Verifiable Aggregation for Heterogeneous Decentralized Identity in Internet of Things

Kai Ding, Tianxiu Xie, Keke Gai, *Senior Member, IEEE,* Jing Yu, Chennan Guo, Zhengkang Fang, Liehuang Zhu, *Senior Member, IEEE,* Weizhi Meng, *Senior Member, IEEE*

***Abstract*—Blockchain-based *Decentralized Identity* (DID) typically employs identity aggregation techniques to support efficient and trustworthy identity authentication in order to meet the requirements of the high volume of service requests in *Internet of Things* (IoT). Due to the lack of effective mechanisms for heterogeneous DID aggregation, a complete aggregated identity authentication often requires multiple rounds of signature verification for different identity attributes. However, this setting brings trust and privacy issues, and one notable threat is the potential disclosure of secret identity information through the linkage of heterogeneous identity attributes when enormous IoT devices/accesses are involved. In this paper, we focus on trustworthy authentication of decentralized identity and propose a novel *Anonymous Verifiable Credential-based Aggregation for heterogeneous Decentralized Identity* (AVCA-hDID). Our AVCA-hDID model supports anonymous ownership verification of DIDs through label randomization, thereby effectively safeguarding identity privacy in IoT. AVCA-hDID involves identifier aggregation and attribute aggregation for heterogeneous DIDs, ensuring both authentication efficiency and balancing trustworthiness and adoptability. We analyze the security and unlinkable of our proposed model and further experiment evaluation demonstrates the efficiency and robustness of AVCA-hDID within a blockchain system.**

***Index Terms*—Heterogeneous decentralized identity, Identity aggregation, Privacy preservation, Trustworthy authentication, Blockchain**

I. Introduction

With the advancement of *Internet of Things* (IoT) technology, the identity verification of users/devices in the cross-heterogeneous system context has become a key factor affecting the overall security of the system [1], [2]. For example, integrating smart grid systems with *Distributed Ledger Technology* (DLT) is considered to be an option for constructing a trustworthy execution environment for power grids; however, a relative open access setting also introduces a broader security issues in identity verification [3], [4]. This type of issue increases significantly in complexity and heterogeneity when devices' identities are involved in the verification. We observe that strengthening the security of user/device identities is urgent, for instance, along with the growing requirements of multi-party collaborative computing and IoT-based value network construction in the future.

The emergence of *Decentralized Identity* (DID) is deemed to be a new paradigm that revolutionizes the way of verifying users' identities by managing identities dispersively across multiple networks [5], [6]. DID harnesses the inherent properties of blockchain [7], such as immutability, transparency, and cryptographic security, to provide a higher-level privacy-preserving user-centric methods for identity verification. In the context of the IoT, we find that identity attributes used for identity verification/authentication of IoT devices have a higher-level heterogeneity, comparing to traditional user-centric identities. This phenomenon mainly derives from the heterogeneity of data sources and implementation scenarios, which causes identity attributes varied as the scenarios switch. Thus, we present a concept of *Heterogeneous Decentralized Identity* (H-DID) in order to address heterogeneity-related issues of identity verification in IoT. To be specific, H-DID emphasizes delivering cross-platform/system mutual authentication and interoperability, by which a higher-level identity portability can be achieved without relying on a specific identity service provider. One of potential merits of implementing H-DID is to reinforce the governance of DID due to the intensified involvement of identity attributes.

Due to large-scale identity-related data within networks, blockchain-based DID systems typically employ identity aggregation techniques to improve the efficiency of DID authentication across various systems in heterogeneous networks [8]. By consolidating multiple DIDs into a unified identifier, identity aggregation streamlines the verification process and reduces the overhead associated with authenticating each DID, separately [9]. Through the application of signature compression, identity aggregation enables the batching of DID verifications, resulting in significant communication and storage cost reductions.

However, H-DIDs presents two key challenge for traditional identity aggregation algorithms. On one hand, since H-DID involves multiple different identity attributes, traditional aggregation algorithms struggle to effectively combine them into a unified identifier representation. To be specific, a comprehensive DID authentication process requires verifiers to individually access and verify the signatures associated

K. Ding, T. Xie, K. Gai, C. Guo, Z. Fang, and L. Zhu are with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China, 100081, {3220185089, 3120215672, gaikeke, 3220231807, 3220245312, liehuangz}@bit.edu.cn.

J. Yu is with the School of Information Engineering, Minzu University of China, China, 100081, jing.yu@muc.edu.cn.

W. Meng is with School of Computing and Communications, Lancaster University, United Kingdom. Email: weizhi.meng@ieee.org.

This work is supported by the National Key Research and Development Program of China (Grant No. 2021YFB2701300), National Natural Science Foundation of China (Grant No. 62372044).

T. Xie is the corresponding author (3120215672@bit.edu.cn).

0000–0000/00$00.00 © 2021 IEEE

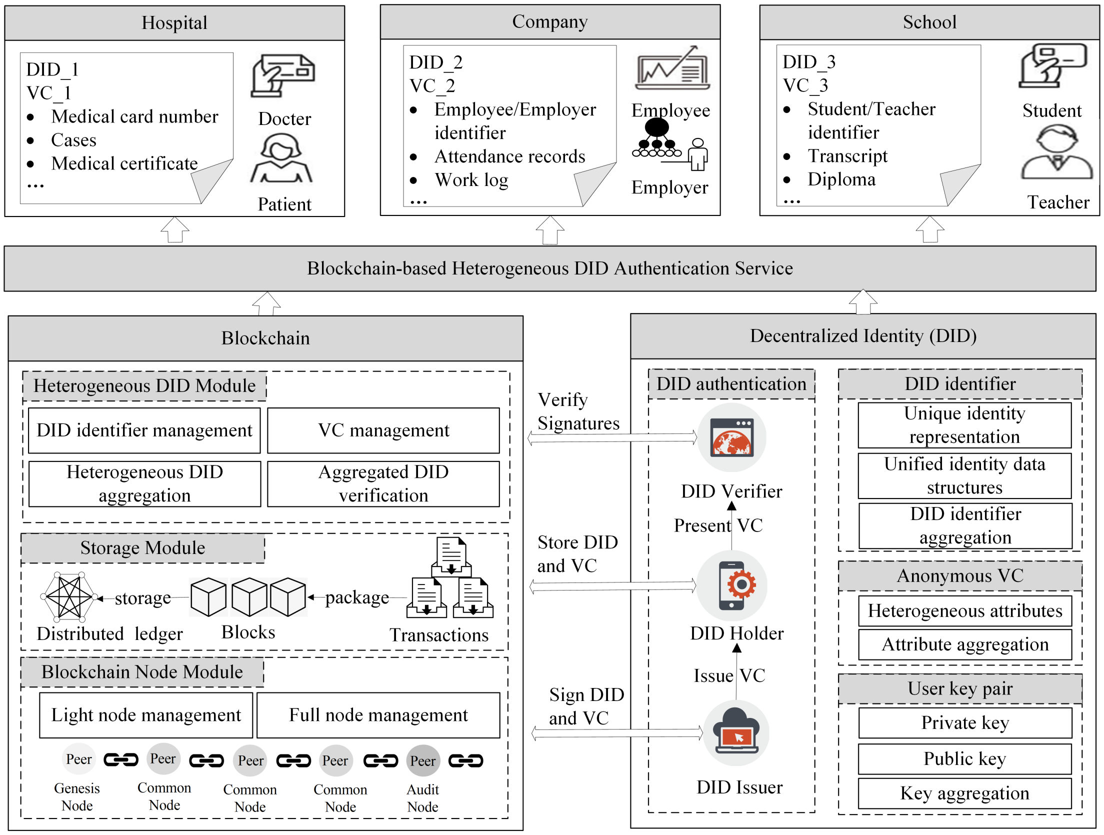


Fig. 1: The architecture of AVCA-hDID model.

with each attribute of the DID. As a result, the linear time complexity associated with verifying each attribute individually persists, limiting the overall efficiency gains that can be achieved through identity aggregation. On the other hand, while the dispersion of identities across multiple networks can provide additional security benefits, the analysis of identity correlations cannot be naturally mitigated in the decentralized environment of blockchain. Adversaries can exploit the inter connection between identities and potentially deduce sensitive user information through multiple rounds of identity authentication, posing a considerable threat to user privacy.

To address the identity verification issue and overcome the aforementioned two key challenges, this paper proposes a *A*nonymous *V*erifiable *C*redential-based *A*ggregation *for* *h*eterogeneous *D*ecentralized *Id*entity (AVCA-hDID) to realize the unlinkability of distributed digital identities. Fig. 1 illustrates an architecture of AVCA-hDID model, which provides a blockchain-based heterogeneous DID authentication service for different institutes. As the core design of AVCA-hDID, DID technology consists of for major components, including DID identifier, anonymous *Verifiable Credential* (VC), user key pair and DID authentication. The DID identifier serves as a unique representation of an entity's identity and is linked to the entity's corresponding identity data. We note that the identity data of different entities adopts a unified data structures. In addition, as a digital credential, the VC is used to prove the authenticity and integrity of the DID. Inspired by [10], we improve the structure of typical VC for H-DIDs and implement a novel anonymous VC based on random tag aggregation. Specifically, the anonymous VC can aggregate heterogeneous identity attributes from different entities while ensuring that the DIDs remain unlinkable. Thus, our proposed AVCA-hDID provides robust privacy preservation while maintaining the trustworthiness and integrity of the DID authentication process. We use the aggregated user public key to verify the validity of the aggregated signatures of DIDs and VCs. AVCA-hDID performs the DID authentication process to verify identities. The DID Verifier validates the VC, the DID Holder requests the VC from the DID Issuer, and the DID Issuer signs the VC. This process utilizes anonymous VCs and heterogeneous attribute aggregation to ensure privacy protection while enhancing verification efficiency. By aggregating users' public keys, AVCA-hDID can verify multiple identity attributes without validating each one separately, improving both security and efficiency.

Furthermore, AVCA-hDID utilizes blockchain technology to facilitate the storage, circulation, and aggregation for DIDs. Through the decentralization and consensus of the blockchain, DID Holder receives DIDs and VCs issued by DID Issuer and stores them in the distributed ledger. Subsequently, DID Verifier invokes smart contracts to verify the labels, aggregated signatures, and verification public keys provided by users, ensuring the effectiveness of DIDs and VCs. Due to the blockchain-based AVCA-hDID, we can support trustworthy authentication of H-DIDs in various identity application scenarios, such as hospitals, companies, and schools. For example, the hospital includes patient-related information such as medical card numbers and certificates, while the company contains employee and employer identifiers, attendance records, and work logs. The school involves student and teacher identifiers, transcripts, diplomas, and so on.

The main contributions of this work are threefold.

1) In this work, we have proposed a new DID mechanism that utilizes heterogeneous identity attributes in the context of IoT for verifying identities of both hardware and users. We consider the heterogeneity of IoT systems to be one of the sufficient conditions for generating heterogeneous identity attributes. Our approach adopts heterogeneous identity attributes to achieve cross-system/platform mutual authentication and interoperability, while guaranteeing the efficiency of the system execution within massive IoT devices.
2) We propose a novel identity aggregation scheme for H-DIDs, which enables multi-attribute authentication of different attributes in a single verification process. Compared to accessing signatures for all heterogeneous attributes associated with a DID, our solution significantly improves the efficiency of H-DID authentication and reduces the storage costs on the blockchain.
3) We propose an anonymous Verifiable Credential structure based on randomized labels. The aggregated signatures in the anonymous VCs possess properties of unforgeability and unlinkability. Thus, AVCA-hDID ensures reliable and secure DID authentication while preserving anonymity and privacy.

The organization of this paper follows the following order. Section II explains preliminaries of our work. Sections III and V present the model design and security analysis of AVCA-hDID, respectively. Section VI provides experimental evaluations. We illustrate the related work in Section VII and our conclusions is given in Section VIII.

II. Preliminaries

Notations. The notations and corresponding descriptions of this paper are shown in Table I as follows.

Cyclic Groups and Generators. A cyclic group is a special type of finite group in which there exists an element (generator) such that repeated group operations with that element can generate all the elements in the group. The order of a generator is defined as the smallest positive integer i such that g^i is equal to the identity element. In a finite group G of order n, if an element g has an order i, then i divides n. This means that the order of any element is a divisor of the group's order. Moreover, if g is an element of order i in G, then $g^x = g^y$ if and only if $x \equiv y \mod i$. This indicates that the sequence of powers of g repeats in cycles of length i. For a group $\mathbb{G}$ of prime order p, it is necessarily cyclic, meaning there exists a generator g such that every element of $\mathbb{G}$ can be represented as some power of g. Moreover, all elements of $\mathbb{G}$ other than the identity can serve as generators, each capable of generating the entire group through their powers.

TABLE I: Descriptions of Notations

Notation	Description
$\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$	Cyclic groups of prime order p in bilinear group
p	Prime order of cyclic groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$
g, $\mathfrak{g}$	Generators of group $\mathbb{G}_1$ and $\mathbb{G}_2$
e	Bilinear pairing map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$
$\mathcal{P}$	Tag domain defined over $\mathbb{G}_1$
h	Generator element selected from $\mathbb{G}_1^*$ during tag generation
$\tilde{\mathrm{p}}$	Secret parameter in $\mathbb{Z}_p$ serving as user's private key
ρ	random tag
t_j, u_j, v_j	Private keys for j-th institution
$r_{j,i}, s_{j,i}$	Private keys for i-th attribute of j-th institution
VP_j	Public key of j-th institution
$VP_{j,i}$	Public key for i-th attribute of j-th institution
σ	Aggregated signature for multiple attributes
sig_j	Signature from j-th institution
$FinalKey$	Aggregated public key
$DID[i]$	i-th decentralized identifier
$FinalDID$	Aggregated DID identifiers
k	Security parameter in Setup algorithm
$\mathcal{H}$	Hash function on $\mathbb{G}_1$ domain
λ	Security parameter in security assumptions
$\mathcal{D}, \mathcal{R}$	Distributions in DDH assumption
$\mathcal{D}_{SDH}, \mathcal{U}$	Distributions in DSqDH assumption
negl(λ)	Negligible function in security parameter λ
$a_{j,i}$	Content of i-th attribute issued by j-th institution
β	Randomization exponent in signature aggregation

Discrete Logarithm. The Discrete Logarithm Assumption asserts that calculating the discrete logarithm in a discrete logarithm group is computationally difficult. In particular, within a group $\mathbb{G}$ defined by a generator g and a large prime p, the challenge is to find x in the equation $g^x \equiv y \pmod{p}$ when g and y are given. Recovering the value of x that satisfies $g^x \equiv y \pmod{p}$ requires substantial computational effort.

The *Square Discrete Logarithm* (SDL) assumption extends the Discrete Logarithm problem. It suggests that computing the square root of an element in a finite field is as hard as solving the discrete logarithm. In a group $\mathbb{G}$ of prime order p with a generator g and an element y, the assumption posits that finding an integer x such that $g^{x^2} \equiv y \pmod{p}$ is as difficult as solving the Discrete Logarithm problem.

Bilinear pairing. A bilinear pairing maps elements from two vector spaces to an element in another vector space. The asymmetric bilinear setup consists of $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$, where $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are cyclic groups of prime order p. Here, g_1 and g_2 are the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. The bilinear pairing e maps elements from $\mathbb{G}_1 \times \mathbb{G}_2$ to $\mathbb{G}_T$ and satisfies the following conditions:

- *Bilinearity.* For any $a, b \in \mathbb{Z}_p$ and $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$, the bilinear pairing satisfies $e(P^a, Q^b) = e(P, Q)^{ab}$. This means that $e(P^a, Q) = e(P, Q)^a$ and $e(P, Q^b) = e(P, Q)^b$.
- *Non-degeneracy.* For any $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, $e(g_1, g_2) \neq 1$.
- *Computability.* For any $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, there exists an efficient polynomial-time algorithm to compute $e(P, Q)$.

Security Assumption. Based on bilinear pairing and discrete logarithm, the following assumptions can be formally defined:

Definition II.1. ***Decisional Diffie-Hellman (DDH) assumption:*** *Let $\mathbb{G}$ be a cyclic group of prime order p with generator g. Given three elements g^a, g^b, and g^c (where a, b, and c*

are random values) . It is difficult to distinguish between the following two distributions through computation.

$$\mathcal{D} = (g, g^a, g^b, g^{ab}) \in \mathbb{G}_4; a, b \in \mathbb{Z}_p$$

$$\mathcal{R} = (g, g^a, g^b, g^c) \in \mathbb{G}_4; a, b, c \in \mathbb{Z}_p$$

Specifically, for any adversary $\mathcal{A}$ with Probabilistic Polynomial Time (PPT) algorithm, the advantage $Adv_{\mathcal{A}}^{DDH}$ is negligible in distinguishing between the random oracle experiment $\mathcal{R}$ and the DDH experiment $\mathcal{D}$:

$$Adv_{DDH}^{\mathcal{A}}(\lambda) = |Pr\left[\mathcal{A}(\mathcal{R}) = 1\right] - Pr\left[\mathcal{A}(\mathcal{D}) = 1\right]| \leq \text{negl}(\lambda)$$

where λ is the security parameter.

Definition II.2. ***Decisional Square Diffie-Hellman(DSqDH) assumption:*** *Given a cyclic group $\mathbb{G}$ of prime order p, a generator g, and two elements g^a, g^b (where a and b are randomnesses), the following two distributions are computationally indistinguishable:*

$$\mathcal{D}_{SDH} = (g, g^a, g^{a^2}) \in \mathbb{G}_3; a, b \in \mathbb{Z}_p$$

$$\mathcal{U} = (g, g^a, g^b) \in \mathbb{G}_3; a, b \in \mathbb{Z}_p$$

Similar to the DDH assumption, DSqDH also assumes that the advantage of $\mathcal{A}$ with PPT algorithms is negligible:

$$Adv_{DSqDH}^{\mathcal{A}}(\lambda) = |Pr\left[\mathcal{A}(\mathcal{U}) = 1\right] - Pr\left[\mathcal{A}(\mathcal{D}_{SDH}) = 1\right]| \\ \leq \text{negl}(\lambda)$$

where λ is the security parameter.

III. Proposed Model

A. Design goals

The design goals of our approach primarily consider the implementation of DID in the context of IoT, which implies that massive DID Holders shall be considered to be a fundamental requirement for this goal. Thus, based on this basic requirement, we aim at achieving following design goals.

Heterogeneous Identity Aggregation. In the context of IoT, a DID Holder can possess multiple DIDs issued by various DID Issuers. Corresponding to VCs, each DID can sign claims regarding multiple attributes of the DID Holder at that DID Issuer. The attributes issued by different DID Issuers may be different. Verifying attributes from different DID Issuers may cause many challenges. Such as low verification efficiency, high system complexity, and significant storage load. Therefore, we should implement the aggregation of heterogeneous identity attributes to reduce storage load and system complexity, while enhancing verification efficiency.

Identity Security and Trustworthiness. Identity is the foundation of access control in an IoT system. To ensure the security of DID Holder identity and maintain the trustworthiness of the system, the system should verify DID Holder identity credentials. This ensures the correctness of DID Holder identity and prevents malicious tampering or forgery of DID Holder identity information.

Privacy Protection. The correlation of heterogeneous identity attributes may reveal a DID Holder's identity information. To protect the privacy of DID Holder data, we need to anonymize DID Holder identity attributes. The system needs to hide the DID Holder's true attribute information and ensure the unlinkability of DID Holders' information.

In a heterogeneous identity aggregation model, the security of identities is paramount. As identity attributes are issued by different DID Issuers, which is complex and variable. Therefore, the security measures must ensure the following aspects.

(i) Correctness: The model should accurately verify the authenticity of identity attributes from various sources. This involves cross-checking the attributes against the original issuers' databases and ensuring that they have not been altered or tampered with. Correctness guarantees that the DID Holder identity being presented is valid and accurate, thus preventing identity fraud. **(ii) Unforgeability:** The model should implement robust cryptographic techniques to ensure that identity attributes cannot be forged. This includes using digital signatures and public key infrastructure to validate.The model need to validate that the attributes were indeed issued by legitimate DID Issuers. Unforgeability is critical in maintaining trust in the model, as it ensures that only genuine attributes are accepted and recognized. **(iii) Unlinkability:** To protect DID Holder privacy, the model must ensure that identity attributes cannot be linked across different DID Issuers. This involves anonymizing DID Holder's data, which allow for the verification of identity attributes without revealing the DID Holder's true identity. Unlinkability ensures that even if identity attributes from multiple sources are aggregated, they cannot be used to track or profile the DID Holder.

By addressing these security aspects, a heterogeneous identity aggregation model can effectively manage and protect DID Holder identities. The model can ensure both the integrity of the system and the privacy of the DID Holders.

