a}_4^{(i)} = \mathsf{Commit}(d_0^{(i)}, u_0^{(i)}) = a_4^{(i)}$.

Case II $(cl = 1)$: To demonstrate $COM \stackrel{?}{=} \mathsf{Commit}(A_0, \widetilde{A}_1, \widetilde{A}_2, A_3, \widetilde{A}_4, \widetilde{A}_5)$, we need to check the correctness of following equalities: $\widetilde{A}_1 = \mathsf{Commit}(\widetilde{a}_1^{(1)}, \ldots, \widetilde{a}_1^{(M)}) = A_1$, $\widetilde{A}_2 = \mathsf{Commit}(\widetilde{a}_2^{(1)}, \ldots, \widetilde{a}_2^{(M)}) = A_2$, $\widetilde{A}_4 = \mathsf{Commit}(\widetilde{a}_4^{(1)}, \ldots, \widetilde{a}_4^{(M)}) = A_4$ and $\widetilde{A}_5 = \mathsf{Commit}(\widetilde{a}_5^{(1)}, \ldots, \widetilde{a}_5^{(M)}) = A_5$. In particular, it is sufficient to verify that $\widetilde{a}_1^{(i)} = a_1^{(i)}$, $\widetilde{a}_2^{(i)} = a_2^{(i)}$, $\widetilde{a}_4^{(i)} = a_4^{(i)}$, and $\widetilde{a}_5^{(i)} = a_5^{(i)}$. Now, it follows that $\widetilde{a}_4^{(i)} = \mathsf{Commit}(r_1^{(i)} - d_1^{(i)}, \mathcal{P}(r_1^{(i)}) - u_1^{(i)}) = \mathsf{Commit}(d_0^{(i)}, \mathcal{P}(r_1^{(i)}) - u_1^{(i)})$ [using $d_1^{(i)} = r_1^{(i)} - d_0^{(i)}$] $= \mathsf{Commit}(d_0^{(i)}, u_0^{(i)})$ [using $u_1^{(i)} = \mathcal{P}(r_1^{(i)}) - u_0^{(i)}$]$= a_4^{(i)}$.

Case III $(cl = 2)$: To check $COM \stackrel{?}{=} \mathsf{Commit}$ $(\widetilde{A}_0, A_1, \widetilde{A}_2, \widetilde{A}_3, A_4, \widetilde{A}_5)$, we need to verify the existence of the following equalities:

$\widetilde{A}_0 = \mathsf{Commit}(\widetilde{a}_0^{(1)}, \ldots, \widetilde{a}_0^{(M)}) = A_0$, $\widetilde{A}_2 = \mathsf{Commit}(\widetilde{a}_2^{(1)}, \ldots, \widetilde{a}_2^{(M)}) = A_2$, $\widetilde{A}_3 = \mathsf{Commit}(\widetilde{a}_3^{(1)}, \ldots, \widetilde{a}_3^{(M)}) = A_3$ and $\widetilde{A}_5 = \mathsf{Commit}(\widetilde{a}_5^{(1)}, \ldots, \widetilde{a}_5^{(M)}) = A_5$.

To be specific, it is sufficient to demonstrate: $\widetilde{a}_0^{(i)} = a_0^{(i)}$, $\widetilde{a}_2^{(i)} = a_2^{(i)}$, $\widetilde{a}_3^{(i)} = a_3^{(i)}$, and $\widetilde{a}_5^{(i)} = a_5^{(i)}$. We then have: $\widetilde{a}_3^{(i)} = \mathsf{Commit}(r_0^{(i)} - t_0^{(i)}, \mathcal{P}(r_0^{(i)}) - e_0^{(i)}) = \mathsf{Commit}(t_1^{(i)}, \mathcal{P}(r_0^{(i)}) - e_0^{(i)})$ [using $t_1^{(i)} = r_0^{(i)} - t_0^{(i)}$] $= \mathsf{Commit}(t_1^{(i)}, e_1^{(i)})$ [using $e_1^{(i)} = \mathcal{P}(r_0^{(i)}) - e_0^{(i)}$]$= a_3^{(i)}$.

