A Secure Authentication Framework to Guarantee the Traceability of Avatars in Metaverse

Kedi Yang, Zhenyong Zhang, Member, IEEE, Youliang Tian, Member, IEEE, and Jianfeng Ma, Member, IEEE

Abstract—Metaverse is a vast virtual environment parallel to the physical world in which users enjoy a variety of services acting as an avatar. To build a secure living habitat, it’s vital to ensure the virtual-physical traceability that tracking a malicious player in the physical world via his avatars in virtual space. In this paper, we propose a two-factor authentication framework based on biometric-based authentication and chameleon signature. First, aiming at disguise in virtual space, we design an avatar’s two-factor identity model to ensure the verifiability of avatar’s virtual identity and physical identity. Second, facing at authentication efficiency and keys holding cost, we propose a chameleon collision signature algorithm to efficiently ensure that the avatar’s virtual identity is associated with its physical identity. Finally, aiming at impersonation in the physical world, we design two decentralized authentication protocols based on the avatar’s identity model and the chameleon collision signature to achieve real-time authentication on the avatar’s identity. Security analysis indicates that the proposed authentication framework guarantees the consistency and traceability of the avatar’s identity. Simulation experiments show that the framework not only completes the decentralized authentication between avatars but also achieves virtual-physical tracking.

Index Terms—Metaverse, avatar, authentication, traceability.

I. INTRODUCTION

METAVERSE, a combination of the prefix “meta” implying transcending with the stem “verse” of the universe [1], means a new type of Internet application and social form beyond the physical world. The word first appeared in the science fiction novel, Snow Crash, written by Neal Stephenson in 1992, which described a vast virtual environment parallel to the physical world, where people communicate and work through digital avatars.

The most representative prototype of the metaverse is the virtual platform Second Life released by Linden Research in 2003. The platform constructs a virtual world that is highly similar to reality in which players can freely socialize, trade, and build facilities via their digital role. IBM once purchased a piece of land and built its own sales center in the complete virtual ecosystem. As a highly digitized world, the metaverse is devoted to building an environment that satisfies immersive interaction and virtual-physical coexistence [2], which breaks through the limitations of the physical world and enables people to perform unimaginable works. For example, in the medical field, the digital representation of a patient is projected into a virtual consultation room, in which experts from around the world discuss treatment options face-to-face without leaving their offices. In the social field, friends in different places are projected into the same virtual environment allowing them to communicate and shop like in the physical world.

Metaverse will become the second living space of human beings’ coexistence with the physical world [3]. At present, leading companies in various countries have turned their attention to the metaverse [4], [5], where the dimension is infinite but the ecosystem is finite. Baidu, the largest Internet company in China, has published an immersive interactive environment, Xirang,2 based on Virtual Reality (VR) and Artificial Intelligence (AI). The environment consists of infinitely connected virtual spaces, each of which is a unique digital metropolis. MATE (formerly Facebook), the most popular social platform in America, has released an open VR social environment Horizon Worlds that enables each player to create his own community and meet strangers from all over the world for leisure and entertainment. However, something as disturbing as the real world is happening in this emerging environment.

---

1https://secondlife.com
2https://vr.baidu.com/product/xirang
In the public beta of Horizon Worlds, a female tester reported that her avatar was sexually harassed by other players. Soon after, it was reported that a researcher from SumOfUs had suffered similar harassment in the virtual cyberspace, which was worse than the previous. It can be seen from the above events that the avatar’s safety is being threatened, which seriously hinders the further development of metaverse. Therefore, there is an urgent need to establish a traceable authentication mechanism that tracks a malicious avatar to its physical manipulator.

**A. Tracking Methods**

Avatar is the virtual representation of a physical manipulator in the metaverse, indicating that an avatar is the unity of a virtual identity and a physical identity. For this reason, the consistency of virtual and physical identities is key to guaranteeing virtual-physical traceability. In the current metaverse application, players are able to create any avatar as their virtual representation, which is not related to the player’s physical identity. This defect enables malicious players to create a similar avatar and launch a disguised attack. During the manipulator tracking process, the server provider mainly utilizes the user’s (account, password) as the core of the avatar’s virtual identity for tracing. Using password-based methods, however, any player knowing the account password can complete login, which enables the malicious player to log in and manipulate a legitimate player’s avatar by stealing the password in some way. Ultimately, the malicious avatar cheating in the metaverse cannot be traced back to its real physical manipulator via the (account, password). Therefore, if an avatar’s identity model can be constructed such that both its virtual identity and physical identity can be verified, virtual-physical tracking can be well achieved.

Some network applications also utilize the device’s IP [9] and MAC [10] as physical identity for authentication and tracing. These methods can realize the traceability of virtual avatars to physical players, but the traced physical identity is potentially inconsistent with the actual manipulator. In IP-based methods, the system combines various factors such as login IP, time, and location to detect abnormal logins to ensure the legitimacy of the manipulator’s physical identity when logging in. However, identity verification is no longer performed in players’ interactions, as a result, the malicious manipulator is able to impersonate the corresponding player via authenticated devices. Ultimately, it cannot trace back to the real physical manipulator through IP. Therefore, if a real-time authentication method can be constructed based on biometrics, it can guarantee the consistency of the avatar’s virtual and physical identities, which supports virtual-physical tracking.

The current metaverse authentication only focuses on mutual authentication when the avatars meet, while ignores the real-time authentication during the avatar’s interaction. Scheme [11] uses a central platform as an intermediary to forward authentication information during avatars’ mutual authentications. If the intermediary is adopted in real-time authentication, the consumption will be unimaginable. Moreover, if the central platform is involved in all authentication processes of the avatar’s life and works in the metaverse, the platform may grasp all the details of the user’s life in the physical world, which will seriously threaten user’s privacy. Therefore, if a decentralized real-time authentication method can be constructed, it can not only reduce platform overhead but also protect users’ privacy.

**B. Main Challenges**

Constructing such an authentication mechanism is non-trivial: (i) it is impossible to directly use biometrics as an identity factor to build the avatar’s identity model because biometrics are vulnerable to replay attack; (ii) the traditional signature algorithm cannot be adopted to verify the avatar’s virtual and physical identities, because the traditional signature is difficult to indicate the internal connection between two signed messages; (iii) it is difficult to complete avatar mutual authentication without a central platform because there is a lack of intermediary to forward trusted parameters.

In this paper, we aim to address the above challenges. For the first challenge, in traditional biometric-based authentication methods [12], the verifier throws a random challenge to the avatar, while the avatar as a prover submits its manipulator’s biological sample as a response. Based on these physical identity parameters, the verifier can check the avatar’s physical identity to guarantee the traceability of the physical manipulator. However, the biometric feature is difficult to avoid replay attack [13] and thus a malicious player may deny that the violation wasn’t performed by him. To solve this problem, we build an avatar’s two-factor identity model based on the player’s biometric template and signature public key, ensuring the verifiability of the avatar’s virtual identity and physical identity while avoiding biometrics replay.

For the second challenge, the traditional signature algorithm [14], [15] can verify the legitimacy of a single signature message, but it cannot indicate the inner connection between two signature messages. Therefore, it is impossible to prove the consistency between the avatar’s virtual identity and the physical identity. Chameleon signature is a one-to-many signature mechanism that signs multiple plaintexts with one signed hash. Based on this feature, the players are able to sign multiple identity parameters, which remains associated with the core identity. Regrettably, the traditional chameleon signatures [16] are cumbersome and inefficient. It consists of two parts, the chameleon hash and the common signature, leading to the players having to hold multiple key pairs. To this end, we improve the chameleon hash function [17] to propose a chameleon collision signature algorithm, which enables players to sign avatars with only a pair of keys. Based on this algorithm, it ensures that the avatar’s virtual identity is associated with its physical identity as well as improves authentication efficiency.

For the third challenge, we assume that there is an authentication mechanism that allows avatars to complete dynamic identity verification without involving a trusted third party,
which achieves the continuous authentication for the avatar’s identity in a decentralized manner. However, the current continuous authentication methods are impacted by scenarios [18] and the methods affect the player’s immersive experience [19]. To this end, we designed two immersive decentralized authentication protocols based on the avatar’s identity model and the chameleon collision signature, including the one-party and the two-party authentication protocol. Before avatar’s interaction, the two-party authentication protocol completes the decentralized mutual authentication. During the interaction, the one-party protocol completes the continuous authentication. These protocols guarantee the consistency of avatars’ virtual and physical identities in real-time and enable virtual-physical tracking.

C. Contributions

To sum up, we construct a decentralized and traceable authentication framework for avatars based on chameleon signatures and biometrics. The contributions are as follows:

- We introduce the notion of virtual-physical traceability for avatar authentication systems and define the security requirement of consistency for avatars’ virtual and physical identities.
- We construct an avatar’s two-factor identity model based on the biometric template and the chameleon public key to realize the verifiability of the avatar’s virtual identity and its physical identity.
- We propose a chameleon collision signature algorithm to effectively generate colliding signatures by only one private key, which ensures that the avatar’s virtual identity is associated with its physical identity.
- We design two sets of decentralized avatar authentication protocols based on the avatar’s identity model and chameleon collision signature to guarantee the consistency of avatar’s virtual and physical identities in real-time.
- We build an avatar authentication system based on blockchain and iris recognition method, which achieves virtual-physical tracking.

The rest of the paper is organized as follows. We introduce the preliminaries in the next section. Section III introduces the secure authentication framework. The consistent avatar’s identity is presented in Section IV. Section V designs the decentralized avatar authentication protocol. The security analysis and the performance evaluation are given in Section VI and Section VII, respectively. We review related works in Section VIII and finally conclude the paper in Section IX.

II. PRELIMINARIES

In this section, we recall the common signature and the traditional chameleon signature, then briefly introduce the design idea of the proposed chameleon signature.

