information

Article

Cross-Domain Identity Authentication Protocol of Consortium Blockchain Based on Face Recognition

Xiang Chen [1], Shouzhi Xu [1,2,*], Kai Ma [1,2] and Peng Chen [1,2]

[1] College of Computer and Information Technology, China Three Gorges University, Yichang 443002, China
[2] Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering, China Three Gorges University, Yichang 443002, China
* Correspondence: xsz@ctgu.edu.cn

Abstract: A consortium system can leverage information to improve workflows, accountability, and transparency through setting up a backbone for these cross-company and cross-discipline solutions, which make it become a hot spot of market application. Users of a consortium system may register and log in different target domains to get the access authentications, so how to access resources in different domains efficiently to avoid the trust-island problem is a big challenge. Cross-domain authentication is a kind of technology that breaks trust islands and enables users to access resources and services in different domains with the same credentials, which reduces service costs for all parties. Aiming at the problems of traditional cross-domain authentication, such as complex certificate management, low authentication efficiency, and being unable to prevent the attack users' accounts, a cross-domain authentication protocol based on face recognition is proposed in this paper. The protocol makes use of the decentralized and distributed characteristics of the consortium chain to ensure the reliable transmission of data between participants without trust relationships, and achieves biometric authentication to further solve the problem of account attack by applying a deep-learning face-recognition model. An asymmetric encryption algorithm is used to encrypt and store the face feature codes on the chain to ensure the privacy of the user's face features. Finally, through security analysis, it is proved that the proposed protocol can effectively prevent a man-in-the-middle attack, a replay attack, an account attack, an internal attack, and other attacks, and mutual security authentication between different domains can be realized with the protocol.

Keywords: cross-domain authentication; consortium blockchain; ArcFace; biometric

Citation: Chen, X.; Xu, S.; Ma, K.; Chen, P. Cross-Domain Identity Authentication Protocol of Consortium Blockchain Based on Face Recognition. *Information* **2022**, *13*, 535. https://doi.org/10.3390/info13110535

Academic Editors: Weizhi Meng, Zheng Yan and Xiaokang Zhou

Received: 4 October 2022
Accepted: 7 November 2022
Published: 10 November 2022

1. Introduction

Since that network resources are more and more widely distributed, most enterprises have established their own independent resource access policies and access domains. The resources and services provided in a single domain cannot meet all the needs of users. If users need to access the resources and services of other domains, they need to register and log in the target domain to get the access to its authorization. This operation will undoubtedly increase the burden of users. Maintaining many temporary users information in the database of the target domain will also increase the maintenance cost. Cross domain authentication is a technology to break the trust island. It not only enables users to access the resources and services of the relevant domain through the same credentials, but also eliminates the need for domain service providers to maintain the account information of each access user, reducing the communication costs of all parties.

Blockchain technology applies a peer-to-peer distributed ledger based on cryptography, and the data in the growing ledger stored in a chain structure with the characteristics of decentralization, anonymity, traceability, and transparency [1], which can provide trusted data delivery for participants or systems without a mutual trust social relationship. The consortium chain is a special form of blockchain technology, which requires that each new

node joins the chain through the audit of legitimate nodes on the chain, and only authorized nodes can access the chain [2]. The problems of multi-party trusted communication and certificate management in cross domain scenarios can be effectively solved. Usually, many cross-domain systems provide account name–password pairs as identity authentication, but there are enormous security risks in this certification method, and this certification method cannot prevent accounts from stealing attacks. Consortium chain needs apply the user's secret key to represent the complete identity of a user in the identity management, since the key content is difficult to carry and record. Human biometric features are the most distinctive features that represent themselves, such as fingerprints, faces, and other innate features are naturally unique, stable and difficult to imitate [3]. The biometric features have unique advantages in both key generation and management. Among those biometric methods, face authentication technology is the most suitable one as an authentication scheme in cross-domain scenarios because of its accessibility and non-contact nature. However, face features as the unique identity of users have long-term stability and universality. Once the face features are leaked or copied, it will bring incalculable losses to users, so it is crucial to ensure the security and confidentiality of face features information [4].

Aiming at the problems of consortium chain account attacks and face features leaked in cross-domain authentication, biometric codes and consortium chains are combined to make a new cross-domain authentication protocol based on face recognition in this paper, which can enable different domain service providers to achieve mutual trusted authentication in insecure public networks. The main contributions of this paper are as follows: (1) comparing and analyzing existing cross-domain authentication protocols, proposing a new cross-domain identity authentication protocol based on face recognition in the consortium chain, which requires only one registration for full-domain authentication and solves the problem of complicated and inefficient traditional cross-domain authentication process; (2) proposing a method for securely storing biometric codes, in which the biometric information extracted by face recognition algorithms is encrypted using asymmetric key encryption and then stored on the chain, only corresponding users can indirectly access the real biometric information through the authentication service center, solving the privacy and security problems commonly found in biometric authentication; (3) running the final authentication process on the chain as a smart contract, which reduces the authentication burden of the domain server and also eliminates the attacks of malicious nodes on the chain.

2. Related Work

Traditional cross-domain authentication schemes are usually implemented using public key infrastructure (PKI), but there are vast differences in communication, protocols, and services between different domains, resulting in complex identity certificate management and extremely low authentication efficiency. However, the emergence of blockchain brings a new direction in cross-domain authentication. In view of the performance and security problems of the existing centralized cross-domain authentication, Ref. [5] proposed a new network cross-domain authentication scheme based on blockchain called Trustroam. In this scheme, it authenticated users and servers in a distributed and anonymous way, which avoids serious problems such as single point of failure and privacy leakage. Ref. [6] proposed an efficient and secure blockchain-assisted authentication mechanism, which supports the authentication of devices located in different Internet of Things domains. The protocol introduces a consortium blockchain to build trust between different domains and designs an identity management mechanism to keep the authenticated nodes anonymous. In [7], aiming at the cross-domain data access problem of product manufacturing, the author proposed a centralized cloud cross-domain data sharing platform based on blockchain with multiple security gateways. The platform uses blockchain to store information in a centralized cloud that can be audited, and apps or data providers found to be misbehaving can be penalized using smart contracts. Ref. [8] proposes a blockchain-based drone intelligent 5G interconnection cross-domain certification plan for the security and privacy issues between drones. This method uses multiple signatures based on threshold sharing

to build a collaborative domain and combines with smart contracts to certify reliable communication between cross-domain devices. Ref. [9] abstracts a general-purpose universal diagram in the certification relationship between IoT intelligent devices, and then converts the certification problem into a signature transitivity problem with the blockchain. Here, the signature only needs to calculate the signatures and witnesses of the relevant edges, which can effectively reduce the pressure of digital signature authentication. All the above studies use key pair as the unique identification of user identity authentication, and the real identity of the current key user cannot be determined during authentication, resulting in the risk of account attack.

