Journal of Multimedia Information System VOL. 12, NO. 4, December 2025 (pp. 149-156): ISSN 2383-7632 (Online)
https://doi.org/10.33851/JMIS.2025.12.4.149

Asymmetric Key Construction Based on Palmprint Hash Coding

Zhengrong Liao[1,2], Jiafeng Hu[1,2], Lu Leng[1,2,*]

Abstract

To address two key challenges: the absence of high-precision hash algorithms in palmprint feature cryptography, and private key secure storage in asymmetric cryptography. This paper proposes an asymmetric key generation method based on palmprint hash coding. First, the method generates a variable-length palmprint hash template through a hash mapping network, and uses the Bose–Chaudhuri–Hocquenghem (BCH) error-correcting algorithm to achieve consistency recovery of each newly generated template. Then, it employs a Token to ensure key revocability, and combines with the Rivest-Shamir-Adleman (RSA) key generation algorithm to finally generate an asymmetric key strongly bound to the user's palmprint. Experimental results show that the method not only ensures the stable generation of keys but also achieves an Equal Error Rate (EER) of 0% on two public palmprint datasets. Additionally, the generated private key requires no additional storage and can be repeatedly regenerated through the palmprint and related processes, balancing security and practicality.

Key Words: Biometric Cryptography, Asymmetric Key Generation, Palmprint Hash Coding.

I. INTRODUCTION

Since their proposal, asymmetric encryption algorithms have been widely applied in fields such as encrypted communication, digital signatures, and blockchain. Common asymmetric encryption algorithms include Rivest-Shamir-Adleman (RSA, based on large integer factorization) and Elliptic curve cryptography (ECC, based on elliptic curve cryptography), etc. [1]. Due to security constraints, the private key in the asymmetric key pair must be secretly stored. However, it is extremely difficult to memorize and store a long key, and even if the key is stored in encrypted hardware, there remains a risk of leakage [2].

In the field of fingerprint contact-based recognition, law enforcement investigators or attackers can fabricate fingerprint replicas using low-cost materials such as gelatin and silicone. Such replicas can successfully deceive traditional fingerprint acquisition devices and feature extraction algorithms. Therefore, generating keys solely based on biometric features poses risks of attacks from the physical world.

To enhance the security of private key storage, the optimal approach is to avoid explicit storage. Consequently, biometric cryptosystems (BCS), which integrate biometric features with traditional cryptosystems, have emerged [3]. Early BCS adopted a key release strategy, where the key is only released upon successful user identity authentication. However, key release is vulnerable to database attacks, enabling attackers to bypass identity authentication and directly obtain the key. Subsequently, more secure and efficient key binding methods were proposed, with Fuzzy Commitment [4] and Fuzzy Vault [5] being two typical techniques. Fuzzy Commitment employs binary error-correcting codes to eliminate intra-class variations of biometric features and combines hash functions to release the exact key, while Fuzzy Vault utilizes polynomial functions to protect enrolled biometric features. Binding biometric features with asymmetric keys via the Fuzzy Commitment framework to achieve non-explicit private key storage constitutes a new challenge. This approach not only ensures the secure storage of private keys but also expands the application scope of biometrics.

In recent years, palmprint recognition has garnered widespread attention due to its advantages of high recognition efficiency and contactless acquisition mode. Neural networks based on scale [6] and attention mechanisms [7] have achieved better performance. However, many high-precision palmprint recognition networks adopt high-dimensional real-valued features as templates [8-10], which not