A. Common Signature

Common signature algorithm consists of four modules: Setup, KeyGen, Sign, and Verify. In the signing phase, the user signs a message \( M \) based on the private key \( sk \) to generate a signature \( \sigma \). In the validation phase, the other user verifies the pair of signature message \((M, \sigma)\) based on the corresponding public key \( pk \). This type of algorithm can verify the legitimacy of a single pair \((M_1, \sigma_1)\) and \((M_2, \sigma_2)\) respectively, but cannot indicate the inherent relationship between \( M_1 \) and \( M_2 \).

- \( \text{Setup}(\lambda) \rightarrow \text{Parm} \). The input of this probabilistic algorithm is the security parameter \( \lambda \) and the output is the system parameter \( \text{Parm} \).
- \( \text{KeyGen}(\text{Parm}) \rightarrow (pk, sk) \). The key generation algorithm takes the system parameter \( \text{Parm} \) as input. It outputs the public-private key pair \((pk, sk)\).
- \( \text{Sign}(sk, M) \rightarrow \sigma \). The signature algorithm signs the hash of \( M \) based on the signature private key \( sk \), and outputs an ordinary signature \( \sigma \).
- \( \text{Verify}(pk, M, \sigma) \rightarrow b \in \{0, 1\} \). The algorithm verifies the legitimacy of the ordinary signature \( \sigma \) based on \((pk, M, \sigma)\) and outputs the verification result \( b \).

B. Traditional Chameleon Signature

Chameleon signature algorithm is modifiable and its traditional schemes are composed of chameleon hash and ordinary signature. In the signing phase, the user first generates a chameleon hash \( h \) and its check parameter \( R \) for a message \( M \) based on the public key \( pk \), forming a chameleon triplet \((h, M, R)\) that can be modified; Then the user signs \( h \) based on the signature private key \( sk \), to generate a signature \( \sigma \). In validation phase, \((h, M, R, \sigma)\) is verified by \( pk \). If the user wants to modify the signature message \( M \) to \( M' \), the user utilizes chameleon hash \( sk \) to generate the check parameter \( R' \) for \( M' \), where \((h, M', R', \sigma)\) is also a legal signature message. Among them, \( h \) indicates that there is a inherent relationship between \( (M, R) \) and \((M', R')\), called collision pairs about \( h \). Since the private key of the traditional chameleon signature consists of two parts, the chameleon hash key \( sk_h \) and the ordinary signature key \( sk_s \), which poses a huge security challenge to key storage and management.

- \( \text{Setup}(\lambda) \rightarrow \text{Parm} \). The same as common signature.
- \( \text{KeyGen}(\text{Parm}) \rightarrow (sk, pk) \). The key generation algorithm takes \( \text{Parm} \) as input. It outputs \((pk, sk)\), where \( sk = \{sk_h, sk_s\} \).
- \( \text{Hash}(pk, M) \rightarrow (h, R) \). The hash generation algorithm takes as input \( pk \) and \( M \), it outputs the chameleon hash \( h \) and the check parameter \( R \) of \( M \).
- \( \text{Check}(pk, h, M, R) \rightarrow b \in \{0, 1\} \). The compatibility detection algorithm takes as input the chameleon triplet \((h, M, R)\) and \( pk \). It outputs a decision \( b \in \{0, 1\} \) indicating whether the \((pk, h, M, R)\) is compatible.
- \( \text{Forge}(sk_h, h, M, R, M') \rightarrow R' \). The forge algorithm takes as input the chameleon private key \( sk_h \), chameleon triplet \((h, M, R)\), and a new message \( M' \). It calculates the check parameter \( R' \) of \( M' \), such that \( \text{Check}(pk, h, M, R) = \text{Check}(pk, h, M', R') = 1 \). Among them, \((M, R)\) and \((M', R')\) are called collision pairs about \( h \).
- \( \text{Sign}(sk_s, h) \rightarrow \sigma \). The signature algorithm signs the chameleon hash \( h \) based on the signature private key \( sk_s \) and outputs the chameleon signature \( \sigma \).
C. The Proposed Chameleon Collision Signature

Since a chameleon collision is forged by the user’s private key, a new collision can be treated as a chameleon signature related to the old collision. Based on this idea, we propose an efficient chameleon collision signature, which reduces the key holding cost and signing overhead. We design $\text{Sign}$ based on $\text{Forge}$ and design $\text{Verify}$ based on $\text{Check}$. The proposed chameleon signature consists of the following six parts:

- $\text{Setup}(\lambda) \rightarrow \text{Parm}$. The same as common signature.
- $\text{KeyGen} (\text{Parm}) \rightarrow (pk, sk)$. The same as common signature.
- $\text{Hash}(pk, M) \rightarrow (h, R)$. The hash generation algorithm takes as input $pk$ and $M$, it outputs the chameleon hash $h$ and the check parameter $R$ of $M$.
- $\text{Check}(pk, h, M, R) \rightarrow b \in \{0, 1\}$. The compatibility detection algorithm takes as input the chameleon triplet $(h, M, R)$ and $pk$. It outputs a decision $b \in \{0, 1\}$ indicating whether the $(pk, h, M, R)$ is compatible.
- $\text{Sign}(sk, h, M, R, M') \rightarrow R'$. To sign a message $M'$, the signing algorithm takes as input $sk$ and $(h, M, R)$. It outputs the check parameter $R'$ of $M'$, where the pairs $(M, R)$ and $(M', R')$ are called a colliding signature about $h$.
- $\text{Verify}(pk, h, M, R, M', R') \rightarrow b$. To verify the colliding signature $(M, R)$ and $(M', R')$ about $h$, the algorithm checks the compatibility of $(h, M, R)$ and $(h, M', R')$ by $\text{Check}$. It outputs a decision $b \in \{0, 1\}$ indicating whether $(M, R)$ and $(M', R')$ is valid.

III. Secure Authentication Framework

To ensure the virtual-physical traceability, we require that players need to obtain a metaverse identity token (MIT) from a trusted identity provider (IDP) before entering the metaverse. Based on this token, players are able to construct their avatar with verifiable virtual and physical identities in the metaverse, which ensures virtual-physical traceability. For this purpose, we construct a secure authentication framework as shown in Fig.1, which is divided into two layers: the physical world layer and the metaverse layer. In the physical world, IDP examines the players’ real identities and grants the corresponding MIT. In the metaverse, players create their avatars based on the MIT and complete decentralized authentication based on the proposed authentication protocols, which ensures the consistency of the avatar’s virtual and physical identities. It is worth noting that the IDP does not participate in the avatars’ authentications in the metaverse. It is only responsible for granting MITs and tracking real identities in the physical world. Here, we introduce the system model, security threats, and design goals of the proposed authentication framework.

A. System Model

- **Avatar**: It is the virtual representation of a physical player. Based on this feature, we regard an avatar as the unity of virtual identity and physical identity, which means that a valid avatar must satisfy the consistency of virtual and physical identities.
- **Player**: It is a specific manipulator of an avatar in the physical world. To ensure traceability, players need to submit their chameleon key and biological samples, such as iris, to complete registration and obtain a two-factor MIT, which enables the player to create an avatar that satisfies the verifiability of virtual and physical identities.
- **Identity provider (IDP)**: As a trusted organization in the physical world, it runs a consortium blockchain to hold identity data and provides identity services for metaverse users. In the process of registration, it examines players’ real identities and grants the corresponding MIT. During the tracking process, it discloses the violator’s real identity through MIT and the related parameters submitted by a whistleblower.
- **Blockchain**: Due to the lack of a trusted third party to forward identity parameters in avatars’ mutual authentication, it is necessary to utilize the blockchain as an intermediary to host the auxiliary data required for avatar authentication. As a consortium blockchain, it stores public information about the avatars’ identity such as MITs, which are open to the whole world to support cross-domain authentication.

B. Security Threats

A malicious player may disguise his avatar as the one of target players to deceive others, may replay outdated information collected from interactions with the target player, and may even get a device authenticated by a legitimate player to manipulate the corresponding avatar. Thus, we assume that the malicious player attempted to mount the following attacks:

- **Disguise**: In the virtual metaverse, the attacker as a malicious avatar disguises his appearance that looks the same as the target avatar to deceive the interactor.
• Replay: Both in the metaverse and the physical world, the attacker collects the outdated identity parameters associated with an honest avatar and submits them to the interactor which claims to be the target avatar.
• Impersonation: In the physical world, the attacker as a malicious manipulator gets a device authenticated by a legitimate player and impersonates the player to manipulate the corresponding avatar.

C. Design Goals

To realize the secure interaction for avatars in the metaverse, the proposed framework should achieve the following goals.
• Consistency: To achieve virtual-physical traceability, the avatar’s virtual and physical identities must be consistent. Therefore, we should check the validity of the avatar’s virtual identity to prevent disguise, check the avatar’s physical identity to avoid impersonation, and check the freshness of the submitted identity parameters to prevent replays.
• Traceability: The traceability refers to tracking an avatar in the virtual metaverse to the corresponding manipulator in the physical world, we call it virtual-physical traceability. It requires that a malicious avatar should be traced back to its manipulator through the avatar’s identity parameters retained in the authentication process, while an honest player should not be framed by fake identity parameters about the suspect avatar.
• Decentralized: Authentication between avatars can be done without the involvement of a trusted third party.
• Immersive: During the authentication process, players do not need to perform specific operations, ensuring the immersive experience.
• Privacy: The player’s physical identity is not disclosed during authentication. At the same time, the interaction relationship of avatars will not be known by the service provider.

IV. VERIFIABLE AVATAR’S IDENTITY

In this section, we first design an avatar’s two-factor identity model based on user biometrics and signature parameters to ensure the verifiability of the avatar’s virtual and physical identities. Then, we propose a chameleon collision signature to ensure that the avatar’s virtual identity is associated with its physical identity. Here, we detail the avatar’s identity model and the chameleon collision signature.