B. System Model

To meet the design goals, we aggregate the heterogeneous identities of DID Holders, which reduces the consumption of the time and resources. Fig. 2 shows the structure and workflow of our model. There are three types of participants in AVCA-hDID, DID Issuer, DID Holder, and DID Verifier. The DID issuer refers to the signing entity within the system, which possesses a public-private key pair and one or more signature algorithms. This entity can generate a signature by applying the signature algorithm to the content, thus producing a identifier that attests to the issuer's endorsement of the content. If the signature and public key pass the validation algorithm, the signature can be considered valid. The DID Holder can apply for an identity certificate from the issuer. A DID Verifier can verify whether the holder's identity corresponds correctly to prevent the occurrence of untrustworthy identity data. As shown in Fig. 2, the DID Issuer issues signatures containing the identity attributes for each DID Holder. Then, the DID Holder uploads these VCs with signatures to the blockchain and aggregates the identifiers and attributes. This process uses

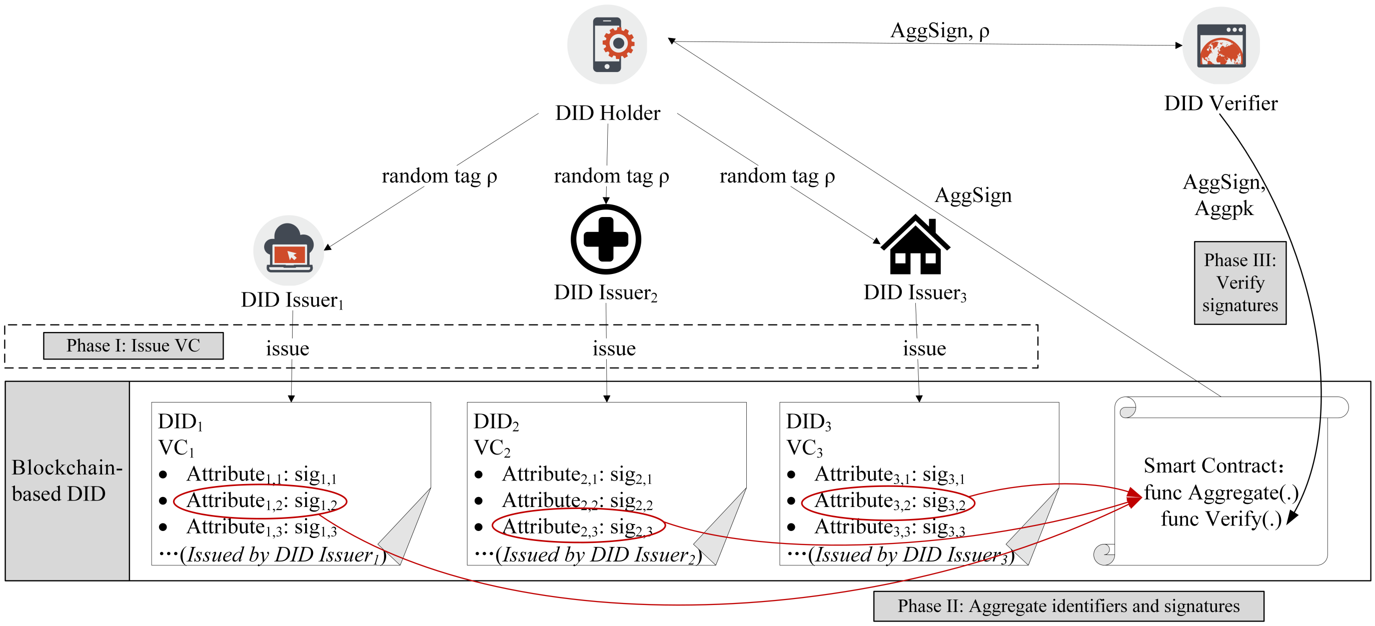

AggSign, ρ
DID Holder
DID Verifier
random tag ρ
random tag ρ
random tag ρ
AggSign
AggSign,
Aggpk
Phase III: Verify signatures
DID Issuer$_1$
DID Issuer$_2$
DID Issuer$_3$
Phase I: Issue VC
issue
issue
issue
Blockchain-based DID
DID$_1$
VC$_1$
Attribute$_{1,1}$: sig$_{1,1}$
Attribute$_{1,2}$: sig$_{1,2}$
Attribute$_{1,3}$: sig$_{1,3}$
···(Issued by DID Issuer$_1$)
DID$_2$
VC$_2$
Attribute$_{2,1}$: sig$_{2,1}$
Attribute$_{2,2}$: sig$_{2,2}$
Attribute$_{2,3}$: sig$_{2,3}$
···(Issued by DID Issuer$_2$)
DID$_3$
VC$_3$
Attribute$_{3,1}$: sig$_{3,1}$
Attribute$_{3,2}$: sig$_{3,2}$
Attribute$_{3,3}$: sig$_{3,3}$
···(Issued by DID Issuer$_3$)
Smart Contract:
func Aggregate(.)
func Verify(.)
Phase II: Aggregate identifiers and signatures

Fig. 2: The workflow of proposed AVCA-hDID model.

random tag methods to ensure the anonymity and unlinkability of identity information, effectively protecting the privacy of the DID Holder. Finally, the DID Verifier uses smart contracts to verify the aggregated signatures and public keys, ensuring the validity of the provided information.

At each signer node, content aggregation of DIDs is performed on identifiers and attribute information. Each signer only needs to complete the signing of multiple distributed digital identities with a single signature. Each signer then signs the previously aggregated DIDs using the signature algorithm originally deployed with their node's chain code. The signer nodes employ a randomization algorithm to abstract the resulting signatures, generating one-time keys with randomness. The signer nodes use the single signature algorithm from the aggregation function to produce single signatures. Public key aggregation is performed by applying linear operations to each signer's public key, combining them into a single aggregated public key that represents all signers. Similarly, signature aggregation is achieved by linearly combining the single signatures of multiple signers into a single aggregated signature. These aggregated signatures and public key parameters are then sent to the credential verifier. Upon receiving the aggregated signature and public key parameters, the credential verifier uses a signature verification function to validate them. The shared parameters are used to compute the verification parameter e_1, which involves operations with the aggregated signature and the elliptic curve base point $\mathbb{G}$ to obtain the verification parameter e_2. The two verification parameters, e_1 and e_2, are then compared. If they are equal, the aggregation is deemed valid; otherwise, it is deemed invalid.

C. Model Design

To protect the privacy of DID Holder, the aggregation of heterogeneous DID Holder identities primarily involves three phases, namely, issuing VCs, aggregating identifiers and attributes, and verifying trusted signatures. As shown in Fig. 2, there are three phases for our AVCA-hDID model:

Phase I: Issuing VC. Each credential issuing authority generates a distributed digital identifier for the DID Holder and signs the attributes the DID Holder holds at that DID Issuer to create legitimate VCs, which are published on the blockchain. The signatures in the VCs generated during this stage can be repeatedly invoked by the DID Holder in subsequent processes for aggregation to meet different verification needs, significantly reducing the system's computational cost.

Phase II: Aggregating DID identifiers and signatures of attributes. DID Holders obtain VCs from the blockchain.And they select signatures for attributes based on specific requirements. They utilize a random function to process corresponding identity and attribute information. And they achievie heterogeneous aggregation of identity information and attributes through invoking aggregate functions in smart contracts deployed on the blockchain.

Phase III: Verifying trusted signatures. The verification DID Issuer validates the identity or attributes that the DID Holder wishes to verify based on the labels, aggregated public key, and aggregated signature provided by the DID Holder. This is achieved by invoking the verification function in the smart contract deployed on the blockchain. If the verification is successful, it indicates that the aggregate signature is legitimate and can be used as a trusted credential for the DID Holder's desired verification attributes. If the verification fails, it indicates that the attributes verified by the DID Holder are not entirely correct, and at least one illegal signature participated in the aggregation operation. In this case, It required a comprehensive inspection of all information aggregated by the DID Holder.

In the process of identity aggregation, based on the different classifications of the aggregated objects, the aggregation can be divided into content aggregation, signature aggregation, and public key aggregation. The process of Content aggregation is using an algorithm to combine several DIDs to be signed into a single entity. All parts can be signed in a single operation. This method of content aggregation improves signature efficiency. Signature aggregation refers to the process of combining multiple signatures into a single signature. Public key aggregation mainly combines multiple public keys into a single public key. In AVCA-hDID, public key aggregation is achieved by applying linear operations to each signer's public key, combining them into a single aggregated public key that represents all signers.

IV. Heterogeneous DID Aggregation

A. Verifiable Aggregation for H-DID

Due to the heterogeneity of identity attributes and decentralized storage, H-DID authentication necessitates cross-issuer collaboration, resulting in inefficiencies, security vulnerabilities, and scalability constraints in DID digital signatures. Furthermore, as a single DID Holder's identity attributes are distributed across multiple DID Issuers, a comprehensive H-DID authentication requires accessing data from various sources, leading to excessive network resource consumption, diminished authentication efficiency, and substantial load pressure on blockchain-based DID systems. To address these challenges, the proposed AVCA-hDID model employs a random tag-based verifiable aggregate signature scheme to achieve efficient H-DID aggregation, thereby streamlining the multi-issuer interactions involved in VC signing and verification while significantly improving signature verification efficiency. Notably, AVCA-hDID incorporates anonymous VC with random tag to ensure unlinkability in H-DID aggregation, not only enhancing the security of blockchain-based digital identity frameworks but also reinforcing privacy preservation mechanisms.

The identity aggregation algorithm for H-DID, presented as Alg. 1, comprises seven distinct phases: parameter initialization, tag generation, public-private key pair generation, DID Issuer signing, public key aggregation, signature aggregation, and aggregated signature verification. The initial phase of parameter initialization establishes the foundational cryptographic parameters for AVCA-hDID by generating an asymmetric bilinear group along with sets of valid and invalid tags. During this phase, AVCA-hDID takes a security parameter as input and constructs an asymmetric bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ where all three are cyclic groups of prime order p, with g and $\mathfrak{g}$ serving as random generators for $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. Additionally, it defines a tag domain over $\mathbb{G}_1$ where each tag consists of three distinct elements from this group. The output of this initialization phase includes both the asymmetric bilinear group configuration and the defined tag domain, which collectively form the common parameters utilized throughout the subsequent phases of the Alg. 1.

The tag generation phase focuses on creating a random secure tag for DIDs. Utilizing the public parameters established in the initialization phase, Alg. 1 randomly selects a generator element h from the multiplicative group $\mathbb{G}_1^*$ and a secret parameter $\tilde{p}$ from the integer field $\mathbb{Z}_p$. These components are combined to produce a structured tag $\rho = \left(h, h^{\tilde{p}}, h^{\tilde{p}^2}\right)$, where the exponentiation operations are performed within the group $\mathbb{G}_1$. The critical parameter $\tilde{p}$ functions as the DID Holder's private key and remains securely stored by the DID Holder. In addition, the random tag ρ is provided to the DID Verifier in the subsequent interactive proof, allowing the DID Verifier to verify the authenticity of the aggregated signature.

Algorithm 1 The Process of DID Aggregation

```
/* Phase 1 : Initialize Parameters*/
$(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g, \mathfrak{g}, e) \leftarrow k$: Input the security parameter $k$, and generate $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g, \mathfrak{g}, e)$, a set of asymmetric bilinear groups, where $g$ and $\mathfrak{g}'$ are random generators of $\mathbb{G}_1$ and $\mathbb{G}_2$.
$\mathcal{P} = \mathbb{G}^3$: Tag Collection
$\mathrm{pp} \leftarrow (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g, \mathfrak{g}, e; \mathcal{P})$: Output initialization parameters.
/* Phase 2 : Generate Tag*/
$h \xleftarrow{\$} \mathbb{G}_1^*, \tilde{p} \xleftarrow{\$} \mathbb{Z}_p$: Randomly select a generator and a parameter.
$\rho = (h, h^{\tilde{p}}, h^{\tilde{p}^2})$: Compute a random tag.
/* Phase 3 : Generate Public and Private Keys*/
$SP_j = [t_j, u_j, v_j], SP'_{j,i} = [r_{j,i}, s_{j,i}] \in \mathbb{Z}_p^5$: Randomly generate a one-time private key of the $j$-th DID Issuer.
$VP_j = [\mathfrak{g}^{t_j}, \mathfrak{g}^{u_j}, \mathfrak{g}^{v_j}], VP'_{j,i} = [\mathfrak{g}^{r_{j,i}}, \mathfrak{g}^{s_{j,i}}]) \in \mathbb{G}_2^5$: Calculate the corresponding public key of the $j$-th DID Issuer.
$sp_{j,i} \leftarrow (SP_j = [t_j, u_j, v_j], SP'_{j,i} = [r_{j,i}, s_{j,i}]), vp_{j,i} \leftarrow (VP_j = [\mathfrak{g}^{t_j}, \mathfrak{g}^{u_j}, \mathfrak{g}^{v_j}], VK'_{j,i} = [\mathfrak{g}^{r_{j,i}}, \mathfrak{g}^{s_{j,i}}])$: Output the public key $(vp_{j,i})$ of the $i$-th attribute of the $j$-th DID Issuer and the pravite key $(sp_{j,i})$ of the $i$-th attribute of the $j$-th DID Issuer.
/* Phase 4 : Sign the Content*/
$sig \leftarrow \rho_1^{t+r+ms} \times \rho_2^u \times \rho_3^v \in \mathbb{G}_1$: To sign the attributes, input the random tag $(\rho)$ and calculate the signature of the $j$-th issuer at index $(j, i)$.
/* Phase 5 : Aggregate Identifiers and Signatures*/
$FinalDID \leftarrow \mathsf{AggID}(DID_1, DID_2, ..., DID_n)$: For identifier aggregation, input the DID identifiers and output the aggregated identifiers.
$\sigma \leftarrow \mathsf{AggSig}(sig_1, sig_2, ..., sig_i)$: For signature aggregation, input the signature of each attribute and output the final aggregated signatures $(\sigma)$.
/* Phase 6 :Aggregate Public Keys*/
$fvp_j \leftarrow VP_j \cup [VP'_{j,i}]_i, FinalKey \leftarrow [fvp_j]_j$: Input the public keys to be aggregated and output the aggregated public key $(FinalKey)$.
/* Phase 7 :Verify the Legality of the Signature*/
$e_1 = e(\sigma, \mathfrak{g})$
$e_2 = e(\rho_1, \prod_j VP_{j,1}^{n_j} \times \prod_i VP'_{j,i,1} \cdot VP_{j,i,2}^{a_{j,i}})$
$e_3 = e(\rho_2, \prod_j VP_{j,2}^{n_j})$
$e_4 = e(\rho_3, \prod_j VP_{j,3}^{n_j})$
**Return** true **if** $e_1 == e_2 \times e_3 \times e_4$ **else return** false
```

A random one-time private key for each attribute and its public key are generated by the public-private key generation algorithm. The input of this phase is the public parameters generated by the above model. Five elements are randomly selected in $\mathbb{Z}_p$. Among them, the first three of which are used as the private key (SP_j) of the j-th DID Issuer, and the latter two are used as the private key ($sp_{j,i}$) of the i-th attribute of the j-th DID Issuer. Therefore, for the same DID Issuer (j), the private key (SP_j) is always consist of three parameters ($[t_j, u_j, v_j]$). In particular, for each i-th attribute of the same DID Issuer (j), another two parameters ($[r_{j,i}, s_{j,i}]$) are independently generated and included in $sp_{j,i}$. Finally, the public key vp is calculated based on the sp: $vp_{j,i} = (VP_j = [\mathfrak{g}^{t_j}, \mathfrak{g}^{u_j}, \mathfrak{g}^{v_j}], VP_{j,i} = [\mathfrak{g}^{r_{j,i}}, \mathfrak{g}^{s_{j,i}}])$, and the public and private keys of the i-th attribute of the j-th DID Issuer are output.

The DID Issuer signing phase combines the tag and the private key corresponding to each attribute to sign the content of the attribute. Input the content of the attribute to be signed and the private key corresponding to the attribute, including the private key of the DID Issuer to which the attribute belongs and the private key of the attribute itself, as well as the public tag ρ. Then, the signature $sig = \rho_1^{t+r+ms} \times \rho_2^{u} \times \rho_3^{v} \in \mathbb{G}_1$ is computed as the output.

To aggregates the signatures of all required attributes, serving as a trusted VC provided by the DID Holder to the verification DID Issuer, Alg. 1 adopts the signature aggregation phase. The input of this phase is several signatures of attributes. Then it outputs signatures which are multiplied sequentially in the $\mathbb{G}_1$ domain to obtain an aggregated signature.

Due to all the public keys are related to the aggregated attributes, we should form an aggregated public key by the public key aggregation phase. Then, verify the authenticity of the aggregated signature by the verification DID Issuer. The related public keys of all attributes involved in the aggregated signature mentioned above is as the input to the algorithm. For different attributes of the same DID Issuer, it is only necessary to record the DID Issuer's own public key once, and then combine it with the public keys corresponding to different attributes sequentially to obtain $fvp_j = VP_j \cup \{VP'_{j,i}\}$. Further, combine the fvp_j of each DID Issuer sequentially to obtain $FinalKey = [fvp_j]$ as the output.

The aggregated signature is verified by the signature verification phase. To prove the legality of the aggregated signature, the algorithm uses the properties of bilinear pairing. Input the public tag ρ, the aggregated public key $FinalKey$, the content vector ($\vec{a} = [a_i]$) of each attribute to be verified, and the aggregated signature σ and then compute the bilinear mappings. If the verification is successful, output true; otherwise, output false.

B. Identifier aggregation for Anonymous VC

Each DID Holder may be issued multiple identities by different DID Issuers, resulting in multiple DID identifiers. To ensure the authenticity of the DID Holder, AVCA-hDID must verify these DID identifiers. However, individually verifying multiple identities introduces significant computational overhead, leading to inefficiency. To address this issue, we propose a DID identifier aggregation method to optimize verification efficiency. The DID Verifier only needs to perform a single verification on the aggregated identity: if successful, it confirms the validity of all the DID Holder's identities across the verified issuers; if failed, it indicates that at least one identity is invalid or potentially tampered with, requiring further investigation. The identifier aggregation method $\mathsf{AggID}(\cdot)$ (line 15 in Alg. 1) is shown in Eq. 1 as follows:

$$FinalDID = \sum_{i=1}^{n} DID[i] \tag{1}$$

where n is the number of DID to be aggregated.

To further enhance authentication efficiency during heterogeneous attribute aggregation, H-DID also aggregates the involved DID identifiers. The preprocessing and verification phases align with Alg. 1, ultimately producing the aggregated identity $FinalDID$.

Complexity. It needs to traverse all n DID identifiers and perform addition operations. Assuming each addition operation is constant time $O(1)$, the time complexity for the $\mathsf{AggID}(\cdot)$ algorithm is $O(n)$.

C. Attribute aggregation for Anonymous VC

In different DID Issuers, the attributes of DID Holders are different. Therefore, to achieve efficient verification of different attributes for multiple DID Holders, AVCA-hDID proposes a novel anonymous VC based on the aggregation of heterogeneous identity attributes, as shown in Alg. 1. The specific computation steps of H-DID attribute aggregation are as follows:

- $\mathtt{pp} \leftarrow \mathsf{Setup}(1^k)$: For parameter initialization (line 1-4 in Alg. 1), given a security parameter k, to generate an asymmetric bilinear group and a tag domain $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g, \mathfrak{g}, e; \mathcal{P})$. At this point, define $\mathtt{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g, \mathfrak{g}, e; \mathcal{P}, \mathcal{H})$, where $\mathcal{H}$ is a hash function on the $\mathbb{G}_1$ domain.
- $\rho \leftarrow \mathsf{UKeyGen}(DID)$: For tag generation (line 6-7 in Alg. 1) of the DID Holder (U), let $h = \mathcal{H}(DID) \in \mathbb{G}_1^*$, select a randomness $\tilde{p} \xleftarrow{\$} \mathbb{Z}_p$ as the private key to participate in subsequent interactive proofs. Then, calculate $\rho \leftarrow (h, h^{\tilde{p}}, h^{\tilde{p}^2}) \in \mathbb{G}_1^3$ as the random tag, which can be presented to the DID Verifier during the verification process.
- $sp_{j,i},\ vp_{j,i} \leftarrow \mathsf{CIKeyGen}(\mathtt{pp})$: For key pair generation (line 8-11 in Alg. 1), the DID Issuer generates private key (sp_j) and public key (vp_j) for the n attributes it possesses. Three elements t_j, u_j, v_j in SP_j are randomly selected from the set $\mathbb{Z}_p$. And for n attributes of a DID Holder, the elements $r_{j,i}, s_{j,i}$ $(i \in n)$ in $SP'_{j,i}$ $(i \in n)$ are randomly selected from $\mathbb{G}_2$. Then, we use these randomnesss to compute vp_j as follows:

$$sp_j^n = (SP_j = [t_j, u_j, v_j], (SP'_{j,i} = [r_{j,i}, s_{j,i}])_{i \in n}) \in \mathbb{Z}_p^{3+2n}$$
$$vp_j^n = \begin{pmatrix} VP_j = [\mathfrak{g}^{t_j}, \mathfrak{g}^{u_j}, \mathfrak{g}^{v_j}], \\ (VP'_{j,i} = [\mathfrak{g}^{r_{j,i}}, \mathfrak{g}^{s_{j,i}}])_{[i \in n]} \end{pmatrix} \in \mathbb{G}_2^{3+2n}$$

- $VC_{sig_j}^{[a_i]} \leftarrow \mathsf{VCIssue}\ (U, \rho, vp, [a_i]_{i \in n}; sp_j^n)$: For attribute signatures (line 13 in Alg. 1), the DID Holder U provides the random tag (ρ) and applies for VCs from a trusted

DID Issuer. The issuer signs the content of each attribute ($[a_i]$) using the private key (sp_j^n) possessed by the DID Holder, which serves as the signature sig_j of the attribute ($[a_i]$) in the trusted VC ($VC_{sig_j}^{[a_i]}$).

- $(FinalKey, (a_{j,i})_{j,i}, \rho', \sigma') \leftarrow \mathsf{VCShow}\ (\rho, VC_{sig_{j,i}}^{[a_j,i]}, (VP_j,\ VP_{j,i}^{'}))$: For key aggregation (line 18 in Alg. 1), the DID Holder (U) aggregates the public keys to obtain $FinalKey \leftarrow [fvp_j]_j = \cup_j (VP_j \cup [VP'_{j,i}]_i)$. For signature aggregation (line 16 in Alg. 1), the DID Holder uses a randomness $\beta \xleftarrow{\$} \mathbb{Z}_p$ to randomize the parameter ρ as $\rho' \leftarrow (\rho_1^\beta, \rho_2^\beta, \rho_3^\beta) \in \mathbb{G}_1$. Then DID Holder aggregates the signatures to obtain $\sigma \leftarrow \prod_{j,i} sig_{j,i} \in \mathbb{G}_1$, and computes $\sigma' \leftarrow \sigma^\beta$.
- $0/1 \leftarrow \mathsf{VCVerify}\ ((FinalKey, (a_{j,i})_{j,i}, \rho', \sigma'), (VP_j, VP_{j,i}^{'}))$: For verification (line 20-24 in Alg. 1), the DID Holder sends $(FinalKey, (a_{j,i})_{j,i}, \rho', \sigma')$ to the DID Verifier. DID Verifier verifies the aggregated signature to prove the legality of the aggregated signature.

Complexity. Within the proposed attribute aggregation scheme for anonymous VC, let N denote the total number of attributes possessed by a DID Holder, n represent the number of attributes requiring aggregated verification, and K signify the number of DID Issuers. The storage and transmission complexity of AVCA-hDID are quantified by the number of elements in the groups $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, and the field $\mathbb{Z}_p$. The total public key ($[vp_j]_{j\in K}$) stored on the blockchain exhibit a size of $(3K + 2N)$ elements in $\mathbb{G}_2$. Regarding verification for n aggregated attributes, the public key ($[vp_j^n]_{j\in K}$) provided during the process require $(3K + 2n)$ elements in $\mathbb{G}_2$. In addition, $\mathsf{VCShow}\ (\cdot)$ involves 4 elements in $\mathbb{G}_1$ (3 elements for tag randomization, and 1 for the aggregated signature), and 1 vector element ($\vec{a}$) in $\mathbb{Z}_p$. For $\mathsf{VCVerify}\ (\cdot)$, the DID Verifier performs $(n + 3K)$ exponentiations in $\mathbb{G}_2$ and 4 bilinear pairing operations. Notably, multiplication operations are omitted from this analysis due to their negligible impact on performance.