Manuscript received November 18, 2025; Revised November 29, 2025; Accepted December 03, 2025. (ID No. JMIS-25M-11-081)
Corresponding Author (*): Lu Leng, +86-791-86453251, leng@nchu.edu.cn
[1]Jiangxi Provincial Key Laboratory of Image Processing and Pattern Recognition, Nanchang Hangkong University, Nanchang 330063, China, fdecad@outlook.com, 13829966706@163.com, leng@nchu.edu.cn
[2]School of Software, Nanchang Hangkong University, 330063, China, fdecad@outlook.com, 13829966706@163.com, leng@nchu.edu.cn

only incur large storage requirements but also rely on cosine distance to measure similarity during matching. Their time complexity and space complexity are significantly higher than those of palmprint templates using short binary codes [11]. Therefore, based on a high-precision palmprint recognition network, this paper constructs a hash mapping network to map original high-dimensional features into short-bit binary codes while maximizing the retention of recognition accuracy. Meanwhile, Bose–Chaudhuri–Hocquenghem (BCH) error-correcting codes are employed to construct a fuzzy commitment for the high-precision binary templates, thereby obtaining an asymmetric key bound to palmprint features. The contributions of this paper are summarized as follows:

(1) A hash mapping network is proposed, adopting the architecture of a backbone network combined with a mapping network. It maximizes the retention of the accuracy of the original high-dimensional feature template and obtains a highly discriminative binary hash template.
(2) An asymmetric key generation framework based on fuzzy commitment is proposed. The extracted binary hash template is constructed through BCH error-correcting coding to ensure that each generated palmprint template is consistent with the original template.
(3) A user Token is employed to encrypt the template, which not only improves the security of the palmprint template but also enables the generated key to be renewable. Combined with the RSA key generation algorithm, an asymmetric key bound to the user's palmprint is constructed.

The rest of this paper is organized as follows. In Section 2, we briefly review the related wor⌞k, including deep hash networks and biometric cryptosystems. Section 3 elaborates on the proposed method in detail. Section 4 describes and discusses the experimental results. In Section 5, we conclude this work.

II. RELATED WORKS

In recent years, deep hash networks have continuously made breakthroughs in biometric recognition and template protection, providing key support for the security and efficiency of biometric cryptosystems. The recent research progress in related fields is summarized as follows:

2.1. Palmprint Hash Network

In 2023, Yang et al. [11] proposed the Competitive Palmprint Hash Network (CCNet), which captures unique texture features through learnable Gabor filters and integrates a comprehensive competition mechanism. Subsequently, Yang et al. [12] optimized the position of the hash layer based on CCNet, placing it after the fully connected layer to construct a new Hash Competition Network, further enhancing template protection capabilities. In 2024, Khan et al. [13] integrated chaotic sequences as weights into the neuron activation layer, supporting dynamic control to generate diverse palmprint feature templates. Chen et al. [14] proposed an enhanced multi-task learning framework that jointly optimizes identity recognition, soft biometric attribute classification, and hash branches, achieving template compression and matching acceleration while maintaining recognition performance. In 2025, Liu et al. [15] proposed a distillation-driven deep hash retrieval scheme, realizing lightweight and efficient retrieval and recognition with short bits in palmprint and finger vein recognition, verifying the feasibility and scalability of binary hash templates in hand biometrics. Liao et al. [16] developed a general framework suitable for texture-based tasks based on the second-order texture feature extraction method. Sai Kishore et al. [17] conducted a performance analysis of distillation-based hash palmprint recognition schemes, compressing deep models into lightweight hash encoders through knowledge distillation, and simultaneously improving recognition accuracy and inference speed in short-bit matching scenarios.

2.2. Biometric Cryptosystem

In the field of biometric cryptosystems, in 2022, Wu et al. [18] used deep hash coding as palmprint templates to construct a fuzzy commitment cryptosystem, and proposed a bit discriminative energy compression algorithm to balance the improvement of hash code accuracy and length compression. Suresh et al. [2] collected features through a fingerprint recognition system and input them into a random number generator to generate RSA public and private keys. In 2023, Barman et al. [19] adopted fuzzy commitment technology to securely store biometric data on remote servers, whose security was confirmed by two protocol verification tools. Lin et al. [20] proposed an error correction-based iris recognition scheme, selecting LDPC codes as error-correcting codes to achieve soft and reliable extraction. Shahreza et al. [21] processed extracted features through a user-specific random weighted Multilayer Perceptron (MLP) and binarized the output to generate protected biometric templates.

