Journal of Multimedia Information System VOL. 11, NO. 1, March 2024 (pp. 17-34): ISSN 2383-7632 (Online)
https://doi.org/10.33851/JMIS.2024.11.1.17

Privacy-Preserving Decentralized Biometric Identity Verification in Car-Sharing System

Saprunov Vadim[1], Muhammad Firdaus[2], Kyung-Hyune Rhee[3*]

Abstract

The growing popularity of car-sharing applications underscores the need for robust security and privacy measures in user authentication. Traditional methods such as passwords and physical keys prove inadequate for safeguarding user privacy, securing data, and verifying the identity of the authorized driver. This paper introduces a decentralized biometric authentication system to address these challenges. The system primarily employs FaceNet for face recognition, effectively capturing facial features. A key feature is the implementation of self-sovereign identity (SSI), which grants users complete control over their personal identification data. The proposed system utilizes an off-chain approach with the InterPlanetary File System (IPFS) to store biometric data, promoting decentralization and reducing storage costs. Blockchain technology securely links to this data, employing decentralized identifiers (DIDs) and content identifiers (CIDs) to enhance traceability and data security. Additionally, the system incorporates fully homomorphic encryption (FHE), ensuring the safety and privacy of biometric data, even when uploaded to public platforms. Aligned with the general data protection regulation (GDPR), this system enhances security and privacy for contactless car-sharing applications. While initially designed for car sharing, its innovative approach holds potential for other industries requiring secure authentication methods.

Key Words: Biometric Authentication, FaceNet, FHE, DIDs.

I. INTRODUCTION

In recent years, the car-sharing platforms emerging as popular and sustainable transportation alternatives. However, traditional authentication methods, such as passwords or physical keys, have proven susceptible to various security risks. For example, they make it impossible to confirm the physical presence of a certified driver in the car and offer limited user control over personal data. Consequently, the identity fraud in a car-sharing environment is a real problem [1]. On the other hand, verifying user identity is a complex issue, not only in car-sharing environments but also in various other contexts. The advancement of biometric technology has facilitated its wide applications in law enforcement, border control, healthcare, and financial identification and verification. Due to the peculiarity of biometric features, such as their immutability, permanence, and uniqueness, ensuring security and privacy is crucial to maintaining integrity, reliability, and availability in biometric-related applications [2]. These contexts all require reliable and secure user verification methods.

Biometric authentication, which relies on unique physiological or behavioral characteristics, offers a more secure and convenient alternative. The uniqueness of biometric identifiers is intrinsically tied to an individual's identity and is distinct for every person. If biometric data are compromised in one system, all other systems relying on the same biometric information are at risk. This could potentially lead to widespread identity theft and significant data security breaches. Hence, despite the advantages offered by biometric systems, this is one of the significant vulnerabilities [2]. Additionally, the European Data Protection Supervisor has noted that centralized databases, as opposed to decentralized ones, inherently raise the risk of misuse. Such central storage systems also tend to attract intentions to repurpose the data for ends not originally intended, thus posing challenges to privacy preservation [3].

To address the above issues, we propose a decentralized framework for ensuring privacy in a biometric identity verification system. The development of a decentralized biometric authentication system would significantly improve the authentication landscape in car-sharing applications.

Manuscript received February 08, 2024; Revised February 29, 2024; Accepted March 09, 2024. (ID No. JMIS-24M-02-002)

Corresponding Author (*): Kyung-Hyune Rhee, +82-51-629-6247, khrhee@pknu.ac.kr
[1]Department of Artificial Intelligence Convergence, Pukyong National University, Busan, Korea, nicknosok@gmail.com
[2]Department of Information Security, Pukyong National University, Busan, Korea, mfirdaus@pukyong.ac.kr
[3]Division of Computer Engineering and AI, Pukyong National University, Busan, Korea, khrhee@pknu.ac.kr

This system aims to foster trust, enhance user experience, and promote the adoption of sustainable and reliable transportation solutions compared to traditional authentication methods, such as Know Your Customer (KYC) systems. KYC processes heavily rely on identity management, and current systems vary from service to service, requiring users to provide personal data and complete the KYC procedure each time they register with a new service. This makes current KYC systems time-consuming, increases the risk of data breaches, lacks a unified or standard solution, and is expensive for companies that must use different services.

In our system, we use blockchain technology in supporting multi-party processes where members do not trust each other [4-5]. Blockchain technology stands as a decentralized and tamper-proof ledger system, which came to prominence with the advent of Bitcoin in 2009 [6]. It answered a longstanding question within the cryptographic community on achieving consensus on financial transactions in a distributed manner without central oversight. Biometric technology, in contrast, focuses on verifying identities using unique physical (like facial features or fingerprints) or behavioral (such as voice patterns or handwriting) characteristics. The fusion of biometrics with blockchain could significantly enhance distributed digital identity systems built upon blockchain technology, introducing a plethora of new applications. For instance, biometrics could refine the authentication processes for smart devices — objects that interact with blockchains to perform actions or make decisions based on the recorded data. A practical example is a vehicle that can be rented or purchased via a smart contract on a blockchain, where the challenge remains to effectively identify the user for such transactions [7].

However, while blockchain provides excellent architecture and practical tools for securing and managing sensitive and private data stored in biometric templates, several limitations exist. These include low transaction processing capacity (around tens of transactions per second), the need to store all system transactions, leading to rapid growth in the required storage space, and insufficiently studied robustness against different types of attacks. Additional limitations encompass the economic cost of executing smart contracts, privacy concerns, and challenges related to scalability and processing capability [7].

To address these issues, we propose a user-centric and privacy-preserving authentication framework that leverages SSI, DIDs, off-chain storage, and Homomorphic encryption. Combining blockchain and biometrics could be beneficial for novel applications in biometrics, such as the PKI mechanism, distributed trusted service, and identity management. It is argued that while blockchain provides immutability, accountability, audibility, and availability to biometric solutions, biometrics can augment blockchain by offering a secure digital identity model to access digital assets or introducing biometric wallets. Creating a digital identity on blockchain enables individuals to control who and how to access their personal information easily. During past years, identity management has been the second most popular topic for blockchain and biometrics integration [3]. In summary, the primary objectives of this research are:

1. Investigating the existing authentication methods in car-sharing platforms and identifying their limitations concerning privacy, security, and user control.
2. Exploring the potential of decentralized identity management systems, specifically DIDs, in enhancing privacy and security in biometric authentication.
3. Examining the role of off-chain storage solutions and Homomorphic Encryption in protecting user privacy and ensuring secure identity verification.
4. Designing a decentralized biometric authentication system that addresses the identified limitations and incorporates user-centric identity management, GDPR-compliant data handling, and artificial intelligence (AI)-driven biometric verification.

This paper is organized as follows: Section 2 discusses the technologies used in this approach and the legal aspects. Section 3 introduces our system and its algorithms. Section 4 covers the implementation and evaluation, including simulation results for better understanding. Finally, Section 5 concludes this paper.

II. PRELIMINARY

2.1. KYC Systems and Their Application in Contactless Car-Sharing Services

In the real world, we rely on governments to issue identity credentials in the form of physical representations such as passports, birth certificates, and driver's licenses. These documents contain information that is kept by a central authority, usually in protected digital storage. However, we use physical copies of these documents independently of those centralized systems for identification purposes because of their convenience and portability. For example, we can open a bank account or travel on an airline by using only our driver's license as it represents a verified set of data about us [8]. However, in a digital world, people tend to use KYC systems to identify themselves. Customer identification, authentication, and verification form the foundation of KYC procedures, which are essential for many businesses, including banking, telecommunications, and shared services like car rentals [9]. The KYC process is designed to prevent identity theft, financial fraud, money laundering and terrorist financing.

However, existing KYC processes are often seen as in-

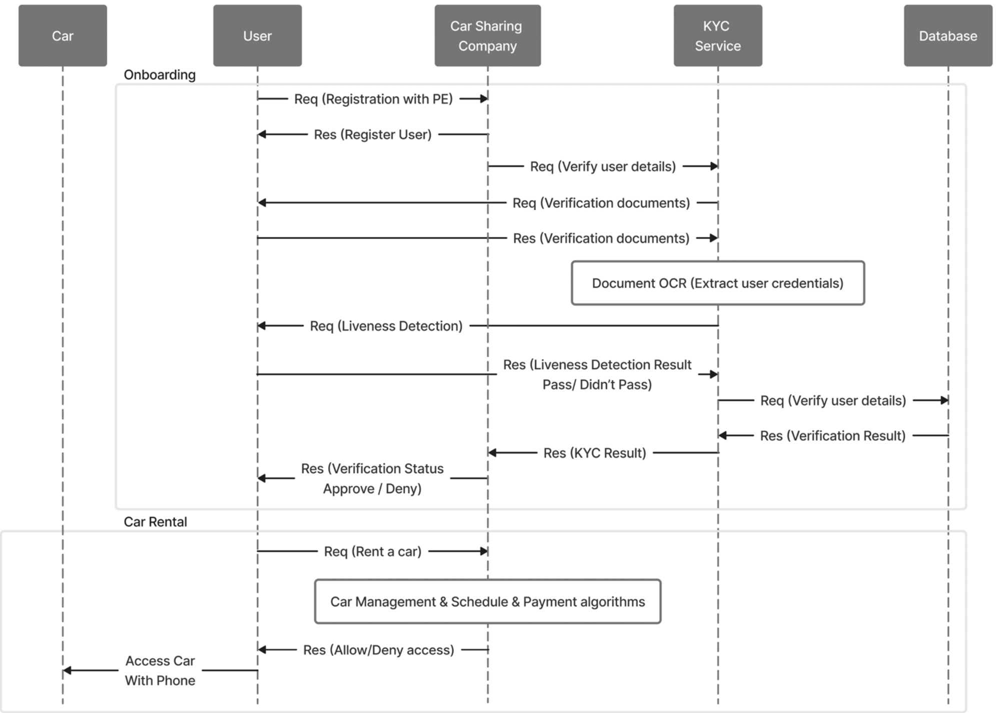

Car
User
Car Sharing Company
KYC Service
Database
Onboarding
Req (Registration with PE)
Res (Register User)
Req (Verify user details)
Req (Verification documents)
Res (Verification documents)
Document OCR (Extract user credentials)
Req (Liveness Detection)
Res (Liveness Detection Result Pass/ Didn't Pass)
Req (Verify user details)
Res (Verification Result)
Res (KYC Result)
Res (Verification Status Approve / Deny)
Car Rental
Req (Rent a car)
Car Management & Schedule & Payment algorithms
Res (Allow/Deny access)
Access Car With Phone