With the enhancement of computer computing capabilities, more and more terminal devices have supported biological characteristics detection, so some researchers have considered introducing biological characteristics to further improve the security of blockchain identity certification [10]. In order to verify user identity, Ref. [11] proposes a new authentication security framework. This framework uses a novel verification secure framework based on fusion algorithm, which combines radio frequency identification (RFID) and finger vein (FV) biometric characteristics to improve the randomness and security of the system, and combines blockchain and steganography technology to ensure the confidentiality, integrity and availability of user information. Ref. [12] Designed a multi-purpose iris authentication system. The system uses homogenic encryption technology to encrypt the iris feature information and save it on the blockchain during the authentication certification and high accuracy. Aiming at the common user privacy problem in the Industrial Internet of Things, Ref. [13] proposed a new intelligent industry identity management system based on blockchain. The system provides participants with anonymous credentials through biometric and fuzzy extractors, and supports selective disclosure, suspension/unfreezing, and revocation of credentials. Aiming at the problems of biometric information leakage risk, unreliable authentication module, and opaque biometric information management in the biometric authentication system, Ref. [14] proposed a biometric authentication system based on blockchain. The system improves the security and reliability of existing biometric authentication systems by fragmenting biometric templates and managing them with the decentralized and tamper-proof mechanism of blockchain. Ref. [15] proposes a blockchain based framework that allows secure, transparent and privacy protected biometric authentication. The framework manages biometric data using distributed DID, and allows users to have autonomous and controllable electronic identities, so that they can fully control their own biometric identity information and ensure the security of user information. In view of the challenges faced by blockchain in storing private files and granting access rights, Ref. [16] proposes a biometric-based blockchain file storage and access authorization scheme. In this scheme, the requests and responses for file storage and access are all executed on the blockchain, and the file owner is not required to store any information locally, so it can be used on devices with limited resources.

In summary, although there are some blockchain-based cross-domain authentication methods, so far less of them can combine security, privacy, versatility, and robustness, making it difficult to apply to complex scenarios in real life. Therefore, it is urgent to research an efficient and universal cross-domain authentication algorithm.

3. Cross Domain Authentication Model

3.1. System Structure

A cross-domain authentication protocol based on face recognition and a consortium chain is designed, which adopts the consortium chain as the basic service; the entire system structure is shown in Figure 1. The system mainly includes the three following major roles: User (U), Certificate Service Center (CA), and Authentication Service Center (AS). Each CA serves each domain separately and is mainly responsible for providing a certificate service for the consortium chain network in the domain, providing the service of issuing, verifying, and revoking the certificate of user and certificate service center. As a part of the core of the consortium chain network, all AS'es in different domains collaborate to process

all the authentication data in the whole consortium chain network, and also runs the core certification program smart contract.

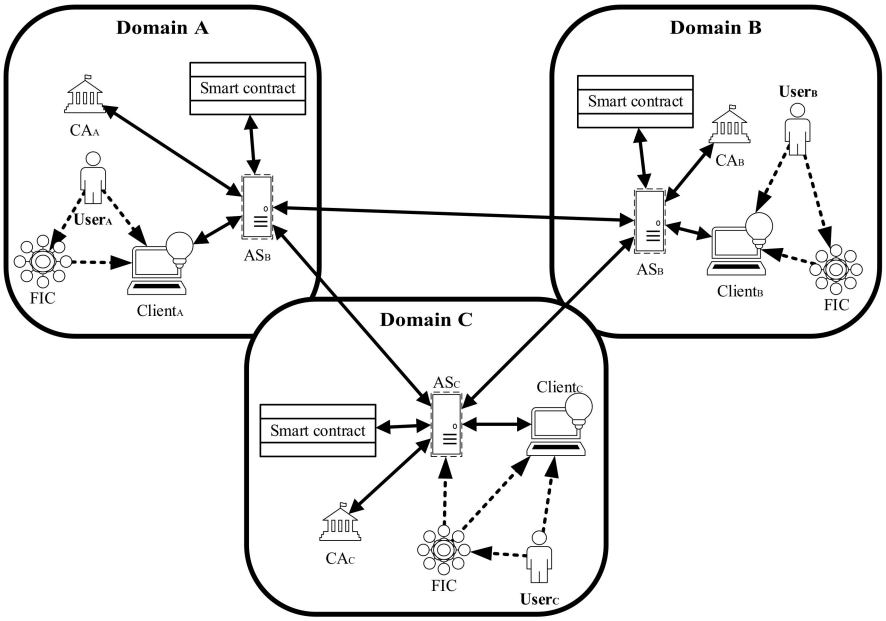

Figure 1. System structure of consortium blockchain. Each domain has an independent user and device, and different domains communicate with each other through the consortium chain.

Besides the above three major roles, this system also includes three important modules: client (C), face-information-collection module (FIC), and smart contract (SC). As the only channel to access the authentication system, the client provides users with a simple visual interface to help complete the identity authentication process, and then provides subsequent system services after the authentication is passed. The face-information-collection module is mainly responsible to collect the primitive face information, which can collect the original information by mobile phone camera, computer camera, and other professional photoing devices. Smart contract is the main authentication program running on the consortium chain maintained by the authentication service center, which mainly includes a feature extraction model and a face recognition algorithm. The face feature extraction model is a model trained by deep learning face recognition technology, which can extract the corresponding biometric code from the primitive face information and store it on the chain. The face feature recognition algorithm is mainly used to compare the registration biometric code and authentication biometric code in the authentication stage, and finally gives the authentication result.

3.2. Face Recognition Model

The identity authentication in the face recognition model comprises two parts: one is the asymmetric key authentication based on the consortium chain, the other is a face recognition model, which includes a deep-learning face feature extraction model and a face feature authentication algorithm. The face feature extraction model takes Deep Residual Networks (ResNet) as the backbone network and ArcFace algorithm [17] as the loss function of the training process. After it completed the training, we deployed the optimal result model on the chain to provide a face feature extraction service for this protocol. Cosine distance measurement is mainly used to realize face feature authentication algorithm, which is mainly used to judge whether the biometric code uploaded during user registration is consistent with the biometric code uploaded during authentication. The face recognition model structure is shown in Figure 2.

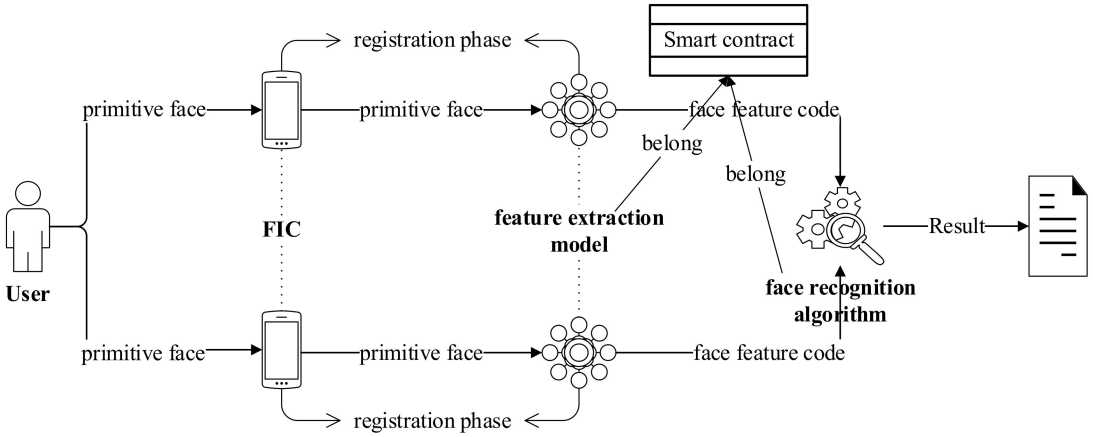

Figure 2. Face recognition model structure for cross-domain blockchain. The primitive face information is collected by FIC module, and then extracted by the face feature extraction module on the AS node. Then the face biometric features are stored in the consortium chain and used in the authentication stage.