In 2024, Jide et al. [22] generated 256-bit private keys from binary features of fingerprint images, generated public keys using the ECDSA algorithm, and applied them to voting procedures. Cao et al. [23] proposed a fixed-length ordered extraction method fusing point features and

direction features, realizing efficient and secure palmprint template protection through index maximization transformation. Tran et al. [24] designed a multi-factor fuzzy extractor for key generation and reconstruction, and constructed a new multi-factor authentication key exchange protocol to enhance system security and key regeneration capabilities. In 2025, Yirga et al. [25] fused deep features of faces and finger veins, combined with fuzzy extractors to achieve reproducible key generation, and published stability and security evaluation indicators. Geißner et al. [26] proposed a multi-finger deep fuzzy commitment scheme, improving key regeneration stability and increasing attack difficulty through multi-sample aggregation. Almola et al. [27] proposed a fingerprint-based key generation system combined with deep learning, using dual CNNs to be responsible for identity verification and feature extraction respectively, enhancing key randomness through particle swarm optimization, and supporting engineering implementation of real-time derivation.

Existing research still faces key technical bottlenecks: most palmprint feature encryption algorithms, in pursuit of high security and recognition accuracy, often rely on complex network architectures. This leads to numerous matrix computations during the encryption process, resulting in high time complexity. The problem is particularly prominent in applications under resource-constrained scenarios. Therefore, there is still a lack of a palmprint feature encryption algorithm that can balance encryption security, feature matching accuracy, and engineering practicality with low time complexity.

III. METHODS

3.1. Hash Network

To address the problem that the high-dimensional output of current feature extraction networks makes it difficult to maintain high accuracy while reducing dimensionality, we have specifically designed a residual hash module to compress high-dimensional palmprint features. Benefiting from the high discriminability of the CCNet [11], this paper adopts it as the backbone network to extract original palmprint features. As shown in Fig. 1, the palmprint ROI generates 2048-dimensional real-valued features after passing through the trained feature extraction network. These features then go through three fully connected layers to further extract the relationships between them, with ReLU activation functions used between the fully connected layers and a Tanh activation function employed in the final hash layer to simulate binary output. The hash module can be expressed as:

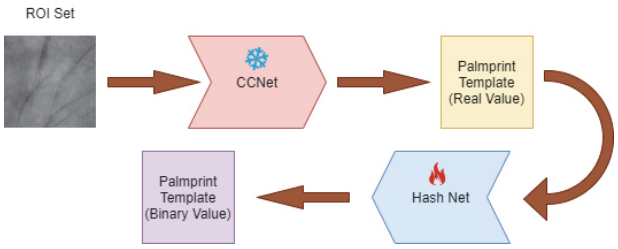


Fig. 1. Hash Network Framework.

$$f(x_i) = H_i, \qquad H \in \{0,1\}^L. \tag{1}$$

Among them, $x_i \in R^N$ denotes the high-dimensional feature vector of a specific palmprint, $f(\cdot)$ represents the mapping function, and L stands for the length of the output binary code.

To ensure the inter-class discriminability and bit randomness of the output binary codes during the training of the mapping network, we adopt a Cryptographically Secure Pseudo-Random Number Generator (CSPRNG) to generate high-entropy random binary codes (SPC), which serve as class centers. To maximize the separation of these class centers and further strengthen the aforementioned design objectives, we additionally set a threshold T as the minimum Hamming distance between different SPCs. Specifically, after being mapped to the Hamming space, the hash codes of the same class gather within a Hamming sphere centered at the corresponding class center with a specific radius as the boundary, ensuring high similarity among intra-class hash codes. For the class centers of different classes, we impose strict constraints to ensure their minimum Hamming distance $d \geq T$ thereby achieving effective inter-class separation.