Fig. 1. Sequence diagram of the current contactless car-sharing system.

trusive, time-consuming, and prone to errors. A major concern is that they typically require customers to share sensitive personal information, posing privacy risks. Additionally, traditional KYC procedures are often manual, which can lead to human errors and inefficiencies [10]. An innovative area where efficient and reliable KYC procedures are critical is in the field of contactless car-sharing applications. These applications allow users to rent cars for short periods, often by the minute. The process is entirely app-based, with users locating, unlocking, and starting cars directly from the app [11]. To ensure secure transactions and prevent unauthorized usage, car-sharing services must implement robust KYC procedures. When a user signs up, they provide their personal details and driver's license information. The system then verifies this information against various databases to ensure its accuracy and validity. Fig. 1 shows the sequence diagram of the current contactless car-sharing system.

Current authentication systems used in contactless car-sharing environments typically authenticate users during enrollment. Mobile app authentication requires users to authenticate themselves through a dedicated car-sharing mobile app, which communicates with the vehicle to unlock the doors and start the engine. However, after a user is enrolled and has booked a car, such applications do not verify the driver's identity, and they have no ability to do so. This can lead to identity fraud, as the car may be booked by a certified driver, but the actual driver may not have a valid driver's license.

2.2. Self-Sovereign Identity (SSI), Decentralized Identifiers (DID), and Verifiable Credentials (VC)

In a digital era, the management of digital identities has become a critical issue. Traditional centralized identity systems pose significant privacy and security concerns, resulting in a growing interest in decentralized identity management systems. SSI, DID, and VC are innovative technologies that could revolutionize the way we manage and verify identities in the digital world. SSI is a concept that allows individuals or organizations to have complete control over their digital identities [12]. Unlike traditional identity systems, where identities are managed by a central authority, in an SSI model, individuals hold their identities and control how their personal information is shared and used. A crucial element of SSI is the DID, a type of identifier that is self-created, globally unique, and not reliant on any centralized registry, identity provider, or certificate authority [13]. DIDs are stored on a decentralized network, such as a blockchain, and can be resolved to DID documents that contain public key material, authentication descriptors, and service endpoints, enabling secure, privacy-preserving communication and interaction. VCs, on the other hand, are digital equivalents of physical credentials, such as driver's licenses, passports, or university degrees [14]. They are tamper-evident and cryptographically verifiable, meaning they can be trusted to be genuine and not manipulated. The

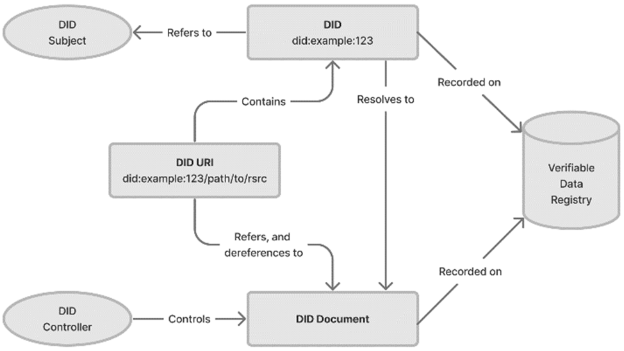


Fig. 2. The traditional DID scheme.

issuer of a VC can be independently verified, and the holder of the VC can prove they are the legitimate owner without exposing the actual data contained in the credential.

These technologies can provide significant benefits. They offer enhanced privacy, as users can control who accesses their data and for what purpose. They improve security by eliminating the need for centralized databases that can be targeted by hackers. One potential application of these technologies is in KYC processes. An individual could hold a VC that proves their identity, issued by a trusted authority. When they need to prove their identity, they present the VC, which can be cryptographically verified by the relying party. The individual doesn't need to provide any additional personal data, and the relying party doesn't need to conduct any further checks. This could streamline KYC processes, reduce privacy risks, and provide a better user experience.

Fig. 2 shows the traditional DID scheme. This structure benefits the biometric system by allowing full control of identifiers of the subject with an individual biometric identifier, independent of different heterogeneous providers. Thus, biometrics could significantly improve currently distributed digital identity schemes based on the blockchain. In terms of on-chain challenges, DIDs help transfer most of the processing load from the main blockchain to the off-chain network. A DID document consists of three fundamental parts: an authentication key, an authentication method, and a service endpoint. Whoever has access to the key is entitled to access the DID document. Therefore, we suggest future research use biometric data only in combination with similar specifications, i.e., off-chain, minimizing both the quantity and sensitivity of personal data stored on-chain.

2.3. Blockchain and IPFS (InterPlanetary File System)

Blockchain technology, initially conceived for the cryptocurrency Bitcoin, has been recognized for its potential to transform various industries [15]. A blockchain is a distributed ledger that allows multiple parties to reach a consensus on a single version of shared digital history. Its immutable nature and transparency make it a highly reliable system for many applications, including authentication and verification [16]. In the context of digital identity, blockchain can be used to store DIDs and their associated public keys securely. This allows individuals to prove their identity without revealing any personal data, enhancing privacy and security. Furthermore, the cryptographic nature of the blockchain ensures the integrity of the DIDs, making it nearly impossible for identities to be forged or altered.

Moreover, blockchain and smart contracts can be used to verify transactions and interactions. For example, in a blockchain-based voting system, a smart contract could automatically verify each vote's validity, ensuring that only eligible voters can vote and that each voter can only vote once [17]. This can significantly increase the transparency and integrity of the voting process, strengthening trust in the system.

In the realm of decentralized data storage and management, a critical evaluation of cost-effectiveness and efficiency is paramount. The choice between utilizing blockchain (or smart contracts) and alternative storage solutions such as the IPFS (InterPlanetary File System) hinges significantly on the associated costs [18]. Analyzing the cost implications, it is observed that storing 1 kB of data via a smart contract execution on the Ethereum blockchain consumes approximately 2,155,120 gas. Given the Ethereum price of 1,574.26 USD as of October 18, 2023, the resultant cost equates to around 47 USD per 1 kB of data. In a practical application where encrypted biometric features approximating 446 kB need to be stored, the financial expenditure escalates to an approximate value of 20,962 USD. Additionally, the storage of a public key, which typically encompasses substantial data, in this case around 47.3 MB, further inflates the cost to an estimated 2,223,100 USD.

Considering such substantial costs, even the exploration of alternative networks does not present a viable economic solution due to the inherently high expenses associated with substantial data storage on blockchain technologies. Thus, an optimized approach involves leveraging IPFS for the storage of substantial content such as encrypted biometric features and public keys. Concurrently, the blockchain can be utilized efficiently by storing only the DID URI linked with the Content Identifier (CID). This strategic approach significantly reduces the data storage requirements on the blockchain to an estimated 100 kB, thereby reducing the execution cost to about 451,325 gas. Based on the pricing as of October 18, 2023, this translates to a cost of approximately 9.9572 USD, representing a pragmatic and cost-efficient strategy for decentralized data storage and management. Consequently, by implementing IPFS and blockchain, we still ensure that there is no single point of failure, and since only encrypted versions of biometrics are stored online, privacy is preserved.

2.4. Homomorphic Encryption

Homomorphic encryption is a transformative cryptographic technique that facilitates computations on encrypted data without necessitating its decryption. This ensures that sensitive data remains confidential throughout computational processes, thereby bolstering its security and privacy. The concept remained largely theoretical until 2009, when Craig Gentry presented the first fully homomorphic encryption (FHE) scheme [19]. This FHE scheme utilized lattice-based cryptography and allowed both addition and multiplication on encrypted data. The significance of homomorphic encryption lies in its potential applications, especially in cloud computing. It enables data owners to outsource computations to cloud servers without compromising the confidentiality of their data. Since Gentry's pioneering work, there has been a surge in research dedicated to optimizing and enhancing the efficiency of homomorphic encryption schemes. Notable advancements include leveled homomorphic encryption, which allows users to set a limit on the depth of computations [20], and bootstrapping techniques that refresh ciphertexts to reduce noise [21]. These innovations are paving the way for the practical implementation of homomorphic encryption in real-world scenarios, such as secure voting systems, private medical data analysis, and encrypted search.

In practical application scenarios like neural networks, the introduction of hierarchical Galois keys proves to be highly efficient. For example, with the use of the proposed hierarchical Galois keys, the implementation of networks such as ResNet-20 and ResNet-18 becomes more feasible, as evidenced by the reduced memory requirements and the maintenance of competitive classification accuracies in encrypted domains. Furthermore, homomorphic encryption, leveraging the innovative use of Galois keys and other optimizations, finds substantial applications in supply chain security, regulatory compliance, and private data analytics. This cryptographic method enables secure third-party computations, adherence to rigorous data protection regulations such as the GDPR, and private and secure data analytics operations.

Galois automorphisms are intrinsic to field theory in mathematics, with profound applications in homomorphic encryption, particularly in schemes like the Brakerski-Fan-Vercauteren (BFV) and Cheon-Kim-Kim-Song (CKKS). A Galois automorphism is defined as an isomorphism from a field to itself that preserves field operations, mathematically represented as

$$\sigma(a+b)=\sigma(a)+\sigma(b). \tag{1}$$

$$\sigma(ab)=\sigma(a)\sigma(b), \tag{2}$$

for all (a, b) in a field (F), where (\sigma: F to F) is a bijective map.