(1) Feature Extraction Model

The ResNet model comprises a series of residual units stacked. In the whole blockchain network, some links are usually added so that the upper-layer data can keep features and transmit them directly to the deeper layer. Meanwhile, the new links will not increase the complexity of the model itself, so it can maintain high operation efficiency. A common residual element is shown in Figure 3. Suppose that the input face image x is output $H(x) = F(x) + x$ after the nonlinear mapping of the residual network, in which $H(x)$ is the sum of input x and residual block output $F(x)$. If x is taken as a real value and $H(x)$ as an estimated value, $F(x)$ represents residual value. The specific calculation formula of a residual unit is given as follows:

$$H(x) = F(x, \{w_i\}) + x \tag{1}$$

where x is the input of the residual unit, $H(x)$ is the output of the residual unit, and $F(x, \{w_i\})$ represents the residual function that the network needs to learn. For example, in Figure 3, the residual function is represented as $F = w_2\varphi(w_1 x)$, φ is the activation function, w_1 is the parameter of the first single-layer network, and w_2 is the parameter of the second single-layer network.

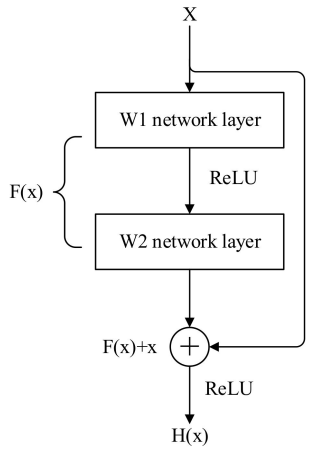

Figure 3. Residual Unit of residual network. It is the smallest unit in the primitive face feature extraction process.

The input face image x will be related with the full connection layer after the residual network processing, and finally the output from multi-dimensional vector to one-dimensional vector, which can further simplify the calculation. Loss functions in many network models are usually implemented Softmax function, but Softmax has difficulty in constraining intra-class distance and inter-class distance, which will reduce the precision of face recognition. So, this paper chooses ArcFace as the loss function, the formula is:

$$L_{ArcFace} = \frac{1}{N}\sum_{i} -\log \frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))} + \sum\limits_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}} \tag{2}$$

where θ_j is the angle between the weight vector w_j and the input vector x_i, s is the normalized result of the input vector x_j, m is the interval between w_{y_i} and x_i, a complete residual network is shown in Figure 4.

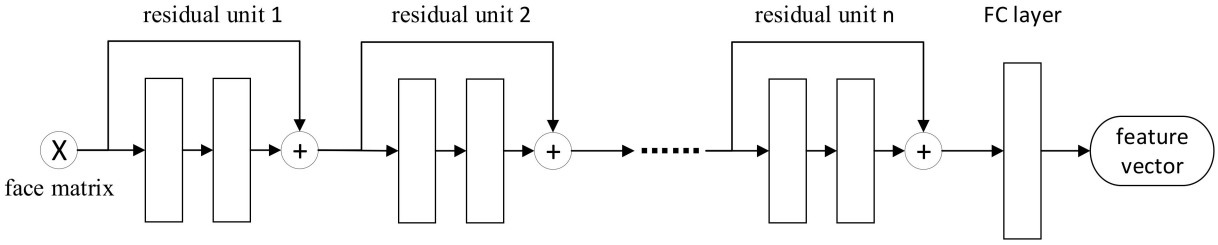

Figure 4. Complete residual network of face recognition. The primitive face information is processed by several residual unit to get the feature matrix.

(2) Face Recognition Algorithm

The face feature authentication algorithm is for comparing the face feature value between the authenticated user and the user on the chain. The judge algorithms include Euclidean measurement, cosine similarity, Chebyshev distance, Pearson correlation coefficient, etc. Considering efficiency and accuracy, cosine similarity is chosen as the face feature authentication algorithm in this paper. Cosine similarity, also known as cosine distance, is used to measure the difference between two individuals based on the cosine value of the angle between two vectors. Suppose that the face feature information stored in the chain by a user during registration is $X = (x_1, x_2 \ldots x_n)$, and the primitive face information got during authentication is $Y = (y_1, y_2 \ldots y_n)$ after processing through the face feature extraction model, the cosine similarity calculation formula is given by:

$$D(X,Y) = \frac{\sum\limits_{i=1}^{n} x_i y_i}{\sqrt{\sum\limits_{i=1}^{n} x_i^2}\sqrt{\sum\limits_{i=1}^{n} y_i^2}} \tag{3}$$

4. Main Protocol Processes

The main protocol processes include three parts: local domain registration, local domain authentication, and cross domain authentication. The specific certification procedures of the cross-domain identity certification agreement is based on face recognition. When a user registers to the system, the user's primitive face information will be sent to a smart contract. The smart contract first calls the ResNet face feature extraction model on it for feature extraction, and then encrypts the extracted person's face features to the chain. In the authentication process, besides verifying the user's private key, the user also needs to send the primitive face to the smart contract. The authentication process first calls the ResNet face feature extraction algorithm on the Authentication Service Center node for feature extraction, and then executes the face feature authentication algorithm on smart contract to compare the extracted feature value with the existing user's feature value on the

chain, and finally draws the conclusion whether the authentication is successful. Table 1 illustrates the specific symbols and meanings used in this agreement.

Table 1. Terminology table.

Notation	Description
U_{i-A}	User i in domain A
AS_A / AS_B	A/B Domain Authentication Service Center
CA_A / CA_B	A/B Domain Certificate Service Center
ID_i	Login ID of user i
PW_i	User i registration password
$P_i / P_i^{'}$	User i registration/authentication phase provided by the primitive face information
T_i	Timestamp during authentication
V_i	Biometrics extracted during registration
$V_i^{'}$	Biometric code at the time of registration after user decryption
V_i^{*}	Biometric codes newly extracted in authentication stage
SK_i / PK_i	Private/Public key of user i
SK_{AS_A} / PK_{AS_A}	Private/public key of A domain authentication service center
$SK_i() / PK_i()$	Encrypt using user i's private/public key
Check()	Parameter validity check
KeyGen()	Key generation function based on ECDSA
BioGet()	Primitive face information collection function
FeatureExtraction()	ResNet face feature extraction model
FaceRecognition()	Cosine similarity facial feature authentication algorithm
ChainBroadcast()	Consortium chain data broadcast function
ChainGet()	Consortium chain data reading function

4.1. Local Domain Registration

The registration process of local domain is shown in Figure 5, which can be divided into three stages: certificate generation stage, face collection stage, and contract registration stage.