Loss Function: In the training scenario of our mapping network, the ground truth is regarded as the class center represented by the high-entropy random binary codes generated by CSPRNG, while the predicted value refers to the hash code actually generated by the network. By minimizing the L2 loss, we can drive the generated hash codes to be as close as possible to the class center of their respective classes. The L2 loss is defined as follows:

$$L_2 = \frac{1}{N}\sum_{i=0}^{N}\|Y_i - H_i\|^2. \tag{2}$$

Y_i, H_i denote the output hash code of the mapping network and the class center binary code, respectively, and N represents the batch size. To prevent network overfitting, we also add a dropout layer to the fully connected layer of the last layer, with the dropout parameter set to 0.5.

3.2. Key Seed Generation

The framework of the entire algorithm is illustrated in

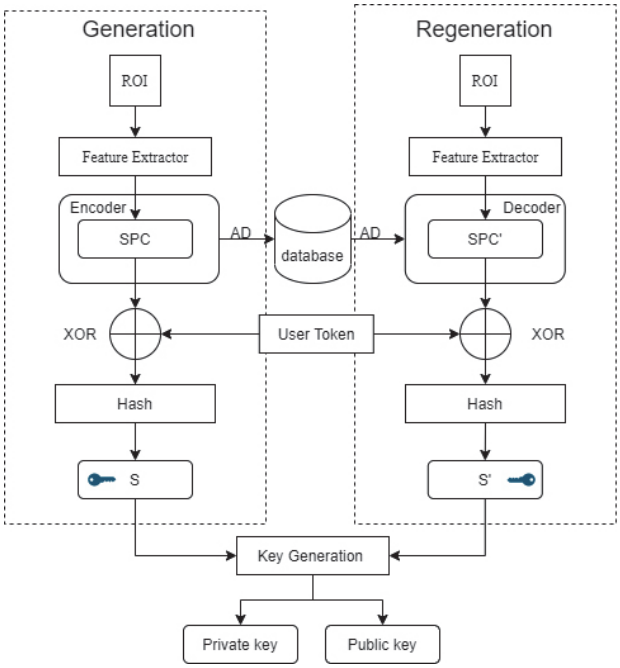


Fig. 2. Asymmetric key generation framework.

Fig. 2, comprising two phases: Key generation and Key regeneration. During enrollment, the user provides a palmprint image to generate the feature vector SPC. After the SPC undergoes the error-correcting coding construction algorithm, auxiliary data (AD) is generated, which is used for decoding during subsequent key regeneration. To further ensure the security of the key, the user needs to provide a Token to encrypt the original feature vector, defending against credential stuffing attacks. Finally, a one-way hash encryption algorithm is adopted to ensure that the generated key seed S cannot be reversed to recover the user's original feature vector even if compromised. Eventually, S is input into the asymmetric key generator to generate the public key and private key. During the key regeneration phase, the feature vector generated by decoding the AD stored in the database is used to recover the SPC from enrollment, and the remaining processes are consistent with those in the generation phase.

The error-correcting coding algorithm adopts BCH codes. Three parameters need to be specified when constructing the error-correcting coding algorithm: code length n, information bit length k, and number of correctable bits t. Typically, $n = 2^m - 1, m \in \mathbb{Z}^+, k \leq n - 2t$. The error-correcting capability of BCH codes is determined by the design parameter t, which can theoretically correct up to t random errors. Its error-correcting capability satisfies the relationship with the minimum Hamming distance: $d_{min} \geq 2t + 1$. To maximize the error-correcting capability of BCH codes and minimize redundancy, the hash mapping network presented in Section 3.1 is used to generate fixed-length SPCs suitable for BCH coding. Table 1 provides reference parameters for n, k, t, and SPC length. During user enrollment, the construction algorithm is invoked to generate parity bits, which are then stored in the database. During key regeneration, the BCH decoding algorithm is utilized.