D. Security Model

Correctness of AVCA-hDID. *Correctness* means that any valid signature aggregation can pass verification under the corresponding tags and the aggregated public key. We remark that $VC_{sig_j}^{[a_i]}$ generated via $\mathsf{VCIssue}\ (\cdot)$ is respectively successfully verified by $\mathsf{VCVerify}(\cdot)$. *Correctness* formally is formulated by Eq. (3) as follows:

$$Pr\left[\mathsf{VCVerify}\begin{pmatrix} FinalKey, (a_{j,i})_{j,i}, \\ \rho', \sigma', VP_j, VP_{j,i}^{'} \end{pmatrix} = 1\right] = 1. \tag{3}$$

Unforgeability of AVCA-hDID. *Unforgeability* means that even if an adversary can intercept tags and signatures previously provided by users and aggregate them, it remains computationally infeasible to forge a valid tag that passes verification. Formally, it is required that for each adversary $\mathcal{A}$ with PPT algorithm, the chance of winning DID Verifier is negligible:

$$Pr\begin{bmatrix} \mathrm{pp} \leftarrow \mathsf{Setup}(1^k), \\ \forall j \in J, (sp_j, vp_j) \leftarrow \mathcal{A}(\mathrm{pp}) : \\ 1 \leftarrow \begin{pmatrix} \mathcal{A}(vp_j), \\ \mathsf{VCVerify}(vp_j, a_j) \end{pmatrix} \end{bmatrix} \leq \mathrm{negl}(\lambda). \tag{4}$$

Unlinkability of AVCA-hDID. Unlinkability, based on the Decisional Diffie-Hellman (DDH) and Decisional Square Diffie-Hellman (DSqDH) assumptions, ensures that an adversary cannot link multiple identities of a user through their tags and signatures. Refer to [10], the tags are random SqDH triplets. When any tag $(\tilde{\rho_1}, \rho_1) \leftarrow GenTag(1^\kappa)$ is randomized to ρ_1', the distributions $(\mathrm{g}_0, \mathrm{g}_0^x, \mathrm{g}_0^{x^2}, \mathrm{g}_1, \mathrm{g}_1^x, \mathrm{g}_1^{x^2})$ and $(\mathrm{g}_0, \mathrm{g}_0^x, \mathrm{g}_0^{x^2}, \mathrm{g}_1, \mathrm{g}_1^y, \mathrm{g}_1^{y^2})$ are indistinguishable on $\mathbb{G}_\$^6$. Based on the assumptions of DDH and DSqDH, *Unlinkability* of AVCA-hDID can be proven that the following two distributions are indistinguishable:

$$\mathcal{D}_0 = (\mathrm{g}_0, \mathrm{g}_0^x, \mathrm{g}_0^{x^2}, \mathrm{g}_1, \mathrm{g}_1^x, \mathrm{g}_1^{x^2}) \tag{5a}$$

$$\mathcal{D}_1 = (\mathrm{g}_0, \mathrm{g}_0^x, \mathrm{g}_0^{x^2}, \mathrm{g}_1, \mathrm{g}_1^y, \mathrm{g}_1^{y^2}) \tag{5b}$$

for $\mathrm{g}_0, \mathrm{g}_1 \in \mathbb{G}$ and $x, y \in \mathbb{Z}_p$.

V. Security Analysis

Lemma V.1. *(Correctness of H-DID aggregation) If DIDs are heterogeneous and implement aggregation, AVCA-hDID model holds* Security*, which means that any legitimate signature can be verified by the corresponding tag and aggregation public key.*

Proof. Our proposed AVCA-hDID model further introduces random tags to associate different messages under different key signatures with the same tag, facilitating aggregation in subsequent processes. In this scheme, the tag serves as a temporary pseudonym used by the user, and its private key is randomly generated by the user and self-managed for interactive verification. The public key is submitted to the verification authority for signature verification. After randomization, it can still maintain the association with the same user but becomes unlinkable, providing anonymity to the user.

For the deterministic aggregation scheme, where the entire scheme requires only one aggregation signature and users do not need to consider the unlinkability between multiple tags, there is no need to randomize the tags and aggregation signature. The correctness proof of the verification process is as follows:

$$\begin{aligned}
e(\sigma, \mathfrak{g}) &= e\left(\prod_{j,i} \sigma_{j,i}, \mathfrak{g}\right) = e\left(\prod_{j,i}\left(\rho_1^{t_j + r^{j,i} + m_{j,i}s_{j,i}} \times \rho_2^{u_j} \times \rho_3^{v_j}\right), \mathfrak{g}\right) \\
&= e\left(\prod_{j,i} \rho_1^{t_j + r_{j,i} + m_{j,i}s_{j,i}}, \mathfrak{g}\right) \times e\left(\prod_{j,i} \rho_2^{u_j}, \mathfrak{g}\right) \times e\left(\prod_{j,i} \rho_3^{v_j}, \mathfrak{g}\right) \\
&= e\left(\prod_j \rho_1^{t_j n_j} \times \prod_i \rho_1^{r_{j,i}} \cdot \rho_1^{m_{j,i}s_{j,i}}\right) \times e\left(\prod_j \rho_2^{u_j n_j}, \mathfrak{g}\right) \times e\left(\prod_j \rho_3^{v_j n_j}, \mathfrak{g}\right) \\
&= e\left(\rho_1, \mathfrak{g}\right)^{\sum_j \left[t_j n_j + \sum_i (r_{j,i} + m_{j,i} sj,i)\right]} \times e\left(\rho_2, \mathfrak{g}\right)^{\sum_j u_j n_j} \times e\left(\rho_3, \mathfrak{g}\right)^{\sum_j v_j n_j} \\
&= e\left(\rho_1, \prod_j \mathfrak{g}^{t_j n_j} \times \prod_i \mathfrak{g}^{r_{j,i}} \cdot \mathfrak{g}^{m_{j,i}s_{j,i}}\right) \times e\left(\rho_2, \prod_j \mathfrak{g}^{u_j n_j}\right) \times e\left(\rho_3, \prod_j \mathfrak{g}^{v_j n_j}\right) \\
&= e\left(\rho_1, \prod_j VP_{j,1}^{n_j} \times \prod_i VP'_{j,i,1} \cdot VP'^{m_{j,i}}_{j,i,2}\right) \times e\left(\rho_2, \prod_j VP_{j,2}^{n_j}\right) \times e\left(\rho_3, \prod_j VP_{j,3}^{n_j}\right)
\end{aligned} \tag{6}$$

Clearly, the deterministic scheme exhibits lower system efficiency since it only utilizes the attribute signatures in the verifiable credential once. To enhance the overall efficiency of the system and enable the credential issuer to reuse signatures for multiple attributes based on actual requirements, our proposed aggregation scheme for heterogeneous digital identities based on blockchain adopts the aggregation signature algorithm based on SqDH with random tags. During the verification process, users randomize the aggregation signature and tags to ensure the unlinkability of the signatures. The correctness proof of the verification process is similar to the deterministic scheme and is as follows:

$$\begin{aligned} e(\sigma',\mathfrak{g}) &= e(\sigma^{\beta},\mathfrak{g}) = e(\sigma,\mathfrak{g})^{\beta} = e(\prod_{j,i}\sigma_{j,i},\mathfrak{g})^{\beta} = e(\prod_{j,i}(\rho_1^{t_j+r^{j,i}+m_{j,i}s_{j,i}}\times\rho_2^{u_j}\times\rho_3^{v_j}),\mathfrak{g})^{\beta} \\ &= e(\prod_{j,i}\rho_1^{t_j+r_{j,i}+m_{j,i}s_{j,i}},\mathfrak{g})^{\beta}\times e(\prod_{j,i}\rho_2^{u_j},\mathfrak{g})^{\beta}\times e(\prod_{j,i}\rho_3^{v_j},\mathfrak{g})^{\beta} \\ &= e(\prod_{j}\rho_1^{t_jn_j}\times\prod_{i}\rho_1^{r_{j,i}}\cdot\rho_1^{m_{j,i}s_{j,i}})^{\beta}\times e(\prod_{j}\rho_2^{u_jn_j},\mathfrak{g})^{\beta}\times e(\prod_{j}\rho_3^{v_jn_j},\mathfrak{g})^{\beta} \\ &= e(\rho_1,\mathfrak{g})^{\beta\sum_j[t_jn_j+\sum_i(r_{j,i}+m_{j,i}sj,i)]}\times e(\rho_2,\mathfrak{g})^{\beta\sum_j u_jn_j}\times e(\rho_3,\mathfrak{g})^{\beta\sum_j v_jn_j} \\ &= e(\rho_1,\prod_{j}\mathfrak{g}^{t_jn_j}\times\prod_{i}\mathfrak{g}^{r_{j,i}}\cdot\mathfrak{g}^{m_{j,i}s_{j,i}})^{\beta}\times e(\rho_2,\prod_{j}\mathfrak{g}^{u_jn_j})^{\beta}\times e(\rho_3,\prod_{j}\mathfrak{g}^{v_jn_j})^{\beta} \\ &= e(\rho_1,\prod_{j}VK_{j,1}^{n_j}\times\prod_{i}VK'_{j,i,1}\cdot VK'^{m_{j,i}}_{j,i,2})^{\beta}\times e(\rho_2,\prod_{j}VK_{j,2}^{n_j})^{\beta}\times e(\rho_3,\prod_{j}VK_{j,3}^{n_j})^{\beta} \\ &= e(\rho_1^{\beta},\prod_{j}VK_{j,1}^{n_j}\times\prod_{i}VK'_{j,i,1}\cdot VK'^{m_{j,i}}_{j,i,2})\times e(\rho_2^{\beta},\prod_{j}VK_{j,2}^{n_j})\times e(\rho_3^{\beta},\prod_{j}VK_{j,3}^{n_j}) \end{aligned} \tag{7}$$

□

Lemma V.2. *(Unforgeability of H-DID aggregation) If DIDs are heterogeneous and implement aggregation, AVCA-hDID model holds* Unforgeability*, which means that even if the attacker can intercept the tags and signatures previously provided by the user for aggregation, it is difficult to forge a legitimate tag through interactive proof.*

Proof. Given an effective SqDH group $(g_i, a_i = g_i^{w_i}, b_i = a_i^{v_i})$, where $g_i \in G^*$, $w_i, v_i \in \mathbb{Z}_p^*$, we need to output at least two non-zero integers α_i such that the new group $(G = \prod g_i^{\alpha_i}, A = \prod a_i^{\alpha_i}, B = \prod b_i^{\alpha_i})$ is an effective new SqDH group with respect to DL. Assume it is difficult to construct a new valid SqDH group based on a set of indices α_i but knowing the logarithm base and the random values. In simple terms, it is difficult to construct a new valid SqDH group based on the linear combination of indices α_i known but random logarithms and values.

HPP signature is a homomorphic signature over G or its exponentiation [10], [11], assuming $\mathbf{m} = (m_1, \ldots, m_n) \in \mathbb{Z}_p$, $g \in G$, it is infeasible to forge the same signature for $g^{\mathbf{m}}$. This means that only signatures with the same tag can be legally combined linearly. Similar to the HPP signature, in this scheme the central tag $\rho = \left(h, h^{\tilde{p}}, h^{\tilde{p}^2}\right)$ is used. When signing, $\sigma = \rho_1^{t+r+ms} \times \rho_2^u \times \rho_3^v = (h^1)^r \times (h^m)^s \times h^t \times (h^{\tilde{p}})^u \times (h^{\tilde{p}^2})^v$ is used, which is exactly the result of signing (r, s, t, u, v) with (h, h^m). Therefore, based on the infeasibility of the HPP signature, the signature in the composite signature scheme of the heterogeneous digital identity cluster proposed in this paper is also infeasible to forge.

To ensure the infeasibility of signature forgery, the generation of each tag must be random, so in the second step of the scheme, each user's signature tag is generated by $H(id)$, thus ensuring that the recipient is aware of the initial state of the tag. Since all tags are randomly generated in the uncertain scheme, this scheme does not have the issue of private key leakage. □

Lemma V.3. *(Unlinkability of H-DID aggregation) If DIDs are heterogeneous and implement aggregation, AVCA-hDID model holds* Unlinkability*, which means that it is difficult for attackers to link multiple users through tags and signatures.*

Proof. For the Unlinkability, it can be proven that both distributions are indistinguishable from a randomly independent 6-tuple (distribution $\mathbb{G}_6$):

$$\begin{aligned} \mathcal{D}_0 &\approx (\mathrm{g}_0, \mathrm{g}_0^x, \mathrm{g}_0^y, \mathrm{g}_1, \mathrm{g}_1^x, \mathrm{g}_1^y), \mathrm{g}_0, \mathrm{g}_1 \in \mathbb{G}, x, y \in \mathbb{Z}_p(\text{DSqDH}) \\ &\approx (\mathrm{g}_0, \mathrm{g}_0^x, \mathrm{g}_0^y, \mathrm{g}_1, \mathrm{g}_1^u, \mathrm{g}_1^v), \mathrm{g}_0, \mathrm{g}_1 \in \mathbb{G}, x, y, u, v \in \mathbb{Z}_p(\text{DDH}) \\ &\approx (\mathrm{g}_0, \mathrm{g}_0^x, \mathrm{g}_0^{x^2}, \mathrm{g}_1, \mathrm{g}_1^u, \mathrm{g}_1^{u^2}), \mathrm{g}_0, \mathrm{g}_1 \in \mathbb{G}, x, u \in \mathbb{Z}_p(\text{DSqDH}) \\ &= \mathcal{D}_1 \end{aligned}$$

For a user $\mathcal{U}$ who needs to complete multi-round signature aggregation, in order to prevent the verifier or attackers from associating different aggregated attributes across multiple rounds and compromising their privacy, the user has two approaches $\mathcal{H}_0$ and $\mathcal{H}_1$.

$$\begin{aligned} &\mathcal{H}_0 : \mathcal{S}_1 = (\rho_1, \rho_2) = (h_0, h_0^x, h_0^{x^2}, h_1, h_1^y, h_1^{y^2}) \\ &\mathcal{H}_1 : \mathcal{S}_2 = (\rho_1, \rho_1') = (h_0, h_0^x, h_0^{x^2}, h_0^{\beta}, (h_0^{\beta})^x, (h_0^{\beta})^{x^2}) \end{aligned}$$

In $\mathcal{H}_0$, each round, the user generates a random tagρ_1, ρ_2, etc., which is completely unrelated to their identity. In the first round, they provide ρ_1 to the verifiable credential issuer for signing the attributes of that round. Then, they invoke a smart contract to aggregate the signatures, which are subsequently submitted to the verification authority for verification. In the second round, ρ_2 is provided to the verifiable credential issuer for another round of signing, repeating the same operations as in the first round. This process is repeated for each additional round.

In $\mathcal{H}_1$, each round, at the initialization, the user generates an initial tagρ_1 and provides it to the verifiable credential issuer for signing all the possible attributes that may be aggregated in the future. These signed attributes are then published on the blockchain. In each round, the tagis randomized to generate ρ_1'. The required attribute signatures are selected for aggregation, and the signatures are also randomized in the same manner. Finally, the aggregated signature is submitted to the verification authority for verification. The same process is followed for subsequent rounds, using tagrandomization instead of generating new tags as in $\mathcal{H}_0$.

Based on the previous proof, it can be concluded that the distribution $\mathcal{S}_1$ and $\mathcal{S}2$ are indistinguishable on $\mathbb{G}_{\$}^6$. The verifier or attacker would find it difficult to associate user information based on the received tags. The randomization operation effectively achieves the unlinkability between tags, ensuring user anonymity and protecting user privacy. It can be observed that H_1 significantly reduces the system's complexity and storage burden compared to H_0, reducing the workload for users and the verifiable credential issuer. The verifiable credential issuer can issue signatures for multiple attributes in a single step, and users can invoke aggregation based on

TABLE II: Experiment Settings

Settings	Metrics	Number of Signatures
1-1	AST	1,000
1-2	AVT	1,000
2-1	MU	1,000
2-2	CU	1,000
3-1	TST	(10, 100, 200, 300, 400, 500, 1,000)
3-2	TVT	(10, 100, 200, 300, 400, 500, 1,000)
3-3	Sub-alg. Latency	(10, 100, 200, 300, 400, 500, 1,000)

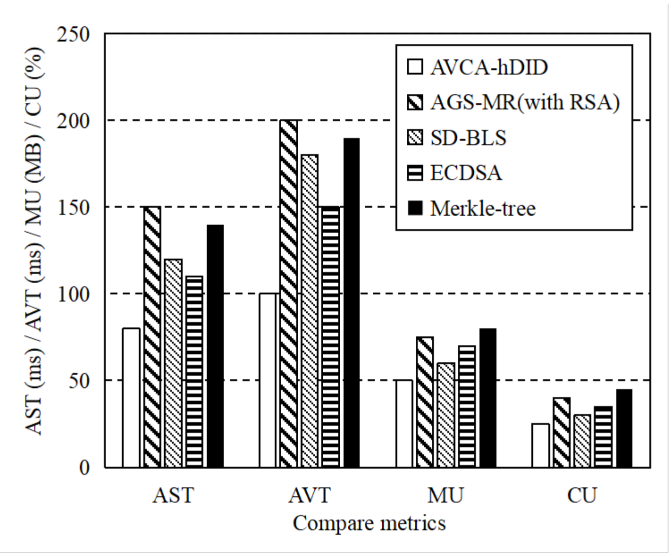


Fig. 3: Comparisons of efficiency between our scheme and other methods.

the same tagfor different attribute signatures according to their actual needs, fulfilling identity authentication requirements in various scenarios. The provided unlinkability greatly protects user privacy and achieves anonymity for user identities. □

VI. Experiment and the Results

A. Experiment Configuration

The principle of our experiment configuration was to simulate the DID Aggregation algorithm in proposed AVCA-hDID mode. The program running environment of our experiment was a host with Intel Core i5-1035G1 CPU, 32.0-GB memory, 1.0-TB hard disk. Windows 11 64-bit operating system has been deployed on it. The programming languages used is Python. We implemented the DID aggregation and verification algorithms in the proposed AVCA-hDID model using the Charm library [1]. Specifically, we utilized the MNT224 bilinear group in Charm and employed the library's group multiplication, exponentiation, and bilinear pairing operations to construct the algorithms. We randomly generated a sample dataset containing 1,000 DIDs, each consisting of a randomly generated public key and a message to be signed. Each algorithm involved in the comparison signed and verified these identities.

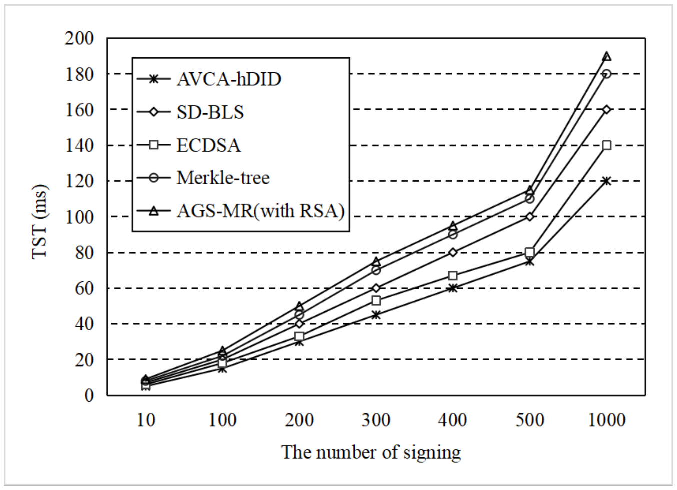


Fig. 4: Comparisons of the total signing time performance.

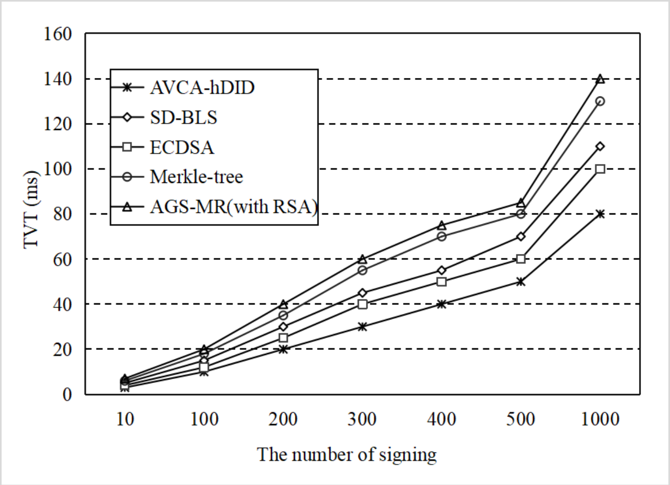
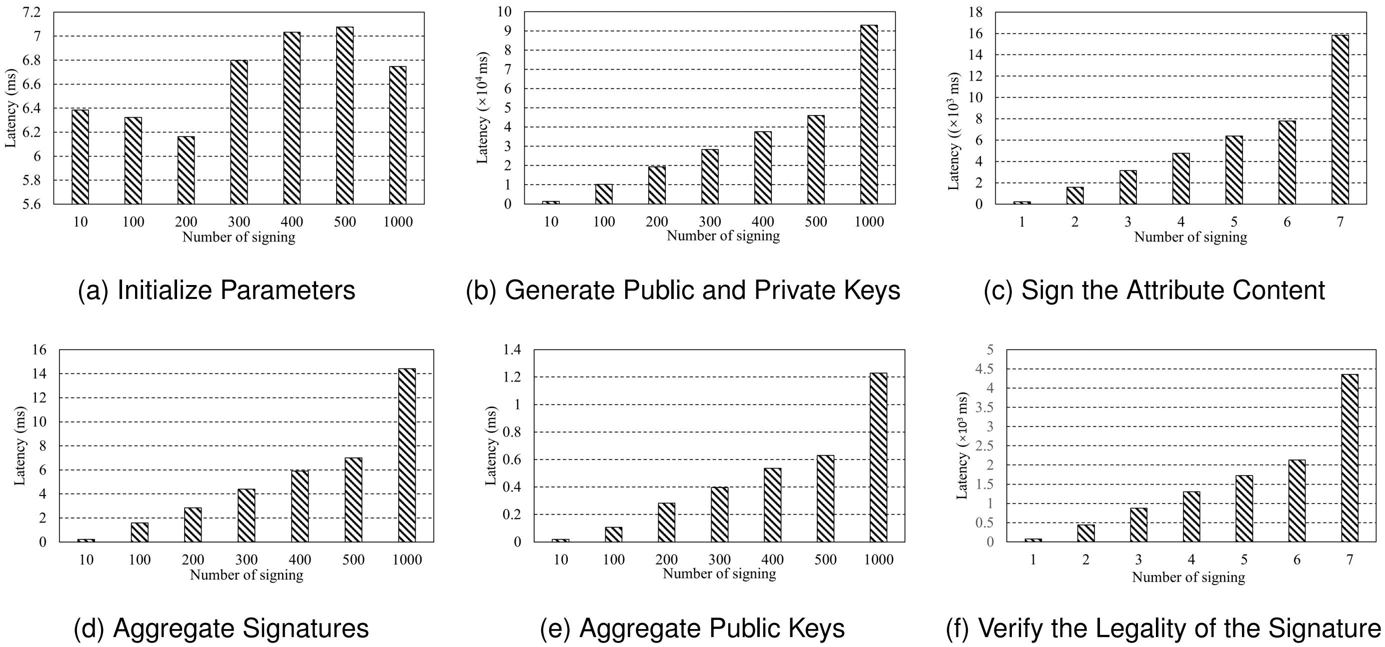


Fig. 5: Comparisons of the total verification time.