1. The fundamental processes of certificate generation are as follows:

Step 1.1: User U_i starts the client and inputs username ID_i, password PW_i, and other necessary information.

Step 1.2: The client generates the current timestamp T_1 and encrypts the registration information (ID_i, PW_i, T_1) with the public key PK_{AS} of the local domain authentication service center AS: $(ID_i, PW_i, T_1) \overset{PK_{AS}}{\rightarrow} PK_{AS}(ID_i, PW_i, T_1)$.

Step 1.3: The client sends the encrypted result $PK_{AS}(ID_i, PW_i, T_1)$ to the local authentication service center AS.

Step 1.4: AS decrypts the registration information with its own private key: $PK_{AS}(ID_i, PW_i, T_1) \overset{SK_{AS}}{\rightarrow} (ID_i, PW_i, T_1)$.

Step 1.5: AS check whether the username ID_i has been registered and whether it meets the user-name specifications. Subsequently, it checks whether the user-password PW_i meets the basic security requirements, and checks if T_i is a valid timestamp lastly. If the timestamp is within three minutes, the request is considered valid and proceeds to the next step.

Step 1.6: AS send registered data (ID_i, PW_i) to the Certificate Service Center CA.

Step 1.7: CA generates the identity certificate (PK_i, SK_i) according to the user information (ID_i, PW_i) through the elliptic curve cryptosystem, and returns it to the local authentication service center AS. Where PK_i is the user's public key and SK_i is the user's private key.

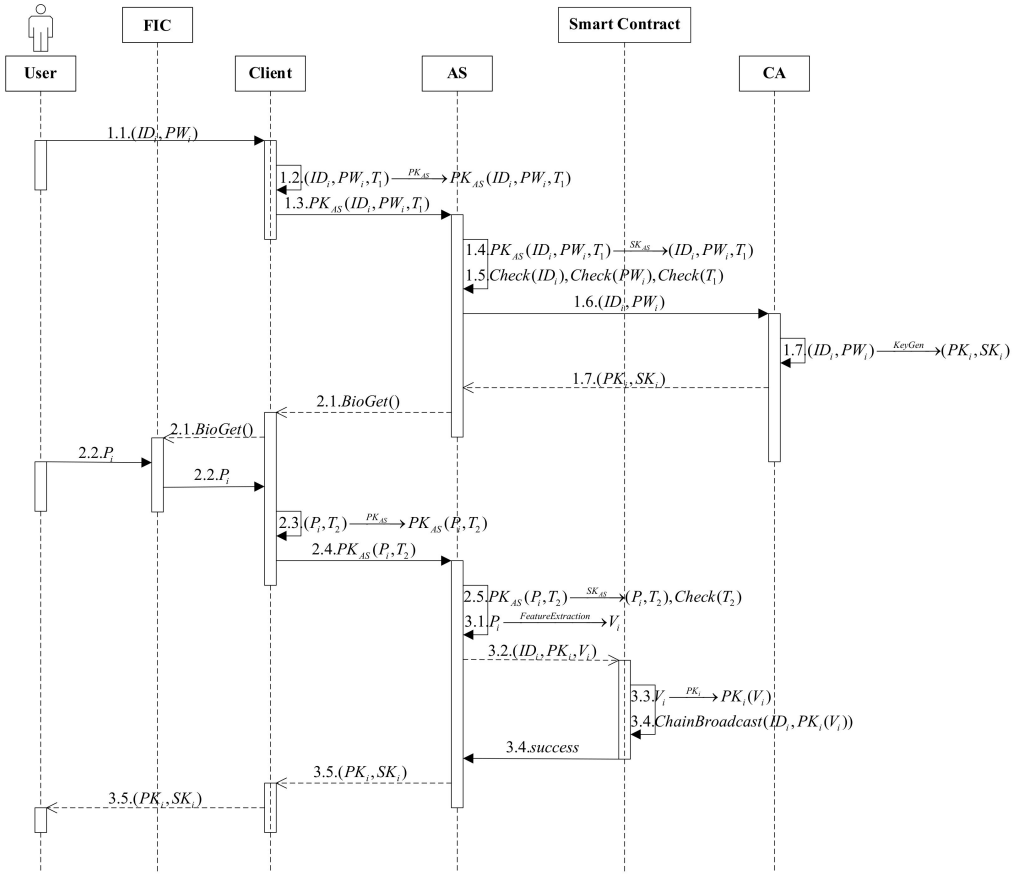

Figure 5. Registration process of local domain. The user sends the registration information to the AS node through the client. As invokes CA to assign a certificate according to registration information, then notifies user to input face information, and finally writes registration information to consortium chain.

2. The face collection stage includes:

Step 2.1: AS stores the received local certificate information temporarily and calls the function BioGet() to notify the client to collect the user's primitive face information.

Step 2.2: The face-information-collection module collects the user's primitive face information P_i through the user's interface and returns it to the client.

Step 2.3: The client generates the current timestamp T_2 and encrypt the information (P_i, T_2) with the public key PK_{AS} of the local domain authentication service center AS: $(P_i, T_2) \overset{PK_{AS}}{\rightarrow} PK_{AS}(P_i, T_2)$.

Step 2.4: The client sends the user encrypted information $PK_{AS}(P_i, T_2)$ to AS.

Step 2.5: AS decrypts the encrypted information $PK_{AS}(P_i, T_2)$ with its own private key SK_{AS}: $PK_{AS}(P_i, T_2) \overset{SK_{AS}}{\rightarrow} (P_i, T_2)$. AS check if T_2 is a valid timestamp. If the timestamp is within three minutes, the request is considered valid and is proceeded to the next step.

3. The contract registration stage includes:

Step 3.1: The Authentication Service Center calls the face feature extraction model to extract the biometric code of face feature P_i: $P_i \overset{\text{FeatureExtraction}}{\rightarrow} V_i$.

Step 3.2: The smart contract on the consortium chain is called for user registration and gets the input parament registration information (ID_i, PK_i, V_i) from AS.

Step 3.3: The smart contract uses user public key PK_i to encrypt biometric code: $V_i \overset{PK_i}{\rightarrow} PK_i(V_i)$.

Step 3.4: The smart contract broadcasts the user's registration information to the entire network: ChainBroadcast$(ID_i, PK_i(V_i))$ and notifies AS the successful registration result.

Step 3.5: The AS sends the user certificate (PK_i, SK_i) to the user to notify that the registration is successful and deletes all local registration information.

4.2. Local Domain Authentication

Figure 6 shows the authentication process in local domain, which can be divided into three stages: certificate generation stage, face collection stage, and contract authentication stage.