Table 1. Reference BCH Parameters.

n	k (SPC)	t
127	57	11
255	99	23
511	211	41

To ensure that the key seed S does not reveal any information about the original palmprint, a User Token is used to perform an XOR operation on it. The User Token has the same length as the codeword encoded by the BCH algorithm. The User Token needs to be generated from a password provided by the user, and the provided password must comply with the NIST [28] protocol. The PBKDF (Password-Based Key Derivation Function) algorithm is adopted to perform multiple iterations and salting calculations on the input password, ultimately generating the User Token.

Untransformed palmprint feature templates are vulnerable to credential stuffing attacks and reconstruction attacks. [29] Therefore, we adopt a hash algorithm to perform one-way mapping on the generated palmprint feature templates, yielding the key seed S or S'.

3.3. Asymmetric Key Construction

The key seed S is split into S1 and S2, which are then input into a Pseudorandom Number Generator (PRNG) to generate two large integers. To ensure the security of the RSA key, the generated large integers are set to 1,024 bits in length, and the last bit is set to 1 to ensure they are odd numbers.

After generating the two large integers, an efficient probabilistic algorithm (Miller-Rabin Primality Test) is adopted to check if they are large primes. If not, the large integer is incremented by 2 and the check is repeated. After iterative verification, two large primes p and q are obtained.

At this point, based on the aforementioned large primes p and q, the RSA asymmetric key pair can be constructed. First, the modulus n is calculated, which is the product of the two large primes and serves as a common component of both the public key and the private key:

$$n = pq. \tag{3}$$

Next, Euler's totient function $\phi(n)$ is calculated. This function describes the number of positive integers coprime with n and serves as a key parameter for key generation:

$$\phi(n) = (p - 1)(q - 1). \tag{4}$$

Subsequently, an integer e is selected as the public key exponent, and its selection must satisfy two conditions:

1. $1 < e < \phi(n)$
2. e is coprime with $\phi(n)$, i.e., $\gcd(e, \phi(n)) = 1$, where gcd denotes the greatest common divisor.

Finally, the private key exponent d is calculated, which is the modular inverse of the public key exponent e under modulo $\phi(n)$. This means that d must satisfy the following congruence equation:

$$d \equiv e^{-1}(mod\ \phi(n)). \quad (5)$$

The final resulting key pair is: public key (n, e) and private key (d, n).

IV. RESULTS AND DISCUSSION

4.1. Dataset and Experimental Settings

The proposed method is evaluated on two public palmprint datasets: PolyU [30] and Tongji [31]. Among them, the Tongji dataset consists of palmprint data collected by a non-contact device, while the PolyU dataset is collected through contact-based acquisition. Fig. 3 shows the ROI example for PolyU and Tongji.

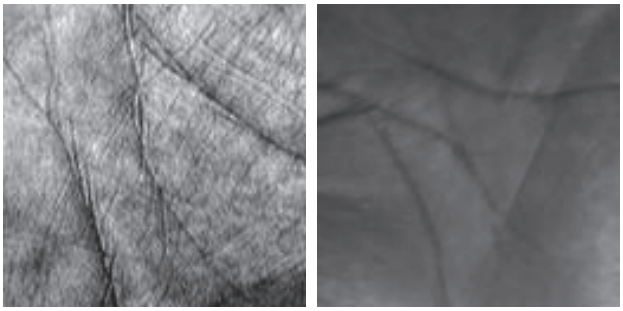
The hash network is implemented using PyTorch. From the state-of-the-art (SOTA) palmprint recognition models based on deep learning, CCNet is selected as the backbone network for palmprint feature extraction. CCNet is trained with the parameter settings as described in the original paper. For all datasets, the number of training epochs is set to 1,000 and the ratio of the training set to the test set is 1:1. The residual mapping network is trained with 500 epochs, using the same training and test sets as CCNet. The Adam optimizer is adopted with a learning rate of 0.001.

The hardware used in this study includes an Intel(R) Xeon(R) Platinum 8222CL CPU @ 3.00 GHz, 64 GB of memory, and an NVIDIA GTX 3090 GPU.