The experiment setups focused on evaluating aggregate signature algorithms in efficiency, resource consumption, and scalability in order to examine performance of the proposed scheme in various IoT environments. The efficiency comparison consists of two settings, which include *Average Signature Time* (AST) and *Average Verification Time* (AVT). In AST, algorithms such as AVCA-hDID, RSA, BLS, Merkle-tree, and ECDMA were evaluated, measuring the time taken to generate signatures for the aggregation of 1,000 signatures. Similarly, AVT assesses the verification time for the same algorithms using 1,000 signatures. The resource consumption comparison analyzes *Memory Usage* (MU) and *CPU Utilization* (CU). MU measures the memory consumed by each algorithm during the generation of 1,000 signatures, while CU evaluates CPU usage as a percentage during execution. Finally, the scalability comparison investigates performance metrics across varying input sizes (10, 100, 200, 300, 400, 500, 1000) for *Total Signature Time* (TST), *Total Verification Time* (TVT) and latency of each sub-algorithm in AVCA-hDID, measuring the time taken to generate and verify signatures as well as the time required for each part of the algorithm, respectively. This

[1] https://github.com/JHUISI/charm



(a) Initialize Parameters

(b) Generate Public and Private Keys

(c) Sign the Attribute Content

(d) Aggregate Signatures

(e) Aggregate Public Keys

(f) Verify the Legality of the Signature

Fig. 6: Latency for AVCA-hDID sub-algorithm.

structured approach ensures robust and meaningful results, facilitating a direct comparison of the algorithms' performance characteristics and their suitability for real-world applications.

B. Experiment Results and Main Findings

Under Experiment Setting 1, the AST and AVT evaluation metrics in Fig. 3 showed the time consumption of AVCA-hDID compared to the other four alternative algorithms (SD-BLS [12], AGS-MR(with RSA) [13], ECDSA [14], Merkle tree) during the aggregation and verification of signatures. This assessed the performance of AVCA-hDID. Considering the process of H-DID identity aggregation, as it involves high-frequency identity verification requirements in large-scale IoT, we quantify the computational latency of batch operations for H-DID signature verification through AST and AVT, thereby evaluating the verification efficiency of different aggregate signature schemes in large-scale DID identity authentication. In addition, for the scalability requirements of H-DID verification in IoT, MU and CU jointly evaluate the deployment feasibility of different aggregate signature schemes on resource-constrained DID nodes.

We implemented Merkle-tree aggregation signature using the hash-generic implementation of Merkle-tree from the Bellman-Bignat library[2]. Additionally, we utilized the hash-generic implementation of the RSA accumulator in the same library to achieve AGS-MR(with RSA) [13]. The Merle-tree aggregation signatures were constructed using a hierarchical structure. We employed a bottom-up approach with the Pedersen hash function to build the tree. For AGS-MR(with RSA), we utilized an RSA accumulator and applied modular exponentiation to implement the signing and verification algorithms. Additionally, we replicated SD-BLS [12] using the open-source project BLS Signature Aggregation[3]. This implementation utilized the BLS12-381 elliptic curve based on bilinear mappings, the Pedersen hash function, and point addition operations on the elliptic curve. Furthermore, we replicated the ECDSA aggregation signature scheme [14] by leveraging the secp256k1 curve from the btcec package [4] to construct public keys and one-time keys. Our implementation combined point multiplication and addition operations on this elliptic curve to achieve aggregation.

In Experiment Settings 1-1 and 1-2, the results consistently demonstrated that AVCA-hDID had the least computational time consumption, with a significant lead. The average aggregation signature time was better than AGS-MR by 46.7%, SD-BLS by 33.3%, ECDSA by 27.3%, and Merkle-tree by 42.9%. The average signature verification time showed advantages of 49.7% over AGS-MR, 44.4% over SD-BLS, 32.1% over ECDSA, and 46.4% over Merkle-tree, respectively.

Under Experiment Setting 2, the computational resource consumption values of AVCA-hDID were compared with the other four algorithms. The MU and CU indicators in Fig. 3 provided specific comparative results for Experiment Settings 2-1 and 2-2. Compared to RSA, BLS, ECDSA, and Merkle-tree, the memory consumption of AVCA-hDID was reduced by an average of 33.3%, 11.7%, 28.6%, and 37.5%, respectively. The CPU utilization percentage was reduced by an average of 37.5%, 16.7%, 27.5%, and 44.3%. The experiment results in Fig. 3 indicated that the method proposed in this work had advantages in both execution time performance and computational resource consumption.

[2] https://github.com/alex-ozdemir/bellman-bignat

[3] https://github.com/gazman-sdk/BLS_signature_aggregation

[4] https://github.com/btcsuite/btcd/btcec

Under Experiment Settings 3-1 and 3-2, Fig. 4 and Fig. 5 show the changes in total aggregation signature time and verification signature time as the number of aggregated signatures increased for each comparison algorithm. The results indicated that as the number of aggregated signatures increased, the computational workload grew, leading to a generally linear increase in both aggregation signature time and verification signature time. However, our solution consistently achieved the shortest time across all numbers of aggregated signatures, with the time advantage growing larger as the number of signatures increased. Therefore, AVCA-hDID exhibits better scalability with increasing signature counts. In IoT scenarios, where the number of connected devices is high, better scalability is beneficial for the aggregation of device DIDs. Although our time also increased linearly with the number of signatures, the latencys are acceptable in IoT contexts.

Under Experiment Setting 3-3, Fig. 6 shows the latency of AVCA-hDID sub-algorithms during the DID aggregation and verification processes. The experimental setup simulates different signature aggregation and verification scenarios to assess the performance of AVCA-hDID when handling a large number of DID identities. Fig. 6 shows the latency for each sub-algorithm as it processes aggregated signatures, including the initialization phase, public/private key generation, signing, signature aggregation, public key aggregation, and signature verification. The experiment results in the Fig. 6 shows that as the number of aggregated signatures increases, the latency for each phase grows linearly. This indicates that although the AVCA-hDID system must handle more signatures and identity information during large-scale identity verification, the overall computational latency remains within acceptable limits, especially in environments with many IoT devices. Furthermore, AVCA-hDID also excels in optimizing resource usage, maintaining high efficiency even in resource-constrained environments. By minimizing unnecessary computations and optimizing signature verification, AVCA-hDID offers an efficient and scalable solution for identity verification in IoT applications.

Based on the above analysis, the proposed AVCA-hDID consumed the least computational time and resources compared to the other algorithms. In the scalability evaluation, it also achieved the best scalability relative to the comparison algorithms. Its execution efficiency was acceptable in systems with a large number of IoT devices.

VII. Related Work

A. Decentralized Identity

DID [15]–[17] permits every individual to establish their unique Identifiers within the blockchain ecosystem. Identity identifiers are securely encapsulated within the blockchain infrastructure, subject to verification and maintenance by a multitude of network nodes. The adoption of DID authentication methodologies [18], [19] ushers in a paradigm shift, extricating users from the dependency on conventional, centralized identity governance bodies. Instead, individuals are empowered to authenticate their identities leveraging their self-sovereign identifiers, thereby fostering a heightened degree of decentralization and autonomy in the identity verification process. The most common use case for DIDs involves user authorization to transfer personal credentials from their devices to websites [20]. The DID Foundation [21] and the W3C's [22] DID work group are actively creating standards and use cases to enable these types of transactions. On the platform side of distributed digital identity, Candid [23] proposed a solution aimed at realizing DID in a practical and user-friendly manner, focusing on empowering users to manage their own credentials. For distributed digital identity in complex and heterogeneous networks, DePTVM [24] proposed a decentralized pseudonym and trust value management scheme for integrated heterogeneous networks. Nereus [25] Integrated smart contracts into *Software Guard Extensions* (SGX) enclaves further expands the scope of DID applications.

B. Identity Aggregation

Aggregate Signature which is initially introduced by Boneh *et al.* [26], enables compressing signatures on distinct messages into a short aggregate signature. To alleviate this issue, Goyal *et al.* [27] pioneered the concept of locally verifiable aggregate signatures. This advanced scheme enables verifiers to validate the authenticity of individual messages within the aggregate without necessitating access to the entire message collection. Signature aggregation can be achieved through two primary methods: general aggregation [28] and sequential aggregation [29]. Many signature schemes are well-known for their signature aggregation techniques, such as Schnorr signatures [30], lattice-based signatures [31], and pairing-based signatures [32]. When it comes to pairing-based signatures, we must highlight the BLS scheme [33], which has been practically used in blockchain projects, such as Dfinity and Algorand.

TFS-ABS [34] combines traceability and anonymity with constant-size signatures, addressing key revocation issues in dynamic IoT networks. Unlike our blockchain-based anonymous VC, TFS-ABS relies on a trusted authority for traceability, which may conflict with decentralization. OABS [35] achieves server-aided verification with constant-size signatures, optimizing for resource-constrained devices. However, such schemes lacks considerations for heterogeneous identity aggregation.

C. Attribute-based Anonymous Certificates

Anonymous credentials were initially introduced by Chaum [36] with the primary aim of enabling users to interact anonymously with organizations, allowing them to subsequently present their credentials to various service providers (verifiers) without revealing their identity. This concept evolved into attribute-based anonymous credential schemes, which facilitated users declaring their attributes to verifiers while preserving anonymity. Such schemes found applications in privacy-centric areas like direct anonymous authentication [37] and anonymous electronic identifiers [38]. It wasn't until Brands [39] introduced a model for single-attribute anonymous credentials, followed by Kampanakis and Lysyanskaya's [40] proposal for multi-attribute versions, that anonymous credentials garnered significant attention. Following these developments, numerous schemes for anonymous credentials have been proposed to enhance their performance. Diverse signature schemes, including CL signatures [40]–[42], malleable signatures [43], structure-preserving signatures [44], and editable signatures [45], have been employed to construct various anonymous credential schemes tailored to different attribute requirements. A common feature among most of these schemes is their design as unlinkable and non-transferable, ensuring that verifiers cannot link interactions involving the same credential to a single user, and simultaneously preventing users from fabricating false credentials or misappropriating others' credentials for unauthorized use.

Existing identity authentication schemes primarily rely on cryptographic techniques, such as anonymous credentials [10], [28], [42], [46], to ensure secure identity verification. Hesse *et al.* [42] employ anonymous credentials to achieve privacy-preserving digital identity authentication, extending the aggregate signature mechanism within Self-sovereign Identity (SSI) frameworks. Doerner *et al.* [46] leverage the BBS+ signature scheme for distributed digital certificate issuance, while Hebant *et al.* [10] apply attribute-based encryption for traceable certificates, supporting multi-party authentication. In addition to cryptographic signatures, Secure Multiparty Computation (SMPC) is frequently used to establish distributed trust across multiple parties in DID authentication. For instance, Tan *et al.* [47] introduce MPCAuth, an efficient protocol proposed for establishing TLS connections within SMPC, enabling multi-factor authentication. However, such identity authentication schemes do not consider the heterogeneous identity aggregation.

VIII. Conclusion

In this work we propose an anonymous VC-based aggregation scheme for heterogeneous DID, which aimed at solving trust and privacy issues of identity authentication in IoT. The proposed scheme can successfully support anonymous ownership verification of DIDs through label randomization, which is theoretically demonstrated in this work. Our evaluations have also evidenced the efficiency and robustness of the proposed scheme.

Practical deployment for AVCA-hDID in large-scale IoT environments faces significant challenges, including blockchain scalability under high-frequency device onboarding, computational constraints of resource-limited IoT devices and interoperability gaps across heterogeneous DID issuers. Addressing these through layer-2 solutions, hardware acceleration and W3C DID compliance constitutes critical future work.

References

[1] Bei Gong, Guiping Zheng, Muhammad Waqas, Shanshan Tu, and Sheng Chen. LCDMA: Lightweight cross-domain mutual identity authentication scheme for internet of things. *IEEE Internet of Things Journal*, 10(14):12590–12602, 2023.

[2] Xuefei Yin, Song Wang, Muhammad Shahzad, and Jiankun Hu. An iot-oriented privacy-preserving fingerprint authentication system. *IEEE Internet of Things Journal*, 9(14):11760–11771, 2021.

[3] Keke Gai, Yulu Wu, Liehuang Zhu, Meikang Qiu, and Meng Shen. Privacy-preserving energy trading using consortium blockchain in smart grid. *IEEE Transactions on Industrial Informatics*, 15(6):3548–3558, 2019.

[4] Keke Gai, Yulu Wu, Liehuang Zhu, Lei Xu, and Yan Zhang. Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks. *IEEE Internet of Things Journal*, 6(5):7992–8004, 2019.

[5] Pawel Szalachowski. Password-authenticated decentralized identities. *IEEE Transactions on Information Forensics and Security*, 16:4801–4810, 2021.

[6] Sandro Rodriguez Garzon, Hakan Yildiz, and Axel Küpper. Decentralized identifiers and self-sovereign identity in 6g. *IEEE Network*, 36(4):142–148, 2022.

[7] Zijian Bao, Debiao He, Muhammad Khurram Khan, Min Luo, and Qi Xie. Pbidm: Privacy-preserving blockchain-based identity management system for industrial internet of things. *IEEE Transactions on Industrial Informatics*, 19(2):1524–1534, 2022.

[8] Chaosheng Feng, Bin Liu, Keping Yu, Sotirios K Goudos, and Shaohua Wan. Blockchain-empowered decentralized horizontal federated learning for 5g-enabled uavs. *IEEE Transactions on Industrial Informatics*, 18(5):3582–3592, 2021.

[9] Siwon Huh, Myungkyu Shim, Jihwan Lee, Simon S Woo, Hyoungshick Kim, and Hojoon Lee. Did we miss anything?: Towards privacy-preserving decentralized id architecture. *IEEE Transactions on Dependable and Secure Computing*, 20(6):4881–4898, 2023.

[10] Chloé Hébant and David Pointcheval. Traceable constant-size multi-authority credentials. *Information and Computation*, 293:105060, 2023.

[11] Chloé Hébant, Duong Hieu Phan, and David Pointcheval. Linearly-homomorphic signatures and scalable mix-nets. In *Public-Key Cryptography–PKC 2020: 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4–7, 2020, Proceedings, Part II 23*, pages 597–627. Springer, 2020.

[12] Denis Roio, Rebecca Selvaggini, Gabriele Bellini, and Andrea Dintino. Sd-bls: Privacy preserving selective disclosure of verifiable credentials with unlinkable threshold revocation. In *2024 IEEE International Conference on Blockchain (Blockchain)*, pages 505–511, 2024.

[13] Khaled Chait, Abdelkader Laouid, Mostefa Kara, Mohammad Hammoudeh, Omar Aldabbas, and Abdullah T Al-Essa. An enhanced threshold rsa-based aggregate signature scheme to reduce blockchain size. *IEEE Access*, 11:110490–110501, 2023.

[14] Jay Bojič Burgos and Matevž Pustišek. Decentralized iot data authentication with signature aggregation. *Sensors*, 24(3):1037, 2024.

[15] K. Wang, J. Gao, Q. Wang, J. Zhang, Y. Li, Z. Guan, and Z. Chen. Hades: Practical decentralized identity with full accountability and fine-grained sybil-resistance. In *Proceedings of the 39th Annual Computer Security Applications Conference*, pages 216–228, 2023.

[16] S. Zheng, Z. Li, J. Luo, Z. Xin, and X. Liu. Idea-dac: Integrity-driven editing for accountable decentralized anonymous credentials via zk-json. *Cryptology ePrint Archive*, 2024.

[17] Jiguo Li, Yu Chen, Jinguang Han, Chengdong Liu, Yichen Zhang, and Huaqun Wang. Decentralized attribute-based server-aid signature in the internet of things. *IEEE Internet of Things Journal*, 9(6):4573–4583, 2021.

[18] J. Chen, F. Lu, Y. Liu, S. Peng, Z. Cai, and F. Mo. Cross trust: A decentralized ma-abe mechanism for cross-border identity authentication. *International Journal of Critical Infrastructure Protection*, 44:100661, 2024.

[19] H.-H. Ou, C.-H. Pan, Y.-M. Tseng, and I.-C. Lin. Decentralized identity authentication mechanism: Integrating fido and blockchain for enhanced security. *Applied Sciences*, 14:3551, 2024.

[20] I.T. Javed, F. Alharbi, B. Bellaj, T. Margaria, N. Crespi, and K.N. Qureshi. Health-id: A blockchain-based decentralized identity management for remote healthcare. In *Healthcare*, volume 9, page 712. MDPI, 2021.

[21] Decentralized Identity Foundation. https://identity.foundation/, 2020.

[22] W3C. Decentralized identifiers (dids) v0.11: Data model and syntaxes for decentralized identifiers. Technical report, 2018.

[23] M. Deepak, M. Harjasleen, F. Zhang, N. Jean-Louis, F. Alexander, et al. Candid: Can-do decentralized identity with legacy compatibility, sybil-resistance, and accountability. In *2021 IEEE SP*, pages 1348–1366. IEEE, 2021.

[24] G. Liu, Z. Yan, D. Wang, H. Wang, and T. Li. Deptvm: Decentralized pseudonym and trust value management for integrated networks. *IEEE Transactions on Dependable and Secure Computing*, 2023.

[25] M. Li, Y. Chen, C. Lal, M. Conti, F. Martinelli, and M. Alazab. Nereus: Anonymous and secure ride-hailing service based on private smart

contracts. *IEEE Transactions on Dependable and Secure Computing*, 2022.

[26] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In Eli Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, pages 416–432, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[27] R. Goyal and V. Vaikuntanathan. Locally verifiable signature and key aggregation. In *Annual International Cryptology Conference*, pages 761–791. Springer, 2022.

[28] Omid Mir, Balthazar Bauer, Scott Griffy, Anna Lysyanskaya, and Daniel Slamanig. Aggregate signatures with versatile randomization and issuer-hiding multi-authority anonymous credentials. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 30–44, New York, NY, USA, 2023.

[29] T. Li, H. Wang, D. He, and J. Yu. Designated-verifier aggregate signature scheme with sensitive data privacy protection for permissioned blockchain-assisted iiot. *IEEE Transactions on Information Forensics and Security*, 2023.

[30] G. Fuchsbauer, A. Plouviez, and Y. Seurin. Blind schnorr signatures and signed elgamal encryption in the algebraic group model. In *Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part II 30*, pages 63–95. Springer, 2020.

[31] W.-K. Lee, R. K. Zhao, R. Steinfeld, A. Sakzad, and S. O. Hwang. High throughput lattice-based signatures on gpus: Comparing falcon and mitaka. *IEEE Transactions on Parallel and Distributed Systems*, 35:675–692, 2024.

[32] M. Brodsky, A. Choudhuri, A. Jain, and O. Paneth. Monotone-policy aggregate signatures. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 168–195. Springer, 2024.

[33] D. Boneh, M. Drijvers, and G. Neven. Bls multi-signatures with public-key aggregation. *URL: https://crypto. stanford. edu/˜ dabo/pubs/papers/BLSmultisig. html*, 2018.

[34] Zhaozhe Kang, Jiguo Li, Jian Shen, Jinguang Han, Yuting Zuo, and Yichen Zhang. Tfs-abs: Traceable and forward-secure attribute-based signature scheme with constant-size. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):9514–9530, 2023.

[35] Zhaozhe Kang, Jiguo Li, Yuting Zuo, Yichen Zhang, and Jinguang Han. Oabs: Efficient outsourced attribute-based signature scheme with constant-size. *IEEE Internet of Things Journal*, 2024.

[36] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[37] Ernie Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 132–145, 2004.

[38] Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 21–30, 2002.

[39] Stefan Brands. *Rethinking public key infrastructures and digital certificates: building in privacy*. Mit Press, 2000.

[40] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Advances in Cryptology—EUROCRYPT 2001: International Conference on the Theory and Application of Cryptographic Techniques Innsbruck, Austria, May 6–10, 2001 Proceedings 20*, pages 93–118. Springer, 2001.

[41] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Advances in Cryptology—CRYPTO 2002: 22nd Annual International Cryptology Conference Santa Barbara, California, USA, August 18–22, 2002 Proceedings 22*, pages 61–76. Springer, 2002.

[42] Julia Hesse, Nitin Singh, and Alessandro Sorniotti. How to bind anonymous credentials to humans. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 3047–3064, 2023.

[43] Sven Schäge. New limits of provable security and applications to elgamal encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 255–285. Springer, 2024.

[44] Renas Bacho, Julian Loss, Stefano Tessaro, Benedikt Wagner, and Chenzhi Zhu. Twinkle: Threshold signatures from ddh with full adaptive security. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 429–459. Springer, 2024.

[45] Zonglun Li, Shuhao Zheng, Junliang Luo, Ziyue Xin, and Xue Liu. Ideadac: Integrity-driven editing for accountable decentralized anonymous credentials. In *The Web Conference 2024*.

[46] Jack Doerner, Yashvanth Kondi, Eysa Lee, Abhi Shelat, and LaKyah Tyner. Threshold bbs+ signatures for distributed anonymous credential issuance. In *IEEE SP '23*, pages 773–789. IEEE, 2023.

[47] Sijun Tan, Weikeng Chen, Ryan Deng, and Raluca Ada Popa. Mpcauth: multi-factor authentication for distributed-trust systems. In *IEEE SP '23*, pages 829–847. IEEE, 2023.