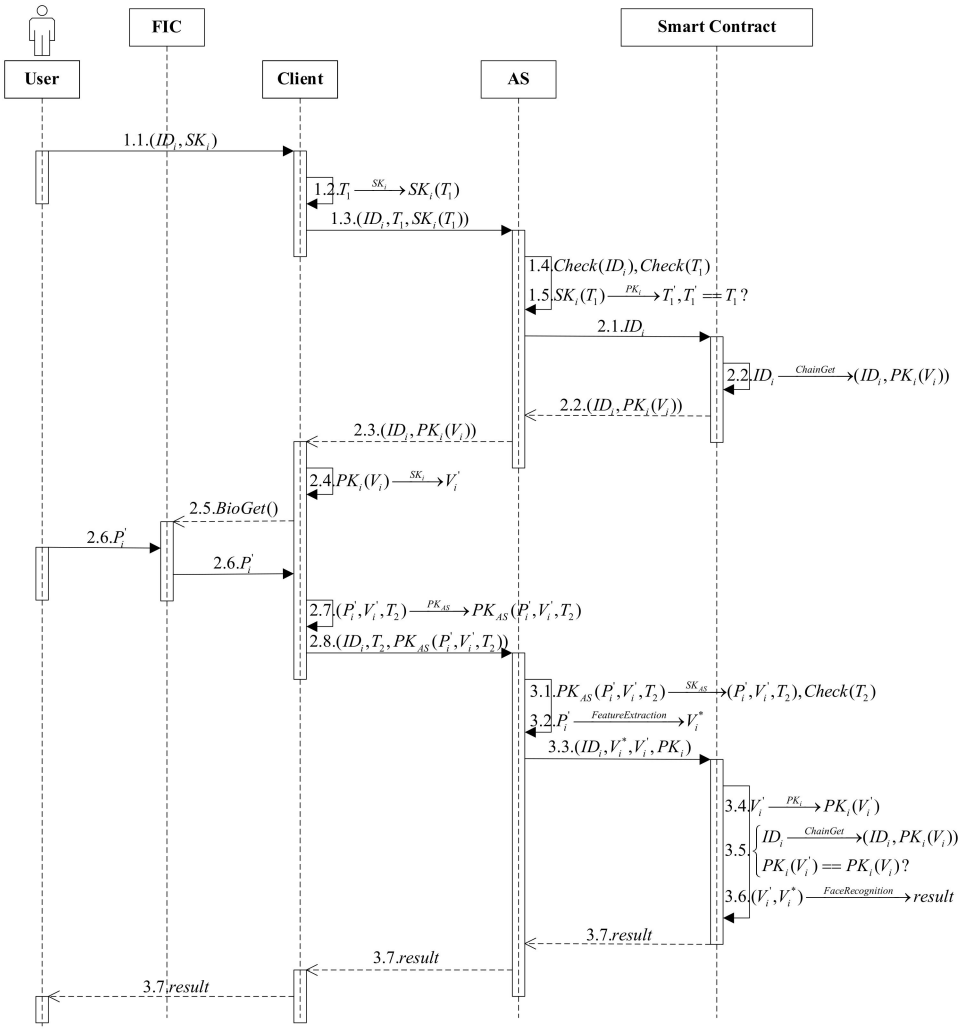

Figure 6. Local domain authentication process. The user sends the login information and face information to the AS node through the client. The AS reads the registration information stored on the consortium chain according to the login information, compares the login information with the registration information, and returns the registration result.

1. The certificate generation stage includes:

Step 1.1: User U_i opens the client, inputs the account ID_i and selects the registered private key SK_i.

Step 1.2: The client generates the current timestamp T_1 and encrypts T_1 using the user's private key SK_i: $T_1 \overset{SK_i}{\rightarrow} SK_i(T_1)$.

Step 1.3: The client sends the certificate authentication information $(ID_i, T_1, SK_i(T_1))$ to the local Authentication Service Center AS.

Step 1.4: AS Checks whether ID_i is a legitimate user. The username must be registered and not revoked. Check if T_1 is a valid timestamp. If the timestamp is within three minutes, the request is considered valid and proceeds to the next step.

Step 1.5: AS uses U_i's public key PK_i to decrypt $SK_i(T_1)$: $SK_i(T_1) \overset{PK_i}{\rightarrow} T_1^{'}$, if $T_1 == T_1^{'}$ is verified and passed.

2. The face collection phase includes:

Step 2.1: AS invokes the smart contract to query face authentication information and passes in parameter ID_i.

Step 2.2: The smart contract calls ChainGet() to get face authentication information $(ID_i, PK_i(V_i))$ according to ID_i and returns the face information to AS.

Step 2.3: AS returns $(ID_i, PK_i(V_i))$ to the client and notifies the client to collect the primitive face information of the user.

Step 2.4: The client uses user's private key SK_i to decrypt $PK_i(V_i)$ to get the face feature code: $PK_i(V_i) \overset{SK_i}{\rightarrow} V_i^{'}$.

Step 2.5: The client invokes the function BioGet() to notify the primitive face-information-collection module to collect the user's primitive face information.

Step 2.6: The face-information-collection module collects the user's primitive face information $P_i^{'}$ from the user and returns it to the client.

Step 2.7: The client generates the current timestamp T_2 and encrypt the authentication information $(P_i^{'}, V_i^{'}, T_2)$ with the public key PK_{AS} of the local domain authentication service center AS: $(P_i^{'}, V_i^{'}, T_2) \overset{PK_{AS}}{\rightarrow} PK_{AS}(P_i^{'}, V_i^{'}, T_2)$.

Step 2.8: The client packages the face authentication request as $(ID_i, T_2, PK_{AS}(P_i^{'}, V_i^{'}, T_2))$ and sends it to AS.

3. The contract certification stage includes:

Step 3.1: AS decrypts the authentication request using its own private key: $PK_{AS}(P_i^{'}, V_i^{'}, T_2) \overset{SK_{AS}}{\rightarrow} (P_i^{'}, V_i^{'}, T_2)$. AS check if T_2 is a valid timestamp. If the timestamp is within three minutes, the request is considered valid and proceeds to the next step.

Step 3.2: AS calls face feature extraction model to extract the biological characteristics of face information $P_i^{'}$: $P_i^{'} \overset{\text{FeatureExtraction}}{\rightarrow} V_i^{*}$.

Step 3.3: AS invokes the smart contract on the consortium chain to authenticate the user's face and input the face authentication information $(ID_i, V_i^{*}, V_i^{'}, PK_i)$.

Step 3.4: Smart contracts use PK_i to encrypt $V_i^{'}$: $V_i^{'} \overset{PK_i}{\rightarrow} PK_i(V_i^{'})$.

Step 3.5: Smart contracts get face authentication information $(ID_i, PK_i(V_i))$ through the function ChainGet(), and then judge whether $PK_i(V_i^{'}) == PK_i(V_i)$. If it is true, the smart contract will continue the next step.

Step 3.6: Smart contract invokes face feature authentication algorithm to judge whether two face features belong to the same person: $(V_i^{'}, V_i^{*}) \overset{\text{FaceRecognition}}{\rightarrow} result$. If the result is *true*, the authentication passes.

Step 3.7: The smart contract notifies both AS and the user of the authentication result.

4.3. Cross-Domain Authentication

Figure 7 shows the cross-domain authentication process, which can be divided into four stages: the local certificate authentication stage, local face collection stage, cross-domain certificate authentication stage and contract face authentication stage.