A. Avatar’s Two-Factor Identity Model

An avatar is the unity of a virtual identity and a physical identity. We combine the player’s biometric feature and keys to construct an avatar’s two-factor identity model Avatar = \{MIT, VID, PID, parameters and corresponding meanings

| Symbol | Description |
|--------|-------------|
| MIT    | Metaverse identity token |
| SN     | The serial number of a MIT |
| pk     | Player’s chameleon public key |
| h      | Chameleon hash |
| T      | Biometric template generated from iris samples |
| R      | The check parameter of T |
| VID    | Avatar’s virtual identity |
| M_a    | The description of avatar’s virtual identity |
| R_a    | The check parameter of M_a |
| PID    | Avatar’s physical identity |
| M’_a   | The biometric feature of avatar’s physical identity |
| R’_a   | The check parameter of M’_a |

1) Metaverse Identity Token (MIT): MIT as a bridge between the virtual space and the physical world enables an avatar’s virtual identity associated with its physical identity. The player’s public key is an important factor to realize the verifiability of the avatar’s virtual identity and the biometric feature is an important factor to ensure the traceability of the avatar’s physical identity. Combining the public key with a biometric feature such as iris, face, and fingerprint can construct a two-factor identity token to support virtual-physical tracking. Since the metaverse platform presents a vivid digital environment via a head-mounted display (HMD), which completely cover the player’s eye area to ensure a perfect visual experience, the iris will be an important biometric feature to achieve metaverse authentication.

Based on the above ideas, we construct a metaverse identity token based on the iris feature and public key to ensure the verifiability of the avatar’s virtual and physical identities. To obtain this token, the player provides ID, T, and pk to IDP, then the IDP generates the corresponding MIT = (SN, pk, h, T, R) and signs it after examining the above information. Among them, the ID is the player’s real identity, T is the biometric template generated from iris samples, SN is the serial number of MIT, pk is the chameleon public key taken from the key pair \((pk, sk) \leftarrow \text{KeyGen}()\), h is the chameleon hash taken from \((h, R) \leftarrow \text{Hash}(pk, M)\), and R is the check parameter of T. Ultimately, the IDP publishes MIT on the blockchain to ensure the public verifiability of the avatar’s identity and records (ID, SN) in a secure database to guarantee the traceability of the player’s real identity.

2) Virtual Identity (VID): VID is the player’s public virtual identity in a metaverse region, which appears as a visible avatar. Before entering the metaverse, players need to create an avatar’s VID = (M_a, R_a) based on the chameleon collision \((T, R) \subseteq MIT\) and his private key sk, where
$M_a$ is the description of the avatar’s virtual identity, $R_a \leftarrow \text{Sign}(sk, h, T, R, M_a)$ is the check parameter of $M_a$.

3) Physical Identity (PID): PID is the player’s biological identity in the physical world, which presents as a processed biometric feature. During metaverse interaction, the avatar provides its $PID = (M_a, R_a)$ to the verifier who checks the validity of physical identity through PID and MIT, where $M_a$ is the player’s biometric feature, $R_a \leftarrow \text{Sign}(sk, h, T, R, M_a')$ is the check parameter of $M_a'$.

B. Chameleon Collision Signature

The methods of traditional chameleon signature involve signing on a chameleon hash and forging collisions, resulting in the signer having to hold two private keys, one for signing and the other for generating collisions. In addition, the traditional methods [17], [20] introduce zero-knowledge proofs to ensure the security of chameleon hash, which seriously reduces the efficiency of the algorithm. Since a collision is forged by the signer’s private key, the new collision can be treated as a chameleon signature. Based on this idea, we propose an efficient chameleon collision signature by modifying the chameleon hash [17], which reduces the cost of holding keys and eliminates the zero-knowledge proofs while ensuring security. The proposed algorithm consists of the following six parts, namely, Setup, KeyGen, Hash, Check, Sign, and Verify.

- Setup($\lambda$) $\rightarrow$ Parm. Let $\lambda$ be a security parameter in the chameleon collision signature system. $G, GT$ are multiplicative cyclic groups of prime order $q \geq 2^k$, where $g$ is a generator of $G$. A pairing $\hat{e} : G \times G \rightarrow GT$ is an efficiently computable bilinear map, which satisfies $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$ for all $a, b \in \mathbb{Z}_q$. The system selects the global anti-collision hash function $H_G : \{0, 1\}^k \rightarrow G$, which maps bit strings of arbitrary length to corresponding elements in $G$. Finally, the algorithm publishes the system parameters $Param = \{G, GT, g, \hat{e}, H_G\}$.

- KeyGen($Param$) $\rightarrow$ $(pk, sk)$. The key generation algorithm takes the system parameter $Param$ as input. The algorithm picks a randomness $x \stackrel{R}{\leftarrow} \mathbb{Z}_q$ as the private key $sk$ and calculates $y = g^{1/x} \in G$ as the public key $pk$. It outputs the following public-private key pair

$sk = x, \quad pk = y$.

- Hash($pk, M$) $\rightarrow$ $(h, R)$. The chameleon hash generation algorithm takes as input the public key $pk = y$ and the message $M$. The algorithm sets $m = H_G(M)$ and picks a randomness $r \stackrel{R}{\leftarrow} \mathbb{Z}_q$. It outputs the chameleon hash $h$ and the corresponding check parameter $R$ as

$h = m \cdot y^r, \quad R = g^r$.

- Check($pk, h, M, R$) $\rightarrow$ $b$. The chameleon hash check algorithm takes as input the public key $pk = y$ and the chameleon triple $(h, M, R)$. The algorithm sets $m = H_G(M)$ and checks the compatibility of $(y, h, M, R)$. It outputs $b = 1$ if

$\hat{e}(h/m, g) = \hat{e}(R, y)$.

- Sign($sk, h, M, R'$) $\rightarrow$ $R'$. To sign a message $M'$, the chameleon sign algorithm takes as input the chameleon private key $sk = x$ and the chameleon triple $(h, M, R)$. The algorithm sets $m' = H_G(M')$ and outputs the check parameter $R'$ as

$R' = (h/m')^x$.

where the pairs $(M, R)$ and $(M', R')$ are called a colliding signature with respect to $h$.

- Verify($pk, h, M, R, R'$) $\rightarrow$ $b$. To verify the colliding signature $(M, R)$ and $(M', R')$ about $h$, the algorithm checks the compatibility of $(pk, h, M, R)$ and $(pk, h, M', R')$. It outputs $b = 1$ if

$Check(pk, h, M, R) = Check(pk, h, M', R') = 1$.

V. Decentralized Avatar Authentication Protocol

In this section, we design two sets of decentralized authentication protocols based on the avatar’s identity model, which are the one-party authentication protocol and the two-party authentication protocol. The feature of the two-party authentication protocol is to establish a session key, while the one-party authentication protocol is to guarantee the consistency of the avatar’s identity in real-time. When two avatars meet, they first execute the two-party protocol to complete decentralized mutual authentication and establish a session key, then each avatar periodically executes the one-party protocol to prove that his own physical identity is consistent with his own virtual identity in real-time.

A. One-Party Authentication Protocol

The one-party authentication protocol implements authentication based on a challenge-response mechanism. As shown in Fig.2: 1) avatar A as a prover claims that his identity is valid; 2) avatar B as a verifier checks the validity of A’s virtual identity and throws a random challenge to A to confirm whether A’s physical identity matches its virtual identity; 3) avatar A’s manipulator provides his biometric feature and corresponding check parameters as a response;
4) avatar B checks the validity of the parameters to determine the consistency of avatar’s virtual and physical identities. Details are as follows:

1) Claim: As shown in Fig.3(a), in phase 1, avatar A submits \( SN \in MIT \) and \( V ID = (M_a, R_a) \) to avatar B, which initiates an identity claim.

2) Challenge: In phase 2, avatar B throws a random challenge \( C_a \) to avatar A after B completes the check on \( MIT \) and \( V I D \). In the process of checking \( MIT \), B gets \( MIT = (SN, pk, h, T, R) \) from blockchain by \( SN \) and verifies IDP’s signature to ensure the validity of \( MIT \). During \( V I D \) verification, B checks the matching of collisions \( (M_a, R_a) = V ID \) and \( (T, R) \subset MIT \) by \( Verify(y, h, T, R, M_a, R_a) \) in IV-B. If both \( MIT \) and \( V I D \) are valid, A’s virtual identity is valid.

3) Response: In phase 3, avatar A’s manipulator submits his biometrics feature \( M'_a \) and the corresponding check parameter \( R'_a \) to prove the validity of A’s \( P ID \), where \( M'_a \) embedded with B’s challenge \( C_a \). The specific steps are as follows: (i) the manipulator randomly samples his biometric information and processes it locally to form a biometric feature \( M' \); (ii) the manipulator embeds \( C_a \) as watermarking into \( M' \) to form a watermarked biometric feature \( M'_a \), which provides conditions for anti-replay attacks; (iii) the manipulator generates the check parameter \( R'_a \leftarrow Sign(sk, h, T, R, M'_a) \) using his chameleon private key sk to ensure secure communication.

4) Verify: In this phase, to verify whether avatar A’s physical identity is consistent with his virtual identity, the avatar B needs to perform the following three steps: (i) extracts the challenge information \( C'_a \) from \( M'_a \) and checks \( C'_a = C_a \) to ensure whether \( M'_a \) is a replay; (ii) checks the match between the biometric feature \( M'_a \) and the biological template \( T \in MIT \) to ensure the validity of A’s physical identity; (iii) verifies the match between the collision pairs \( (M'_a, R'_a) = P ID \) and \( (T, R) \subset MIT \) to ensure the consistency of avatar’s virtual and physical identities. If all the above steps are passed, the verifier can determine that A’s physical identity is consistent with its virtual identity.

B. Two-Party Authentication Protocol

In this part, we designed a two-party authentication protocol to realize the decentralized mutual authentication between avatars. As shown in Fig.3(b), the two-party protocol adds a phase of session key negotiation based on the one-party protocol to achieve secure communication between avatar A and avatar B. Details are as follows:

Round 1: In phase 1 and phase 2, first avatar A submits an identity claim to avatar B, then B submits a claim of B’s identity and a challenge to A. The specific steps are as follows: (i) A submits B with \( SN_a \) and \( V ID_a \) to initiate a claim about A’s identity; (ii) upon B checking the validity of A’s virtual identity based on \( MIT_a \) and \( V ID_a \), B sends \( SN_b, V ID_b \) and \( C_a \) to A, where \( SN_b \) and \( V ID_b \) is the claim of B’s identity, \( C_a \) is a random challenge to A’s physical identity.