Verifiable Aggregation for Heterogeneous Decentralized Identity in Internet of Things

Kai Ding, Tianxiu Xie, Keke Gai, *Senior Member, IEEE,* Jing Yu, Chennan Guo, Zhengkang Fang, Liehuang Zhu, *Senior Member, IEEE,* Weizhi Meng, *Senior Member, IEEE*

Abstract—**Blockchain-based *Decentralized Identity* (DID) typically employs identity aggregation techniques to support efficient and trustworthy identity authentication in order to meet the requirements of the high volume of service requests in *Internet of Things* (IoT). Due to the lack of effective mechanisms for heterogeneous DID aggregation, a complete aggregated identity authentication often requires multiple rounds of signature verification for different identity attributes. However, this setting brings trust and privacy issues, and one notable threat is the potential disclosure of secret identity information through the linkage of heterogeneous identity attributes when enormous IoT devices/accesses are involved. In this paper, we focus on trustworthy authentication of decentralized identity and propose a novel *Anonymous Verifiable Credential-based Aggregation for heterogeneous Decentralized Identity* (AVCA-hDID). Our AVCA-hDID model supports anonymous ownership verification of DIDs through label randomization, thereby effectively safeguarding identity privacy in IoT. AVCA-hDID involves identifier aggregation and attribute aggregation for heterogeneous DIDs, ensuring both authentication efficiency and balancing trustworthiness and adoptability. We analyze the security and unlinkable of our proposed model and further experiment evaluation demonstrates the efficiency and robustness of AVCA-hDID within a blockchain system.**

Index Terms—**Heterogeneous decentralized identity, Identity aggregation, Privacy preservation, Trustworthy authentication, Blockchain**

I. Introduction

With the advancement of *Internet of Things* (IoT) technology, the identity verification of users/devices in the cross-heterogeneous system context has become a key factor affecting the overall security of the system [1], [2]. For example, integrating smart grid systems with *Distributed Ledger Technology* (DLT) is considered to be an option for constructing a trustworthy execution environment for power grids; however, a relative open access setting also introduces a broader security issues in identity verification [3], [4]. This type of issue increases significantly in complexity and heterogeneity when devices' identities are involved in the verification. We observe that strengthening the security of user/device identities is urgent, for instance, along with the growing requirements of multi-party collaborative computing and IoT-based value network construction in the future.

The emergence of *Decentralized Identity* (DID) is deemed to be a new paradigm that revolutionizes the way of verifying users' identities by managing identities dispersively across multiple networks [5], [6]. DID harnesses the inherent properties of blockchain [7], such as immutability, transparency, and cryptographic security, to provide a higher-level privacy-preserving user-centric methods for identity verification. In the context of the IoT, we find that identity attributes used for identity verification/authentication of IoT devices have a higher-level heterogeneity, comparing to traditional user-centric identities. This phenomenon mainly derives from the heterogeneity of data sources and implementation scenarios, which causes identity attributes varied as the scenarios switch. Thus, we present a concept of *Heterogeneous Decentralized Identity* (H-DID) in order to address heterogeneity-related issues of identity verification in IoT. To be specific, H-DID emphasizes delivering cross-platform/system mutual authentication and interoperability, by which a higher-level identity portability can be achieved without relying on a specific identity service provider. One of potential merits of implementing H-DID is to reinforce the governance of DID due to the intensified involvement of identity attributes.

Due to large-scale identity-related data within networks, blockchain-based DID systems typically employ identity aggregation techniques to improve the efficiency of DID authentication across various systems in heterogeneous networks [8]. By consolidating multiple DIDs into a unified identifier, identity aggregation streamlines the verification process and reduces the overhead associated with authenticating each DID, separately [9]. Through the application of signature compression, identity aggregation enables the batching of DID verifications, resulting in significant communication and storage cost reductions.

However, H-DIDs presents two key challenge for traditional identity aggregation algorithms. On one hand, since H-DID involves multiple different identity attributes, traditional aggregation algorithms struggle to effectively combine them into a unified identifier representation. To be specific, a comprehensive DID authentication process requires verifiers to individually access and verify the signatures associated

K. Ding, T. Xie, K. Gai, C. Guo, Z. Fang, and L. Zhu are with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China, 100081, {3220185089, 3120215672, gaikeke, 3220231807, 3220245312, liehuangz}@bit.edu.cn.

J. Yu is with the School of Information Engineering, Minzu University of China, China, 100081, jing.emy.yu01@gmail.com.

W. Meng is with School of Computing and Communications, Lancaster University, United Kingdom. Email: weizhi.meng@ieee.org.

This work is supported by the National Key Research and Development Program of China (Grant No. 2021YFB2701300), National Natural Science Foundation of China (Grant No. 62372044).

T. Xie is the corresponding author (3120215672@bit.edu.cn).

0000–0000/00$00.00 © 2021 IEEE

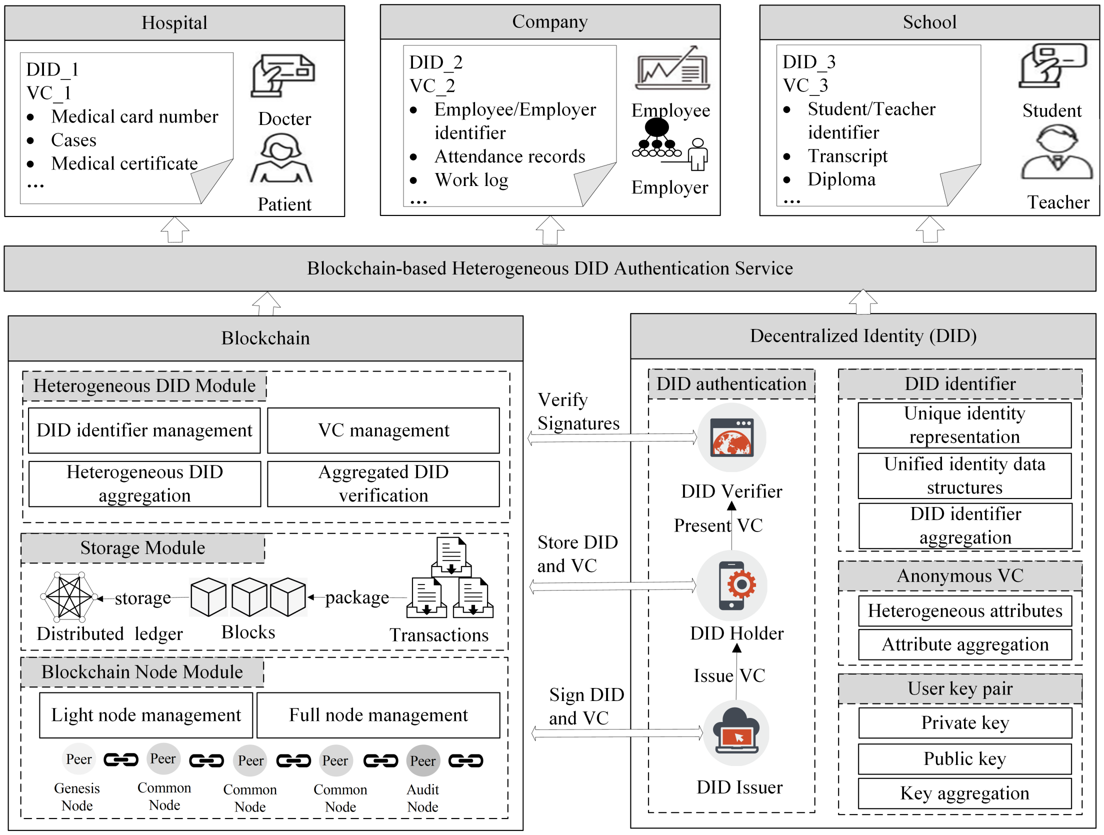


Fig. 1: The architecture of AVCA-hDID model.

with each attribute of the DID. As a result, the linear time complexity associated with verifying each attribute individually persists, limiting the overall efficiency gains that can be achieved through identity aggregation. On the other hand, while the dispersion of identities across multiple networks can provide additional security benefits, the analysis of identity correlations cannot be naturally mitigated in the decentralized environment of blockchain. Adversaries can exploit the inter connection between identities and potentially deduce sensitive user information through multiple rounds of identity authentication, posing a considerable threat to user privacy.

To address the identity verification issue and overcome the aforementioned two key challenges, this paper proposes a *Anonymous Verifiable Credential-based Aggregation for heterogeneous Decentralized Identity* (AVCA-hDID) to realize the unlinkability of distributed digital identities. Fig. 1 illustrates an architecture of AVCA-hDID model, which provides a blockchain-based heterogeneous DID authentication service for different institutes. As the core design of AVCA-hDID, DID technology consists of for major components, including DID identifier, anonymous *Verifiable Credential* (VC), user key pair and DID authentication. The DID identifier serves as a unique representation of an entity's identity and is linked to the entity's corresponding identity data. We note that the identity data of different entities adopts a unified data structures. In addition, as a digital credential, the VC is used to prove the authenticity and integrity of the DID. Inspired by [10], we improve the structure of typical VC for H-DIDs and implement a novel anonymous VC based on random tag aggregation. Specifically, the anonymous VC can aggregate heterogeneous identity attributes from different entities while ensuring that the DIDs remain unlinkable. Thus, our proposed AVCA-hDID provides robust privacy preservation while maintaining the trustworthiness and integrity of the DID authentication process. We use the aggregated user public key to verify the validity of the aggregated signatures of DIDs and VCs. AVCA-hDID performs the DID authentication process to verify identities. The DID Verifier validates the VC, the DID Holder requests the VC from the DID Issuer, and the DID Issuer signs the VC. This process utilizes anonymous VCs and heterogeneous attribute aggregation to ensure privacy protection while enhancing verification efficiency. By aggregating users' public keys, AVCA-hDID can verify multiple identity attributes without validating each one separately, improving both security and efficiency.

Furthermore, AVCA-hDID utilizes blockchain technology to facilitate the storage, circulation, and aggregation for DIDs. Through the decentralization and consensus of the blockchain, DID Holder receives DIDs and VCs issued by DID Issuer and stores them in the distributed ledger. Subsequently, DID Verifier invokes smart contracts to verify the labels, aggregated signatures, and verification public keys provided by users, ensuring the effectiveness of DIDs and VCs. Due to the blockchain-based AVCA-hDID, we can support trustworthy authentication of H-DIDs in various identity application scenarios, such as hospitals, companies, and schools. For example, the hospital includes patient-related information such as medical card numbers and certificates, while the company contains employee and employer identifiers, attendance records, and work logs. The school involves student and teacher identifiers, transcripts, diplomas, and so on.

The main contributions of this work are threefold.

1) In this work, we have proposed a new DID mechanism that utilizes heterogeneous identity attributes in the context of IoT for verifying identities of both hardware and users. We consider the heterogeneity of IoT systems to be one of the sufficient conditions for generating heterogeneous identity attributes. Our approach adopts heterogeneous identity attributes to achieve cross-system/platform mutual authentication and interoperability, while guaranteeing the efficiency of the system execution within massive IoT devices.
2) We propose a novel identity aggregation scheme for H-DIDs, which enables multi-attribute authentication of different attributes in a single verification process. Compared to accessing signatures for all heterogeneous attributes associated with a DID, our solution significantly improves the efficiency of H-DID authentication and reduces the storage costs on the blockchain.
3) We propose an anonymous Verifiable Credential structure based on randomized labels. The aggregated signatures in the anonymous VCs possess properties of unforgeability and unlinkability. Thus, AVCA-hDID ensures reliable and secure DID authentication while preserving anonymity and privacy.

The organization of this paper follows the following order. Section II explains preliminaries of our work. Sections III and V present the model design and security analysis of AVCA-hDID, respectively. Section VI provides experimental evaluations. We illustrate the related work in Section VII and our conclusions is given in Section VIII.

II. Preliminaries

Notations. The notations and corresponding descriptions of this paper are shown in Table I as follows.

Cyclic Groups and Generators. A cyclic group is a special type of finite group in which there exists an element (generator) such that repeated group operations with that element can generate all the elements in the group. The order of a generator is defined as the smallest positive integer i such that g^i is equal to the identity element. In a finite group G of order n, if an element g has an order i, then i divides n. This means that the order of any element is a divisor of the group's order. Moreover, if g is an element of order i in G, then $g^x = g^y$ if and only if $x \equiv y \mod i$. This indicates that the sequence of powers of g repeats in cycles of length i. For a group $\mathbb{G}$ of prime order p, it is necessarily cyclic, meaning there exists a generator g such that every element of $\mathbb{G}$ can be represented as some power of g. Moreover, all elements of $\mathbb{G}$ other than the identity can serve as generators, each capable of generating the entire group through their powers.

TABLE I: Descriptions of Notations

Notation	**Description**
$\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$	Cyclic groups of prime order p in bilinear group
p	Prime order of cyclic groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$
g, $\mathfrak{g}$	Generators of group $\mathbb{G}_1$ and $\mathbb{G}_2$
e	Bilinear pairing map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$
$\mathcal{P}$	Tag domain defined over $\mathbb{G}_1$
h	Generator element selected from $\mathbb{G}_1^*$ during tag generation
$\tilde{p}$	Secret parameter in $\mathbb{Z}_p$ serving as user's private key
ρ	random tag
t_j, u_j, v_j	Private keys for j-th institution
$r_{j,i}, s_{j,i}$	Private keys for i-th attribute of j-th institution
VP_j	Public key of j-th institution
$VP_{j,i}$	Public key for i-th attribute of j-th institution
σ	Aggregated signature for multiple attributes
sig_j	Signature from j-th institution
$FinalKey$	Aggregated public key
$DID[i]$	i-th decentralized identifier
$FinalDID$	Aggregated DID identifiers
k	Security parameter in Setup algorithm
$\mathcal{H}$	Hash function on $\mathbb{G}_1$ domain
λ	Security parameter in security assumptions
$\mathcal{D}, \mathcal{R}$	Distributions in DDH assumption
$\mathcal{D}_{SDH}, \mathcal{U}$	Distributions in DSqDH assumption
$\text{negl}(\lambda)$	Negligible function in security parameter λ
$a_{j,i}$	Content of i-th attribute issued by j-th institution
β	Randomization exponent in signature aggregation

Discrete Logarithm. The Discrete Logarithm Assumption asserts that calculating the discrete logarithm in a discrete logarithm group is computationally difficult. In particular, within a group $\mathbb{G}$ defined by a generator g and a large prime p, the challenge is to find x in the equation $g^x \equiv y \pmod p$ when g and y are given. Recovering the value of x that satisfies $g^x \equiv y \pmod p$ requires substantial computational effort.

The *Square Discrete Logarithm* (SDL) assumption extends the Discrete Logarithm problem. It suggests that computing the square root of an element in a finite field is as hard as solving the discrete logarithm. In a group $\mathbb{G}$ of prime order p with a generator g and an element y, the assumption posits that finding an integer x such that $g^{x^2} \equiv y \pmod p$ is as difficult as solving the Discrete Logarithm problem.

Bilinear pairing. A bilinear pairing maps elements from two vector spaces to an element in another vector space. The asymmetric bilinear setup consists of $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$, where $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are cyclic groups of prime order p. Here, g_1 and g_2 are the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. The bilinear pairing e maps elements from $\mathbb{G}_1 \times \mathbb{G}_2$ to $\mathbb{G}_T$ and satisfies the following conditions:

- *Bilinearity.* For any $a, b \in \mathbb{Z}_p$ and $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$, the bilinear pairing satisfies $e(P^a, Q^b) = e(P, Q)^{ab}$. This means that $e(P^a, Q) = e(P, Q)^a$ and $e(P, Q^b) = e(P, Q)^b$.
- *Non-degeneracy.* For any $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, $e(g_1, g_2) \neq 1$.
- *Computability.* For any $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, there exists an efficient polynomial-time algorithm to compute $e(P, Q)$.

Security Assumption. Based on bilinear pairing and discrete logarithm, the following assumptions can be formally defined:

Definition II.1. ***Decisional Diffie-Hellman (DDH) assumption:*** *Let $\mathbb{G}$ be a cyclic group of prime order p with generator g. Given three elements g^a, g^b, and g^c (where a, b, and c*

are random values) . It is difficult to distinguish between the following two distributions through computation.

$$\mathcal{D} = (g, g^a, g^b, g^{ab}) \in \mathbb{G}_4; a, b \in \mathbb{Z}_p$$

$$\mathcal{R} = (g, g^a, g^b, g^c) \in \mathbb{G}_4; a, b, c \in \mathbb{Z}_p$$

Specifically, for any adversary $\mathcal{A}$ with Probabilistic Polynomial Time (PPT) algorithm, the advantage $Adv_{\mathcal{A}}^{DDH}$ is negligible in distinguishing between the random oracle experiment $\mathcal{R}$ and the DDH experiment $\mathcal{D}$:

$$Adv^{\mathcal{A}}_{DDH}(\lambda) = \left|Pr\left[\mathcal{A}(\mathcal{R}) = 1\right] - Pr\left[\mathcal{A}(\mathcal{D}) = 1\right]\right| \leq \text{negl}(\lambda)$$

where λ is the security parameter.

Definition II.2. ***Decisional Square Diffie-Hellman(DSqDH) assumption:*** *Given a cyclic group $\mathbb{G}$ of prime order p, a generator g, and two elements g^a, g^b (where a and b are randomnesses), the following two distributions are computationally indistinguishable:*

$$\mathcal{D}_{SDH} = (g, g^a, g^{a^2}) \in \mathbb{G}_3; a, b \in \mathbb{Z}_p$$

$$\mathcal{U} = (g, g^a, g^b) \in \mathbb{G}_3; a, b \in \mathbb{Z}_p$$

Similar to the DDH assumption, DSqDH also assumes that the advantage of $\mathcal{A}$ with PPT algorithms is negligible:

$$\begin{aligned} Adv^{\mathcal{A}}_{DSqDH}(\lambda) = &\left|Pr\left[\mathcal{A}(\mathcal{U}) = 1\right] - Pr\left[\mathcal{A}(\mathcal{D}_{SDH}) = 1\right]\right| \\ &\leq \text{negl}(\lambda) \end{aligned}$$

where λ is the security parameter.

III. Proposed Model

A. Design goals

The design goals of our approach primarily consider the implementation of DID in the context of IoT, which implies that massive DID Holders shall be considered to be a fundamental requirement for this goal. Thus, based on this basic requirement, we aim at achieving following design goals.

Heterogeneous Identity Aggregation. In the context of IoT, a DID Holder can possess multiple DIDs issued by various DID Issuers. Corresponding to VCs, each DID can sign claims regarding multiple attributes of the DID Holder at that DID Issuer. The attributes issued by different DID Issuers may be different. Verifying attributes from different DID Issuers may cause many challenges. Such as low verification efficiency, high system complexity, and significant storage load. Therefore, we should implement the aggregation of heterogeneous identity attributes to reduce storage load and system complexity, while enhancing verification efficiency.

Identity Security and Trustworthiness. Identity is the foundation of access control in an IoT system. To ensure the security of DID Holder identity and maintain the trustworthiness of the system, the system should verify DID Holder identity credentials. This ensures the correctness of DID Holder identity and prevents malicious tampering or forgery of DID Holder identity information.

Privacy Protection. The correlation of heterogeneous identity attributes may reveal a DID Holder's identity information. To protect the privacy of DID Holder data, we need to anonymize DID Holder identity attributes. The system needs to hide the DID Holder's true attribute information and ensure the unlinkability of DID Holders' information.

In a heterogeneous identity aggregation model, the security of identities is paramount. As identity attributes are issued by different DID Issuers, which is complex and variable. Therefore, the security measures must ensure the following aspects.

(i) Correctness: The model should accurately verify the authenticity of identity attributes from various sources. This involves cross-checking the attributes against the original issuers' databases and ensuring that they have not been altered or tampered with. Correctness guarantees that the DID Holder identity being presented is valid and accurate, thus preventing identity fraud. **(ii) Unforgeability:** The model should implement robust cryptographic techniques to ensure that identity attributes cannot be forged. This includes using digital signatures and public key infrastructure to validate.The model need to validate that the attributes were indeed issued by legitimate DID Issuers. Unforgeability is critical in maintaining trust in the model, as it ensures that only genuine attributes are accepted and recognized. **(iii) Unlinkability:** To protect DID Holder privacy, the model must ensure that identity attributes cannot be linked across different DID Issuers. This involves anonymizing DID Holder's data, which allow for the verification of identity attributes without revealing the DID Holder's true identity. Unlinkability ensures that even if identity attributes from multiple sources are aggregated, they cannot be used to track or profile the DID Holder.

By addressing these security aspects, a heterogeneous identity aggregation model can effectively manage and protect DID Holder identities. The model can ensure both the integrity of the system and the privacy of the DID Holders.