4.2. Verification Experiment

The hash network proposed in Section 3.1 is used for hash mapping on two public palmprint datasets (PolyU and Tongji). Table 2 presents the Equal Error Rate (EER%) as well as the Minimum and Maximum Hamming distances for Genuine users and Imposters under different datasets and various bit lengths.

As can be seen from Table 2, the overlap of matching results between Genuine and Imposter only occurs for the 57-bit hash codes of the PolyU dataset. The Equal Error Rate (EER) is determined when the False Acceptance Rate (FAR) equals the False Rejection Rate (FRR), and the corresponding threshold



(a) PolyU (b) Tongji

Fig. 3. The ROI image of PolyU and Tongji.

Table 2. Hash mapping results.

Bits	Category	PolyU		
		Min-d	Max-d	EER%
57	Genuine	0	20	0.04
	Imposter	16	45	
99	Genuine	0	6	0
	Imposter	36	73	
211	Genuine	0	15	0
	Imposter	82	138	
		Tongji		
57	Genuine	0	6	0
	Imposter	15	45	
99	Genuine	0	12	0
	Imposter	34	73	
211	Genuine	0	24	0
	Imposter	79	139	

is adopted as the value of parameter t for BCH code construction. If the EER is 0, the maximum Hamming distance of Genuine users is directly used as the threshold, i.e., parameter t. Table 3 presents the thresholds and recommended BCH code parameters for different datasets under various bit lengths. With a fixed information bit length k, if the threshold is less than the number of correctable bits t, it indicates that the BCH code can fully recover the original palmprint feature template without causing overlap.

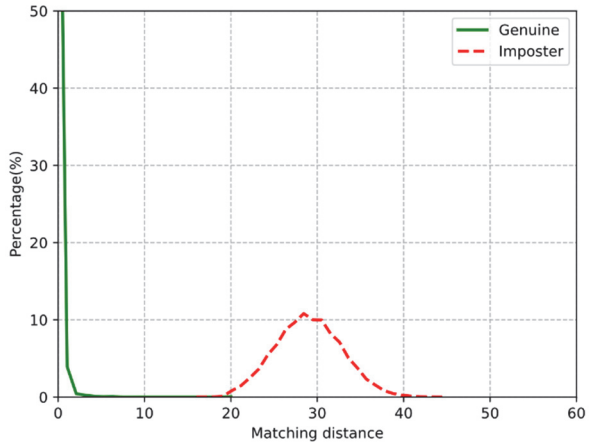
Fig. 4 presents the matching distance distribution diagram of the two datasets at a bit length of 57. It can be observed from the figure that only the PolyU dataset exhibits an overlap between Genuine (genuine samples) and Imposter (imposter samples), while the Tongji dataset is completely separated with a significant degree of separation between the two curves. This demonstrates that our method remains highly effective even at short bit lengths.

4.3. Security of palmprint template

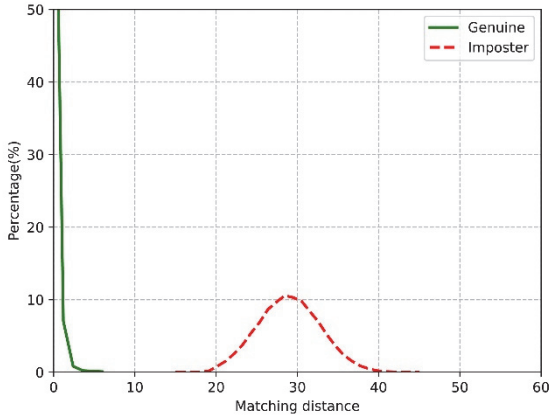
Only the parity bits generated during the BCH encoding process is stored in the database. These parity bits do not

Table 3. Recommended parameters and EER.