In the context of HE, Galois automorphisms facilitate essential operations such as rotation and permutation on encrypted data, enhancing the utility of encryption schemes for complex operations without decryption. This is pivotal in algorithms related to AI and machine learning (ML), where encrypted data can be manipulated efficiently, preserving privacy and security.

2.5. Biometric Verification (FaceNet)

Schroff et al. [22] introduced FaceNet, a remarkable system proficient at directly learning a mapping from face images to a compact Euclidean space where distances authentically signify face similarity. In this well-constructed space, conventional tasks like face recognition, verification, and clustering become straightforward, utilizing FaceNet embeddings as pivotal feature vectors. Employing a deep convolutional network, this methodology is distinct, optimizing the embedding directly and bypassing the intermediate bottleneck layer optimization common in previous deep learning strategies. A unique online triplet mining method is used for training, employing triplets of nearly aligned matching/non-matching face patches. This innovative approach enhances representational efficiency, enabling exemplary face recognition performance while requiring only 128 bytes per face.

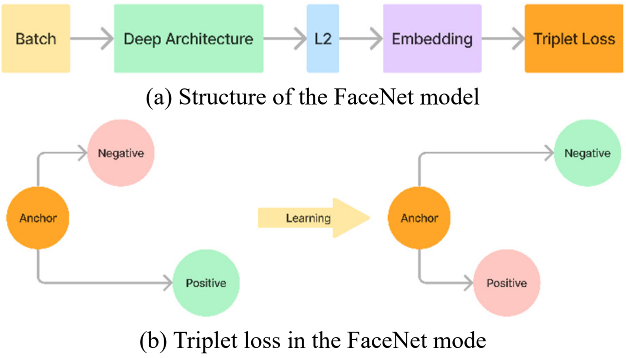
The FaceNet model architecture, as depicted in Fig. 3(a), encompasses a batch input layer and a profound CNN, succeeded by L_2 normalization, culminating in the generation of face embeddings. The influential triplet loss is applied during the training phase.

Triplet Loss plays a crucial role in diminishing the distance between an anchor and a positive (of identical identity) while concurrently maximizing the distance with a negative from a different identity. The process is shown in Fig. 3(b). Triplet Loss, essential for optimization during training, is computed using the equation:

$$\mathcal{L}=\sum_i^N\left[\left\|f(x_i^a)-f\left(x_i^p\right)\right\|_2^2-\|f(x_i^a)-f(x_i^n)\|_2^2+\ \alpha\right], \tag{3}$$

where:

· $(f(x))$ signifies the embedding of image (x),



(a) Structure of the FaceNet model

(b) Triplet loss in the FaceNet mode

Fig. 3. FaceNet structure and model.

- x_i^a is the anchor image,
- x_i^p represents a positive image sharing the anchor's identity,
- x_i^n denotes a negative image with a distinct identity from the anchor,
- α is a margin incorporated between positive and negative pairs.

The squared L_2 distance metric is crucial in measuring similarities between embeddings. This metric computes the square of the Euclidean distance between two points in the embedding space, ensuring effective differentiation between various face embeddings.

$$D(x, y) = \|f(x) - f(y)\|_2^2. \tag{4}$$

Here, $D(x, y)$ symbolizes the squared L_2 distance between the embeddings of images (x) and (y).

III. PROPOSED SYSTEM

In this section, we present a pioneering decentralized biometric authentication system designed for car-sharing services, integrating DID, off-chain data storage solutions, and AI to enhance the accuracy of biometric verification. Our system incorporates FHE and leverages the IPFS for secure and efficient data handling, thereby allowing temporary biometric verification access while guaranteeing the privacy and security of user data. By storing encrypted biometric data off-chain, we significantly mitigate privacy risks. The FaceNet model is utilized to refine the verification process, ensuring a smooth and efficient user experience. Our approach to secure identity verification within the car-sharing context aims to instill a higher degree of trust among stakeholders in the ecosystem.

Our framework is designed with a focus on user-centric identity management, compliance with GDPR through off-chain data storage, and the application of AI to streamline biometric verification processes. Users initiate their interaction with our system through an enrollment process, during which they create a DID, and the DMV issues a VC that is securely stored on the user's device. To rent a car, users present their digital driver's license, authenticated by the DMV. Biometric features are encrypted on the user's device using Homomorphic Encryption and subsequently stored on IPFS. The corresponding content identifier (CID) is linked to the user's DID URI, ensuring that once the authority verifies the biometric data, it cannot be altered. With the encryption keys held solely by the user, the privacy of the biometric information is preserved, as only the user has the capability to decrypt the data.

During the verification process, users authenticate themselves using in-car biometric sensors. The car-sharing service retrieves the encrypted biometric features from IPFS and computes the Euclidean distance between these and the features extracted during the rental process. Based on this encrypted comparison, the system determines whether to grant access to the vehicle. This innovative method enhances the security and privacy of the car-sharing service while maintaining a user-friendly interface. The detailed framework can be seen in Fig. 4.

Table 1. Summary of notations.

Symbol	Description
DID	Decentralized identifier
CID	Content identifier
FHE	Fully homomorphic encryption
IPFS	InterPlanetary file system
FaceNet	Facial recognition model
GDPR	General data protection regulation
DMV	Department of motor vehicles
GOV	Government
DL	Driver's license
VC	Verifiable credential
DID_URI	DID uniform resource identifier
DID_Sk	DID secret key
DID_Pk	DID public key
HE_Sk	Homomorphic encryption secret key
HE_Pk	Homomorphic encryption public key
DKMS	Decentralized key management system
VC_Pk	Verifiable credential public key

3.1. Onboarding

The onboarding process consists of two phases: registration and verification.

3.1.1. Registration

Initially, government entities participating in our system, such as the DMV, must also register, enabling them to issue VCs subsequently. During user registration, the user creates an account within the DKMS application and generates their DID, which includes a DID URI, public key DID_Pk, and secret key DID_Sk. The DKMS then publishes the DID_Pk and DID_URI on the blockchain. Subsequently, the user captures a biometric image using their phone, and the DKMS application extracts features employing the method described in Algorithm 1 (see Section IV). The system then generates homomorphic encryption keys HE_Sk and HE_Pk, following the procedures outlined in Algorithms 2 and 3. The biometric features are encrypted with the HE_Sk like shown in Algorithm 4, and the encrypted