Round 2: In phase 3 and phase 4, first avatar A submits his physical identity parameters as a response and throws a challenge to avatar B, then B submits his own physical identity parameters as a response and sends a session parameter to A. The specific steps are as follows: (i) after A checks the validity of B’s virtual identity based on \( MIT_b \) and \( V ID_b \), A submits \( P ID_a \) and \( C_b \) to B, where \( C_b \) is a random challenge to B’s physical identity; (ii) upon B checking the consistency of A’s physical identity based on \( MIT_a \) and \( P ID_a \), B responses \( P ID_b \) and \( g^w \) to A, among them, \( g^w \) is the secure parameter for establishing a session key \( K = y^w \) between A and B.

1) Session Key Establishment: After A checking the validity of B’s physical identity based on \( MIT_b \) and \( P ID_b \), A establishes the session key \( K = (g^w)^{1/3} = y^w \) using his private key \( sk = x \) to realize secure communication.

VI. Security Analysis

The proposed authentication framework provides a security guarantee of virtual-physical tracking for avatars. Before the interaction, the traceability depends on the two-party authentication protocol; During the interaction, the consistency of the avatar’s identity depends on the one-party authentication protocol; In the process of verifying the avatar’s virtual identity and physical identity, the verifiability depends on the unforgeability of chameleon signature. In this light, the security core of the authentication framework lies in the security of the proposed chameleon signature. If a malicious player is capable of forging chameleon collisions, he is able to generate any valid physical identity based on outdated biometrics. In this way, the one-party authentication protocol cannot guarantee that the physical identity is consistent with the virtual identity, which enables the malicious avatars to communicate with others via the two-party authentication protocol. Ultimately, a malicious player in the authentication framework cannot be traced through the avatar’s identity parameters.

In this section, we first analyze the security of the chameleon collision signature, including the existential unforgeability under adaptive chosen message attacks (EUF-CMA) and the key holding cost. Then, we analyze the security of the authentication protocol, including resist disguise, resist replay, resist impersonation, and consistency. Finally, we analyze the security of the authentication framework, involving traceability, decentralized authentication, immersive mutual authentication, key escrow problem, and privacy protection.

A. Security of Chameleon Signature

Suppose a malicious attacker \( \mathcal{A} \) knows a public chameleon triplet \( (h, M, R) \) of player B. During the forgery attack, \( \mathcal{A} \) can query collisions from B within a limited number of times, the purpose of \( \mathcal{A} \) is to forge a new collision \( (M^*, R^*) \) such that \( Verify(pk, h, M, R, M^*, R^*) = 1 \), which means that \( \mathcal{A} \) successfully forged a valid colliding signature. Here, we demonstrate that the proposed chameleon signature is resistant to this attack.

1) EUF-CMA: The security of the proposed chameleon signature is based on the Divisible Computation Diffie-Hellman (DCDH) assumption [21]. Its security model is based on the existential unforgeability game \( EUF^A_{\text{DCDH}}(K) \) under chosen message attack. The game as shown in Fig.4 contains an
adversary $A$ and a challenger $B$, simulating queries initiated from $A$ and answers from $B$.

**Theorem 1:** Let $G$ be a multiplicative cyclic group and $H_G$ be a collision-resistant hash on $G$, if the DCDH assumption holds on $G$, the chameleon collision signature is EUF-CMA.

2) **Analysis:** In the following, we briefly analyze the security of the proposed chameleon signature algorithm, while the detailed proof of Theorem 1 is in the Appendix A.

The security of the proposed chameleon signature algorithm depends on the DCDH assumption, that is, it is hard to output $g^{a/b} \in G$ on given random triples $g, g^a, g^b \in G$. According to the game model at Fig. 4, the adversary attempts to forge a check parameter $R^* = (h/m)^x$ with respect to a new message $M^*$ through $(pk, h, M, R)$. Let $pk = g^{1/x} = g^A$, $(h/m)^x = g^B$, where $m^* = H(M^*)$. If the polytime adversary is able to output $R^* = (h/m)^x = g^{B/A}$, then the DCDH problem can be solved (it contradicts with the DCDH assumption). Therefore, the proposed chameleon signature is EUF-CMA.

3) **Keys Holding Cost:** We compare the keys holding cost of our proposed chameleon signature with that of Khalili’s [17] and Gamenisch’s [20]. From Section IV-B we get that the public and private keys of our scheme contain one element in $G_1$ and one element in $Z_p$, respectively. Khalili’s scheme constructs two sets of chameleon hash methods in which construction 2 is more efficient, therefore, we only analyze it. Since construction 2 leverages the signature of knowledge [22] to ensure the security of the chameleon hash, the elements contained in the keys are greatly increased. By analyzing this construction, the elements contained in the key pair are shown in TABLE II. Its public key involves 3 group elements $(G_1^2, G_2)$ and a set of common reference string $crs(m + 2n + 7 G_1, n + 3 G_2)$. Therefore, Khalili’s public key contains $(m + 2n + 7 G_1, n + 4 G_2)$ and private key contains 3 parameters in $Z_p$, which are $x \in Z_p$ in chameleon hash and $a, b \in Z_p$ in zero-knowledge-proofs. Gamenisch’s scheme builds a chameleon hash algorithm in Gap-Groups, which is the same as the group of our scheme. By analyzing this construction, we get that its public key contains $(1 G_1 + 2 G_2)$ and private key contains 2 $Z_p$. Thus, our scheme has low element cost in public-private keys. Additionally, Khalili’s and Gamenisch’s construction has to add a pair of public-private keys for signing because their keys can only generate a hash. But our proposed algorithm only needs a pair of keys for hashing and signing. Therefore, our scheme has fewer key pairs to generate signatures. From the above analysis, we can conclude that our proposed chameleon signature has obvious advantages in terms of key holding cost under the premise of security.

---

**TABLE II**

| Construction | Hash PK | Hash SK | Additional Key |
|--------------|---------|---------|----------------|
| Khalili[17]  | $m + 2n + 7 G_1$ | $n + 4 G_2$ | 3 $Z_p$ | Yes |
| Camenisch[20] | $1 G_1 + 2 G_2$ | $2 Z_p$ | Yes |
| This work    | $1 G_1$ | $1 Z_p$ | No |

1 We utilize $G_1$, $G_2$, $Z_p$ to denote the number of elements contained in groups $G_1$, $G_2$, and finite field $Z_p$, respectively.

2 It indicates whether an additional key is required to implement the chameleon signature.
The security core of the proposed authentication framework lies in the chameleon signature while the security core of the chameleon signature lies in the signature key. The proposed chameleon signature requires players to hold a pair of elements \((1 \in \mathbb{G}_1, 1 \in \mathbb{Z}_1)\) as a public-private key. Among them, the public key is published on the blockchain and the private key is held by the player, which requires players to hold one element in \(\mathbb{Z}_1\). Taking the current blockchain signature algorithm as a reference, Secp256k1 is an elliptic curve digital signature algorithm (ECDSA) used in Bitcoin and Hyperledger Fabric, which sets the private key length to 256 bits. From this, the authentication framework demands setting the private key length to 256 bits. In practical applications, it is necessary to construct a 256-bit proprietary chip for HMD to store the player’s private key, which avoids the leak of the private keys.

B. Security of Authentication Protocol

The proposed authentication protocol includes the one-party authentication protocol and the two-party authentication protocol, where the essence of the two-party protocol is to add a session key negotiation on the basis of the one-party protocol. Therefore, the security of the two-party protocol lies in the one-party. The goal of the one-party protocol is to ensure the consistency of avatars’ virtual and physical identities, which needs to avoid the three attacks of disguise, replay, and impersonation. Though phase 1–2 of the one-party protocol as shown in Fig. 3(a), it avoids disguise attack by identifying abnormal virtual identity between the attacker’s \(SN^a\) and the fake appearance \(M_a \in V I D^f\). In phase 3–4, it avoids replay attack by checking the matching of the fake \((M'_a, R'_a) = P I D'_a\) and \((T, R) \in M I T_a\). Finally, it avoids impersonation attacks by periodically executing the one-party protocol.

1) Disguise: In disguise attack, a malicious avatar \(C\) wants to disguise as avatar \(A\)’s appearance to deceive avatar \(B\). In this attack, what \(C\) can do is to create a similar virtual identity \(V I D^f_a = (M_a, R_a)\) based on its own private key \(sk_c\), where the appearance \(M_a \in V I D^f_a\) is the same as \(M_a \in V I D_a\). In phase 1 of the one-party protocol, \(C\) submits \(\{S N_c, V I D^f_a\}\) to \(B\). In phase 2, although \(B\) cannot confirm that the virtual identity is forged by \(M_a \in V I D_a\), \(B\) can determine that the identity is abnormal according to \(S N_c\) and \(M_a\). Then, \(B\) gives up getting \(M I T_c\) and terminates the interaction. In another case, if \(B\) is not aware of the abnormality of \(S N_c\) and \(M_a\), \(C\) needs to submit its \(P I D^f = (M'_a, R'_a)\) for physical identity verification. Based on \(S N_c \in M I T_c, V I D^f\), and \(P I D^f\), the malicious avatar \(C\) can be tracked. Therefore, the one-party protocol can resist disguise attacks.

2) Replay: In replay attack, a malicious player \(C\) has an outdated biological sample \(M'\) of avatar \(A\)’s. In this attack, \(C\) wants to replay \(M'\) to deceive avatar \(B\). In phase 1–2 of the one-party protocol, \(C\) submits \(\{S N_a, V I D_a\}\), while \(B\) returns challenge \(C_a\). In phase 3, \(C\) embeds \(C_a\) into \(M'\) to form \(M''\) and generates check parameter \(R'_a\) to form \(P I D'_a = (M''_a, R'_a)\). In phase 4, since the proposed chameleon signature satisfies EUF-CMA, \((M'_a, R'_a) = P I D'_a\) cannot match \((T, R) \in M I T_a\), meaning that \(P I D'_a\) failed to pass the authentication. Therefore, the one-party protocol can resist replay attacks.