B. System Model

To meet the design goals, we aggregate the heterogeneous identities of DID Holders, which reduces the consumption of the time and resources. Fig. 2 shows the structure and workflow of our model. There are three types of participants in AVCA-hDID, DID Issuer, DID Holder, and DID Verifier. The DID issuer refers to the signing entity within the system, which possesses a public-private key pair and one or more signature algorithms. This entity can generate a signature by applying the signature algorithm to the content, thus producing a identifier that attests to the issuer's endorsement of the content. If the signature and public key pass the validation algorithm, the signature can be considered valid. The DID Holder can apply for an identity certificate from the issuer. A DID Verifier can verify whether the holder's identity corresponds correctly to prevent the occurrence of untrustworthy identity data. As shown in Fig. 2, the DID Issuer issues signatures containing the identity attributes for each DID Holder. Then, the DID Holder uploads these VCs with signatures to the blockchain and aggregates the identifiers and attributes. This process uses

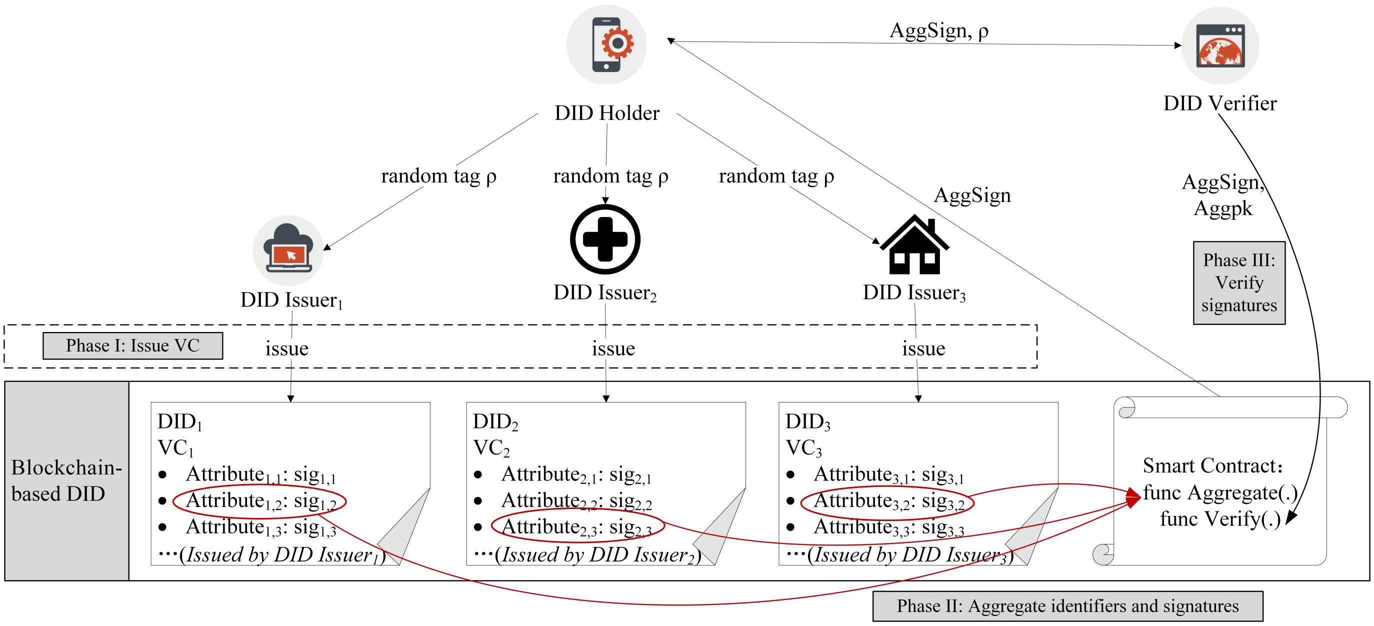

AggSign, ρ
DID Holder
DID Verifier
random tag ρ
random tag ρ
random tag ρ
AggSign
AggSign, Aggpk
Phase III: Verify signatures
DID Issuer$_1$
DID Issuer$_2$
DID Issuer$_3$
Phase I: Issue VC
issue
issue
issue
Blockchain-based DID
DID$_1$
VC$_1$
Attribute$_{1,1}$: sig$_{1,1}$
Attribute$_{1,2}$: sig$_{1,2}$
Attribute$_{1,3}$: sig$_{1,3}$
···(Issued by DID Issuer$_1$)
DID$_2$
VC$_2$
Attribute$_{2,1}$: sig$_{2,1}$
Attribute$_{2,2}$: sig$_{2,2}$
Attribute$_{2,3}$: sig$_{2,3}$
···(Issued by DID Issuer$_2$)
DID$_3$
VC$_3$
Attribute$_{3,1}$: sig$_{3,1}$
Attribute$_{3,2}$: sig$_{3,2}$
Attribute$_{3,3}$: sig$_{3,3}$
···(Issued by DID Issuer$_3$)
Smart Contract:
func Aggregate(.)
func Verify(.)
Phase II: Aggregate identifiers and signatures

Fig. 2: The workflow of proposed AVCA-hDID model.

random tag methods to ensure the anonymity and unlinkability of identity information, effectively protecting the privacy of the DID Holder. Finally, the DID Verifier uses smart contracts to verify the aggregated signatures and public keys, ensuring the validity of the provided information.

At each signer node, content aggregation of DIDs is performed on identifiers and attribute information. Each signer only needs to complete the signing of multiple distributed digital identities with a single signature. Each signer then signs the previously aggregated DIDs using the signature algorithm originally deployed with their node's chain code. The signer nodes employ a randomization algorithm to abstract the resulting signatures, generating one-time keys with randomness. The signer nodes use the single signature algorithm from the aggregation function to produce single signatures. Public key aggregation is performed by applying linear operations to each signer's public key, combining them into a single aggregated public key that represents all signers. Similarly, signature aggregation is achieved by linearly combining the single signatures of multiple signers into a single aggregated signature. These aggregated signatures and public key parameters are then sent to the credential verifier. Upon receiving the aggregated signature and public key parameters, the credential verifier uses a signature verification function to validate them. The shared parameters are used to compute the verification parameter e_1, which involves operations with the aggregated signature and the elliptic curve base point $\mathbb{G}$ to obtain the verification parameter e_2. The two verification parameters, e_1 and e_2, are then compared. If they are equal, the aggregation is deemed valid; otherwise, it is deemed invalid.

C. Model Design

To protect the privacy of DID Holder, the aggregation of heterogeneous DID Holder identities primarily involves three phases, namely, issuing VCs, aggregating identifiers and attributes, and verifying trusted signatures. As shown in Fig. 2, there are three phases for our AVCA-hDID model:

Phase I: Issuing VC. Each credential issuing authority generates a distributed digital identifier for the DID Holder and signs the attributes the DID Holder holds at that DID Issuer to create legitimate VCs, which are published on the blockchain. The signatures in the VCs generated during this stage can be repeatedly invoked by the DID Holder in subsequent processes for aggregation to meet different verification needs, significantly reducing the system's computational cost.

Phase II: Aggregating DID identifiers and signatures of attributes. DID Holders obtain VCs from the blockchain.And they select signatures for attributes based on specific requirements. They utilize a random function to process corresponding identity and attribute information. And they achievie heterogeneous aggregation of identity information and attributes through invoking aggregate functions in smart contracts deployed on the blockchain.

Phase III: Verifying trusted signatures. The verification DID Issuer validates the identity or attributes that the DID Holder wishes to verify based on the labels, aggregated public key, and aggregated signature provided by the DID Holder. This is achieved by invoking the verification function in the smart contract deployed on the blockchain. If the verification is successful, it indicates that the aggregate signature is legitimate and can be used as a trusted credential for the DID Holder's desired verification attributes. If the verification fails, it indicates that the attributes verified by the DID Holder are not entirely correct, and at least one illegal signature participated in the aggregation operation. In this case, It required a comprehensive inspection of all information aggregated by the DID Holder.

In the process of identity aggregation, based on the different classifications of the aggregated objects, the aggregation can be divided into content aggregation, signature aggregation, and public key aggregation. The process of Content aggregation is using an algorithm to combine several DIDs to be signed into a single entity. All parts can be signed in a single operation. This method of content aggregation improves signature efficiency. Signature aggregation refers to the process of combining multiple signatures into a single signature. Public key aggregation mainly combines multiple public keys into a single public key. In AVCA-hDID, public key aggregation is achieved by applying linear operations to each signer's public key, combining them into a single aggregated public key that represents all signers.

IV. Heterogeneous DID Aggregation

A. Verifiable Aggregation for H-DID

Due to the heterogeneity of identity attributes and decentralized storage, H-DID authentication necessitates cross-issuer collaboration, resulting in inefficiencies, security vulnerabilities, and scalability constraints in DID digital signatures. Furthermore, as a single DID Holder's identity attributes are distributed across multiple DID Issuers, a comprehensive H-DID authentication requires accessing data from various sources, leading to excessive network resource consumption, diminished authentication efficiency, and substantial load pressure on blockchain-based DID systems. To address these challenges, the proposed AVCA-hDID model employs a random tag-based verifiable aggregate signature scheme to achieve efficient H-DID aggregation, thereby streamlining the multi-issuer interactions involved in VC signing and verification while significantly improving signature verification efficiency. Notably, AVCA-hDID incorporates anonymous VC with random tag to ensure unlinkability in H-DID aggregation, not only enhancing the security of blockchain-based digital identity frameworks but also reinforcing privacy preservation mechanisms.

The identity aggregation algorithm for H-DID, presented as Alg. 1, comprises seven distinct phases: parameter initialization, tag generation, public-private key pair generation, DID Issuer signing, public key aggregation, signature aggregation, and aggregated signature verification. The initial phase of parameter initialization establishes the foundational cryptographic parameters for AVCA-hDID by generating an asymmetric bilinear group along with sets of valid and invalid tags. During this phase, AVCA-hDID takes a security parameter as input and constructs an asymmetric bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ where all three are cyclic groups of prime order p, with g and $\mathfrak{g}$ serving as random generators for $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. Additionally, it defines a tag domain over $\mathbb{G}_1$ where each tag consists of three distinct elements from this group. The output of this initialization phase includes both the asymmetric bilinear group configuration and the defined tag domain, which collectively form the common parameters utilized throughout the subsequent phases of the Alg. 1.

The tag generation phase focuses on creating a random secure tag for DIDs. Utilizing the public parameters established in the initialization phase, Alg. 1 randomly selects a generator element h from the multiplicative group $\mathbb{G}_1^*$ and a secret parameter $\tilde{p}$ from the integer field $\mathbb{Z}_p$. These components are combined to produce a structured tag $\rho = \left(h, h^{\tilde{p}}, h^{\tilde{p}^2}\right)$, where the exponentiation operations are performed within the group $\mathbb{G}_1$. The critical parameter $\tilde{p}$ functions as the DID Holder's private key and remains securely stored by the DID Holder. In addition, the random tag ρ is provided to the DID Verifier in the subsequent interactive proof, allowing the DID Verifier to verify the authenticity of the aggregated signature.

Algorithm 1 The Process of DID Aggregation

```
/* Phase 1 : Initialize Parameters*/
$(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g, \mathfrak{g}, e) \leftarrow k$: Input the security parameter $k$, and generate $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g, \mathfrak{g}, e)$, a set of asymmetric bilinear groups, where $g$ and $\mathfrak{g}'$ are random generators of $\mathbb{G}_1$ and $\mathbb{G}_2$.
$\mathcal{P} = \mathbb{G}^3$: Tag Collection
$\mathtt{pp} \leftarrow (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g, \mathfrak{g}, e; \mathcal{P})$: Output initialization parameters.
/* Phase 2 : Generate Tag*/
$h \xleftarrow{\$} \mathbb{G}_1^*, \tilde{p} \xleftarrow{\$} \mathbb{Z}_p$: Randomly select a generator and a parameter.
$\rho = (h, h^{\tilde{p}}, h^{\tilde{p}^2})$: Compute a random tag.
/* Phase 3 : Generate Public and Private Keys*/
$SP_j = [t_j, u_j, v_j], SP'_{j,i} = [r_{j,i}, s_{j,i}] \in \mathbb{Z}_p^5$: Randomly generate a one-time private key of the $j$-th DID Issuer.
$VP_j = [\mathfrak{g}^{t_j}, \mathfrak{g}^{u_j}, \mathfrak{g}^{v_j}], VP'_{j,i} = [\mathfrak{g}^{r_{j,i}}, \mathfrak{g}^{s_{j,i}}]) \in \mathbb{G}_2^5$: Calculate the corresponding public key of the $j$-th DID Issuer.
$sp_{j,i} \leftarrow (SP_j = [t_j, u_j, v_j], SP'_{j,i} = [r_{j,i}, s_{j,i}]), vp_{j,i} \leftarrow (VP_j = [\mathfrak{g}^{t_j}, \mathfrak{g}^{u_j}, \mathfrak{g}^{v_j}], VK'_{j,i} = [\mathfrak{g}^{r_{j,i}}, \mathfrak{g}^{s_{j,i}}])$: Output the public key $(vp_{j,i})$ of the $i$-th attribute of the $j$-th DID Issuer and the pravite key $(sp_{j,i})$ of the $i$-th attribute of the $j$-th DID Issuer.
/* Phase 4 : Sign the Content*/
$sig \leftarrow \rho_1^{t+r+ms} \times \rho_2^u \times \rho_3^v \in \mathbb{G}_1$: To sign the attributes, input the random tag $(\rho)$ and calculate the signature of the $j$-th issuer at index $(j, i)$.
/* Phase 5 : Aggregate Identifiers and Signatures*/
$FinalDID \leftarrow \mathsf{AggID}(DID_1, DID_2, ..., DID_n)$: For identifier aggregation, input the DID identifiers and output the aggregated identifiers.
$\sigma \leftarrow \mathsf{AggSig}(sig_1, sig_2, ..., sig_i)$: For signature aggregation, input the signature of each attribute and output the final aggregated signatures $(\sigma)$.
/* Phase 6 :Aggregate Public Keys*/
$fvp_j \leftarrow VP_j \cup [VP'_{j,i}]_i, FinalKey \leftarrow [fvp_j]_j$: Input the public keys to be aggregated and output the aggregated public key $(FinalKey)$.
/* Phase 7 :Verify the Legality of the Signature*/
$e_1 = e(\sigma, \mathfrak{g})$
$e_2 = e(\rho_1, \prod_j VP_{j,1}^{n_j} \times \prod_i VP'_{j,i,1} \cdot VP_{j,i,2}^{a_{j,i}})$
$e_3 = e(\rho_2, \prod_j VP_{j,2}^{n_j})$
$e_4 = e(\rho_3, \prod_j VP_{j,3}^{n_j})$
**Return** true **if** $e_1 == e_2 \times e_3 \times e_4$ **else return** false
```

A random one-time private key for each attribute and its public key are generated by the public-private key generation algorithm. The input of this phase is the public parameters generated by the above model. Five elements are randomly selected in $\mathbb{Z}_p$. Among them, the first three of which are used as the private key (SP_j) of the j-th DID Issuer, and the latter two are used as the private key ($sp_{j,i}$) of the i-th attribute of the j-th DID Issuer. Therefore, for the same DID Issuer (j), the private key (SP_j) is always consist of three parameters ($[t_j, u_j, v_j]$). In particular, for each i-th attribute of the same DID Issuer (j), another two parameters ($[r_{j,i}, s_{j,i}]$) are independently generated and included in $sp_{j,i}$. Finally, the public key vp is calculated based on the sp: $vp_{j,i} = (VP_j = [\mathfrak{g}^{t_j}, \mathfrak{g}^{u_j}, \mathfrak{g}^{v_j}], VP_{j,i} = [\mathfrak{g}^{r_{j,i}}, \mathfrak{g}^{s_{j,i}}])$, and the public and private keys of the i-th attribute of the j-th DID Issuer are output.

The DID Issuer signing phase combines the tag and the private key corresponding to each attribute to sign the content of the attribute. Input the content of the attribute to be signed and the private key corresponding to the attribute, including the private key of the DID Issuer to which the attribute belongs and the private key of the attribute itself, as well as the public tag ρ. Then, the signature $sig = \rho_1^{t+r+ms} \times \rho_2^u \times \rho_3^v \in \mathbb{G}_1$ is computed as the output.

To aggregates the signatures of all required attributes, serving as a trusted VC provided by the DID Holder to the verification DID Issuer, Alg. 1 adopts the signature aggregation phase. The input of this phase is several signatures of attributes. Then it outputs signatures which are multiplied sequentially in the $\mathbb{G}_1$ domain to obtain an aggregated signature.

Due to all the public keys are related to the aggregated attributes, we should form an aggregated public key by the public key aggregation phase. Then, verify the authenticity of the aggregated signature by the verification DID Issuer. The related public keys of all attributes involved in the aggregated signature mentioned above is as the input to the algorithm. For different attributes of the same DID Issuer, it is only necessary to record the DID Issuer's own public key once, and then combine it with the public keys corresponding to different attributes sequentially to obtain $fvp_j = VP_j \cup \{VP'_{j,i}\}$. Further, combine the fvp_j of each DID Issuer sequentially to obtain $FinalKey = [fvp_j]$ as the output.

The aggregated signature is verified by the signature verification phase. To prove the legality of the aggregated signature, the algorithm uses the properties of bilinear pairing. Input the public tag ρ, the aggregated public key $FinalKey$, the content vector ($\vec{a} = [a_i]$) of each attribute to be verified, and the aggregated signature σ and then compute the bilinear mappings. If the verification is successful, output true; otherwise, output false.

B. Identifier aggregation for Anonymous VC

Each DID Holder may be issued multiple identities by different DID Issuers, resulting in multiple DID identifiers. To ensure the authenticity of the DID Holder, AVCA-hDID must verify these DID identifiers. However, individually verifying multiple identities introduces significant computational overhead, leading to inefficiency. To address this issue, we propose a DID identifier aggregation method to optimize verification efficiency. The DID Verifier only needs to perform a single verification on the aggregated identity: if successful, it confirms the validity of all the DID Holder's identities across the verified issuers; if failed, it indicates that at least one identity is invalid or potentially tampered with, requiring further investigation. The identifier aggregation method AggID$(\cdot)$ (line 15 in Alg. 1) is shown in Eq. 1 as follows:

$$FinalDID = \sum_{i=1}^{n} DID[i] \tag{1}$$

where n is the number of DID to be aggregated.

To further enhance authentication efficiency during heterogeneous attribute aggregation, H-DID also aggregates the involved DID identifiers. The preprocessing and verification phases align with Alg. 1, ultimately producing the aggregated identity $FinalDID$.

Complexity. It needs to traverse all n DID identifiers and perform addition operations. Assuming each addition operation is constant time $O(1)$, the time complexity for the AggID$(\cdot)$ algorithm is $O(n)$.

C. Attribute aggregation for Anonymous VC

In different DID Issuers, the attributes of DID Holders are different. Therefore, to achieve efficient verification of different attributes for multiple DID Holders, AVCA-hDID proposes a novel anonymous VC based on the aggregation of heterogeneous identity attributes, as shown in Alg. 1. The specific computation steps of H-DID attribute aggregation are as follows:

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^k)$: For parameter initialization (line 1-4 in Alg. 1), given a security parameter k, to generate an asymmetric bilinear group and a tag domain $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g, \mathfrak{g}, e; \mathcal{P})$. At this point, define $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g, \mathfrak{g}, e; \mathcal{P}, \mathcal{H})$, where $\mathcal{H}$ is a hash function on the $\mathbb{G}_1$ domain.
- $\rho \leftarrow \mathsf{UKeyGen}(DID)$: For tag generation (line 6-7 in Alg. 1) of the DID Holder (U), let $h = \mathcal{H}(DID) \in \mathbb{G}_1^*$, select a randomness $\tilde{p} \xleftarrow{\$} \mathbb{Z}_p$ as the private key to participate in subsequent interactive proofs. Then, calculate $\rho \leftarrow (h, h^{\tilde{p}}, h^{\tilde{p}^2}) \in \mathbb{G}_1^3$ as the random tag, which can be presented to the DID Verifier during the verification process.
- $sp_{j,i},\ vp_{j,i} \leftarrow \mathsf{CIKeyGen}(\mathsf{pp})$: For key pair generation (line 8-11 in Alg. 1), the DID Issuer generates private key (sp_j) and public key (vp_j) for the n attributes it possesses. Three elements t_j, u_j, v_j in SP_j are randomly selected from the set $\mathbb{Z}_p$. And for n attributes of a DID Holder, the elements $r_{j,i}, s_{j,i}$ $(i \in n)$ in $SP'_{j,i}$ $(i \in n)$ are randomly selected from $\mathbb{G}_2$. Then, we use these randomnesss to compute vp_j as follows:

$$sp_j^n = (SP_j = [t_j, u_j, v_j], (SP'_{j,i} = [r_{j,i}, s_{j,i}])_{i \in n}) \in \mathbb{Z}_p^{3+2n}$$

$$vp_j^n = \begin{pmatrix} VP_j = [\mathfrak{g}^{t_j}, \mathfrak{g}^{u_j}, \mathfrak{g}^{v_j}], \\ (VP'_{j,i} = [\mathfrak{g}^{r_{j,i}}, \mathfrak{g}^{s_{j,i}}])_{[i \in n]} \end{pmatrix} \in \mathbb{G}_2^{3+2n}$$

- $VC_{sig_j}^{[a_i]} \leftarrow \mathsf{VCIssue}\ (U, \rho, vp, [a_i]_{i \in n}; sp_j^n)$: For attribute signatures (line 13 in Alg. 1), the DID Holder U provides the random tag (ρ) and applies for VCs from a trusted

DID Issuer. The issuer signs the content of each attribute ($[a_i]$) using the private key (sp_j^n) possessed by the DID Holder, which serves as the signature sig_j of the attribute ($[a_i]$) in the trusted VC ($VC_{sig_j}^{[a_i]}$).

- $(FinalKey, (a_{j,i})_{j,i}, \rho', \sigma') \leftarrow \mathsf{VCShow}\ (\rho, VC_{sig_{j,i}}^{[a_j,i]}, (VP_j,\ VP'_{j,i}))$: For key aggregation (line 18 in Alg. 1), the DID Holder (U) aggregates the public keys to obtain $FinalKey \leftarrow [fvp_j]_j = \cup_j (VP_j \cup [VP'_{j,i}]_i)$. For signature aggregation (line 16 in Alg. 1), the DID Holder uses a randomness $\beta \xleftarrow{\$} \mathbb{Z}_p$ to randomize the parameter ρ as $\rho' \leftarrow (\rho_1^\beta, \rho_2^\beta, \rho_3^\beta) \in \mathbb{G}_1$. Then DID Holder aggregates the signatures to obtain $\sigma \leftarrow \prod_{j,i} sig_{j,i} \in \mathbb{G}_1$, and computes $\sigma' \leftarrow \sigma^\beta$.
- $0/1 \leftarrow \mathsf{VCVerify}\ ((FinalKey, (a_{j,i})_{j,i}, \rho', \sigma'), (VP_j, VP'_{j,i}))$: For verification (line 20-24 in Alg. 1), the DID Holder sends $(FinalKey, (a_{j,i})_{j,i}, \rho', \sigma')$ to the DID Verifier. DID Verifier verifies the aggregated signature to prove the legality of the aggregated signature.

Complexity. Within the proposed attribute aggregation scheme for anonymous VC, let N denote the total number of attributes possessed by a DID Holder, n represent the number of attributes requiring aggregated verification, and K signify the number of DID Issuers. The storage and transmission complexity of AVCA-hDID are quantified by the number of elements in the groups $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, and the field $\mathbb{Z}_p$. The total public key ($[vp_j]_{j\in K}$) stored on the blockchain exhibit a size of $(3K+2N)$ elements in $\mathbb{G}_2$. Regarding verification for n aggregated attributes, the public key ($[vp_j^n]_{j\in K}$) provided during the process require $(3K+2n)$ elements in $\mathbb{G}_2$. In addition, $\mathsf{VCShow}\ (\cdot)$ involves 4 elements in $\mathbb{G}_1$ (3 elements for tag randomization, and 1 for the aggregated signature), and 1 vector element ($\vec{a}$) in $\mathbb{Z}_p$. For $\mathsf{VCVerify}\ (\cdot)$, the DID Verifier performs $(n+3K)$ exponentiations in $\mathbb{G}_2$ and 4 bilinear pairing operations. Notably, multiplication operations are omitted from this analysis due to their negligible impact on performance.