1. Certificate authentication in the local domain includes:

Step 1.1: User U_{i-A} of domain A opens the client, inputs the account ID_i and selects the registered private key SK_i.

Step 1.2: The client in domain A generates the current timestamp T_1 and encrypts T_1 with the private key SK_i: $T_1 \overset{SK_i}{\rightarrow} SK_i(T_1)$.

Step 1.3: The client sends authentication information $(ID_i, T_1, SK_i(T_1))$ to the authentication service center AS_A of the consortium chain in domain A.

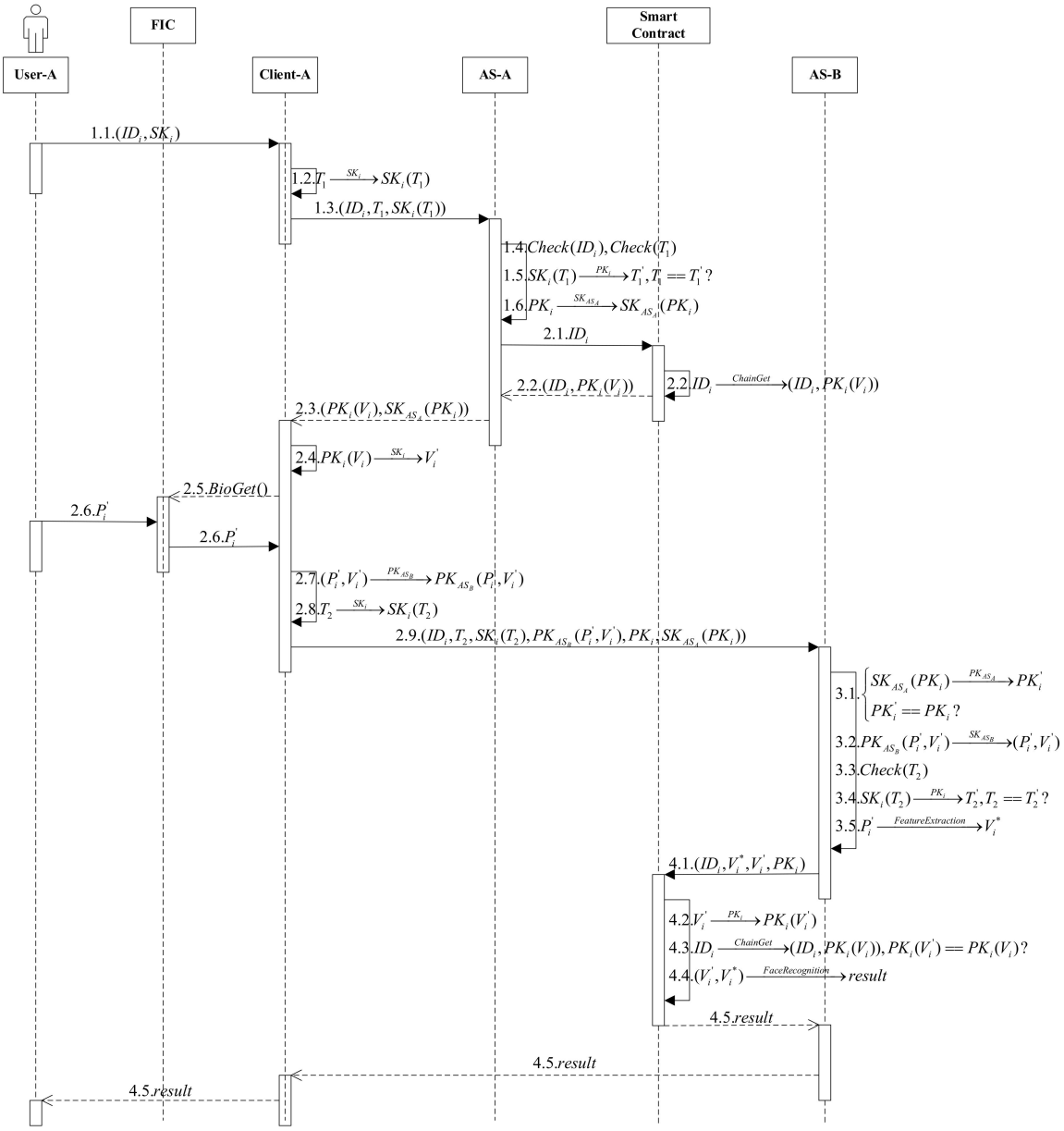

Figure 7. Cross-domain authentication process. The user sends the cross-domain authentication request to the local domain AS node, and the local AS node signs the user certificate and returns. Then the user sends the face information and the certificate signature to the target domain AS for cross-domain authentication. The target domain authenticates the user's certificate signature and the face information and finally returns the authentication result.

Step 1.4: AS_A check whether ID_i is a legitimate user. The username must be registered and not revoked. Check if T_1 is a valid timestamp. If the timestamp is within three minutes, the request is considered valid and proceeds to the next step.

Step 1.5: AS_A decrypts $SK_i(T_1)$ using ID_i's public key PK_i: $SK_i(T_1) \xrightarrow{PK_i} T_1'$, and check if $T_1 == T_1'$. The equal value means the timestamp with user's signature is accepted, then the domain certificate of the user is authenticated.

Step 1.6: AS_A generates a unique endorsement for the user. AS_A encrypts the public key of User U_{i-A} with its private key SK_{AS_A}: $PK_i \xrightarrow{SK_{AS_A}} SK_{AS_A}(PK_i)$, and takes the encrypted result as user's endorsement.

2. The local domain face collection stage includes:

Step 2.1: AS_A invokes the smart contract to query face authentication information and passes in parameter ID_i.

Step 2.2: Smart contract gets face authentication information $(ID_i, PK_i(V_i))$ by calling the function ChainGet(), and returns human face information to AS_A.

Step 2.3: AS_A package the face authentication information and endorsement encryption information of the user on the chain as $(PK_i(V_i), SK_{AS_A}(PK_i))$, and send the package to the client.

Step 2.4: The client uses user's private key SK_i to decrypt $PK_i(V_i)$ to get the face feature code: $PK_i(V_i) \overset{SK_i}{\rightarrow} V_i^{'}$.

Step 2.5: The client invokes the function BioGet() to notify the primitive face-information-collection module to collect the user's primitive face information.

Step 2.6: The above face-information-collection module collects the user's primitive face information $P_i^{'}$ from the user and returns the information to the client.

Step 2.7: The client uses the public key PK_{AS_B} of authentication service center AS_B in domain B to encrypt $P_i^{'}$ and $V_i^{'}$: $(P_i^{'}, V_i^{'}) \overset{PK_{AS_B}}{\rightarrow} PK_{AS_B}(P_i^{'}, V_i^{'})$.

Step 2.8: The client generates the current timestamp T_2 and encrypts T_2 with the private key SK_i: $T_2 \overset{SK_i}{\rightarrow} SK_i(T_2)$.