3) Impersonation: In impersonation attack, a malicious player \(C\) can obtain an authenticated device to manipulate the corresponding avatar. In this attack, the manipulator’s physical identity is inconsistent with the avatar’s virtual identity for a certain period of time. To resist this attack, the interactor needs to verify the avatar’s virtual and real identities in real-time. According to the Theorem 2, the proposed one-party protocol can ensure the real-time consistency of the avatar’s identity. Therefore, the one-party protocol can resist impersonation attacks.

Definition 1 (Consistency): An avatar’s identity satisfies consistency if the avatar’s virtual identity and physical identity match a metaverse identity token.

Theorem 2: If the chameleon signature satisfies EUF-CMA, the one-party authentication protocol can guarantee consistency in real-time.

Proof: To guarantee consistency via the one-party authentication protocol, three aspects should be checked according to the Definition 1: the validity of the avatar’s virtual identity, the validity of the avatar’s physical identity, and the freshness of the submitted biometric feature.

During an interaction between avatar \(A\) as the prover and avatar \(B\) as the verifier, \(B\) dynamically executes the one-party protocol to check the consistency of \(A\)’s identity. (i) \(B\) checks the match of \((M_a, R_a) = V I D_a\) and \((T, R) \subset M I T_a\) to ensure the validity of \(A\)’s virtual identity; (ii) \(B\) checks the match of \((T, R) \subset M I T_a\) and \((M'_a, R'_a) = P I D_a\) sampled from \(A\)’s manipulator to ensure the validity of \(A\)’s physical identity; (iii) \(B\) first checks the match of \(M'_a\) and \(T \in M I T\), then it extracts the watermarking \(C^b_a\) from \(M'_a\) and compares \(C_a\) with the challenge \(C_a\) throwing by \(B\) to ensure that the biometric feature is freshness.

Since the chameleon signature satisfies EUF-CMA, only the legal \(V I D_a\) satisfies \(Verify(pk, h, T, R, M_a, R_a) = 1\) in step i, and only the legal \(P I D_a\) satisfies \(Verify(pk, h, T, R, M'_a, R'_a) = 1\) in step ii. If step iii is passed, then \(M'_a \in P I D_a\) is fresh, meaning that the \(V I D_a\) and the \(P I D_a\) match the \(M I T_a\). Therefore, the one-party protocol is able to guarantee that only the legal \(V I D_a\) and \(P I D_a\) match with \(M I T_a\). Thus, by executing a round of the one-party protocol, the \(A\)’s virtual and physical identities can be temporarily consistent. To ensure the real-time consistency of \(A\)’s virtual and physical identities, \(A\) dynamically executes the one-party authentication protocol to submit \(B\) with \(S N_a \in M I T_a, V I D_a,\) and \(P I D_a\), where the biometric feature \(M'_a \in P I D_a\) containing a new challenge. Through periodic execution of the one-party protocol, it guarantees the consistency of the avatar’s virtual and physical identities in real-time.

C. Security of Authentication Framework

1) Traceability: The virtual-physical traceability means that a malicious manipulator should be traced back through the avatar’s identity parameters, and an honest manipulator should not be framed by fake identity parameters.

To track back a malicious manipulator, the whistleblower submits IDP with the avatar’s identity parameters \(M I T = (S N, pk, h, T, R), V I D = (M_a, R_a),\) and \(P I D = (M'_a, R'_a)\).
After IDP completes the check on these parameters, it discloses the manipulator’s real identity based on \([ID, SN]\) reserved registration, which achieves the virtual-physical tracking.

In some cases, the manipulator may deny that the above parameters are forged by others to frame him. Based on this statement, if a malicious whistleblower wants to frame the honest avatar, the whistleblower needs to forge the avatar’s \(MIT, VID,\) and \(PID\). Among them, the \(MIT\) and the \(VID\) are the avatar’s public information in the metaverse. Thus, the target parameter whistleblower attempts to forge is the \(PID\), that is, the biometric feature \(M_{a}'\) and the corresponding check parameter \(R_{a}'\). For \(M_{a}'\), the whistleblower is able to embed the fresh challenge \(C_{a}\) into outdated biometrics, which is easy to accomplish. But for \(R_{a}'\), since the whistleblower lacks the corresponding chameleon private key, at the same time, the proposed chameleon signature is EUF-CMA, it is hard for a whistleblower to forge \(R_{a}'\). Therefore, the excuse of being framed is untenable, indicating that the proposed authentication framework satisfies the virtual-physical traceability.

2) Decentralized Authentication: The decentralization authentication of the proposed framework relies on the MIT signed by IDP. In the process of authentication, the verifier checks the validity of the avatar’s virtual identity based on \(MIT\) and \(VID\), and checks the consistency of the avatar’s virtual and physical identities based on \(MIT\) and \(PID\). The entire authentication process does not involve a trusted third party, realizing decentralized authentication between avatars.

3) Immersive Mutual Authentication: In the process of authentication, the verifier throws a random challenge to the prover, then the prover automatically generates physical identity parameters as a response, such as capturing biometrics via an HMD device, ensuring that the player’s interactive experience is not disturbed in any way.

4) No Key Escrow Problem: During authentication, the check on the avatar’s virtual and physical identities is completed only by a pair of chameleon keys \((sk = x, pk = y)\), where \(pk\) is published on the blockchain and the \(sk\) is held by the player himself. Therefore, the player’s keys are not escrowed with any trusted third party, which avoids the key escrow problem.

5) Privacy Protection: In the process of authentication between avatars, the prover initiates an anonymous claim through \(SN \in MIT\) and \(VID\), then the verifier achieves anonymous authentication based on the locally processed biometric feature \(M_{a}'\) and the corresponding check parameter \(R_{a}'\). The whole process of authentication doesn’t reveal the actual information about the prover’s identity. In addition, the whole process doesn’t involve an intermediary. Thus, the proposed framework does not leak the player’s identity privacy and social privacy during the metaverse authentication process.

VII. PERFORMANCE EVALUATION

In this section, we first evaluate the computation cost of the chameleon signature. Then, we evaluate the performance of the proposed authentication framework, including the impact of embedding watermarks into biometrics, the computational cost of authentication, the authenticating consumption on different platforms, the consumption on continuous authentication, and the virtual-physical tracking consumption.

A. Performance of Chameleon Signature

1) Computation Cost: We compare the computation cost of our proposed chameleon collision signature with that of Khalili’s [17] and Gamenisch’s [20] as shown in the TABLE III. It can be seen that the proposed algorithm has obvious computational advantages. The reason is that Khalili’s and Camenisch’s construction involves the signature of knowledge and zero-knowledge proof, respectively, while our algorithm eliminates the above time-consuming steps under the premise of security.

To further analyze the performance of the proposed chameleon collision signature, we implemented our scheme and Khalili’s construction based on the Java programming language. The simulation set the batch size of data processing to 10, 20, 30, 40, and 50 respectively. The average time consumption of these two methods is shown in Fig. 5. It can be seen from the figure that the consumption of our proposed algorithm is within 50ms, which has obvious advantages compared with Khalili’s construction.

| Construction  | Hash 1 | Check | Forge 2 |
|--------------|--------|-------|--------|
| Khalili [17] | \(3 + m + 2n - 1 E_1\) | \(1 E_1\) | \(2 + m + 2n - 1 E_1\) |
| Gamenisch [20] | \(4 E_1 + 4 E_2\) | \(1 E_1\) | \(3 E_1 + 1 E_2\) |
| This work    | \(2 E_1\) | \(1 M_1\) | \(1 E_1\) |
|              | \(1 M_1\) | \(2 P\) | \(1 M_1\) |

1 We utilize \(E_1, E_2,\) and \(E_T\) to denote the exponential operation on \(G_1, G_2,\) and \(G_T\), \(M_1\) to denote the multiplication operation on the group \(G_1, P\) to denote the bilinear map.
2 We utilize Forge to denote the \(HC\) in Khalili’s construction, the \(Adapt\) in Camenisch’s construction, and the \(Sign\) in our scheme because the algorithms \(HC, Adapt,\) and \(Sign\) are essentially forging chameleon collisions.
B. Performance of Authentication Framework

1) Watermarked Biometrics: To analyze the impact of embedding watermarks into biometrics, we take the player’s native iris feature as the biometric template T in the avatar’s MIT and take the native and the watermarked iris features as the response parameters, respectively. During simulation, we utilize the public dataset CASIA-IrisV4-Thousand\(^5\) and CET2005\(^6\) to build the iris biometrics, the OSIRIS algorithm [23] to extract the iris-encoded, and the adaptive image watermarking algorithm [24] to embed 128-bit random challenge into the iris-encoded image. Through simulations, we obtain the False Rejection Rate (FRR) and the False Acceptance Rate (FAR) of iris recognition as shown in Fig. 6. It can be seen that although the proposed authentication framework embeds a watermark into the iris-encoded, the FRR and FAR of iris recognition are almost the same as that of the native iris features under the two types of public datasets. Therefore, our method further prevents replay attacks without affecting recognition efficiency.

2) Computational Cost of the Protocol: The proposed authentication protocol contains one party and two parties. Since the latter includes all the steps of the former, we analyze the computational cost of the latter. The two-party protocol involves verifying the signature on MIT, embedding and extracting watermarks in biometric features, and generating and checking chameleon parameters. To analyze the core computational cost of our proposed protocol, we only consider the generating and checking of chameleon parameters. From the two-party protocol in Section V-B, the costs of each phrase are shown in the TABLE IV.