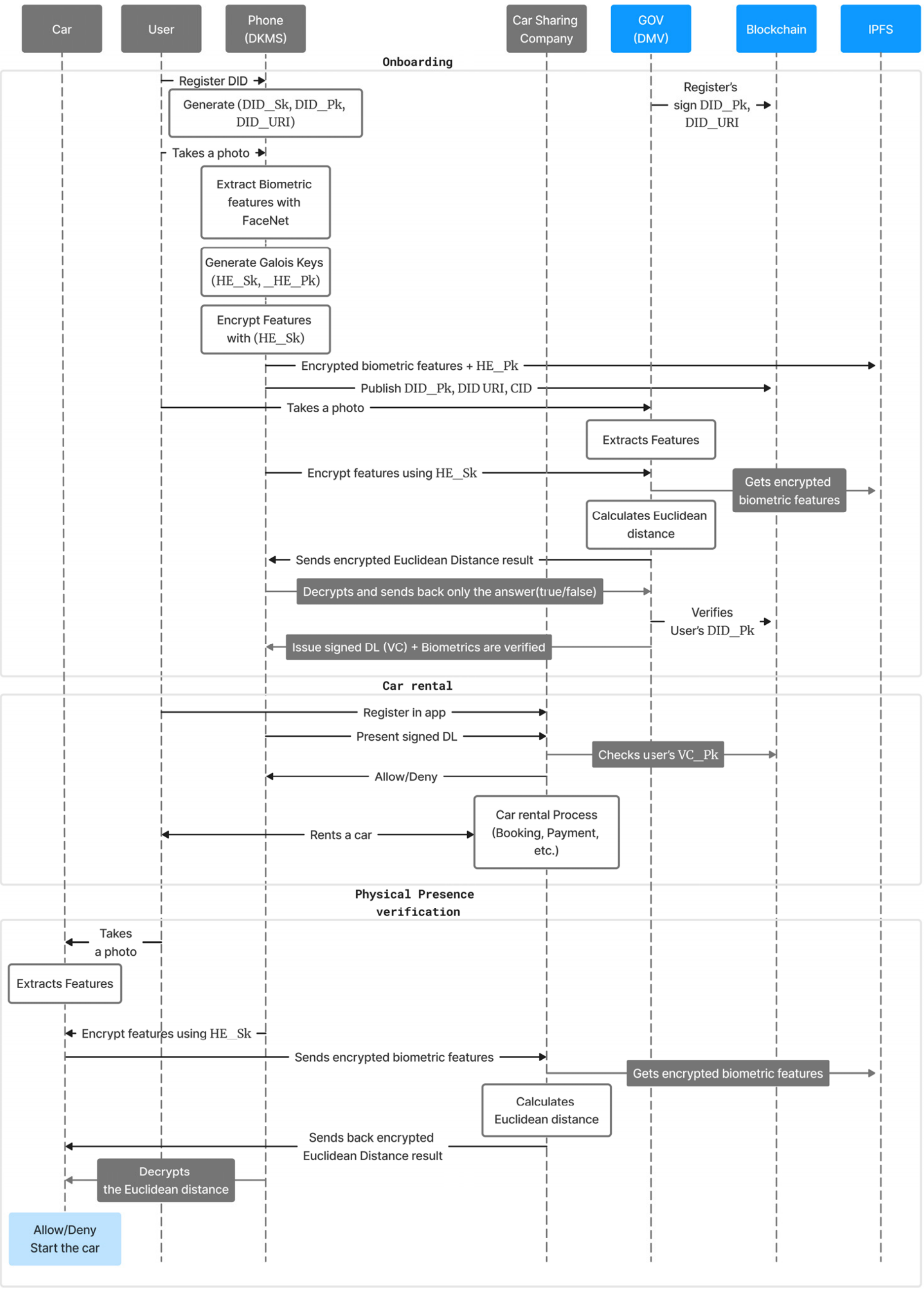

Car
User
Phone (DKMS)
Car Sharing Company
GOV (DMV)
Blockchain
IPFS
Onboarding
Register DID
Generate (DID_Sk, DID_Pk, DID_URI)
Register's sign DID_Pk, DID_URI
Takes a photo
Extract Biometric features with FaceNet
Generate Galois Keys (HE_Sk, _HE_Pk)
Encrypt Features with (HE_Sk)
Encrypted biometric features + HE_Pk
Publish DID_Pk, DID URI, CID
Takes a photo
Extracts Features
Encrypt features using HE_Sk
Gets encrypted biometric features
Calculates Euclidean distance
Sends encrypted Euclidean Distance result
Decrypts and sends back only the answer(true/false)
Verifies User's DID_Pk
Issue signed DL (VC) + Biometrics are verified
Car rental
Register in app
Present signed DL
Checks user's VC_Pk
Allow/Deny
Rents a car
Car rental Process (Booking, Payment, etc.)
Physical Presence verification
Takes a photo
Extracts Features
Encrypt features using HE_Sk
Sends encrypted biometric features
Gets encrypted biometric features
Calculates Euclidean distance
Sends back encrypted Euclidean Distance result
Decrypts the Euclidean distance
Allow/Deny Start the car

Fig. 4. Sequence diagram of the proposed framework for privacy preserving car sharing system.

data, along with the HE_Pk, is stored on IPFS. The user receives a CID that references the storage location, allowing other system participants to access the data. At this juncture, the registration phase concludes. However, the government must still manually verify that the biometric data belongs to the registering user.

3.1.2. Verification

For verification, the user visits a government institution and captures a biometric image on a designated device. This device extracts features using the same algorithm as before (Algorithm 1). The user then encrypts the extracted features with their HE_Sk. The DMV accesses the encrypted biometric data and the user's HE_Pk from IPFS. It computes the Euclidean distance over the encrypted data and sends the encrypted result back to the user's phone. Only the user can decrypt this result with their HE_Sk, and the user returns only the decrypted result to the DMV. Once verified, the DMV issues the digital driver's license (VC) to the user and records it on the blockchain. Despite its complexity, this verification process ensures security and is typically a one-time procedure.

3.2. Car Rental

The car rental process is streamlined compared to traditional methods. By enhancing the security of the onboarding process, reliance on additional KYC services is eliminated, as falsifying a digital license becomes infeasible. Car-sharing companies can verify users by accessing records on the blockchain. Thus, while the basic car rental process remains unchanged, the verification component is simplified. Users apply for a rental through the car-sharing app, presenting their digital driver's license. The car-sharing company verifies the VC_Pk against the blockchain and approves or denies the user's application. Other processes such as booking and payment, are outside the scope of this thesis and remain unchanged.

3.3. Physical Presence Verification

The user, upon entering the rented vehicle, captures their biometric image using the in-car camera. The vehicle extracts features using the prescribed method (Algorithm 1) and encrypts them with the user's HE_Sk. These encrypted features are sent to the car-sharing company, which then retrieves the previously stored encrypted features from IPFS to compute the Euclidean distance as outlined in Algorithm 8. The car-sharing company sends the encrypted result back to the vehicle, where the user decrypts it with their HE_Sk, enabling the car to decide whether to grant engine access. This crucial phase of the authentication process is delineated in Algorithm 10.

IV. IMPLEMENTATION AND EVALUATION

The evaluation of the system's performance was conducted on a computational setup possessing the following specifications:

The assessment entailed a comparative analysis of 280 unlabeled images, during which True Positives, False Positives, True Negatives, and False Negatives were meticulously determined through manual inspection. Moreover, the temporal duration required to execute each function for the respective pair of images was meticulously documented and stored in a tabular format. On the other hand, our experiment focused on evaluating the cost-effectiveness and efficiency of decentralized data storage solutions, particularly for the storage of encrypted biometric data and public keys. Given the high costs associated with storing large amounts of data directly on the Ethereum blockchain, we desire a hybrid approach using the Ethereum testnet and IPFS. We leveraged the Ethereum testnet to store DID URIs securely. Ethereum's testnet was chosen for its compatibility with smart contracts, allowing us to manage DIDs efficiently while leveraging the blockchain's security and decentralization features. Simultaneously, we employed IPFS for storing the actual data, including encrypted biometric features and public keys. IPFS was selected due to its cost-effectiveness for storing large files compared to direct storage on the blockchain. It provides a decentralized storage solution without a single point of failure, maintaining privacy by storing only encrypted data.

Table 2. Simulation parameters.

Parameter	Specification
Processor	AMD Ryzen 5 3600, 6 cores, 12 threads, 3.6 GHz
Memory	32 GB DDR4 RAM
Operating System	Windows 10
ML Framework	Keras 2.14, TensorFlow 2.14
Programming Language	Python 3.6.4
Deep Learning Library	deepface 0.0.79
HE Library	tenseal 0.3.14
DID_URI DID_Sk	DID Uniform Resource Identifier DID Secret Key

4.1. System Implementation

4.1.1. Facial Features Extraction

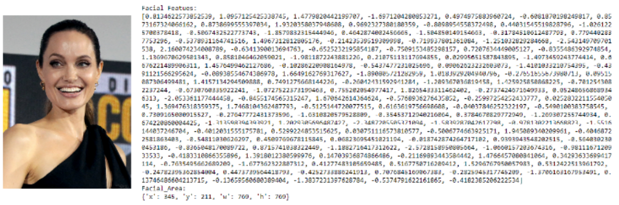
After importing images to compare, we extract features from them with DeepFace library [23]. The 'represent' function from the DeepFace library was utilized to derive vectorial representations, known as embeddings, of the facial images. The output of Algorithm 1, delineated in Fig. 5, elucidates the facial coordinates within the image and subsequently procures the feature sets as an array.

4.1.2. Initialization of Cryptographic Context: Galois Key Generation

Algorithm 2 delineates the procedure for initializing the

Algorithm 1. Embedding an image using the DeepFace with the Facenet model.

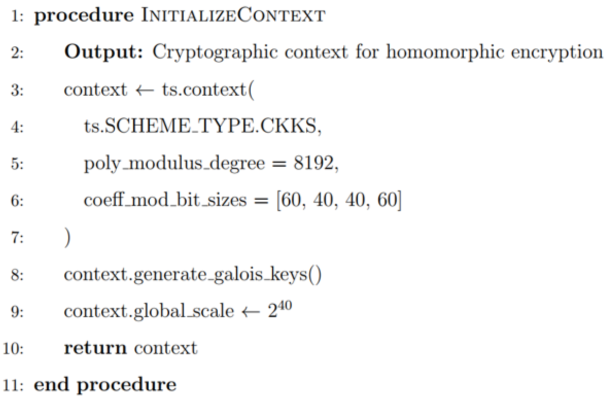
```
procedure ImageEmbedding(img1_path)
    Input: Path to the image (img1_path)
    Output: Embedded representation of the image
    // Utilizing the DeepFace.represent function to get the embedded representation
    img1_embedding ← DeepFace.represent(img1_path, model_name = 'Facenet')
    return img1_embedding
end procedure
```



```
Facial Featues:
[0.013462257385253906, 1.0957125425338745, 1.4779820442199707, -1.6971204280853271, 0.4974975883960724, -0.6081870198249817, 0.85
73167324066162, 0.8738699555397034, 1.9320358037948608, 0.9692327380180359, -0.8098954558372498, 0.44031545519828796, -1.026122
5708378418, -0.5067432522773743, -1.8579832315444946, 0.4642874002456665, -1.504850149154663, -0.31784510612487793, 0.779440283
7753296, -0.5378931164741516, 1.4967312812805176, -0.21423539519309998, -0.719937801361084, -1.251032829284668, -2.543149709701
538, 2.160074234008789, -0.6341390013694763, -0.6525232195854187, -0.7509153485298157, 0.7207634449005127, -0.8355486392974854,
1.1369670629501343, 0.8581046462059021, -1.9811872243881226, 0.21075113117694855, 0.029056515878438950, -1.4073459243774414, 0.0
67622148990631l, 1.4576499462127686, -0.10286220908164978, -0.5437477231025696, 0.09062623232603073, -1.4101033210754395, -0.43
91121566295624, -0.00930554674306978, 1.6649162769317627, -1.890005721282959, 1.0183929204940796, -0.27651655673980713, 0.09515
887306499481, 1.4157134294509888, 0.7491275668144226, -0.26042431592941284, -1.289367036819458, 1.425925850868225, -0.781254308
2237244, -0.6730760335922241, -1.0727522373199463, 0.755202054977417, 1.8205433311462402, -0.2737424671649933, 0.05248656868934
6313, -2.053361177444458, -0.0455174565315247, 1.670642614364624, -0.5760936276435852, -0.25997254252433777, 0.0252832211554050
45, 1.369476318359375, 1.7468104362487793, -0.5125144720077515, 0.6163619756698608, -0.04037846252322197, -0.5490100383758545,
0.7309165000915527, -0.27647772431373596, -1.6310820579528809, -0.35453712940216064, 0.3784678829772949, -1.269387255744934, 0.
674220860004425, -1.313598394393921, 1.2029306960487427, -2.3487205505371094, -1.5839287042617798, -0.9781302213668823, -1.5516
144037246704, -0.4012031555175781, 0.5299224853515625, 0.030751116573810577, -0.5006774663925171, 1.945089340209961, -0.4046872
2581863403, -0.54811030626297, 0.4509760678115845, 0.06823695451021194, -0.018742874264717102, 0.9393943548202515, -0.564030230
0453186, -0.8365048170089722, 0.8715741038322449, -1.1882716417312622, -2.5728158950805664, -1.0660157203674316, -0.98111671209
33533, -0.4183310866355896, 1.3918012380599976, 0.14703936874866486, -0.21169893443584442, 1.4766457080841064, 0.34293633699417
114, -0.7635405662689209, -1.677362322807312, 0.41277483105659485, 0.5167758716289412, 1.5296767950057983, 0.5312422513961792,
-0.24782395362854004, 0.4473739564418793, -0.4252733886241913, 0.7070845169067383, -0.2825945317745209, -1.3706163167953491, 0.
13746406604213715, -0.13659560680389404, -1.3837231397628784, -0.5374791622161865, -0.4182385206222534]
Facial_Area:
{'x': 345, 'y': 211, 'w': 769, 'h': 769}
```

Fig. 5. FaceNet output using DeepFace algorithm.