Step 2.9: The client packages the cross-domain authentication request as $(ID_i, T_2, SK_i(T_2), PK_{AS_B}(P_i^{'}, V_i^{'}), PK_i, SK_{AS_A}(PK_i))$ and sends it to AS_B.

3. Cross-domain certificate authentication includes:

Step 3.1: After receiving the authentication data, the authentication service center AS_B in domain B decrypts $SK_{AS_A}(PK_i)$ with the public key of the authentication service center AS_A in domain A: $SK_{AS_A}(PK_i) \overset{PK_{AS_A}}{\rightarrow} PK_i^{'}$, if $PK_i^{'} == PK_i$, the procedure continues to the next step.

Step 3.2: AS_B decrypts $PK_{AS_B}(P_i^{'}, V_i^{'})$ with its own private key: $PK_{AS_B}(P_i^{'}, V_i^{'}) \overset{SK_{AS_B}}{\rightarrow} (P_i^{'}, V_i^{'})$.

Step 3.3: AS_B checks whether T_2 is a valid timestamp. If the value is a timestamp within three minutes, the request is considered valid and proceeds to the next step.

Step 3.4: AS_B uses the received user public key PK_i to decrypt $SK_i(T_2)$: $SK_i(T_2) \overset{PK_i}{\rightarrow} T_2^{'}$. If $T_2 == T_2^{'}$, the cross-domain certificate authentication succeeds.

Step 3.5: The AS_B invokes the face feature extraction model to extract the biometric code V_i^* of face feature: $P_i^{'} \overset{\text{FeatureExtraction}}{\rightarrow} V_i^*$.

4. The contract face authentication stage includes:

Step 4.1: AS_B invokes the smart contract on the consortium chain to authenticate the user's face and input the face authentication information $(ID_i, V_i^*, V_i^{'}, PK_i)$.

Step 4.2: Smart contracts use PK_i encryption $V_i^{'}$: $V_i^{'} \overset{PK_i}{\rightarrow} PK_i(V_i^{'})$.

Step 4.3: The smart contract gets the face authentication information $(ID_i, PK_i(V_i))$ through the function ChainGet(), and then judges whether $PK_i(V_i^{'}) == PK_i(V_i)$. If it is true, the smart contract will continue the next step.

Step 4.4: The smart contract invokes the face feature authentication algorithm to judge whether the two face features belong to the same person: $(V_i^{'}, V_i^*) \overset{\text{FaceRecognition}}{\rightarrow} result$, if the result is *true*, the authentication passes; otherwise, the authentication fails, and the contract face authentication phase ends.

Step 4.5: The smart contract notifies both AS_B and the user of the authentication results.

5. Security Analysis

In order to prove the effectiveness of the cross-domain identity authentication protocol based on face recognition, this section analyzes the security of the protocol in the specific application process and compares it with three typical schemes cited in [6,18,19].

5.1. Attack Models

In the proposed face recognition-based consortium chain cross-domain authentication protocol, we adopt the following assumptions to analyze the attacks on the system.

(1) The consortium blockchain is jointly maintained by multiple authorized nodes throughout the network, and transaction data and smart contract data are transparent to all taking part nodes.
(2) Each node in the same consortium can always synchronize information according to the protocol Raft in [20].
(3) The environment of the authorized nodes in the consortium chain is relatively secure.
(4) Attackers can intercept and modify communication data through network lines to client or authorized nodes.

Based on the above assumptions, the system may be attacked as follows:

(1) Man-in-the-Middle attack: an attacker captures communication data and uses a false identity to deceive both parties
(2) Replay Attack: the attacker intercepts and saves the normal communication data and sends the data to the service node again after the normal communication finishes.
(3) Account Theft Attack: an attacker illegally gets a user identity key file through technical means and attempts to use the file to use the corresponding user's identity.
(4) Biometric Confidentiality: the attacker attempts to obtain the biometric information of other users from the authorization node of the consortium chain.

5.2. Security Analysis

Based on the above security risks, we conduct a security analysis of this system.

5.2.1. Man-in-the-Middle Attack

At each stage of the system, the key privacy information involved mainly includes account password PW_i, account private key SK_i, primitive face information P_i, and face biometric code V_i. Each time the client communicates with the authentication service center AS, the key privacy information transmitted will be encrypted with the other party's public key. The encryption process is carried out locally without the risk of a man-in-the-middle attack. Suppose that the key privacy information $PK_{AS}(ID_i, PW_i, P_i, V_i)$ transmitted by the client in the communication process with AS is eavesdropped and intercepted by the intermediary through hacker technology. Because of the asymmetric encryption feature, the content encrypted by the public key can only be decrypted by the corresponding private key, the intermediary cannot get the specific privacy content. Therefore, all stages of this protocol can effectively prevent man-in-the-middle attacks.

5.2.2. Replay Attack

In the process of local domain identity authentication and cross domain identity authentication of the system, the client will generate the request parameter T_i according to the current timestamp of the system, and then encrypt T_i to $SK_i(T_i)$ with the user's private key. The timestamp information will be attached to each stage of communication with the authentication service center AS. AS will decrypt $SK_i(T_i)$ with the user's public key after receiving the authentication request, and then check whether the timestamp is within the current time range. If the value is a timestamp within three minutes, the request is considered valid and proceed to the next step. Suppose an attacker intercepts a request $SK_i(T_i)$ with an encrypted timestamp. Since the attacker does not have a user's private key, the timestamp content in the request cannot be changed. Even if the attacker sends the request directly to the server, the request will be considered invalid because the timestamp expires. The replay attack failed.

5.2.3. Account Theft Attacks

In the process of local and cross-domain identity authentication of the system, the authentication service AS will check the identity certificate of the current authenticated user. AS will notify the user to cooperate in collecting face information $P_i^{'}$ after the certificate is checked. Then, the smart contract invokes the face feature extraction model to extract the biometric code V_i^* of face information, and finally compare whether the face features V_i^* at the time of authentication are consistent with the face features V_i saved on the chain. If these are consistent, the authentication passes. Suppose that the attacker robs a user's identity key pair by illegal methods and wanted to use his identity to log in the system to steal the user's property. The attacker can successfully pass the certificate authentication stage of the authentication process by using the user ID_i and private key SK_i. However, the attacker cannot provide the primitive face information needed for authentication in the face collection stage, which leads to the final identity authentication failure. Therefore, this protocol can prevent account theft attacks.

5.2.4. Internal Attack

Internal attacks may mainly come from two aspects: system malicious users and authentication service centers. If a malicious user is successfully registered with the system, it can access all the local domain authentication and cross-domain authentication process of the system. However, since the user can only access their own data during the entire process of identity authentication, so it cannot pose any threat to the system. As for the malicious authentication service center, it may maliciously tamper with the data and authentication results on the consortium chain. However, because the network on the consortium chain is maintained by multiple authentication services using the Raft protocol, tampering by a single malicious node can be quickly detected and processed. In conclusion, this protocol can effectively prevent internal attacks.