In round 1, avatar A only submits the identity parameter to B which involves zero calculation consumption, while the avatar B needs to check the validity of avatar A’s identity through Verify in which involves the computational cost of \((1 M_1 + 2 P) \times 2\). In round 2, A needs to check the validity of B’s identity through Verify and generates a chameleon signature by Sign, which involves the calculation cost of \((1 M_1 + 2 P) \times 2 + 1 E_1 + 1 M_1\). B needs to check the validity of A’s response by Verify, generates a chameleon signature by Sign, and submits a random parameter \(g^w\) to A, which involves the computation cost of \((1 M_1 + 2 P) \times 2 + 2 E_1 + 1 M_1\). In session key establishment, A needs to check the validity of B’s response through Verify and generates the session key \(K = (g^w)^{1/2}\) by his chameleon private key \(x\), which involves the computation cost of \((1 M_1 + 2 P) \times 2 + 1 E_1\). But for B, he establishes the session key \(K = y^w\) by the random number \(w\), which involves the computational cost of \(1 E_1\). To sum up, the calculation costs of A and B in the two-party authentication protocol are \((5 M_1 + 2 E_1 + 8 P)\) and \((5 M_1 + 3 E_1 + 8 P)\), respectively.

3) Authenticating Consumption on Different Platforms: To further analyze the time consumption of the proposed authentication framework, we conduct simulations on three different platforms, PC (CPU is Intel Core i5 7500, memory is 8 GB, the operating system is Windows 10), Smart Phone(CPU is Hisilicon Kirin 659, memory is 4 GB, the operating system is Android 9), and Raspberry Pi (CPU is Quad-Core ARM Cortex-A72, memory is 4 GB). The time consumption of one-party authentication protocol and two-party authentication protocol on different platforms are shown in Fig. 7(a), where Sign is the time consumption to generate the chameleon signature, EmWater is to embed random challenge as a watermark into the biometric feature, ExWater is to extract the watermark from the biometric feature, and Verify is to check the validity of avatar’s virtual and physical identities. It can be seen from Fig. 7(a) that the total consumption of the two-party authentication protocol on PC and Smart Phone is within 1000ms, while that on Raspberry is about 2000ms. In general, the proposed authentication framework is efficient on PC and Smart Phone.

4) Consumption on Continuous Authentication: The proposed decentralized authentication framework offloads the authentication task from the central platform to the interactive devices, which reduces platform resource consumption to support large-scale virtual events. Avatar achieves continuous authentication through periodic execution of the one-party authentication protocol. During actual continuous authentication, an avatar only needs to read the MIT once from the consortium blockchain by \(SN \in MIT\), where \(SN\) is the serial number of MIT. The current consortium blockchains such as Hyperledger Fabric are read at almost the same speed as a traditional central database. Moreover, the blockchain can increase the speed beyond the traditional central database by adding edge nodes and setting indexes. Therefore, the performance of the current blockchains will not have a noticeable impact on the proposed authentication framework.

\(^{5}\)http://biometrics.idealtest.org

\(^{6}\)https://www.nist.gov/itl/iad

TABLE IV

| Phase       | avatar A                 | avatar B                    |
|-------------|--------------------------|------------------------------|
| Round 1     | –                        | \(2 M_1 + 4 P\)              |
| Round 2     | \(3 M_1 + 4 P + 1 E_1\)  | \(3 M_1 + 4 P + 2 E_1\)      |
| Session     | \(2 M_1 + 4 P + 1 E_1\)  | \(1 E_1\)                    |
| Total       | \(5 M_1 + 8 P + 2 E_1\)  | \(5 M_1 + 8 P + 3 E_1\)      |
We further analyze the authentication protocol and optimize execution steps to improve the efficiency of continuous authentication. For a verifier $B$, he performs mutual authentication first and then continuous authentication, meaning that $B$ holds valid $VID_a$ and $MIT_a$ at the end of the two-party authentication protocol. Therefore, in the execution of the one-party protocol, $B$ only needs to periodically send a random challenge to $A$, which forces $A$ to submit $PID_a$ as responses. To analyze the consumption of continuous authentication, we use the Raspberry Pi with similar hardware performance to the HMD for simulation, the hardware parameters are shown as Table V. In the simulation, we set the authentication frequency to 1 time/minute, 3 times/minute, and 6 times/minute respectively. From the results, as shown in Fig. 7(b), it can be seen that the time consumption of the original one-party authentication protocol is about 1s in each round and the consumption of optimized execution steps is reduced by 30%. Therefore, to meet the practical application, the proposed framework recommends adopting the optimized steps and setting the frequency to 1 time/minute to achieve continuous authentication.

5) Virtual-physical Tracking Consumption on Core Steps: The identity parameters retained in the authentication process provide conditions for virtual-physical tracking. The specific steps are as follows: (i) the whistleblower submits IDP with the avatar’s identity parameters including $MIT_a$, $VID_a$, and $PID_a$; (ii) the IDP checks the validity of virtual identity based on $MIT$ and $VID$, checks the consistency of physical-virtual identity based on $MIT$ and $PID$; (iii) after the above check is passed, the IDP reveals the manipulator’s real identity via $\{ID, SN\}$, which achieves virtual-physical tracking. The time consumption of the above process mainly involves the feature matching between the iris-encode and biometric template, the extracting watermark from the iris-encode, and the verifying signature on the chameleon collision pair. To analyze this consumption, we set the report times for malicious avatars to be 10, 20, 30, and 40 respectively. Through simulations on the PC platform, we get the tracking consumption for virtual-physical tracking as shown in Fig. 8. What can be seen from the figure is that the average consumption of feature matching is about 75ms, the extracting watermark is about 35ms, and the verifying signature is about 65ms. From the above cost analysis, we can get that the designed authentication framework can not only achieve virtual-physical tracking but also has a low consumption cost.

| Type       | Device            | CPU             | ARM   |
|------------|-------------------|-----------------|-------|
| HMD        | Oculus quest2 VR  | Snapdragon XR2 (1.8GHz) | 6 GB  |
| Raspberry Pi | Raspberry Pi 4B   | Cortex-A72 (1.5GHz) | 4 GB  |

We further analyze the authentication protocol and optimize execution steps to improve the efficiency of continuous authentication. For a verifier $B$, he performs mutual authentication first and then continuous authentication, meaning that $B$ holds valid $VID_a$ and $MIT_a$ at the end of the two-party authentication protocol. Therefore, in the execution of the one-party protocol, $B$ only needs to periodically send a random challenge to $A$, which forces $A$ to submit $PID_a$ as responses. To analyze the consumption of continuous authentication, we use the Raspberry Pi with similar hardware performance to the HMD for simulation, the hardware parameters are shown as Table V. In the simulation, we set the authentication frequency to 1 time/minute, 3 times/minute, and 6 times/minute respectively. From the results, as shown in Fig. 7(b), it can be seen that the time consumption of the original one-party authentication protocol is about 1s in each round and the consumption of optimized execution steps is reduced by 30%. Therefore, to meet the practical application, the proposed framework recommends adopting the optimized steps and setting the frequency to 1 time/minute to achieve continuous authentication.

5) Virtual-physical Tracking Consumption on Core Steps: The identity parameters retained in the authentication process provide conditions for virtual-physical tracking. The specific steps are as follows: (i) the whistleblower submits IDP with the avatar’s identity parameters including $MIT_a$, $VID_a$, and $PID_a$; (ii) the IDP checks the validity of virtual identity based on $MIT$ and $VID$, checks the consistency of physical-virtual identity based on $MIT$ and $PID$; (iii) after the above check is passed, the IDP reveals the manipulator’s real identity via $\{ID, SN\}$, which achieves virtual-physical tracking. The time consumption of the above process mainly involves the feature matching between the iris-encode and biometric template, the extracting watermark from the iris-encode, and the verifying signature on the chameleon collision pair. To analyze this consumption, we set the report times for malicious avatars to be 10, 20, 30, and 40 respectively. Through simulations on the PC platform, we get the tracking consumption for virtual-physical tracking as shown in Fig. 8. What can be seen from the figure is that the average consumption of feature matching is about 75ms, the extracting watermark is about 35ms, and the verifying signature is about 65ms. From the above cost analysis, we can get that the designed authentication framework can not only achieve virtual-physical tracking but also has a low consumption cost.

VIII. RELATED WORK

Metaverse players manipulate their avatars based on head-mounted display devices. In building an authentication framework, if the avatar is checked only by the device keys, it is hard to ensure that the avatar’s virtual identity is consistent with the manipulator’s physical identity; if the avatar is checked only by the player’s biometrics, it is difficult to avoid replay; Therefore, combining player biometrics and device...
keys is critical for building a traceable authentication framework. At present, researchers mainly design authentication protocols for users or devices through two types of methods, knowledge-based and biometric-based. In this section, we sort out the above authentication methods to build a traceable authentication framework for avatars.

A. Knowledge-Based Authentication

Knowledge-based authentication methods discriminate the identities of users or devices based on information known only to the entity. For example, the network platform authenticates the login user via their password, and the Internet of Things authenticates the device based on its key.

1) Password-Authenticated: The most widespread authentication method over the Internet is the password-authenticated mechanism. The mechanism relies on server operators as trusted parties to offer secure authentication, giving them full control over users’ identities. Zhang et al. [25] constructed a password-based threshold single-sign-on authentication scheme to thwart adversaries from compromising the identity servers. To mitigate the threat of centralized authentication, Szalachowski [26] designed a password-authenticated decentralized identity by which users register their self-sovereign username-password pairs and use them as universal credentials. This kind of password-authenticated method can realize the security authentication between the device and the service provider. However, these methods require the user to enter a password during the authentication process, which fails to achieve immersive authentication. In addition, anyone can pass the login authentication as long as he knows the password, making this method impossible to guarantee consistency between the avatar and its manipulator.