Algorithm 2. Initialization of cryptographic context for homomorphic encryption using TenSEAL.



```
procedure InitializeContext
    Output: Cryptographic context for homomorphic encryption
    context ← ts.context(
        ts.SCHEME_TYPE.CKKS,
        poly_modulus_degree = 8192,
        coeff_mod_bit_sizes = [60, 40, 40, 60]
    )
    context.generate_galois_keys()
    context.global_scale ← $2^{40}$
    return context
end procedure
```

cryptographic context requisite for the generation of secret and public keys.

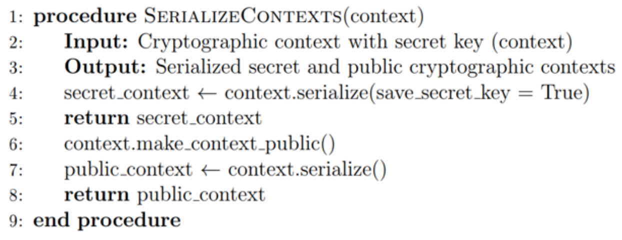
- A cryptographic context for homomorphic encryption is initialized using the tenseal library.
- ts SCHEME TYPE CKKS specifies the use of the CKKS scheme, which allows computations on encrypted real or complex numbers.
- Parameters such as poly modulus degree and coeff mod bit sizes are specified for the encryption scheme.
- Galois keys are generated, and a global scale is set, which are configurations related to the encryption and computation processes.

Algorithm 3 details the process of generating the secret and public keys. Within this cryptographic context, the secret key undergoes serialization—a conversion of the object's state into a storable or transferrable format, resulting in a size of approximately 940 kB. Subsequently, the public context is serialized, amounting to 47.3 MB.

4.1.3. Applying Homomorphic Encryption to Facial Embeddings

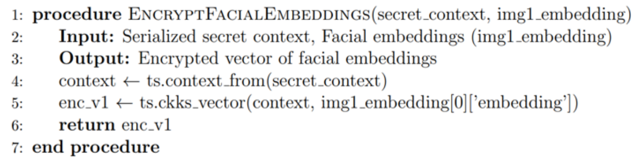
For subsequent cryptographic operations, such as encryption, the secret context is retrieved from "secret.txt." The facial embeddings are encrypted employing the CKKS [1] encryption scheme, sourced from the 'tenseal' library, and the resultant encrypted vector is assigned to the variable enc_v1.

Algorithm 3. Serialization of cryptograph contexts.



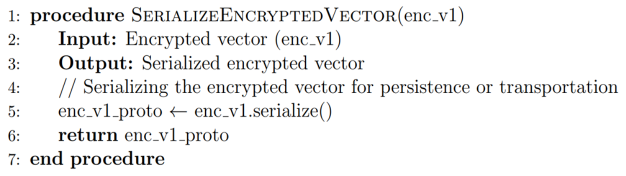
```
procedure SerializeContexts(context)
    Input: Cryptographic context with secret key (context)
    Output: Serialized secret and public cryptographic contexts
    secret_context ← context.serialize(save_secret_key = True)
    return secret_context
    context.make_context_public()
    public_context ← context.serialize()
    return public_context
end procedure
```

Algorithm 4. Loading secret context and encrypting facial embeddings.



```
procedure EncryptFacialEmbeddings(secret_context, img1_embedding)
    Input: Serialized secret context, Facial embeddings (img1_embedding)
    Output: Encrypted vector of facial embeddings
    context ← ts.context_from(secret_context)
    enc_v1 ← ts.ckks_vector(context, img1_embedding[0]['embedding'])
    return enc_v1
end procedure
```

Algorithm 5. Serialization of encrypted vector.



```
procedure SerializeEncryptedVector(enc_v1)
    Input: Encrypted vector (enc_v1)
    Output: Serialized encrypted vector
    // Serializing the encrypted vector for persistence or transportation
    enc_v1_proto ← enc_v1.serialize()
    return enc_v1_proto
end procedure
```

The encrypted vector enc_v1 is then serialized, facilitating its storage or transmission. This serialized vector is conserved within enc_v1_proto. The size of the serialized, encrypted biometric features is approximately 446 kB.

4.1.4. Decentralized Identity Verification via Blockchain and IPFS

In the proposed decentralized identity ecosystem, three core entities interact: holder, verifier, and issuer. The Holder, equipped with a Decentralized Key Management System (DKMS), has the autonomy to create a unique DID and manage its corresponding DID document. This document, stored on the blockchain, includes a CID that points to encrypted biometric features held on the IPFS, allowing for off-chain data storage while maintaining on-chain referential integrity. Holders retain granular control over access to this CID-linked information, ensuring privacy and consent in data sharing. Verifiers, such as contactless car-sharing companies, can authenticate the Holder's credentials by resolving the DID to its document and, with permission, access the detailed data via the CID from IPFS. Issuers like the department of motor vehicles are responsible for issuing credentials embedded with CIDs, maintaining the veracity and currentness of credential information, and managing revocation lists or statuses accessible through IPFS. This architecture not only empowers the Holder with control over their identity but also enables the secure, efficient verification of credentials by Verifiers, underpinned by the trustworthiness of the Issuers. Each party client application participating in this system should have following functionality.

4.1.4.1. Holder

- Manage DID and DID Document: Generate and update their DID and DID document, including updating the CID that points to additional information stored on IPFS.

- Control Access: Determine which entities have access to the CID and, consequently, to the information on IPFS.
- Grant and Revoke Permissions: Use access control mechanisms to grant or revoke permission for Verifiers to retrieve the IPFS-stored information via the CID.
- Share Verifiable Credentials: Provide verifiable credentials, including the relevant CIDs, to Verifiers when needed and consent to access the extended information on IPFS.
- IPFS Interaction: Upload and manage data on IPFS, ensuring that the CID in the DID document is always current and points to the correct data.

4.1.4.2. Verifier (contactless car-sharing company)

- Resolve DID to DID Document: Use the DID to access the DID document on the blockchain, retrieving the CID for additional information.
- Request Access: Request permission from the holder to access the information linked via the CID on IPFS.
- Retrieve Data from IPFS: Once granted permission, use the CID to retrieve additional information from IPFS needed for verification.
- Verify Credentials with Extended Data: Verify the Holder's credentials using both the DID document and the additional data retrieved from IPFS, ensuring all provided information is consistent and valid.

4.1.4.3. Issuer (department of motor vehicles)

- Enhanced Credential Issuance: When issuing credentials, include a CID that points to additional data on IPFS, if necessary, for the credential's use case.
- Update CID in DID Document: Provide tools or services to help Holders update the CIDs in their DID documents, ensuring that only current and valid information is linked.
- Maintain Revocation Lists: Potentially use IPFS to host revocation lists or other credential status information, making it accessible via CIDs in issued credentials.

4.1.5. Generating Encrypted Facial Embedding From Car

Upon completion of the aforementioned process, the car-sharing company will be granted access to encrypted biometric features hosted on the IPFS, accessible via links embedded in the smart contract. To compute the Euclidean distance, it is imperative for the car-sharing company to possess two sets of encrypted features. Consequently, the company should deploy specialized kiosks within vehicles to capture images. Users will present their secret key, possibly in the form of a QR code or an alternative mechanism, allowing the kiosk to encrypt the facial features and transmit them to the company's servers for computation. By design, these kiosks are engineered to preclude the retention of any data, ensuring user privacy. Post-process, the encrypted features remain secure for transport.

Algorithm 6. Processing second image: embedding, encryption, and serialization.

```
procedure PROCESSSECONDIMAGE(img2_path, secret_context)
    Input: Path to the second image (img2_path), Serialized secret context (secret_context)
    Output: Serialized encrypted vector of the second image's embeddings
    // Getting the embeddings of the second image
    img2_embedding ← DeepFace.represent(img2_path, model_name = 'Facenet'
    // Loading the secret context back for encryption
    context ← ts.context_from(secret_context)
    // Encrypting the embeddings of the second image
    enc_v2 ← ts.ckks_vector(context, img2_embedding[0]['embedding'])
    // Serializing the encrypted vector for persistence or transportation
    enc_v2_proto ← enc_v2.serialize()
    return enc_v2_proto
end procedure
```

Algorithm 7. Linking encrypted vectors with public context.

```
procedure LINKVECTORSWITHCONTEXT(public, enc_v1, enc_v2)
    Input: Public cryptographic context (public), Serialized encrypted vectors (enc_v1, enc_v2)
    Output: Encrypted vectors linked with the public context
    // Loading the public cryptographic context
    context ← ts.context_from(public)
    // Reading the serialized encrypted vectors from data
    enc_v1_proto ← read_data(enc_v1)
    enc_v2_proto ← read_data(enc_v2)
    // Creating lazy encrypted vectors from the serialized data
    enc_v1 ← ts.lazy_ckks_vector_from(enc_v1_proto)
    enc_v2 ← ts.lazy_ckks_vector_from(enc_v2_proto)
    // Linking the encrypted vectors with the loaded public context
    enc_v1.link_context(context)
    enc_v2.link_context(context)
    return enc_v1, enc_v2
end procedure
```

4.1.6. Loading the Public Context and Calculating the Euclidean Distance between Encrypted Vectors

The encrypted vectors are converted into lazy CKKS vectors, priming them for homomorphic operations sans immediate decryption. These vectors are associated with a cryptographic context vital for computational tasks on encrypted data, absent of the secret key to maintain security during operations not necessitating decryption. The *enc_v1_proto* signifies the encrypted facial embeddings. The subsequent step involves aligning the encrypted vector with this context to facilitate computations.