Case IV $(cl = 3)$: To show $COM \stackrel{?}{=} \mathsf{Commit}(\widetilde{A}_0, A_1, A_2, \widetilde{A}_3, \widetilde{A}_4, A_5)$, we need to ensure the validity of following equalities:

$\widetilde{A}_0 = \mathsf{Commit}(\widetilde{a}_0^{(1)}, \ldots, \widetilde{a}_0^{(M)}) = A_0$, $\widetilde{A}_3 = \mathsf{Commit}(\widetilde{a}_3^{(1)}, \ldots, \widetilde{a}_3^{(M)}) = A_3$, $\widetilde{A}_4 = \mathsf{Commit}(\widetilde{a}_4^{(1)}, \ldots, \widetilde{a}_4^{(M)}) = A_4$.

Particularly, it is enough to show that $\widetilde{a}_0^{(i)} = a_0^{(i)}$, $\widetilde{a}_3^{(i)} = a_3^{(i)}$, $\widetilde{a}_4^{(i)} = a_4^{(i)}$.

We then have: $\widetilde{a}_0^{(i)} = \mathsf{Commit}(r_0^{(i)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(r_0^{(i)}) - \mathcal{G}(r_0^{(i)}, d_0^{(i)}) - u_0^{(i)}) = \mathsf{Commit}(r_0^{(i)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(r_0^{(i)}) - \mathcal{G}(r_0^{(i)}, r_1^{(i)} - d_1^{(i)}) - u_0^{(i)})$ [as $d_1^{(i)} = r_1^{(i)} - d_0^{(i)}$] $= \mathsf{Commit}(r_0^{(i)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(r_0^{(i)}) - \mathcal{G}(r_0^{(i)}, r_1^{(i)}) + \mathcal{G}(r_0^{(i)}, d_1^{(i)}) - u_0^{(i)})$ [using the bilinearity of $\mathcal{G}$] $= \mathsf{Commit}(r_0^{(i)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(r_0^{(i)}) - \mathcal{G}(r_0^{(i)}, r_1^{(i)}) + \mathcal{G}(r_0^{(i)}, d_1^{(i)}) - \mathcal{P}(r_1^{(i)}) + u_1^{(i)})$ [since $u_1^{(i)} = \mathcal{P}(r_1^{(i)}) - u_0^{(i)}$]$= \mathsf{Commit}(r_0^{(i)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(r_0^{(i)} + r_1^{(i)}) + \mathcal{G}(r_0^{(i)}, d_1^{(i)}) + u_1^{(i)})$ [by expanding $\mathcal{G}(r_0^{(i)}, r_1^{(i)})$] $= \mathsf{Commit}(r_0^{(i)}, \mathsf{k}_{\mathsf{Id}i} - \mathcal{P}(\mathsf{u}_{\mathsf{Id}i}) + \mathcal{G}(r_0^{(i)}, d_1^{(i)}) + u_1^{(i)})$ [since $r_1^{(i)} = \mathsf{u}_{\mathsf{Id}i} - r_0^{(i)}$] $= \mathsf{Commit}(r_0^{(i)}, \mathcal{G}(r_0^{(i)}, d_1^{(i)}) + u_1^{(i)})$ [as $\mathsf{k}_{\mathsf{Id}i} = \mathcal{P}(\mathsf{u}_{\mathsf{Id}i})$]$= a_0^{(i)}$.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the Associate Editor for their valuable suggestions and feedback on the article.