2) Key-Authenticated: With the large-scale application of IoT devices, researchers have been working on building a secure and efficient authentication framework [27] based on public-private key pairs. Guo et al. [28] designed an asymmetric cryptographic algorithm based on elliptic curve cryptography (ECC) to construct a distributed and trusted authentication system for smart terminals. To achieve that the user’s identifying information is controlled by themselves, Xu et al. [29] constructed blockchain-based identity management and authentication scheme. Shen et al. [30] presented an efficient blockchain-assisted secure device authentication mechanism for cross-domain IIoT to alleviate the cost of key management overhead. These public key-based authentication methods facilitate one-party authentication without involving a trusted third party, but it is difficult to efficiently realize Device-to-Device (D2D) mutual authentication. Certificateless cryptographic algorithms are widely used to build mutual authentication methods between devices [31] because they avoid key escrow issues. Shang et al. [32] designed an authentication and key agreement protocol to negotiate a group session key securely and effectively in D2D group communications, which merges the advantages of certificates public key cryptography (CL-PKC) and ECC. Gope et al. [33] presents a lightweight and privacy-preserving two-factor authentication scheme based on physically unclonable functions (PUF).

Li et al. [34] proposes an end-to-end mutual authentication and key exchange protocol for IoT by combining PUF with CL-PKC.

The above scheme realizes efficient mutual authentication between devices, but the authentication process is separated from the user. If this kind of method is directly used to build the metaverse authentication framework, it is difficult to ensure the consistency of an avatar and its manipulator. Therefore, we prepare to construct an efficient signature algorithm based on the ideas of chameleon hash [17] and short signature [15] efficiently guarantee that the avatar is associated with its manipulator.

B. Biometric-Based Authentication

Biometrics are inherent information of the human body and there is no risk of being forgotten or lost, making these methods widely used in the verification of the user’s physical identity. The type of biometric-based authentication method can be divided into two sub-categories, physiological biometric-based and behavioral biometric-based. The physiological biometric-based methods are primarily based on face [35], iris [36], fingerprint [37], and other characteristics to complete static authentication. The behavioral biometric-based methods rely on walking gait [38], eye movements [39], and response to vibrations [40] to achieve dynamic authentication.

1) Physiological Biometric-Based: Owing to its simplicity and accuracy, physiological-based authentication technology has attracted the extensive attention of many experts and scholars [12], [13], [41]. Papadamou et al. [42] proposed a privacy-preserving federated architecture for device-centric authentication (DCA) in which the core authentication functionality resides on a trusted entity. To prevent the attacker from knowing the user’s biometrics after the database is compromised, Chatterjee et al. [43] proposed practical secure sketches to conceal the correspondence between users and their biometric templates. The above methods realize efficient authentication of device and user identities, but these methods are all one-time authentication methods, which are difficult to guarantee that the user’s identity is consistent with the avatar’s identity.

2) Behavioral Biometrics-based: To mitigate the vulnerabilities of static biometrics, the behavior-based authentication method realizes dynamic authentication by continuously collecting user biometrics. Zhu et al. [44] designed an unobtrusive real-time user authentication system by which each biometric sample is aggregated into users’ implicit events, such as raising hands to check the time on the watch. Li [45] leverages the response of hand-surface vibration to construct a resilient user authentication system. Similarly, Lee et al. [40] proposed a usable method for user authentication through the response to vibration challenges on the user’s smartwatches. Wu [46] designed a multimodal biometrics system on smartphones that leverages lip movements and voice for authentication. It can be seen from the above scheme that the behavioral-mentioned authentication mainly solves the problem of dynamic authentication without involving mutual authentication between users. Moreover, these methods are easily limited by physical scenarios, such as, the vibration-response method may be invalid.
when the user has some items in their hands because the newly added item may interfere with the response value [18].

The multi-factor authentication mechanism is conducive to building an authentication mechanism that meets any scenario, as it eliminates dependence on trusted third parties and avoids the involvement of user behavior. Gunasinghe and Bertino [47] introduces a three-factor authentication scheme based on a signed identity token that encodes the user’s biometric identifier and the password entered by the user into the token. Although this method requires users to enter a password during the authentication process, which disturbs the immersive experience of metaverse players, this method is facilitative to the realization of decentralized mutual authentication for avatars based on the player’s biometrics and his keys. Inspired by this idea, we designed a decentralized authentication framework based on two factors to achieve the verifiability of the avatar’s virtual and physical identities and guarantee virtual-physical traceability.

IX. CONCLUSION

Metaverse is a virtual-physical coexistence environment where people communicate and work via digital avatars. In the emerging social ecosystems, however, malicious players frequently violate the safety of other avatars, posing a huge challenge to the healthy development of the metaverse. For this issue, we design a two-factor authentication framework based on chameleon signature and iris biometrics, which guarantees the virtual-physical traceability that tracks an avatar in virtual space to its manipulator in the physical world. To the best of our knowledge, our method is the first work for avatar tracking in the metaverse field. We hope that the proposed framework could bring a little reference to researchers in related fields.

APPENDIX A

SECURITY PROOF FOR CHAMELEON COLLISION SIGNATURE

Theorem 3: Let $G$ be a multiplicative cyclic group and $H_G$ be a collision-resistant hash on $G$, if the DCDH assumption holds on $G$, the chameleon collision signature is EUF-CMA.

Proof: suppose there is a polytime adversary $A$ that breaks the chameleon collision signature with the advantage of $\epsilon(K)$, then there must be an adversary $B$ to solve the DCDH on $G$ at least by the advantage of $Adv_B^{DCDH} \geq \epsilon(K) / e \cdot q_H$.

Where $e$ is the base of the natural logarithm, $q_H$ is the maximum number of queries to $H_G$.

From section VI, the process of the game $EUF_A^{DCDH}(K)$ between $A$ and $B$ is as follows:

1. Adversary $B$ runs $Setup(K)$ and $KeyGen(Parm)$ to select a random function $H_G \in (H : \{0, 1\}^* \rightarrow G)$, generate the key pair $(pk, sk)$, calculate the original chameleon parameters $(y, h, M, R)$, and send adversary $A$ with the system parameters and the chameleon parameters.

2. The adversary $A$ queries the adversary $B$ for the hash and the corresponding signature with any $M'$. Adversary $B$ response $r'$ and $R'$ as the corresponding answer. During this process, $q_H$ is the maximum number of times $A$ queries $H_G(.)$.

3. The adversary $A$ outputs a chameleon collision $(M^*, R^*)$ as a forged signature. If $Check(y, h, M, R) = Check(y, h, M^*, R^*) = 1$, the adversary’s attack is successful, where the check parameter $R^*$ of $M^*$ has not been queried by it before.

From the above process, the adversary $A$ wants to find a certain $r^*$ related to $M^*$ such that $(r^*)^2 = (h/m)^x = R^*$, then, a fake collision $(M^*, R^*)$ can be successfully output. During the hash query, if $m^x$ is the hash value of a certain message $M^*$, then $(h/m)^x = R^*$ is the check parameter for $M^*$. In step (3), $(M^*, R^*)$ is generated by adversary $A$, but $H_G(M^*)$ is generated by $B$, so $B$ is able to set $r^* = h/H_G(M^*) = H_t(M^*)$. When $B$ lets $r^*$ be the potential hash of a targeted message, his goal is to call $A$ to calculate $(r^*)^x$ based on the triple $(g, g^{1/3}, r^*)$, which is to solve the DCDH problem. As already proven in [21], the DCDH problem is equivalent to the CDH problem. Thus, we convert $B$’s attack target to the CDH problem, that is, calling $A$ to calculate $(r^*)^x$ based on the triple $(g, g^x, r^*)$. Throughout the game, $B$ doesn’t know which message will be generated by $A$ to forge a check parameter. Therefore, $B$ has to make a guess that the $j$-th query $H_t$ corresponds to the final forged result from $A$.

For simplicity without loss of generality, we assume that: (i). Adversary $A$ will not initiate the same query $H_t(\cdot)$ twice to $M'$; (ii). Adversary $A$ must have asked $H_t(M')$ before querying the check parameter $R'$; (iii). Adversary $A$ must have asked $H_t(M^*)$ before he outputs $(M^*, R^*)$.

In the actual process, $B$ implicitly regards $u = g^a$ in the known tuple $(g, u = g^a, r^*)$ as its own public key (in fact, $B$ doesn’t know the specific value of $a$, then $(r^*)^a$ is a forged check parameter of a certain message, namely $R^* = (r^*)^a = (h/H_G(M^*))^a = (H_t(M^*))^a$, where $(r^*)^a$ is forged by $B$. To hide instance $u = g^a$, $B$ needs to select a randomness $t \sim Z_p$ and send $u \cdot g^t$ to $A$ as the public key of $B$.

| Category          | Factors       | Virtual | Physical | Immersive | Continuous | Mutual | SessionKey |
|-------------------|---------------|---------|----------|-----------|------------|--------|------------|
| Knowledge         | Password [26] | ☑       |          | ☑         | ☑          | ☑      | ☑          |
| Knowledge         | Key [33]      | ☑       | ☑        | ☑         | ☑          | ☑      | ☑          |
| Biometric         | Physiology [13]| ☑      | ☑        | ☑         | ☑          | ☑      | ☑          |
| Biometric         | Behavior [40] | ☑       |          | ☑         | ☑          | ☑      | ☑          |
| Knowledge-Biometric| Multi-Factor [47]| ☑   | ☑        | ☑         | ☑          | ☑      | ☑          |

We utilize Virtual and Physical to denote the verifiability of an entity’s virtual identity and physical identity respectively, Immersive indicates the immersive authentication between entities, Continuous represents the continuous authentication, Mutual as the mutual authentication, and SessionKey as the negotiation of session keys.

We compare the advantages of different solutions in Table VI.

| Category          | Factors       | Virtual | Physical | Immersive | Continuous | Mutual | SessionKey |
|-------------------|---------------|---------|----------|-----------|------------|--------|------------|
| Knowledge         | Password [26] | ☑       |          | ☑         | ☑          | ☑      | ☑          |
| Knowledge         | Key [33]      | ☑       | ☑        | ☑         | ☑          | ☑      | ☑          |
| Biometric         | Physiology [13]| ☑      | ☑        | ☑         | ☑          | ☑      | ☑          |
| Biometric         | Behavior [40] | ☑       |          | ☑         | ☑          | ☑      | ☑          |
| Knowledge-Biometric| Multi-Factor [47]| ☑   | ☑        | ☑         | ☑          | ☑      | ☑          |
The following proves that the EUF-CMA game $EUF_{\mathcal{A}}^{\text{DCDH}}(\mathcal{K})$ of the chameleon collision signature can be reduced to the CDH problem.