With the encrypted vector duly linked to the cryptographic context, the calculation of the Euclidean distance is feasible, enabling secure and private comparison of biometric data.

The operation commences by computing the discrepancy between two encrypted vectors, *enc_v1* and *enc_v2*. Following this, the dot product of the resultant differential vector with itself is ascertained, yielding the squared Euclidean distance. This value is then encapsulated within the variable *euclidean_squared*, enabling the quantification of similarity between the biometric features represented by the encrypted vectors, all while ensuring that the data remains encrypted and secure throughout the process.

The encrypted squared Euclidean distance, denoted as $euclidean\_squared$, undergoes serialization to enable its storage and transmission. Without the secret key, decryption attempts by the car-sharing company will be futile, ensuring that computational outcomes remain confidential until the user opts to decrypt them.

Consequently, the car-sharing company is tasked with relaying these computations back to the in-car kiosk. Upon reception, the user, with their secret key, can decrypt the results. The kiosk then determines whether to grant or deny vehicle access based on the decrypted outcome, maintaining a secure and user-controlled access system.

The in-car system initiates by loading the secret cryptographic context and the encrypted squared Euclidean distance. Subsequently, the encrypted value is prepped for decryption by integrating it with the secret context. Once decrypted, the plain squared Euclidean distance is obtained. The in-car system then adjudicates: a value below 100 signifies recognition of the individual, granting access; a value at or above 100 denotes a non-match, thus access is denied. This procedure upholds the integrity of personal biometric data, allowing only the user to decrypt and evaluate the comparison outcome.

4.2. Facial Recognition System Evaluation

This report presents a comprehensive evaluation of an authentication system based on various metrics and processing times of image comparisons. The evaluation metrics include the false acceptance rate (FAR), false rejection rate (FRR), acceptance rate (AR), and rejection rate (RR). Additionally, the report analyzes the average processing times involved in each step of the image comparison process.

4.2.1. Evaluation Metrics

The evaluation metrics are calculated based on the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) observed during the authentication process. Fig.6 shows the system evaluation metrics, where the formulas for the metrics are as follows:

- False acceptance rate (FAR):

$$\frac{FP}{FP+TN} = 0.0429 \;\; (or \;\; 4.29\%)$$

- False rejection rate (FRR):

$$\frac{FN}{FN+TP} = 0.3214 \;\; (or \;\; 32.14\%)$$

- Acceptance rate (AR):

$$\frac{TP}{TP+FN} = 0.6786 \;\; (or \;\; 67.86\%)$$

- Rejection rate (RR):

$$\frac{TN}{TN+FP} = 0.9571 (or \;\; 95.71\%)$$

Algorithm 8. Calculating squared euclidean distance between encrypted vectors.

```
procedure CalculateSquaredEuclidean(enc_v1, enc_v2)
    Input: Encrypted vectors (enc_v1, enc_v2)
    Output: Squared Euclidean distance between the encrypted vectors
    // Subtracting one encrypted vector from the other
    euclidean_squared ← enc_v1 - enc_v2
    // Calculating the dot product of the resulting vector with itself
    euclidean_squared ← euclidean_squared.dot(euclidean_squared)
    return euclidean_squared
end procedure
```

Algorithm 9. Serialization of squared euclidean distance.

```
procedure SerializeEuclideanSquared(euclidean_squared)
    Input: Squared Euclidean distance between encrypted vectors (euclidean_squared)
    Output: Serialized squared Euclidean distance
    // Serializing the squared Euclidean distance for persistence or transportation
    serialized_euclidean ← euclidean_squared.serialize()
    return serialized_euclidean
end procedure
```

Algorithm 10. Decrypting and verifying the squared euclidean distance.

```
procedure VerifyDistance(secret, euclidean_squared)
    Input: Serialized secret context (secret), Serialized squared Euclidean distance (euclidean_squared)
    Output: Verification result
    // Loading the secret cryptographic context
    context ← ts.context_from(read_data(secret))
    // Reading the serialized squared Euclidean distance
    euclidean_squared_proto ← read_data(euclidean_squared)
    // Creating a lazy encrypted vector and linking it with the secret context
    euclidean_squared ← ts.lazy_ckks_vector_from(euclidean_squared_proto)
    euclidean_squared.link_context(context)
    // Decrypting the squared Euclidean distance
    euclidean_squared_plain ← euclidean_squared.decrypt()[0]
    // Making a decision based on the decrypted value if $euclidean\_squared\_plain < 100$ then
        return "Allow (Verification passed)" else
        return "Deny (Verification failed)"

end procedure
```

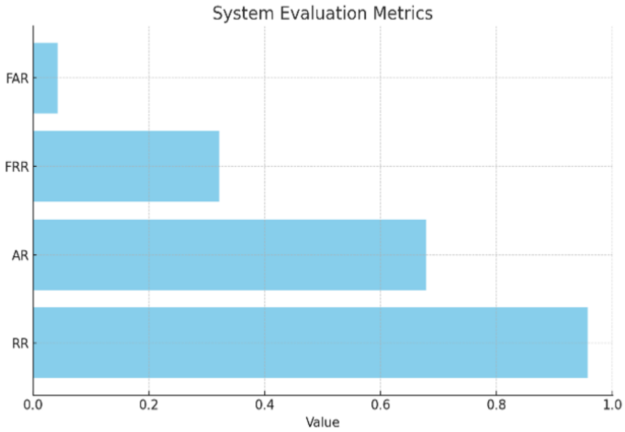


Fig. 6. System evaluation metrics.

4.2.2. Processing Time

In the examination of system efficiency, a detailed temporal analysis was conducted for each discrete stage of the image comparison workflow. The constituent processes scrutinized were the feature extraction, the encryption of these features, the computation of the Euclidean distance in an encrypted form, and the final decryption and result determination. Fig. 7 shows the time per two image execution. For each of these operations, the average time required was meticulously recorded, facilitating a comprehensive understanding of the time allocation across the entire procedure. These averages were then aggregated to determine the total average duration necessary to complete a single image comparison. The average time per step with average total final time can be seen in Fig. 8. The findings from this temporal evaluation are meticulously tabulated, providing insights into the performance and scalability of the system, and are expounded upon in the ensuing discussion.

4.2.3. System Performance Evaluation for Different Threshold

In the analysis of the facial recognition system's performance across various thresholds of Euclidean distance, four key metrics were evaluated: false acceptance rate (FAR), false rejection rate (FRR), acceptance rate (AR), and rejection rate (RR). Fig. 9 illustrates how these metrics vary as the threshold level changes and values are displayed in the Table 3. As observed, with increasing thresholds, FAR generally tends to decrease, indicating fewer instances where the system incorrectly accepts a non-face as a face. Conversely, FRR typically increases with higher thresholds, reflecting more instances of the system failing to recognize a true face. The acceptance rate (AR), representing the proportion of true faces correctly identified, tends to increase with lower thresholds. In contrast, the rejection rate (RR), depicting the proportion of non-faces correctly rejected, usually decreases as the threshold is lowered.

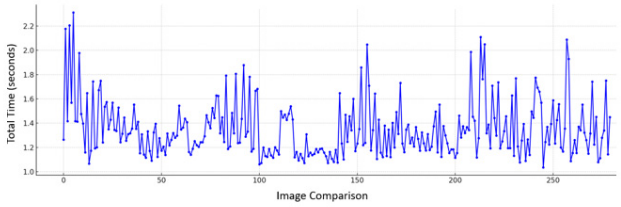


Fig. 7. Time per two image execution.

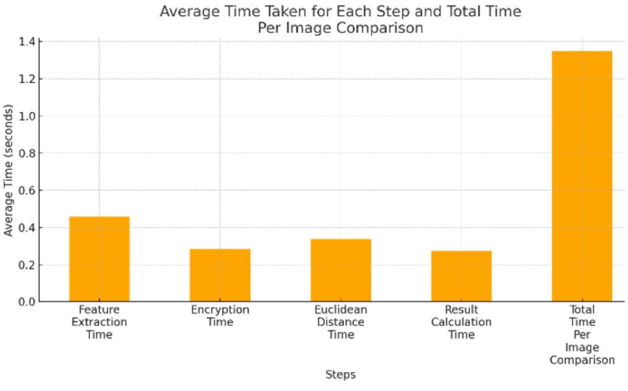


Fig. 8. Average time per step with average total final time.