REFERENCES

[1] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibanez, "Internet of Vehicles: Architecture, protocols, and security," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3701–3709, Oct. 2018.
[2] K. Z. Ghafoor, K. Abu Bakar, J. Lloret, R. H. Khokhar, and K. C. Lee, "Intelligent beaconless geographical forwarding for urban vehicular environments," *Wireless Netw.*, vol. 19, no. 3, pp. 345–362, 2013.
[3] A. DeekshaKumar and M. Bansal, "A review on VANET security attacks and their countermeasure," in *Proc. 4th Int. Conf. Signal Process., Comput. Control*, 2017, pp. 580–585.
[4] F. Qu, Z. Wu, F. Wang, and W. Cho, "A security and privacy review of VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 2985–2996, Dec. 2015.
[5] A. Samad, S. Alam, M. Shuaib, and M. Bokhari, "Internet of Vehicles (IoV) requirements, attacks and countermeasures," in *Proc. 5th Int. Conf. Comput. Sustainable Global Develop.*, 2018, pp. 4037–4040.
[6] Y. Sun *et al.*, "Security and privacy in the internet of vehicles," in *Proc. Int. Conf. Identification, Inf., Knowl. Internet Things*, 2015, pp. 116–121.

[7] B. Bera, S. Saha, A. K. Das, N. Kumar, P. Lorenz, and M. Alazab, "Blockchain-envisioned secure data delivery and collection scheme for 5G-Based IoT-Enabled internet of drones environment," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9097–9111, Aug. 2020.

[8] A. Vangala, B. Bera, S. Saha, A. K. Das, N. Kumar, and Y. Park, "Blockchain-enabled certificate-based authentication for vehicle accident detection and notification in intelligent transportation systems," *IEEE Sensors J.*, vol. 21, no. 14, pp. 15824–15838, Jul. 2021.

[9] B. Bera, S. Saha, A. K. Das, and A. V. Vasilakos, "Designing blockchain-based access control protocol in IoT-Enabled smart-grid system," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5744–5761, Apr. 2021.

[10] A. K. Das, B. Bera, and D. Giri, "AI and blockchain-based cloud-assisted secure vaccine distribution and tracking in IoMT-Enabled COVID-19 environment," *IEEE Internet Things Mag.*, vol. 4, no. 2, pp. 26–32, Jun. 2021.

[11] D. Chattaraj, B. Bera, A. K. Das, S. Saha, P. Lorenz, and Y. Park, "Block-CLAP: Blockchain-assisted certificateless key agreement protocol for internet of vehicles in smart transportation," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 8092–8107, Aug. 2021.

[12] A. K. Das, B. Bera, S. Saha, N. Kumar, I. You, and H.-C. Chao, "AI-envisioned blockchain-enabled signature-based key management scheme for industrial cyber-physical systems," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6374–6388, May 2022.

[13] D. Chattaraj, B. Bera, A. K. Das, J. J. P. C. Rodrigues, and Y. Park, "Designing fine-grained access control for software-defined networks using private blockchain," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1542–1559, Jan. 2022.

[14] R. Kakkar, R. Gupta, S. Tanwar, and J. J. P. C. Rodrigues, "Coalition game and blockchain-based optimal data pricing scheme for ride sharing beyond 5G," *IEEE Syst. J.*, to be published, doi: 10.1109/JSYST.2021.3126620.

[15] S. Wang, S. Sun, X. Wang, Z. Ning, and J. J. P. C. Rodrigues, "Secure crowdsensing in 5G internet of vehicles: When deep reinforcement learning meets blockchain," *IEEE Consum. Electron. Mag.*, vol. 10, no. 5, pp. 72–81, Sep. 2021.

[16] R. Saha *et al.*, "The blockchain solution for the security of internet of energy and electric vehicle interface," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 7495–7508, Aug. 2021.

[17] A. Khanna, P. Rani, T. H. Sheikh, D. Gupta, V. Kansal, and J. J. P. C. Rodrigues, "Blockchain-based security enhancement and spectrum sensing in cognitive radio network," *Wireless Pers. Commun.*, to be published, doi: 10.1007/s11277-021-08729-0.

[18] C. Gentry and Z. Ramzan, "Identity-based aggregate signatures," in *Proc. 9th Int. Conf. Theory Pract. Public-Key Cryptogr.*, 2006, pp. 257–273.