Dataset	Bits	Threshold	EER%	BCH parameter (n, k, t)
PolyU	57	**18**	0.0352	(127, 57, 11)
	99	**6**	0	(255, 99, 23)
	211	**15**	0	(511, 211, 41)
Tongji	57	**6**	0	(127, 57, 11)
	99	**12**	0	(255, 99, 23)
	211	**24**	0	(511, 211, 41)



(a) PolyU



(b) Tongji

Fig. 4. The matching distance distribution of PolyU and Tongji with 57 bits length.

contain any information related to biometrics, thereby ensuring the security of the biometric template and resisting brute-force search attacks and cross-matching attacks.

In a brute-force search attack, the attacker has no knowledge of the key generation method or the auxiliary information stored in the database. To find the correct key, the number of searches required by the attacker equals the length of the generated key. If the key length is L, the number of searches for the attacker is 2^{L-1}. The RSA key generated in this paper has a length of 2,048 bits, which is sufficient to resist such attacks.

In a cross-matching attack, the attacker may access the user's palmprint features stored in other databases. Even if the attacker obtains the user's palmprint features, they still need the user's Token to generate the key. Without acquiring the user's Token, the attacker remains unable to obtain the key.

V. CONCLUSION

This paper proposes an asymmetric key generation algorithm based on palmprint hash coding, which successfully addresses the security storage challenge of private keys in asymmetric cryptography. The scheme eliminates the need for additional private key storage and enables on-demand key regeneration solely through the user's palmprint and Token, significantly reducing the risk of private key leakage. The designed hash mapping network features both low time complexity and high accuracy, efficiently completing the hash coding and key generation processes. In addition, the adoption of a variable-length hash template design not only allows flexible adjustment of the template length to adapt to different application environments but also enables personalized adaptation to BCH error-correcting code parameters, enhancing the algorithm's practicality and compatibility.

ACKNOWLEDGEMENT

This study was funded by National Natural Science Foundation of China (62466038), Jiangxi Provincial Key Laboratory of Image Processing and Pattern Recognition (2024SSY03111), Jiangxi Provincial Natural Science Foundation (Key Program) (No. 20242BAB26015), Open Foundation of Jiangxi Provincial Key Laboratory of Image Processing and Pattern Recognition (ET202404437), and Innovation Foundation for Postgraduate Students of Nanchang Hangkong University (YC2024-117).

REFERENCES

[1] F. Mallouli, A. Hellal, N. S. Saeed and F. A. Alzahrani, "A survey on cryptography: Comparative study between RSA vs ECC algorithms and RSA vs El-Gamal algorithms," in *Proceedings of the IEEE International Conference on Cyber Security and Cloud Computing*, Dubai, UAE, 2019, pp. 173-17.

[2] K. Suresh, P. Rajarshi, and S. Balasundaram, "Two-factor-based RSA key generation from fingerprint biometrics and password for secure communication," *Journal of Complex & Intelligent Systems*, vol. 8, no. 4, pp.

3247-3261, 2022.

[3] K. Prabhjot, K. Nitin, and S. Maheep, "Biometric cryptosystems: A comprehensive survey," *Multimedia Tools and Applications*, vol. 82, no. 11, pp. 16635-16690, 2022.

[4] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proceedings of the ACM Conference on Computer and Communications Security*, Singapore, 1999, pp. 28-36.

[5] A. Juels and M. Sudan, "A fuzzy vault scheme," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 237-257, 2006.

[6] Y. Lee and B. Kim, "Attention-based scale sequence network for small object detection," *Heliyon*, vol. 10, p. e32931, 2024.

[7] H. Park, J. Kang, and B. Kim, "ssFPN: Scale sequence (s2) feature-based feature pyramid network for object detection," *Sensors*, vol. 23, no. 9, p. 4432, 2023.

[8] H. Lu, C. Sheng, and W. Jia, "Palmprint recognition method based on orientation features: A survey," in *Presented at the 18th Chinese Conference on Biometric Recognition*, Singapore, Feb. 2025.

[9] Z. Yang, W. Xia, Y. Qiao, Z. Lu, B. Zhang, and L. Leng, et al., "CO3Net: Coordinate-aware contrastive competitive neural network for palmprint recognition," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1-14, 2023.