D. Security Model

Correctness of AVCA-hDID. *Correctness* means that any valid signature aggregation can pass verification under the corresponding tags and the aggregated public key. We remark that $VC_{sig_j}^{[a_i]}$ generated via $\mathsf{VCIssue}\ (\cdot)$ is respectively successfully verified by $\mathsf{VCVerify}(\cdot)$. *Correctness* formally is formulated by Eq. (3) as follows:

$$Pr\left[\mathsf{VCVerify}\begin{pmatrix} FinalKey, (a_{j,i})_{j,i}, \\ \rho', \sigma', VP_j, VP'_{j,i} \end{pmatrix} = 1\right] = 1. \quad (3)$$

Unforgeability of AVCA-hDID. *Unforgeability* means that even if an adversary can intercept tags and signatures previously provided by users and aggregate them, it remains computationally infeasible to forge a valid tag that passes verification. Formally, it is required that for each adversary $\mathcal{A}$ with PPT algorithm, the chance of winning DID Verifier is negligible:

$$Pr\left[\begin{array}{c} \mathtt{pp} \leftarrow \mathsf{Setup}(1^k), \\ \forall j \in J, (sp_j, vp_j) \leftarrow \mathcal{A}(\mathtt{pp}): \\ 1 \leftarrow \begin{pmatrix} \mathcal{A}(vp_j), \\ \mathsf{VCVerify}(vp_j, a_j) \end{pmatrix} \end{array}\right] \leq \mathrm{negl}(\lambda). \quad (4)$$

Unlinkability of AVCA-hDID. Unlinkability, based on the Decisional Diffie-Hellman (DDH) and Decisional Square Diffie-Hellman (DSqDH) assumptions, ensures that an adversary cannot link multiple identities of a user through their tags and signatures. Refer to [10], the tags are random SqDH triplets. When any tag $(\tilde{\rho_1}, \rho_1) \leftarrow GenTag(1^\kappa)$ is randomized to ρ'_1, the distributions $(\mathrm{g}_0, \mathrm{g}_0^x, \mathrm{g}_0^{x^2}, \mathrm{g}_1, \mathrm{g}_1^x, \mathrm{g}_1^{x^2})$ and $(\mathrm{g}_0, \mathrm{g}_0^x, \mathrm{g}_0^{x^2}, \mathrm{g}_1, \mathrm{g}_1^y, \mathrm{g}_1^{y^2})$ are indistinguishable on $\mathbb{G}_\$^6$. Based on the assumptions of DDH and DSqDH, *Unlinkability* of AVCA-hDID can be proven that the following two distributions are indistinguishable:

$$\mathcal{D}_0 = (\mathrm{g}_0, \mathrm{g}_0^x, \mathrm{g}_0^{x^2}, \mathrm{g}_1, \mathrm{g}_1^x, \mathrm{g}_1^{x^2}) \quad (5a)$$

$$\mathcal{D}_1 = (\mathrm{g}_0, \mathrm{g}_0^x, \mathrm{g}_0^{x^2}, \mathrm{g}_1, \mathrm{g}_1^y, \mathrm{g}_1^{y^2}) \quad (5b)$$

for $\mathrm{g}_0, \mathrm{g}_1 \in \mathbb{G}$ and $x, y \in \mathbb{Z}_p$.

V. Security Analysis

Lemma V.1. *(Correctness of H-DID aggregation) If DIDs are heterogeneous and implement aggregation, AVCA-hDID model holds* Security, *which means that any legitimate signature can be verified by the corresponding tag and aggregation public key.*

Proof. Our proposed AVCA-hDID model further introduces random tags to associate different messages under different key signatures with the same tag, facilitating aggregation in subsequent processes. In this scheme, the tag serves as a temporary pseudonym used by the user, and its private key is randomly generated by the user and self-managed for interactive verification. The public key is submitted to the verification authority for signature verification. After randomization, it can still maintain the association with the same user but becomes unlinkable, providing anonymity to the user.

For the deterministic aggregation scheme, where the entire scheme requires only one aggregation signature and users do not need to consider the unlinkability between multiple tags, there is no need to randomize the tags and aggregation signature. The correctness proof of the verification process is as follows:

$$\begin{aligned}
e(\sigma, \mathfrak{g}) &= e\left(\prod_{j,i} \sigma_{j,i}, \mathfrak{g}\right) = e\left(\prod_{j,i}\left(\rho_1^{t_j + r^{j,i} + m_{j,i}s_{j,i}} \times \rho_2^{u_j} \times \rho_3^{v_j}\right), \mathfrak{g}\right) \\
&= e\left(\prod_{j,i} \rho_1^{t_j + r_{j,i} + m_{j,i}s_{j,i}}, \mathfrak{g}\right) \times e\left(\prod_{j,i} \rho_2^{u_j}, \mathfrak{g}\right) \times e\left(\prod_{j,i} \rho_3^{v_j}, \mathfrak{g}\right) \\
&= e\left(\prod_j \rho_1^{t_j n_j} \times \prod_i \rho_1^{r_{j,i}} \cdot \rho_1^{m_{j,i}s_{j,i}}\right) \times e\left(\prod_j \rho_2^{u_j n_j}, \mathfrak{g}\right) \times e\left(\prod_j \rho_3^{v_j n_j}, \mathfrak{g}\right) \\
&= e\left(\rho_1, \mathfrak{g}\right)^{\sum_j \left[t_j n_j + \sum_i (r_{j,i} + m_{j,i} s j,i)\right]} \times e\left(\rho_2, \mathfrak{g}\right)^{\sum_j u_j n_j} \times e\left(\rho_3, \mathfrak{g}\right)^{\sum_j v_j n_j} \\
&= e\left(\rho_1, \prod_j \mathfrak{g}^{t_j n_j} \times \prod_i \mathfrak{g}^{r_{j,i}} \cdot \mathfrak{g}^{m_{j,i}s_{j,i}}\right) \times e\left(\rho_2, \prod_j \mathfrak{g}^{u_j n_j}\right) \times e\left(\rho_3, \prod_j \mathfrak{g}^{v_j n_j}\right) \\
&= e\left(\rho_1, \prod_j VP_{j,1}^{n_j} \times \prod_i VP'_{j,i,1} \cdot VP'^{m_{j,i}}_{j,i,2}\right) \times e\left(\rho_2, \prod_j VP_{j,2}^{n_j}\right) \times e\left(\rho_3, \prod_j VP_{j,3}^{n_j}\right)
\end{aligned} \quad (6)$$

Clearly, the deterministic scheme exhibits lower system efficiency since it only utilizes the attribute signatures in the verifiable credential once. To enhance the overall efficiency of the system and enable the credential issuer to reuse signatures for multiple attributes based on actual requirements, our proposed aggregation scheme for heterogeneous digital identities based on blockchain adopts the aggregation signature algorithm based on SqDH with random tags. During the verification process, users randomize the aggregation signature and tags to ensure the unlinkability of the signatures. The correctness proof of the verification process is similar to the deterministic scheme and is as follows:

$$\begin{aligned}
e(\sigma', \mathfrak{g}) &= e(\sigma^{\beta}, \mathfrak{g}) = e(\sigma, \mathfrak{g})^{\beta} = e(\prod_{j,i} \sigma_{j,i}, \mathfrak{g})^{\beta} = e(\prod_{j,i}(\rho_1^{t_j + r^{j,i} + m_{j,i}s_{j,i}} \times \rho_2^{u_j} \times \rho_3^{v_j}), \mathfrak{g})^{\beta} \\
&= e(\prod_{j,i} \rho_1^{t_j + r_{j,i} + m_{j,i}s_{j,i}}, \mathfrak{g})^{\beta} \times e(\prod_{j,i} \rho_2^{u_j}, \mathfrak{g})^{\beta} \times e(\prod_{j,i} \rho_3^{v_j}, \mathfrak{g})^{\beta} \\
&= e(\prod_{j} \rho_1^{t_j n_j} \times \prod_{i} \rho_1^{r_{j,i}} \cdot \rho_1^{m_{j,i}s_{j,i}})^{\beta} \times e(\prod_{j} \rho_2^{u_j n_j}, \mathfrak{g})^{\beta} \times e(\prod_{j} \rho_3^{v_j n_j}, \mathfrak{g})^{\beta} \\
&= e(\rho_1, \mathfrak{g})^{\beta \sum_j [t_j n_j + \sum_i (r_{j,i} + m_{j,i} sj,i)]} \times e(\rho_2, \mathfrak{g})^{\beta \sum_j u_j n_j} \times e(\rho_3, \mathfrak{g})^{\beta \sum_j v_j n_j} \\
&= e(\rho_1, \prod_{j} \mathfrak{g}^{t_j n_j} \times \prod_{i} \mathfrak{g}^{r_{j,i}} \cdot \mathfrak{g}^{m_{j,i}s_{j,i}})^{\beta} \times e(\rho_2, \prod_{j} \mathfrak{g}^{u_j n_j})^{\beta} \times e(\rho_3, \prod_{j} \mathfrak{g}^{v_j n_j})^{\beta} \\
&= e(\rho_1, \prod_{j} VK_{j,1}^{n_j} \times \prod_{i} VK'_{j,i,1} \cdot VK'^{m_{j,i}}_{j,i,2})^{\beta} \times e(\rho_2, \prod_{j} VK_{j,2}^{n_j})^{\beta} \times e(\rho_3, \prod_{j} VK_{j,3}^{n_j})^{\beta} \\
&= e(\rho_1^{\beta}, \prod_{j} VK_{j,1}^{n_j} \times \prod_{i} VK'_{j,i,1} \cdot VK'^{m_{j,i}}_{j,i,2}) \times e(\rho_2^{\beta}, \prod_{j} VK_{j,2}^{n_j}) \times e(\rho_3^{\beta}, \prod_{j} VK_{j,3}^{n_j})
\end{aligned} \tag{7}$$

□

Lemma V.2. *(Unforgeability of H-DID aggregation) If DIDs are heterogeneous and implement aggregation, AVCA-hDID model holds* Unforgeability*, which means that even if the attacker can intercept the tags and signatures previously provided by the user for aggregation, it is difficult to forge a legitimate tag through interactive proof.*

Proof. Given an effective SqDH group $(g_i, a_i = g_i^{w_i}, b_i = a_i^{v_i})$, where $g_i \in G^*$, $w_i, v_i \in \mathbb{Z}_p^*$, we need to output at least two non-zero integers α_i such that the new group $(G = \prod g_i^{\alpha_i}, A = \prod a_i^{\alpha_i}, B = \prod b_i^{\alpha_i})$ is an effective new SqDH group with respect to DL. Assume it is difficult to construct a new valid SqDH group based on a set of indices α_i but knowing the logarithm base and the random values. In simple terms, it is difficult to construct a new valid SqDH group based on the linear combination of indices α_i known but random logarithms and values.

HPP signature is a homomorphic signature over G or its exponentiation [10], [11], assuming $\mathbf{m} = (m_1, \ldots, m_n) \in \mathbb{Z}_p$, $g \in G$, it is infeasible to forge the same signature for $g^{\mathbf{m}}$. This means that only signatures with the same tag can be legally combined linearly. Similar to the HPP signature, in this scheme the central tag $\rho = \left(h, h^{\tilde{\rho}}, h^{\tilde{\rho}^2}\right)$ is used. When signing, $\sigma = \rho_1^{t+r+ms} \times \rho_2^{u} \times \rho_3^{v} = (h^1)^r \times (h^m)^s \times h^t \times (h^{\tilde{\rho}})^u \times (h^{\tilde{\rho}^2})^v$ is used, which is exactly the result of signing (r, s, t, u, v) with (h, h^m). Therefore, based on the infeasibility of the HPP signature, the signature in the composite signature scheme of the heterogeneous digital identity cluster proposed in this paper is also infeasible to forge.

To ensure the infeasibility of signature forgery, the generation of each tag must be random, so in the second step of the scheme, each user's signature tag is generated by $H(id)$, thus ensuring that the recipient is aware of the initial state of the tag. Since all tags are randomly generated in the uncertain scheme, this scheme does not have the issue of private key leakage. □

Lemma V.3. *(Unlinkability of H-DID aggregation) If DIDs are heterogeneous and implement aggregation, AVCA-hDID model holds* Unlinkability*, which means that it is difficult for attackers to link multiple users through tags and signatures.*

Proof. For the Unlinkability, it can be proven that both distributions are indistinguishable from a randomly independent 6-tuple (distribution $\mathbb{G}_6$):

$$\begin{aligned}
\mathcal{D}_0 &\approx (\mathrm{g}_0, \mathrm{g}_0^x, \mathrm{g}_0^y, \mathrm{g}_1, \mathrm{g}_1^x, \mathrm{g}_1^y), \mathrm{g}_0, \mathrm{g}_1 \in \mathbb{G}, x, y \in \mathbb{Z}_p(\text{DSqDH}) \\
&\approx (\mathrm{g}_0, \mathrm{g}_0^x, \mathrm{g}_0^y, \mathrm{g}_1, \mathrm{g}_1^u, \mathrm{g}_1^v), \mathrm{g}_0, \mathrm{g}_1 \in \mathbb{G}, x, y, u, v \in \mathbb{Z}_p(\text{DDH}) \\
&\approx (\mathrm{g}_0, \mathrm{g}_0^x, \mathrm{g}_0^{x^2}, \mathrm{g}_1, \mathrm{g}_1^u, \mathrm{g}_1^{u^2}), \mathrm{g}_0, \mathrm{g}_1 \in \mathbb{G}, x, u \in \mathbb{Z}_p(\text{DSqDH}) \\
&= \mathcal{D}_1
\end{aligned}$$

For a user $\mathcal{U}$ who needs to complete multi-round signature aggregation, in order to prevent the verifier or attackers from associating different aggregated attributes across multiple rounds and compromising their privacy, the user has two approaches $\mathcal{H}_0$ and $\mathcal{H}_1$.

$$\begin{aligned}
&\mathcal{H}_0 : \mathcal{S}_1 = (\rho_1, \rho_2) = (h_0, h_0^x, h_0^{x^2}, h_1, h_1^y, h_1^{y^2}) \\
&\mathcal{H}_1 : \mathcal{S}_2 = (\rho_1, \rho_1') = (h_0, h_0^x, h_0^{x^2}, h_0^{\beta}, (h_0^{\beta})^x, (h_0^{\beta})^{x^2})
\end{aligned}$$

In $\mathcal{H}_0$, each round, the user generates a random tagρ_1, ρ_2, etc., which is completely unrelated to their identity. In the first round, they provide ρ_1 to the verifiable credential issuer for signing the attributes of that round. Then, they invoke a smart contract to aggregate the signatures, which are subsequently submitted to the verification authority for verification. In the second round, ρ_2 is provided to the verifiable credential issuer for another round of signing, repeating the same operations as in the first round. This process is repeated for each additional round.

In $\mathcal{H}_1$, each round, at the initialization, the user generates an initial tagρ_1 and provides it to the verifiable credential issuer for signing all the possible attributes that may be aggregated in the future. These signed attributes are then published on the blockchain. In each round, the tagis randomized to generate ρ_1'. The required attribute signatures are selected for aggregation, and the signatures are also randomized in the same manner. Finally, the aggregated signature is submitted to the verification authority for verification. The same process is followed for subsequent rounds, using tagrandomization instead of generating new tags as in $\mathcal{H}_0$.

Based on the previous proof, it can be concluded that the distribution $\mathcal{S}_1$ and $\mathcal{S}2$ are indistinguishable on $\mathbb{G}_{\$}^6$. The verifier or attacker would find it difficult to associate user information based on the received tags. The randomization operation effectively achieves the unlinkability between tags, ensuring user anonymity and protecting user privacy. It can be observed that H_1 significantly reduces the system's complexity and storage burden compared to H_0, reducing the workload for users and the verifiable credential issuer. The verifiable credential issuer can issue signatures for multiple attributes in a single step, and users can invoke aggregation based on

TABLE II: Experiment Settings

Settings	Metrics	Number of Signatures
1-1	AST	1,000
1-2	AVT	1,000
2-1	MU	1,000
2-2	CU	1,000
3-1	TST	(10, 100, 200, 300, 400, 500, 1,000)
3-2	TVT	(10, 100, 200, 300, 400, 500, 1,000)
3-3	Sub-alg. Latency	(10, 100, 200, 300, 400, 500, 1,000)

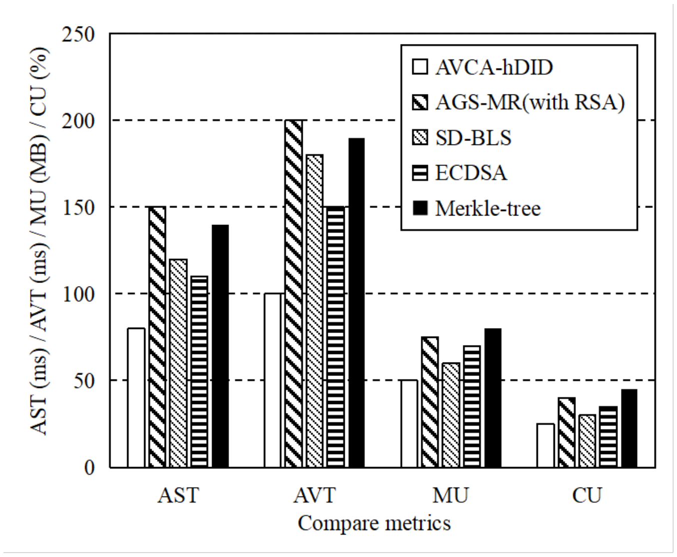


Fig. 3: Comparisons of efficiency between our scheme and other methods.

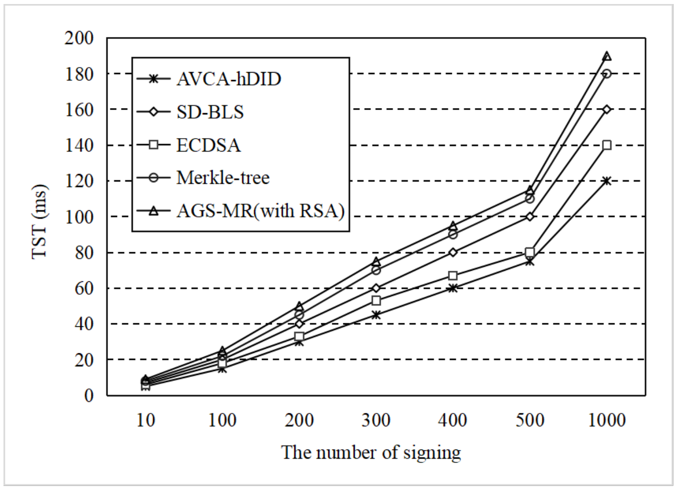


Fig. 4: Comparisons of the total signing time performance.

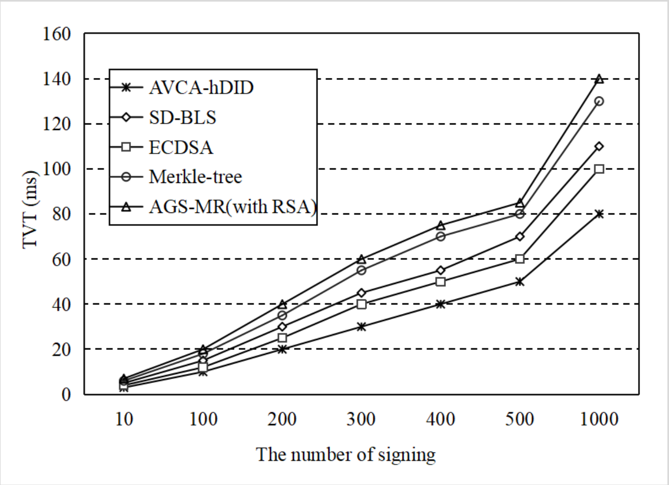


Fig. 5: Comparisons of the total verification time.

the same tagfor different attribute signatures according to their actual needs, fulfilling identity authentication requirements in various scenarios. The provided unlinkability greatly protects user privacy and achieves anonymity for user identities.

□

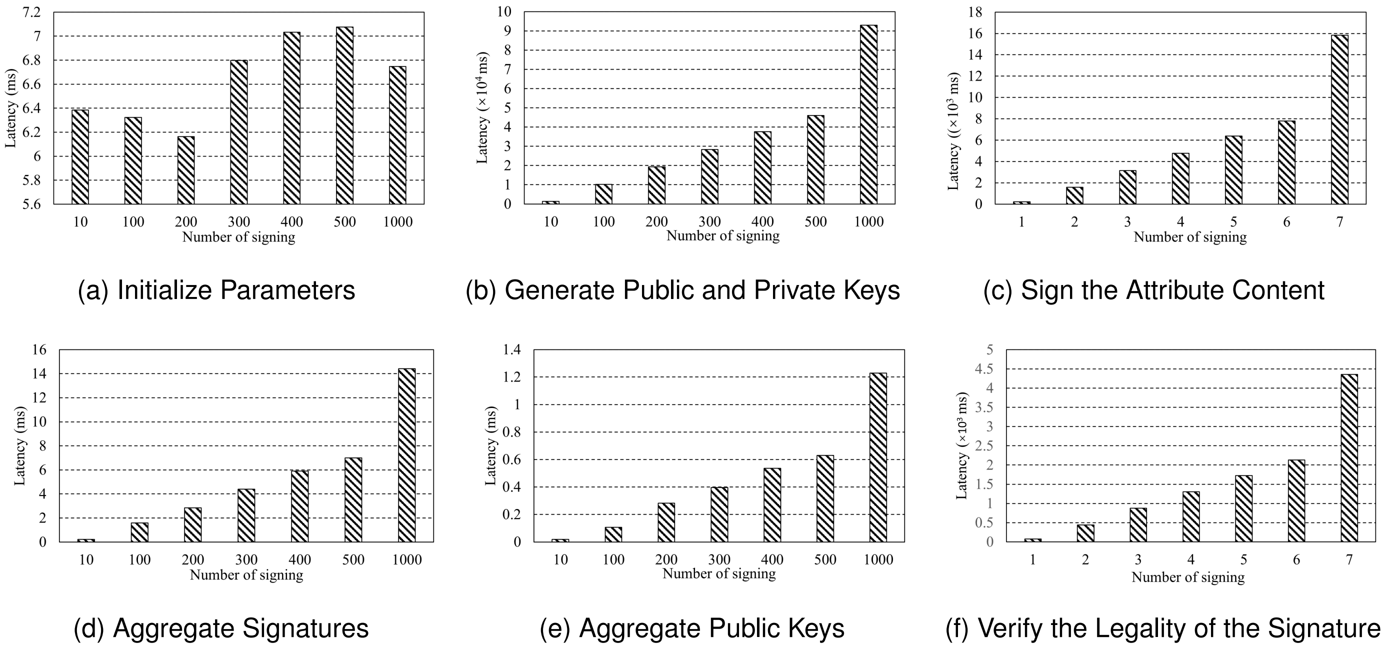
VI. Experiment and the Results

A. Experiment Configuration

The principle of our experiment configuration was to simulate the DID Aggregation algorithm in proposed AVCA-hDID mode. The program running environment of our experiment was a host with Intel Core i5-1035G1 CPU, 32.0-GB memory, 1.0-TB hard disk. Windows 11 64-bit operating system has been deployed on it. The programming languages used is Python. We implemented the DID aggregation and verification algorithms in the proposed AVCA-hDID model using the Charm library [1]. Specifically, we utilized the MNT224 bilinear group in Charm and employed the library's group multiplication, exponentiation, and bilinear pairing operations to construct the algorithms. We randomly generated a sample dataset containing 1,000 DIDs, each consisting of a randomly generated public key and a message to be signed. Each algorithm involved in the comparison signed and verified these identities.