[19] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, 1999.

[20] M. R. Garey and D. S. Johnson, *Computers and Intractability: A. Guide to the Theory of NP-Completeness* (Series of Books in the Mathematical Sciences). San Francisco, CA, USA: Freeman, 1979.

[21] G. Maxwell, A. Poelstra, Y. Seurin, and P. Wuille, "Simple Schnorr multi-signatures with applications to Bitcoin," *Designs, Codes Cryptogr.*, vol. 87, no. 9, pp. 2139–2164, 2019.

[22] M. Kansal and R. Dutta, "Round optimal secure multisignature schemes from lattice with public key aggregation and signature compression," in *Proc. 12th Int. Conf. Prog. Cryptology —AFRICACRYPT*, 2020, pp. 281–300.

[23] M. Fukumitsu and S. Hasegawa, "A tightly-secure lattice-based multisignature," in *Proc. 6th ASIA Public-Key Cryptography Workshop*, 2019, pp. 3–11.

[24] M. Kansal, A. K. Singh, and R. Dutta, "Efficient multi-signature scheme using lattice," *Comput. J.*, to be published, doi: 10.1093/comjnl/bxab077.

[25] S. Tanwar and A. Kumar, "An efficient multi-receiver certificate less digital multisignature scheme with anonymity," *CSI Trans. ICT*, vol. 8, no. 3, pp. 311–318, 2020.

[26] D. Galindo and J. Liu, "Robust subgroup multi-signatures for consensus," *IACR Cryptol. ePrint Arch.*, vol. 2020, 2020, Art. no. 1478.

[27] D. Boneh, M. Drijvers, and G. Neven, "Compact multi-signatures for smaller blockchains," in *Proc. 24th Int. Conf. Theory Appl. Cryptology Inf. Secur.*, 2018, pp. 435–464.

[28] R. E. Bansarkhani and J. Sturm, "An efficient lattice-based multisignature scheme with applications to bitcoins," in *Proc. Int. Conf. Cryptology Netw. Secur.*, 2016, pp. 140–155.

[29] M. Fukumitsu and S. Hasegawa, "A lattice-based provably secure multisignature scheme in quantum random oracle model," in *Proc. Provable Practical Secur.*, 2020, pp. 45–64.

[30] L. Hou, W. Liu, L. Yao, X. Liang, and G.-Q. Zeng, "Practical SM2-based multisignature scheme with applications to vehicular networks," *Secur. Commun. Netw.*, vol. 2021, 2021, Art. no. 7897527.

[31] R. Saha *et al.*, "A blockchain framework in post-quantum decentralization," *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2021.3116896.

[32] Z. Cai, S. Liu, Z. Han, R. Wang, and Y. Huang, "A quantum blind multi-signature method for the industrial blockchain," *Entropy*, vol. 23, no. 11, 2021. [Online]. Available: https://www.mdpi.com/1099-4300/23/11/1520.

[33] C. Jiao and X. Xiang, "Anti-quantum lattice-based ring signature scheme and applications in VANETs," *Entropy*, vol. 23, no. 10, 2021, Art. no. 1364.

[34] F. S. Monteiro, D. H. Goya, and R. Terada, "Improved identification protocol based on the MQ problem," *IEICE, Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. E98.A, no. 6, pp. 1255–1265, 2015.

[35] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proc. 3rd Symp. Operating Syst. Des. Implementation*, 1999, pp. 173–186.

[36] M. S. Kakkasageri and S. S. Manvi, "Multiagent driven dynamic clustering of vehicles in VANETs," *J. Netw. Comput. Appl.*, vol. 35, no. 6, pp. 1771–1780, 2012.

[37] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[38] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, 2002, pp. 337–351.

[39] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[40] K. Sakumoto, T. Shirai, and H. Hiwatari, "Public-key identification schemes based on multivariate quadratic polynomials," in *Proc. 31st Annual Cryptology Conf.*, 2011, pp. 706–723.