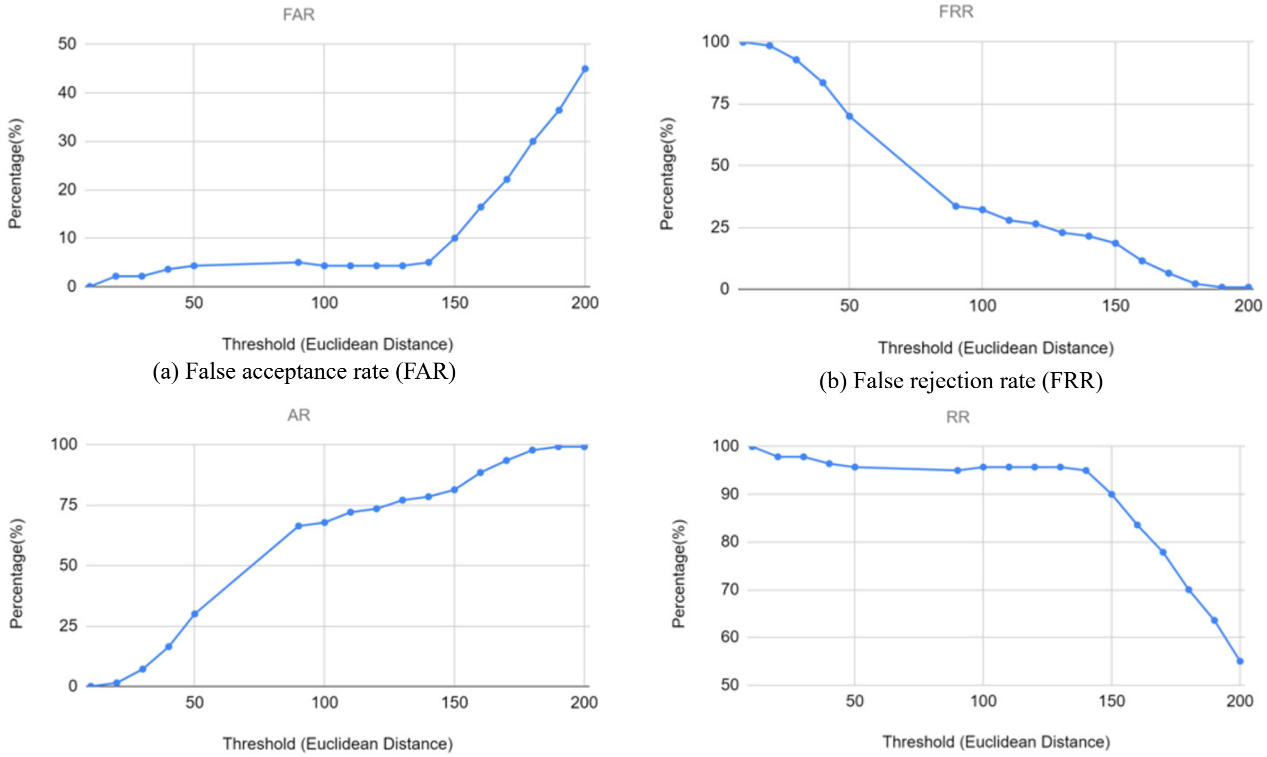
The Fig. 10 focuses on determining the equal error rate (EER) of the system, a crucial metric in biometric systems, representing the point where FAR and FRR are equal. This balance point is vital in evaluating the overall reliability and accuracy of the system. The EER was computed by identifying the threshold at which the absolute difference between FAR and FRR was minimal. In this analysis, the EER was found to be approximately 13.93% at a threshold of 160. This value indicates the system's trade-off point, where both types of errors (accepting false positives and rejecting true positives) are equally probable. The lower the EER, the higher the accuracy of the system. Thus, in this context, an EER of 13.93% suggests a moderate level of accuracy for the facial recognition system under evaluation, highlighting a significant consideration for practical applications and further optimization.

4.3. Comparison with Other Identity Management Solutions

In Table 4, we present a comprehensive comparison of our proposed system with key related works in the field. This comparison focuses on several vital parameters that are pivotal in assessing the sophistication and robustness of contemporary biometric systems.

Firstly, the aspect of HE, a critical component for ensuring data privacy and security, is universally adopted across all the compared systems, including ours. This approach is evident in the works of Sperling et al. [24], Gomez Barrero et al. [25], Yasuda et al. [26], and Alberto Torres et al.[27] , underscoring its significance in the field.

A significant distinction arises in the decentralized approach parameter. Our system uniquely integrates a decentralized framework, enhancing data integrity and reducing reliance on centralized data control, setting it apart from the approaches taken by other related works. This decentralization not only bolsters security but also aligns with modern trends in data management and privacy [28]. In the realm of AI-driven verification, our system stands out by leveraging advanced artificial intelligence algorithms for verification processes, a feature not presented in the referenced works. This integration signifies a leap forward in verification accuracy and efficiency. On the other hand, the work from [29] presents an innovative approach to anti-theft solutions for vehicles. While our methodology addresses a similar challenge, it offers distinct advantages in terms of



(a) False acceptance rate (FAR)

(b) False rejection rate (FRR)

(c) Acceptance rate (AR)

(d) Rejection rate (RR)

Fig. 9. System performance evaluation for different thresholds.

Table 3. System performance evaluation for different threshold.

Euclidean distance	FAR (%)	FRR (%)	AR (%)	RR (%)
10	0.00	100.00	0.00	100.00
20	2.14	98.57	1.43	97.86
30	2.14	92.86	7.14	97.86
40	3.57	83.57	16.43	96.43
50	4.29	70.00	30.00	95.71
90	5.00	33.5	66.43	95.00
100	4.29	32.14	67.86	95.71
110	4.29	27.86	72.14	95.71
120	4.29	26.43	73.57	95.71
130	4.29	22.86	77.14	95.71
140	5.0	21.43	78.57	95.00
150	10.00	18.57	81.43	90.00
160	16.43	11.43	88.57	83.57
170	22.14	6.43	93.57	77.86
180	30.00	2.14	97.86	70.00
190	36.43	0.71	99.29	63.57
200	45.00	0.71	99.29	55.00

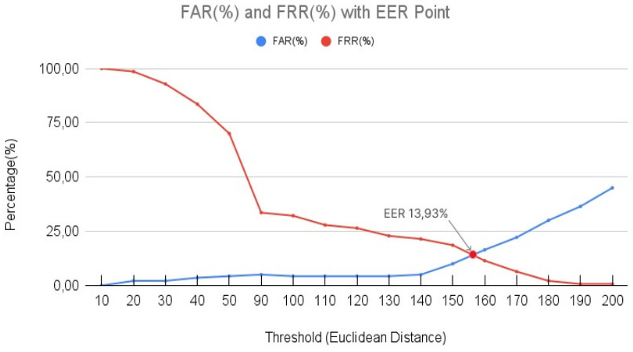


Fig. 10. Equal error rate (EER).

applicability across different industries where user verification is crucial. A key feature of our solution is that it does not require data storage within the vehicle, allowing for greater flexibility and adaptability. Additionally, while [29] highlights the challenges facial recognition technology faces, such as varying lighting conditions, angles, and other environmental factors, we contend that physiological biometrics offer a more reliable and immutable form of identification. These biometric markers are inherently more difficult to replicate or bypass compared to behavioral patterns, which can exhibit a significant degree of similarity among different individuals. This underscores the potential for error in systems relying solely on behavioral data, further emphasizing the robustness of physiological biometrics as a security measure.

User-centric Management, which empowers users with greater control over their biometric data, is another unique feature of our system, distinguishing it from the other cited works. This approach aligns with the growing emphasis on user privacy and autonomy in data management. Regarding Off-chain Data Storage, all the systems under comparison,

Table 4. Comparing proposed system with related works.

Key parameters	[24]	[25]	[26]	[27]	[29]	This work
Homomorphic encryption	✓	✓	✓	✓	✗	✓
Decentralized approach	✗	✗	✗	✗	✗	✓
AI-driven verification	✗	✗	✗	✗	✓	✓
User-centric management	✗	✗	✗	✗	✗	✓
Off-chain data storage	✓	✓	✓	✓	✓	✓
Temporary access facilitation	✗	✗	✗	✗	✗	✓
Protection against adversary attacks	✓	✗	✗	✗	✗	✓

including ours, employ this method. This is indicative of a shared recognition of the importance of secure and efficient data storage solutions outside the blockchain. Our system also uniquely facilitates Temporary Access, a feature not seen in the other works. This adds a layer of flexibility and user control, especially useful in scenarios requiring time-bound access permissions. Finally, in terms of Protection Against Adversary Attacks, our system, along with the system proposed by Sperling et al., demonstrates robust mechanisms to safeguard against various cyber threats, an aspect where some of the other compared systems fall short. Overall, this comparative analysis underscores the advanced features and enhanced security measures of our proposed system, setting it apart from existing works in significant ways.

In comparing our system's performance with that of Alberto Torres et al. [27], a substantial enhancement in processing time is evident. Our system demonstrates an impressive average time of only 1.4 seconds per image pair comparison. This duration encompasses the entire gamut of operations, including key generation, encryption, decryption, and Euclidean distance calculation. In contrast, the system evaluated by Alberto Torres et al. [27] required 10.4 minutes for each comparison of two biometric templates in an encrypted domain. This marked reduction in processing time represents a significant advancement in the efficiency of biometric systems. However, it is crucial to acknowledge that the computations in our study were conducted on a different biometric template. This distinction is important as it might influence the direct comparability of the two systems' performance metrics. Nevertheless, the improvement in processing speed in our system is a noteworthy achievement in the realm of biometric template comparison, potentially contributing to more efficient and practical applications in the field.

4.4. Discussion and Remarks

Our research methodology is focused primarily on the mechanisms of authentication and verification, without delving into the intricacies of the car rental process itself or the operational aspects of in-car hardware. It's important to clarify several key aspects of our approach to ensure a comprehensive understanding:

- *Verification Image Deletion*: Our approach mandates that images used for verification purposes be deleted by the car sharing company's hardware immediately after the process is completed. This step is crucial in mitigating risks associated with the unauthorized access to or misuse of biometric data.
- *Biometric Data Privacy*: The privacy of biometric features is a foundation of our methodology. By storing the private key exclusively on the user's device, we ensure that biometric data remains inaccessible to any third party, including the car rental company. This design principle safeguards users' biometric infor-mation against external breaches or internal misuse.
- *Limited Data Disclosure*: The car sharing application is designed to only communicate a boolean value indicating the success or failure of the verification process. This minimal data exchange ensures that sensitive information, such as the user's biometric data or the specifics of their verification, remains confi-dential.
- *Homomorphic Encryption*: We employ homomorphic encryption specifically to preserve the privacy of biometric features. This advanced cryptographic technique ensures that even the result of the Euclidean distance calculation, used in the verification process, is disclosed only upon successful private key verification. Essentially, this method enables the secure processing of encrypted data, thereby ensuring that sensitive information remains private and inaccessible throughout the process.