(1) $\mathcal{B}$ sends the generator $g$ of group $\mathbb{G}$ and the public key $y$ to $\mathcal{A}$, where the private key corresponding to $y$ is $a + t$, $t \xleftarrow{\$} \mathbb{Z}_q^*$. That is

$$y = u \cdot g^t = g^{a + t} \in \mathbb{G}.$$  

At the same time, $\mathcal{B}$ randomly selects $j \xleftarrow{\$} \{1, 2, \ldots, q_H\}$ as the hypothetical index of the forged parameters, that is, the $j$-th query of $H_r$ from $\mathcal{A}$ corresponds to the hash of the target message $M^*$.

(2) $H_r$ query (at most $q_H$ times). $\mathcal{B}$ creates an empty list $H^{\text{list}}$ and lets the five-tuple $(h, M_i, r_i, b_i, r'_i)$ be the element in it, which means that $\mathcal{B}$ has set

$$H_r(M_i) = h / H_{G}(M_i) = r_i.$$

When $\mathcal{A}$ makes the $i$-th inquiry to $H_r(\cdot)$, $\mathcal{B}$ randomly selects $b_i \xleftarrow{\$} \mathbb{Z}_p^*$ and answers as follows:

- If $i = j$, return $r'_i = r_i \cdot g^{b_i} \in \mathbb{G}$;
- Otherwise, calculate $r'_i = g^{b_i} \in \mathbb{G}$.

$\mathcal{B}$ takes $r'_i$ as the answer to the query $H_r(M_i)$ and appends $(h, M_i, r_i, b_i, r'_i)$ to the list $H^{\text{list}}$.

(3) $\text{Sign}^{\text{f}}$ query (at most $q_H$ times). In the process of $\mathcal{A}$ requesting the check parameter of a message $M'$, $\mathcal{B}$ lets $M' = M_i$ be the $i$-th $H_r$ query, and answers the query in the following way:

- If $i \neq j$, $\mathcal{B}$ retrieves the tuple $(h, M_i, r_i, b_i, r'_i)$ in the $H^{\text{list}}$ by $M_i$, computes $R'_i = \langle u \cdot g^{t} \rangle^{b_i}$, and returns $R'_i$ to $\mathcal{A}$, where $R'_i$ is the check parameter constructed by $(r'_i, M_i)$ with secret key $a + t$.

$$R'_i = \langle u \cdot g^{t} \rangle^{b_i} = (g^{a + t})^{b_i} = (g^{b_i})^{a + t} = (r'_i)^{a + t}.$$  

- Otherwise, interrupt.

(4) $\mathcal{A}$ outputs $(M^*, R^*)$. If $M^* \neq M_j$, interrupt; otherwise, $\mathcal{B}$ outputs $R^* \leftarrow \frac{R^*}{r_i^{(a + t) \cdot g^{b_i}}} \pmod{g}$ as $(r^*)^a$. For

$$R^* = \langle r'_j \rangle^{(a + t)} = \langle r^* \cdot g^{b_i} \rangle^{(a + t)}$$

$$= \langle r^* \rangle^{(a + t)} \cdot g^{b_i^{(a + t)}}$$

$$= \langle r^* \rangle^{a + t} \cdot \langle g^{b_i} \rangle^{a + t}$$

$$= \langle r^* \rangle^{a + t} \cdot \langle g^{(a + t)b_j} \rangle$$

$$= \langle r^* \rangle^{a + t} \cdot u^b g^{b_j}.$$  

From this, we can get the following result:

$$R^* \leftarrow \frac{R^*}{r_i^{(a + t) \cdot g^{b_i}}} \pmod{g} = \langle r^* \rangle^a.$$  

If the guess $j$ from $\mathcal{B}$ is correct and $\mathcal{A}$ finds a correct forgery, $\mathcal{B}$ successfully solves the given CDH problem, that is, $\mathcal{B}$ finds $R^* = \langle r^* \rangle^a$ based on $(g, g^a, r^*)$ through $\mathcal{A}$. The successful output $R^* = \langle r^* \rangle^a$ from $\mathcal{B}$ is determined by the following three events:

- $\mathcal{E}_1$: No interruption was encountered during the interaction between $\mathcal{A}$ and $\mathcal{B}$.
- $\mathcal{E}_2$: $\mathcal{A}$ produces a valid “message-parameter” pair $(M^*, R^*)$.
- $\mathcal{E}_3$: $\mathcal{E}_2$ occurs and the subscript of $M^*$ in the corresponding five-tuple $(h, M_i, r_i, b_i, r'_i)$ is $i = j$. Then

$$Pr[\mathcal{E}_1] = (1 - \frac{1}{q_H^*})^j,$$

$$Pr[\mathcal{E}_2|\mathcal{E}_1] = \epsilon(\mathcal{K}),$$

$$Pr[\mathcal{E}_3|\mathcal{E}_1, \mathcal{E}_2] = Pr[i = j|\mathcal{E}_1, \mathcal{E}_2] = \frac{1}{q_H}.$$  

So the advantage of $\mathcal{A}$ is:

$$Pr[\mathcal{E}_3] = Pr[\mathcal{E}_1] \cdot Pr[\mathcal{E}_2|\mathcal{E}_1] \cdot Pr[\mathcal{E}_3|\mathcal{E}_1, \mathcal{E}_2]$$

$$= (1 - \frac{1}{q_H^*})^j \cdot \frac{1}{q_H} \cdot \epsilon(\mathcal{K})$$

$$\approx \epsilon(\mathcal{K}) \cdot q_H.$$  

Since the DCDH assumption holds on $\mathbb{G}$ and the DCDH problem equals the CDH problem, the advantage $\epsilon(\mathcal{K})/q_H$ of polytime adversary $\mathcal{B}$ is negligible, so the chameleon sign algorithm is EUF-CMA. (Theorem is proved)

REFERENCES

[1] L.-H. Lee et al., “All one needs to know about metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda,” 2021, arXiv:2110.05352.

[2] R. Cheng, N. Wu, S. Chen, and B. Han, “Will metaverse be NextG internet? Vision, hype, and reality,” 2022, arXiv:2201.12894.

[3] H. Duan, J. Li, S. Fan, Z. Lin, X. Wu, and W. Cai, “Metaverse for social good: A university campus prototype,” in Proc. 29th ACM Int. Conf. Multimedia, Oct. 2021, pp. 153–161.

[4] S. Kraus, D. K. Kanbach, P. M. Krysta, M. M. Steinhoff, and N. Tomini, “Facebook and the creation of the metaverse: Radical business model innovation or incremental transformation?” Int. J. Entrepreneurial Behav. Res., vol. 28, no. 5, pp. 52–77, Feb. 2022.

[5] Q. Yang, Y. Zhao, H. Huang, Z. Xiong, J. Kang, and Z. Zheng, “Fusing blockchain and AI with metaverse: A survey,” IEEE Open J. Comput. Soc., vol. 3, pp. 122–136, 2022.

[6] Y. Wang et al., “A survey on metaverse: Fundamentals, security, and privacy,” 2022, arXiv:2203.02662.

[7] K. Yu, G. Gorbachev, U. Eck, F. Pankratz, N. Navab, and D. Roth, “Avatars for teleconsultation: Effects of avatar embodiment techniques on user perception in 3D asymmetric telepresence,” IEEE Trans. Vis. Comput. Graphics, vol. 27, no. 11, pp. 4129–4139, Nov. 2021.

[8] Y. Li, Y. Cheng, W. Meng, Y. Li, and R. H. Deng, “Designing leakage-resilient password entry on head-mounted smart wearable glass devices,” IEEE Trans. Inf. Forensic Security, vol. 16, pp. 307–321, 2021.

[9] L. Cheng, D. M. Divakaran, A. W. K. Ang, W. Y. Lim, and V. L. L. Thing, “FACT: A framework for authentication in cloud-based IP traceback,” IEEE Trans. Inf. Forensic Security, vol. 12, no. 3, pp. 604–616, Mar. 2017.

[10] M. Adil et al., “Hash-MAC-DSDV: Mutual authentication for intelligent IoT-based cyber-physical systems,” IEEE Internet Things J., vol. 9, no. 22, pp. 22173–22183, Nov. 2022.

[11] J. Ryu, S. Son, J. Lee, Y. Park, and Y. Park, “Design of secure mutual authentication scheme for metaverse environments using blockchain,” IEEE Access, vol. 10, pp. 99844–99858, Feb. 2022.

[12] S. Agrawal, S. Badrinarayanan, P. Mukherjee, and P. Rindal, “Game-set-MATCH: Using mobile devices for seamless external-facing biometric matching,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., Oct. 2020, pp. 1351–1370.

[13] S. Luo, A. Nguyen, C. Song, F. Lin, W. Xu, and Z. Yan, “OcuLock: Exploring human visual system for authentication in virtual reality head-mounted display,” in Proc. Netw. Distrib. Syst. Secur. Symp., 2020, pp. 23–26.

[14] S. A. K. Thyagarajan, A. Bhat, G. Malavolta, N. Döttling, A. Kate, and D. Schröder, “Verifiable timed signatures made practical,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., Oct. 2020, pp. 1733–1750.

[15] D. Boneh, X. Boyen, and H. Shacham, “Short group signatures,” in Advances in Cryptology—CRYPTO, vol. 3152. Berlin, Germany: Springer, 2004, pp. 41–55.

[16] N. Mohassel, “One-time signatures and Chameleon hash functions,” in Proc. 17th Int. Conf. Sel. Areas Cryptography, Aug. 2010, pp. 302–319.