5.2.5. Biometric Confidentiality

In this protocol, all information related to user biological information is encrypted, and the biological information can only be viewed by the user. For example, in the face collection phase of local domain identity registration, the client will encrypt it with the public key of the AS to get $PK_{AS}(P_i)$ after the client collects the primitive face information P_i. Only the private key corresponding to the public key can decrypt and read the information, without risk of disclosure. In the contract registration phase of local identity registration, the smart contract uses the feature extraction model to extract the biometric code V_i of the primitive face information P_i, and then uses the user's public key PK_i to store the encrypted result $PK_i(V_i)$ in the chain. Since then, all other participants in the system except for the user himself cannot get the biometric code of the user. Therefore, the security of the user's face features can be ensured.

According to the comparison results in Table 2, this protocol is more secure than other schemes and can resist more complex network attacks.

Table 2. Comparison of protocol security.

Security Performance	Our Protocol	[6]	[18]	[19]
Resist man-in-the-middle attack	√	√	×	√
Cross-domain authentication	√	√	√	×
Resist replay attack	√	√	×	√
Resist account theft attacks	√	×	×	×
Resist internal attack	√	×	√	√
Biometric confidentiality	√	×	×	×

6. Conclusions

In order to solve the problems of complex identity certificate management, low authentication efficiency, and being unable to prevent account theft in distributed scenarios, this paper proposes a cross-domain identity authentication protocol based on face recognition. The protocol uses consortium chain as the underlying architecture to ensure the stability and decentralization of authentication services. ResNet is a face feature extraction model for face authentication, which is deployed on intelligent contract to ensure the security and non-repudiation of privacy information such as biometrics. Then through security analysis, it is proved that the protocol can effectively prevent man-in-the-middle attacks, replay attacks, account embezzlement, internal attacks, and other attacks, and finally achieve mutual security authentication between different domains.

Author Contributions: Conceptualization, S.X. and X.C.; methodology, X.C.; validation, X.C.; formal analysis, K.M.; supervision, P.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Major Science and Technology Projects in Hubei Province of China (Grant No. 2020AEA012).

Data Availability Statement: The data included in this study are available upon request by contact with the first author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Taylor, P.J.; Dargahi, T.; Dehghantanha, A.; Parizi, R.M.; Choo, K.-K.R. A systematic literature review of blockchain cyber security. *Digit. Commun. Netw.* **2020**, *6*, 147–156. [CrossRef]
2. Zheng, P.; Xu, Q.; Zheng, Z.; Zhou, Z.; Yan, Y.; Zhang, H. Meepo: Sharded Consortium Blockchain. In Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE), Chania, Greece, 19–22 April 2021; pp. 1847–1852.
3. Das, R.; Piciucco, E.; Maiorana, E.; Campisi, P. Convolutional Neural Network for Finger-Vein-Based Biometric Identification. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 360–373. [CrossRef]
4. Meden, B.; Rot, P.; Terhörst, P.; Damer, N.; Kuijper, A.; Scheirer, W.J.; Ross, A.; Peer, P.; Štruc, V. Privacy–Enhancing Face Biometrics: A Comprehensive Survey. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 4147–4183. [CrossRef]
5. Li, C.; Wu, Q.; Li, H.; Liu, J. Trustroam: A Novel Blockchain-Based Cross-Domain Authentication Scheme for Wi-Fi Access. In *Proceedings of the 14th International Conference on Wireless Algorithms, Systems, and Applications*; Springer: Cham, Switzerland, 2019; Volume 11604 LNCS, pp. 149–161.
6. Shen, M.; Liu, H.; Zhu, L.; Xu, K.; Yu, H.; Du, X.; Guizani, M. Blockchain-Assisted Secure Device Authentication for Cross-Domain Industrial IoT. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 942–954. [CrossRef]
7. Singh, P.; Masud, M.; Hossain, M.S.; Kaur, A. Cross-domain secure data sharing using blockchain for industrial IoT. *J. Parallel Distrib. Comput.* **2021**, *156*, 176–184. [CrossRef]
8. Feng, C.; Liu, B.; Guo, Z.; Yu, K.; Qin, Z.; Choo, K.-K.R. Blockchain-Based Cross-Domain Authentication for Intelligent 5G-Enabled Internet of Drones. *IEEE Internet Things J.* **2022**, *9*, 6224–6238. [CrossRef]
9. Wang, L.; Tian, Y.; Zhang, D. Toward Cross-Domain Dynamic Accumulator Authentication Based on Blockchain in Internet of Things. *IEEE Trans. Ind. Inform.* **2022**, *18*, 2858–2867. [CrossRef]
10. Delgado-Mohatar, O.; Fierrez, J.; Tolosana, R.; Vera-Rodriguez, R. Biometric Template Storage with Blockchain: A First Look Into Cost and Performance Tradeoffs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2829–2837.
11. Mohsin, A.H.; Zaidan, A.A.; Zaidan, B.B.; Albahri, O.S.; Albahri, A.S.; Alsalem, M.A.; Mohammed, K.I. Based blockchain-PSO-AES techniques in finger vein biometrics: A novel verification secure framework for patient authentication. *Comput. Stand. Interfaces* **2019**, *66*, 103343. [CrossRef]
12. Mahesh Kumar, M.; Prasad, M.V.N.K.; Raju, U.S.N. BMIAE: Blockchain-based multi-instance Iris authentication using additive ElGamal homomorphic encryption. *IET Biom.* **2020**, *9*, 165–177. [CrossRef]
13. Sarier, N.D. Efficient biometric-based identity management on the Blockchain for smart industrial applications. *Pervasive Mob. Comput.* **2021**, *71*, 101322. [CrossRef]
14. Lee, Y.K.; Jeong, J. Securing biometric authentication system using blockchain. *ICT Express* **2021**, *7*, 322–326. [CrossRef]
15. Mishra, P.; Modanwal, V.; Kaur, H.; Varshney, G. Pseudo-Biometric Identity Framework: Achieving Self-Sovereignity for Biometrics on Blockchain. In Proceedings of the 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne, Australia, 17–20 October 2021; pp. 945–951.

16. Ma, J.; Qi, B.; Lv, K. BSA: Enabling Biometric-Based Storage and Authorization on Blockchain. In Proceedings of the 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Shenyang, China, 20–22 October 2021; pp. 1077–1084.
17. Deng, J.; Guo, J.; Yang, J.; Xue, N.; Cotsia, I.; Zafeiriou, S.P. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 5962–5979. [CrossRef] [PubMed]
18. Guo, S.; Wang, F.; Zhang, N.; Qi, F.; Qiu, X. Master-slave chain based trusted cross-domain authentication mechanism in IoT. *J. Netw. Comput. Appl.* **2020**, *172*, 102812. [CrossRef]
19. Zhang, L.; Wu, Q.; Domingo-Ferrer, J.; Qin, B.; Hu, C. Distributed Aggregate Privacy-Preserving Authentication in VANETs. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 516–526. [CrossRef]
20. Zhang, Y.; Han, B.; Zhang, Z.-L.; Gopalakrishnan, V. Network-Assisted Raft Consensus Algorithm. In *Proceedings of the SIGCOMM Posters and Demos*; Association for Computing Machinery: New York, NY, USA, 2017; pp. 94–96.