V. CONCLUSION

The proposed decentralized biometric authentication system for car-sharing applications has potential positive impacts on privacy, security, convenience, and personal data control. Utilizing advanced cryptographic techniques, AI-driven biometric verification, and a user-centric design, the system aims to enhance security, streamline identity verification, comply with data protection regulations, and promote fairness. In the context of car-sharing, the approach is expected to improve security, user experience, competitiveness, regulatory compliance, scalability, interoperability, and corporate social responsibility. Additionally, the system could have significant implications for governments, impacting regulatory compliance, public trust, infrastructure, policy development, national security, and sustainable transportation. The research concludes by presenting a novel decentralized biometric authentication system, highlighting its accuracy, reliability, usability, scalability, and GDPR compliance. Future research can explore additional biometric modalities, advanced security measures, alternative storage and access control mechanisms, and improved interoperability to further enhance the performance and applicability of the system. By continuously refining and adapting the system, we can contribute to the development of a more secure, privacy-preserving, and user-centric authentication landscape for contactless car-sharing and other industries. Additionally, the data security and integrity protocols of car sharing companies need to be considered. There are privacy concerns regarding the data stored in the vehicle's black box camera and GPS device, particularly concerning car companies' access to information about lessees' locations, thus, renders them hesitant to utilize the car-sharing system, as it equates to sharing their data without their explicit knowledge.

ACKNOWLEDGMENT

This work was supported by a Research Grant of Pukyong National University (2023).

REFERENCES

[1] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Proceedings Advances in Cryptology-ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security*, Hong Kong, China, Dec. 2017, pp. 409-437.

[2] W. Yang, S. Wang, H. Cui, Z. Tang, and Y. Li, "A review of homomorphic encryption for privacy-preserving biometrics," *Sensors*, vol. 23, no. 7, p. 3566, Mar. 2023.

[3] M. Ghafourian, B. Sumer, R. Vera-Rodiguez, J. Fierrez, R. Tolosana, and A. Moralez, et al., "Combining blockchain and bio-metrics: A survey on technical aspects and a first legal analysis," Feb. 2023. https://arxiv.org/abs/2302.10883.

[4] N. Kapsoulis, A. Psychas, G. Palaiokrassas, A. Marinakis, A. Litke, and Varvarigou, T "Know your customer (KYC) implementation with smart contracts on a privacy-oriented decentralized architecture," *Future Internet*, vol. 12, no. 2, p. 41, Feb. 2020.

[5] M. Firdaus, S. Noh, Z. Qian, H. T. Larasati, and K. H. Rhee, "Personalized federated learning for heterogeneous data: A distributed edge clustering approach," *Mathematical Biosciences and Engineering*, vol. 20, no. 6, pp. 10725-10740, 2023.

[6] M. Firdaus, H. T. Larasati, and K. H. Rhee, "A secure federated learning framework using blockchain and differential privacy," in *Proceedings 2022 IEEE 9th International Conference on Cyber Security and Cloud Computing (CSCloud)/2022 IEEE 8th International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, Xi'an, China, Jun. 2022, pp. 18-23.

[7] O. Delgado-Mohatar, J. Fierrez, R. Tolosana, and R. Vera-Rodriguez, "Blockchain and biometrics: A first look into opportunities and challenges," in *Proeedings Blockchain and Applications: International Congress*, Avila, Spain, Jun. 2019, pp. 169-177.

[8] A. Othman and J. Callahan, "The horcrux protocol: A method for decentralized biometric-based self-sovereign identity," in *Proceedings 2018 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2018, pp. 1-7.

[9] J. E. Boritz, "IS practitioners' views on core concepts of information integrity," *International Journal of Accounting Information Systems*, vol. 6, no. 4, pp. 260-279, Dec. 2005.

[10] Y. Chen, "Blockchain tokens and the potential democratization of entrepreneurship and innovation," *Business horizons*, vol. 61, no. 4, pp. 567-575, Jul. 2018.

[11] S. A. Shaheen, A. P. Cohen, and J. D. Roberts, "Carsharing in North America," *Transportation Re-search Record*, vol. 1986, no. 1, pp. 116-124, Jan. 2006.

[12] A. Mühle, A. Grüner, T. Gayvoronskaya, and C. Meinel, "A survey on essential components of a self-sovereign identity," *Computer Science Review*, vol. 30, pp. 80-86, Nov. 2018.

[13] O. Avellaneda, A. Bachmann, A. Barbir, J. Brenan, P. Dingle, and K. H. Duffy, et al., "Decentralized identity: Where did it come from and where is it going?," *IEEE Communications Standards Magazine*, vol. 3, no. 4, pp. 10-13, Dec. 2019.

[14] J. Sedlmeir, R. Smethurst, A. Rieger, and G. Fridgen, "Digital identities and verifiable credentials," *Business & Information Systems Engineering*, vol. 63, no. 5, pp. 603-613, Oct. 2021.

[15] M. Firdaus and K. H. Rhee, "Towards trustworthy collaborative healthcare data sharing," in *Proceedings 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Istanbul, Türkiye, Dec. 2023, pp. 4059-4064.

[16] N. Radziwill, "Blockchain revolution: How the technology behind bitcoin is changing money, business, and the world," *The Quality Management Journal*, vol. 25, no. 1, pp. 64-65, Jan. 2018.

[17] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in *Proceedings Financial Cryptography and Data Security: 21st International Conference*, Sliema, Malta, Apr. 2017, pp. 357-375.

[18] M. Firdaus, H. Tatimma Larasati, and K. H. Rhee, "A blockchain-assisted distributed edge intelligence for privacy-preserving vehicular networks," *Computers, Materials, and Continua*, vol. 76, no. 3, pp. 2959-2978, 2003

[19] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the forty-first annual ACM symposium on Theory of Computing*, May 2009, pp. 169-178.

[20] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (Standard) LWE," *SIAM Journal on computing*, vol. 43, no. 2, pp. 831-871, Jan. 2014.

[21] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Proceedings Advances in Cryptology-CRYPTO 2013: 33rd Annual Cryptology Conference*, Santa Barbara, CA, Aug. 2013, pp. 75-92.

[22] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, Jun. 2015, pp. 815-823.

[23] S. I. Serengil and A. Ozpinar, "LightFace: A hybrid deep face recognition framework," in *Proceedings 2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Oct. 2020, pp. 1-5.

[24] L. Sperling, N. Ratha, A. Ross, and V. N. Boddeti, "HEFT: Homomorphically encrypted fusion of biometric templates," in *Proceedings 2022 IEEE International Joint Conference on Biometrics (IJCB)*, Istanbul, Türkiye, Oct. 2022, pp. 1-10.

[25] M. Gomez-Barrero, E. Maiorana, J. Galbally, P. Campisi, and J. Fierrez, "Multi-biometric template protection based on homomorphic encryption," *Pattern Recognition*, vol. 67, pp. 149-163, Jul. 2017.

[26] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, "Packed homomorphic encryption based on ideal lattices and its application to biometrics," in *Proceedings Security Engineering and Intelligence Informatics: CD-ARES 2013 Workshops: MoCrySEn and SeCIHD*, Regensburg, Germany, Sep. 2013, pp. 55-74.

[27] W. A. Alberto Torres, N. Bhattacharjee, and B. Srinivasan, "Privacy-preserving biometrics authentication systems using fully homomorphic encryption," *International Journal of Pervasive Computing and Communications,* vol. 11, no. 2, pp. 151-168, Jun. 2015.

[28] M. Firdaus and K. H. Rhee, "A joint Framework to privacy-preserving edge intelligence in vehicular networks," in *Proceedings International Conference on Information Security Applications,* Jeju, Korea, Aug. 2022, pp. 156-167.

[29] M. A. Alawami, D. Jung, Y. Park, Y. Ku, G. Choi, and K. W. Park, "The car is safe: A fast and accurate pressure-based authentication system for identifying car drivers," *Presented at the the 7th International Conference on Mobile Internet Security (MobiSec 2023)*, Okinawa, Japan, Dec. 2023.

AUTHORS

Saprunov Vadim received his bachelor's degree in Electrical Engineering from Pukyoung National University in 2021, followed by a master's degree in the Artificial Intelligence Convergence faculty at the same university in 2024. His research interests primarily focus on areas such as image detection, Know Your Customer (KYC), Anti-Money Laundering (AML), and full-stack development in general. Presently, he is employed at a company specializing in the development of AML solutions for banks.

Muhammad Firdaus received the M.S. degree from the School of Electrical Engineering and Informatics, Institut Teknologi Bandung (ITB), Indonesia, in 2019, and the Ph.D. degree from the Department of Artificial Intelligence (AI) Convergence at Pukyong National University (PKNU), Republic of Korea, in 2023. He is currently a Postdoctoral Researcher at the Lab. of Information Security and Internet Application (LISIA), PKNU. His research interests include applied cryptography, blockchain, federated learning, edge intelligence, vehicular networks, and security and privacy protection in wireless communications and networking.

Kyung-Hyune Rhee received his M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea, in 1985 and 1992, respectively. He was a senior researcher at the Electronic and Telecommunications Research Institute (ETRI) in the Republic of Korea from 1985 to 1993. He also worked as a visiting scholar at the University of Adelaide, the University of Tokyo, and the University of California, Irvine. He served as the Chairman of the Div. of Information and Communication Technology, Colombo Plan Staff College for Technician Education, Manila, Philippines. He is currently a professor at the Department of IT Convergence and Application Engineering, Pukyong National University, Republic of Korea. His research interests center on key management and its applications, mobile communication security, and the security evaluation of cryptographic algorithms.