[41] A. Bagherzandi and S. Jarecki, "Identity-based aggregate and multi-signature schemes based on RSA," in *Proc. Public Key Cryptography*, 2010, pp. 480–498.

[42] M.-S. Chen, A. Hülsing, J. Rijneveld, S. Samardjiska, and P. Schwabe, "From 5-pass MQ-based identification to MQ-based signatures," 2016. [Online]. Available: https://eprint.iacr.org/2016/708

[43] A. Dua, N. Kumar, A. K. Das, and W. Susilo, "Secure message communication protocol among vehicles in smart city," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4359–4373, May 2018.

[44] H. Zhang, J. Wang, and Y. Ding, "Blockchain-based decentralized and secure keyless signature scheme for smart grid," *Energy*, vol. 180, pp. 955–967, 2019.

[45] W. E. May, "Secure hash standard," 2015, *FIPS PUB 180-1*, National Institute of Standards and Technology (NIST), U. S. Department of Commerce, Apr. 1995. Accessed: Dec. 2021. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

[46] W. Shen, S. Tang, and L. Xu, "A. IBUOVProvably secure identity-based UOV signature scheme," in *Proc. IEEE 16th Int. Conf. Comput. Sci. Eng.*, 2013, pp. 388–395.

[47] L. V. Luyen, "An improved identity-based multivariate signature scheme based on rainbow," *Cryptography*, vol. 3, no. 1, 2019, Art. no. 8.

[48] J. Chen, J. Ling, J. Ning, and J. Ding, "Identity-based signature schemes for multivariate public key cryptosystems," *Comput. J.*, vol. 62, no. 8, pp. 1132–1147, 2019.

[49] R. Dutta, S. K. Debnath, and C. Biswas, "Storage friendly provably secure multivariate identity-based signature from isomorphism of polynomials problem," in *Proc. 18th Int. Conf. Secur. Cryptography*, 2021, pp. 595–602.

[50] A. Petzoldt, "Selecting and reducing key sizes for multivariate cryptography," Ph.D. dissertation, 2013. Accessed: Dec. 4, 2021. [Online]. Available: https://tuprints.ulb.tu-darmstadt.de/3523/1/thesis.pdf

[51] D. Boneh, M. Drijvers, and G. Neven, "Compact multi-signatures for smaller blockchains," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2018, pp. 435–464.

[52] M. Drijvers *et al.*, "On the security of two-round multi-signatures," in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 1084–1101.

[53] M. Drijvers, S. Gorbunov, G. Neven, and H. Wee, "Pixel: Multi-signatures for consensus," in *Proc. USENIX Secur. Symp.*, 2020, pp. 2093–2110.

[54] R. E. Bansarkhani and J. Sturm, "An efficient lattice-based multisignature scheme with applications to bitcoins," in *Proc. Int. Conf. Cryptol. Netw. Secur.*, 2016, pp. 140–155.

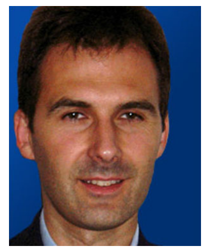
Vikas Srivastava received the B.S.-M.S. dual degree in mathematics from the Indian Institute of Science Education and Research, Mohali, India, in 2017. He is currently a Research Scholar with the Department of Mathematics, National Institute of Technology, Jamshedpur, India. His research interests include cryptography, network security, and blockchain technology.

Sumit Kumar Debnath received the M.Sc. degree in mathematics from the Indian Institutes of Technology Kharagpur (IIT Kharagpur), Kharagpur, India, in 2012, and the Ph.D. degree in cryptology and network security from the Department of Mathematics, IIT Kharagpur, in 2017. He is currently an Assistant Professor with the Department of Mathematics, National Institute of Technology, Jamshedpur, India. He has authored or coauthored more than 27 papers in international journals and conferences in his research areas, which include multivariate cryptography, lattice-based cryptography, network security, and blockchain. He is a Life Member of the Cryptology Research Society of India.