The experiment setups focused on evaluating aggregate signature algorithms in efficiency, resource consumption, and scalability in order to examine performance of the proposed scheme in various IoT environments. The efficiency comparison consists of two settings, which include *Average Signature Time* (AST) and *Average Verification Time* (AVT). In AST, algorithms such as AVCA-hDID, RSA, BLS, Merkle-tree, and ECDMA were evaluated, measuring the time taken to generate signatures for the aggregation of 1,000 signatures. Similarly, AVT assesses the verification time for the same algorithms using 1,000 signatures. The resource consumption comparison analyzes *Memory Usage* (MU) and *CPU Utilization* (CU). MU measures the memory consumed by each algorithm during the generation of 1,000 signatures, while CU evaluates CPU usage as a percentage during execution. Finally, the scalability comparison investigates performance metrics across varying input sizes (10, 100, 200, 300, 400, 500, 1000) for *Total Signature Time* (TST), *Total Verification Time* (TVT) and latency of each sub-algorithm in AVCA-hDID, measuring the time taken to generate and verify signatures as well as the time required for each part of the algorithm, respectively. This

[1] https://github.com/JHUISI/charm



(a) Initialize Parameters

(b) Generate Public and Private Keys

(c) Sign the Attribute Content

(d) Aggregate Signatures

(e) Aggregate Public Keys

(f) Verify the Legality of the Signature

Fig. 6: Latency for AVCA-hDID sub-algorithm.

structured approach ensures robust and meaningful results, facilitating a direct comparison of the algorithms' performance characteristics and their suitability for real-world applications.

B. Experiment Results and Main Findings

Under Experiment Setting 1, the AST and AVT evaluation metrics in Fig. 3 showed the time consumption of AVCA-hDID compared to the other four alternative algorithms (SD-BLS [12], AGS-MR(with RSA) [13], ECDSA [14], Merkle tree) during the aggregation and verification of signatures. This assessed the performance of AVCA-hDID. Considering the process of H-DID identity aggregation, as it involves high-frequency identity verification requirements in large-scale IoT, we quantify the computational latency of batch operations for H-DID signature verification through AST and AVT, thereby evaluating the verification efficiency of different aggregate signature schemes in large-scale DID identity authentication. In addition, for the scalability requirements of H-DID verification in IoT, MU and CU jointly evaluate the deployment feasibility of different aggregate signature schemes on resource-constrained DID nodes.

We implemented Merkle-tree aggregation signature using the hash-generic implementation of Merkle-tree from the Bellman-Bignat library[2]. Additionally, we utilized the hash-generic implementation of the RSA accumulator in the same library to achieve AGS-MR(with RSA) [13]. The Merle-tree aggregation signatures were constructed using a hierarchical structure. We employed a bottom-up approach with the Pedersen hash function to build the tree. For AGS-MR(with RSA), we utilized an RSA accumulator and applied modular exponentiation to implement the signing and verification algorithms. Additionally, we replicated SD-BLS [12] using the open-source project BLS Signature Aggregation[3]. This implementation utilized the BLS12-381 elliptic curve based on bilinear mappings, the Pedersen hash function, and point addition operations on the elliptic curve. Furthermore, we replicated the ECDSA aggregation signature scheme [14] by leveraging the secp256k1 curve from the btcec package [4] to construct public keys and one-time keys. Our implementation combined point multiplication and addition operations on this elliptic curve to achieve aggregation.

In Experiment Settings 1-1 and 1-2, the results consistently demonstrated that AVCA-hDID had the least computational time consumption, with a significant lead. The average aggregation signature time was better than AGS-MR by 46.7%, SD-BLS by 33.3%, ECDSA by 27.3%, and Merkle-tree by 42.9%. The average signature verification time showed advantages of 49.7% over AGS-MR, 44.4% over SD-BLS, 32.1% over ECDSA, and 46.4% over Merkle-tree, respectively.

Under Experiment Setting 2, the computational resource consumption values of AVCA-hDID were compared with the other four algorithms. The MU and CU indicators in Fig. 3 provided specific comparative results for Experiment Settings 2-1 and 2-2. Compared to RSA, BLS, ECDSA, and Merkle-tree, the memory consumption of AVCA-hDID was reduced by an average of 33.3%, 11.7%, 28.6%, and 37.5%, respectively. The CPU utilization percentage was reduced by an average of 37.5%, 16.7%, 27.5%, and 44.3%. The experiment results in Fig. 3 indicated that the method proposed in this work had advantages in both execution time performance and computational resource consumption.

[2] https://github.com/alex-ozdemir/bellman-bignat

[3] https://github.com/gazman-sdk/BLS_signature_aggregation

[4] https://github.com/btcsuite/btcd/btcec

Under Experiment Settings 3-1 and 3-2, Fig. 4 and Fig. 5 show the changes in total aggregation signature time and verification signature time as the number of aggregated signatures increased for each comparison algorithm. The results indicated that as the number of aggregated signatures increased, the computational workload grew, leading to a generally linear increase in both aggregation signature time and verification signature time. However, our solution consistently achieved the shortest time across all numbers of aggregated signatures, with the time advantage growing larger as the number of signatures increased. Therefore, AVCA-hDID exhibits better scalability with increasing signature counts. In IoT scenarios, where the number of connected devices is high, better scalability is beneficial for the aggregation of device DIDs. Although our time also increased linearly with the number of signatures, the latencys are acceptable in IoT contexts.

Under Experiment Setting 3-3, Fig. 6 shows the latency of AVCA-hDID sub-algorithms during the DID aggregation and verification processes. The experimental setup simulates different signature aggregation and verification scenarios to assess the performance of AVCA-hDID when handling a large number of DID identities. Fig. 6 shows the latency for each sub-algorithm as it processes aggregated signatures, including the initialization phase, public/private key generation, signing, signature aggregation, public key aggregation, and signature verification. The experiment results in the Fig. 6 shows that as the number of aggregated signatures increases, the latency for each phase grows linearly. This indicates that although the AVCA-hDID system must handle more signatures and identity information during large-scale identity verification, the overall computational latency remains within acceptable limits, especially in environments with many IoT devices. Furthermore, AVCA-hDID also excels in optimizing resource usage, maintaining high efficiency even in resource-constrained environments. By minimizing unnecessary computations and optimizing signature verification, AVCA-hDID offers an efficient and scalable solution for identity verification in IoT applications.

Based on the above analysis, the proposed AVCA-hDID consumed the least computational time and resources compared to the other algorithms. In the scalability evaluation, it also achieved the best scalability relative to the comparison algorithms. Its execution efficiency was acceptable in systems with a large number of IoT devices.

VII. Related Work

A. Decentralized Identity

DID [15]–[17] permits every individual to establish their unique Identifiers within the blockchain ecosystem. Identity identifiers are securely encapsulated within the blockchain infrastructure, subject to verification and maintenance by a multitude of network nodes. The adoption of DID authentication methodologies [18], [19] ushers in a paradigm shift, extricating users from the dependency on conventional, centralized identity governance bodies. Instead, individuals are empowered to authenticate their identities leveraging their self-sovereign identifiers, thereby fostering a heightened degree of decentralization and autonomy in the identity verification process. The most common use case for DIDs involves user authorization to transfer personal credentials from their devices to websites [20]. The DID Foundation [21] and the W3C's [22] DID work group are actively creating standards and use cases to enable these types of transactions. On the platform side of distributed digital identity, Candid [23] proposed a solution aimed at realizing DID in a practical and user-friendly manner, focusing on empowering users to manage their own credentials. For distributed digital identity in complex and heterogeneous networks, DePTVM [24] proposed a decentralized pseudonym and trust value management scheme for integrated heterogeneous networks. Nereus [25] Integrated smart contracts into *Software Guard Extensions* (SGX) enclaves further expands the scope of DID applications.

B. Identity Aggregation

Aggregate Signature which is initially introduced by Boneh *et al.* [26], enables compressing signatures on distinct messages into a short aggregate signature. To alleviate this issue, Goyal *et al.* [27] pioneered the concept of locally verifiable aggregate signatures. This advanced scheme enables verifiers to validate the authenticity of individual messages within the aggregate without necessitating access to the entire message collection. Signature aggregation can be achieved through two primary methods: general aggregation [28] and sequential aggregation [29]. Many signature schemes are well-known for their signature aggregation techniques, such as Schnorr signatures [30], lattice-based signatures [31], and pairing-based signatures [32]. When it comes to pairing-based signatures, we must highlight the BLS scheme [33], which has been practically used in blockchain projects, such as Dfinity and Algorand.

TFS-ABS [34] combines traceability and anonymity with constant-size signatures, addressing key revocation issues in dynamic IoT networks. Unlike our blockchain-based anonymous VC, TFS-ABS relies on a trusted authority for traceability, which may conflict with decentralization. OABS [35] achieves server-aided verification with constant-size signatures, optimizing for resource-constrained devices. However, such schemes lacks considerations for heterogeneous identity aggregation.

C. Attribute-based Anonymous Certificates

Anonymous credentials were initially introduced by Chaum [36] with the primary aim of enabling users to interact anonymously with organizations, allowing them to subsequently present their credentials to various service providers (verifiers) without revealing their identity. This concept evolved into attribute-based anonymous credential schemes, which facilitated users declaring their attributes to verifiers while preserving anonymity. Such schemes found applications in privacy-centric areas like direct anonymous authentication [37] and anonymous electronic identifiers [38]. It wasn't until Brands [39] introduced a model for single-attribute anonymous credentials, followed by Kampanakis and Lysyanskaya's [40] proposal for multi-attribute versions, that anonymous credentials garnered significant attention. Following these developments, numerous schemes for anonymous credentials have been proposed to enhance their performance. Diverse signature schemes, including CL signatures [40]–[42], malleable signatures [43], structure-preserving signatures [44], and editable signatures [45], have been employed to construct various anonymous credential schemes tailored to different attribute requirements. A common feature among most of these schemes is their design as unlinkable and non-transferable, ensuring that verifiers cannot link interactions involving the same credential to a single user, and simultaneously preventing users from fabricating false credentials or misappropriating others' credentials for unauthorized use.

Existing identity authentication schemes primarily rely on cryptographic techniques, such as anonymous credentials [10], [28], [42], [46], to ensure secure identity verification. Hesse *et al.* [42] employ anonymous credentials to achieve privacy-preserving digital identity authentication, extending the aggregate signature mechanism within Self-sovereign Identity (SSI) frameworks. Doerner *et al.* [46] leverage the BBS+ signature scheme for distributed digital certificate issuance, while Hebant *et al.* [10] apply attribute-based encryption for traceable certificates, supporting multi-party authentication. In addition to cryptographic signatures, Secure Multi-party Computation (SMPC) is frequently used to establish distributed trust across multiple parties in DID authentication. For instance, Tan *et al.* [47] introduce MPCAuth, an efficient protocol proposed for establishing TLS connections within SMPC, enabling multi-factor authentication. However, such identity authentication schemes do not consider the heterogeneous identity aggregation.

VIII. Conclusion

In this work we propose an anonymous VC-based aggregation scheme for heterogeneous DID, which aimed at solving trust and privacy issues of identity authentication in IoT. The proposed scheme can successfully support anonymous ownership verification of DIDs through label randomization, which is theoretically demonstrated in this work. Our evaluations have also evidenced the efficiency and robustness of the proposed scheme.

Practical deployment for AVCA-hDID in large-scale IoT environments faces significant challenges, including blockchain scalability under high-frequency device onboarding, computational constraints of resource-limited IoT devices and interoperability gaps across heterogeneous DID issuers. Addressing these through layer-2 solutions, hardware acceleration and W3C DID compliance constitutes critical future work.

References

[1] Bei Gong, Guiping Zheng, Muhammad Waqas, Shanshan Tu, and Sheng Chen. LCDMA: Lightweight cross-domain mutual identity authentication scheme for internet of things. *IEEE Internet of Things Journal*, 10(14):12590–12602, 2023.

[2] Xuefei Yin, Song Wang, Muhammad Shahzad, and Jiankun Hu. An iot-oriented privacy-preserving fingerprint authentication system. *IEEE Internet of Things Journal*, 9(14):11760–11771, 2021.

[3] Keke Gai, Yulu Wu, Liehuang Zhu, Meikang Qiu, and Meng Shen. Privacy-preserving energy trading using consortium blockchain in smart grid. *IEEE Transactions on Industrial Informatics*, 15(6):3548–3558, 2019.

[4] Keke Gai, Yulu Wu, Liehuang Zhu, Lei Xu, and Yan Zhang. Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks. *IEEE Internet of Things Journal*, 6(5):7992–8004, 2019.

[5] Pawel Szalachowski. Password-authenticated decentralized identities. *IEEE Transactions on Information Forensics and Security*, 16:4801–4810, 2021.

[6] Sandro Rodriguez Garzon, Hakan Yildiz, and Axel Küpper. Decentralized identifiers and self-sovereign identity in 6g. *IEEE Network*, 36(4):142–148, 2022.

[7] Zijian Bao, Debiao He, Muhammad Khurram Khan, Min Luo, and Qi Xie. Pbidm: Privacy-preserving blockchain-based identity management system for industrial internet of things. *IEEE Transactions on Industrial Informatics*, 19(2):1524–1534, 2022.

[8] Chaosheng Feng, Bin Liu, Keping Yu, Sotirios K Goudos, and Shaohua Wan. Blockchain-empowered decentralized horizontal federated learning for 5g-enabled uavs. *IEEE Transactions on Industrial Informatics*, 18(5):3582–3592, 2021.

[9] Siwon Huh, Myungkyu Shim, Jihwan Lee, Simon S Woo, Hyoungshick Kim, and Hojoon Lee. Did we miss anything?: Towards privacy-preserving decentralized id architecture. *IEEE Transactions on Dependable and Secure Computing*, 20(6):4881–4898, 2023.

[10] Chloé Hébant and David Pointcheval. Traceable constant-size multi-authority credentials. *Information and Computation*, 293:105060, 2023.

[11] Chloé Hébant, Duong Hieu Phan, and David Pointcheval. Linearly-homomorphic signatures and scalable mix-nets. In *Public-Key Cryptography–PKC 2020: 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4–7, 2020, Proceedings, Part II 23*, pages 597–627. Springer, 2020.

[12] Denis Roio, Rebecca Selvaggini, Gabriele Bellini, and Andrea Dintino. Sd-bls: Privacy preserving selective disclosure of verifiable credentials with unlinkable threshold revocation. In *2024 IEEE International Conference on Blockchain (Blockchain)*, pages 505–511, 2024.

[13] Khaled Chait, Abdelkader Laouid, Mostefa Kara, Mohammad Hammoudeh, Omar Aldabbas, and Abdullah T Al-Essa. An enhanced threshold rsa-based aggregate signature scheme to reduce blockchain size. *IEEE Access*, 11:110490–110501, 2023.

[14] Jay Bojič Burgos and Matevž Pustišek. Decentralized iot data authentication with signature aggregation. *Sensors*, 24(3):1037, 2024.

[15] K. Wang, J. Gao, Q. Wang, J. Zhang, Y. Li, Z. Guan, and Z. Chen. Hades: Practical decentralized identity with full accountability and fine-grained sybil-resistance. In *Proceedings of the 39th Annual Computer Security Applications Conference*, pages 216–228, 2023.

[16] S. Zheng, Z. Li, J. Luo, Z. Xin, and X. Liu. Idea-dac: Integrity-driven editing for accountable decentralized anonymous credentials via zk-json. *Cryptology ePrint Archive*, 2024.

[17] Jiguo Li, Yu Chen, Jinguang Han, Chengdong Liu, Yichen Zhang, and Huaqun Wang. Decentralized attribute-based server-aid signature in the internet of things. *IEEE Internet of Things Journal*, 9(6):4573–4583, 2021.

[18] J. Chen, F. Lu, Y. Liu, S. Peng, Z. Cai, and F. Mo. Cross trust: A decentralized ma-abe mechanism for cross-border identity authentication. *International Journal of Critical Infrastructure Protection*, 44:100661, 2024.

[19] H.-H. Ou, C.-H. Pan, Y.-M. Tseng, and I.-C. Lin. Decentralized identity authentication mechanism: Integrating fido and blockchain for enhanced security. *Applied Sciences*, 14:3551, 2024.

[20] I.T. Javed, F. Alharbi, B. Bellaj, T. Margaria, N. Crespi, and K.N. Qureshi. Health-id: A blockchain-based decentralized identity management for remote healthcare. In *Healthcare*, volume 9, page 712. MDPI, 2021.

[21] Decentralized Identity Foundation. https://identity.foundation/, 2020.

[22] W3C. Decentralized identifiers (dids) v0.11: Data model and syntaxes for decentralized identifiers. Technical report, 2018.

[23] M. Deepak, M. Harjasleen, F. Zhang, N. Jean-Louis, F. Alexander, et al. Candid: Can-do decentralized identity with legacy compatibility, sybil-resistance, and accountability. In *2021 IEEE SP*, pages 1348–1366. IEEE, 2021.

[24] G. Liu, Z. Yan, D. Wang, H. Wang, and T. Li. Deptvm: Decentralized pseudonym and trust value management for integrated networks. *IEEE Transactions on Dependable and Secure Computing*, 2023.

[25] M. Li, Y. Chen, C. Lal, M. Conti, F. Martinelli, and M. Alazab. Nereus: Anonymous and secure ride-hailing service based on private smart

contracts. *IEEE Transactions on Dependable and Secure Computing*, 2022.

[26] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In Eli Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, pages 416–432, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[27] R. Goyal and V. Vaikuntanathan. Locally verifiable signature and key aggregation. In *Annual International Cryptology Conference*, pages 761–791. Springer, 2022.

[28] Omid Mir, Balthazar Bauer, Scott Griffy, Anna Lysyanskaya, and Daniel Slamanig. Aggregate signatures with versatile randomization and issuer-hiding multi-authority anonymous credentials. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 30–44, New York, NY, USA, 2023.

[29] T. Li, H. Wang, D. He, and J. Yu. Designated-verifier aggregate signature scheme with sensitive data privacy protection for permissioned blockchain-assisted iiot. *IEEE Transactions on Information Forensics and Security*, 2023.

[30] G. Fuchsbauer, A. Plouviez, and Y. Seurin. Blind schnorr signatures and signed elgamal encryption in the algebraic group model. In *Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part II 30*, pages 63–95. Springer, 2020.

[31] W.-K. Lee, R. K. Zhao, R. Steinfeld, A. Sakzad, and S. O. Hwang. High throughput lattice-based signatures on gpus: Comparing falcon and mitaka. *IEEE Transactions on Parallel and Distributed Systems*, 35:675–692, 2024.

[32] M. Brodsky, A. Choudhuri, A. Jain, and O. Paneth. Monotone-policy aggregate signatures. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 168–195. Springer, 2024.

[33] D. Boneh, M. Drijvers, and G. Neven. Bls multi-signatures with public-key aggregation. *URL: https://crypto. stanford. edu/˜ dabo/pubs/papers/BLSmultisig. html*, 2018.

[34] Zhaozhe Kang, Jiguo Li, Jian Shen, Jinguang Han, Yuting Zuo, and Yichen Zhang. Tfs-abs: Traceable and forward-secure attribute-based signature scheme with constant-size. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):9514–9530, 2023.

[35] Zhaozhe Kang, Jiguo Li, Yuting Zuo, Yichen Zhang, and Jinguang Han. Oabs: Efficient outsourced attribute-based signature scheme with constant-size. *IEEE Internet of Things Journal*, 2024.

[36] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[37] Ernie Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 132–145, 2004.

[38] Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 21–30, 2002.

[39] Stefan Brands. *Rethinking public key infrastructures and digital certificates: building in privacy*. Mit Press, 2000.

[40] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Advances in Cryptology—EUROCRYPT 2001: International Conference on the Theory and Application of Cryptographic Techniques Innsbruck, Austria, May 6–10, 2001 Proceedings 20*, pages 93–118. Springer, 2001.

[41] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Advances in Cryptology—CRYPTO 2002: 22nd Annual International Cryptology Conference Santa Barbara, California, USA, August 18–22, 2002 Proceedings 22*, pages 61–76. Springer, 2002.

[42] Julia Hesse, Nitin Singh, and Alessandro Sorniotti. How to bind anonymous credentials to humans. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 3047–3064, 2023.

[43] Sven Schäge. New limits of provable security and applications to elgamal encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 255–285. Springer, 2024.

[44] Renas Bacho, Julian Loss, Stefano Tessaro, Benedikt Wagner, and Chenzhi Zhu. Twinkle: Threshold signatures from ddh with full adaptive security. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 429–459. Springer, 2024.

[45] Zonglun Li, Shuhao Zheng, Junliang Luo, Ziyue Xin, and Xue Liu. Ideadac: Integrity-driven editing for accountable decentralized anonymous credentials. In *The Web Conference 2024*.

[46] Jack Doerner, Yashvanth Kondi, Eysa Lee, Abhi Shelat, and LaKyah Tyner. Threshold bbs+ signatures for distributed anonymous credential issuance. In *IEEE SP '23*, pages 773–789. IEEE, 2023.

[47] Sijun Tan, Weikeng Chen, Ryan Deng, and Raluca Ada Popa. Mpcauth: multi-factor authentication for distributed-trust systems. In *IEEE SP '23*, pages 829–847. IEEE, 2023.