Basudeb Bera received the M.Sc. degree in mathematics and computing from the Indian Institute of Technology (IIT) (ISM) Dhanbad, India, in 2014, and the M.Tech. degree in computer science and data processing from IIT Kharagpur, Kharagpur, India, in 2017. He is currently working toward the Ph.D. degree in computer science and engineering from the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. He has authored or coauthored more than 25 papers in international journals and conferences in his research areas, which include cryptography, network security, and blockchain technology.

Ashok Kumar Das (Senior Member, IEEE) received the M.Sc. degree in mathematics, the M.Tech. degree in computer science and data processing, and the Ph.D. degree in computer science and engineering, from the Indian Institute of Technology Kharagpur, Kharagpur, India. He is currently an Associate Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India, and also a Visiting Faculty with Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Suffolk, VA, USA. He has authored more than 300 papers in international journals and conferences in his research areas, including more than 260 reputed journal papers, which include cryptography, system, and network security, including security in smart grid, Internet of Things, Internet of Drones, Internet of Vehicles, cyber-physical systems, and cloud computing, intrusion detection, blockchain, and AI/ML security. He was the recipient of the Institute Silver Medal from IIT Kharagpur. He is/was on the Editorial Board of IEEE SYSTEMS JOURNAL, *Journal of Network and Computer Applications* (Elsevier), *Computer Communications* (Elsevier), *Journal of Cloud Computing* (Springer), *Cyber Security and Applications*, *IET Communications*, *KSII Transactions on Internet and Information Systems*, and *International Journal of Internet Technology and Secured Transactions* (Inderscience), and was a Program Committee Member in many international conferences. He was also one of the Technical Program Committee Chair of the first International Congress on Blockchain and Applications (BLOCKCHAIN'19), Avila, Spain, June 2019, and second International Congress on Blockchain and Applications (BLOCKCHAIN'20), L'Aquila, Italy, October 2020. His Google scholar citations include more than 12,400 citations with h-index: 66 and i10-index: 192.

Youngho Park (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering, Kyungpook National University, Daegu, South Korea in 1989, 1991, and 1995, respectively. He is currently a Professor with School of Electronics Engineering, Kyungpook National University. During 1996–2008, he was a Professor with the School of Electronics and Electrical Engineering, Sangju National University, Sangju, South Korea. During 2003–2004, he was a Visiting Scholar with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA. His research interests include information security, computer networks, and multimedia.



Pascal Lorenz (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees from the University of Nancy, Nancy, France, in 1990 and 1994, respectively. Between 1990 and 1995, he was a Research Engineer with WorldFIP Europe and with Alcatel-Alsthom. Since 1995, he has been a Professor with the University of Haute-Alsace, Brunstatt-Didenheim, France. He is the author or coauthor of three books, three patents, and 200 international publications in refereed journals and conferences. His research interests include QoS, wireless networks, and high-speed networks. He was the Technical Editor of the *IEEE Communications Magazine* Editorial Board (during 2000–2006), has been the Technical Editor of *IEEE Networks Magazine* since 2015 and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY since 2017, and was the Chair of IEEE ComSoc France (during 2014–2018), Financial Chair of IEEE France (during 2017–2019), Chair of Vertical Issues in Communication Systems Technical Committee Cluster (during 2008–2009), Chair of the Communications Systems Integration and Modeling Technical Committee (during 2003–2009), Chair of the Communications Software Technical Committee (during 2008–2010), and Chair of the Technical Committee on Information Infrastructure and Networking (during 2016–2017). He is an Associate Editor for the *International Journal of Communication Systems* (IJCS-Wiley), *Journal on Security and Communication Networks* (SCN-Wiley), and *International Journal of Business Data Communications and Networking*, *Journal of Network and Computer Applications* (JNCA-Elsevier). He is an IARIA Fellow and member of many international program committees.