[10] Y. Liu, L. Leng, Z. Yang, A. B. J. Teoh, and B. Zhang, "SF2Net: Sequence feature fusion network for palmprint verification," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 9936-9949, 2025.

[11] Z. Yang, H. Huang, L. Leng, B. Zhang, A. B. J. Teoh, and Y. Zhang, "Comprehensive competition mechanism in palmprint recognition," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 5160-5170, 2023.

[12] Z. Yang, M. Kang, A. B. J. Teoh, C. R. Gao, W. Chen, and B. Zhang, et al., "A dual-level cancelable framework for palmprint verification and hack-proof data storage," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 8587-8599, 2024.

[13] K. Md, H. Li, and Z. Chuan, "Deep secure PalmNet: A novel cancelable palmprint template protection scheme with deep attention network and randomized hashing security mechanism," *Computers & Security*, vol. 142, p. 103863, 2024.

[14] L. Chen, L. Leng, Z. Yang, and A. B. J. Teoh, "Enhanced multitask learning for hash code generation of palmprint biometrics," *Neural Networks*, vol. 34, no. 4, pp. 1-15, 2024.

[15] C. Liu, L. Yang, W. Zhou, Y. Li, and F. Hao, "Deep distillation hashing for palmprint and finger vein retrieval," *IET Biometrics*, vol. 14, no. 1, pp. 1-12, 2025.

[16] F. Liao, L. Leng, Z. Yang and B. Zhang, "Multi-order extension codes for palmprint recognition," *Neural Networks*, vol. 35, no. 8, p. 2550039, 2025.

[17] T. S. Kishore, and S. M. Kumar, "Performance analysis of biometric palm print identification system using novel distilled hashing technique," in *AIP Conference Proceedings on Biometrics & Security Technology*, Bangalore, India, 2025, vol. 3318, pp. 020241-020241.

[18] T. Wu, L. Leng, and M. Muhan, "A multi-spectral palmprint fuzzy commitment based on deep hashing code with discriminative bit selection," *Artificial Intelligence Review*, vol. 56, no. 7, pp. 6169-6186, 2022.

[19] S. Barman, S. Chattopadhyay, and D. Samanta, "Toward design a secure protocol for updating remotely stored credentials of a crypto-biometric framework for multi-server environment," *Security and Privacy*, vol. 7, no. 1, p. e339, 2023.

[20] K. Kuochun and Y. Yenming, "A high-security-level iris cryptosystem based on fuzzy commitment and soft reliability extraction," *IEEE Internet of Things Journal*, vol. 10, no. 12, pp. 1-15, 2023.

[21] H. Shahreza, V. Vedrana, and S. Marcel, "MLP-hash: Protecting face templates via hashing of randomized multi-layer perceptron," in *Proceedings of the European Signal Processing Conference*, Seville, Spain, pp. 605-609, 2023.

[22] J. K. Adeniyi, S. A. Ajagbe, E. A. Adeniyi, P. Mudali, M. O. Adigun, and T. T. Adeniyi, et al., "A biometrics-generated private/public key cryptography for a blockchain-based e-voting system," *Egyptian Informatics Journal*, vol. 25, p. 100447, 2024.

[23] Z. Cao, W. Zhao, H. Zhao, and L. Pang, "Composite fixed-length ordered features with index-of-max transformation for high-performing and secure palmprint template protection," *Information and Intelligence*, vol. 8, no. 2, pp. 1-10, 2024.

[24] H. Y. Tran, J. Hu, and W. Hu, "Biometrics-based authenticated key exchange with multi-factor fuzzy extractor," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 9344-9358, 2024.

[25] T. G. Yirga, H. G. Yirga, and E. G. Addisu, "Cryptographic key generation using deep learning with biometric face and finger vein data," *Frontiers in Artificial Intelligence*, vol. 8, p. 1545946, 2025.

[26] H. Geißner, V. Fohr, and C. Rathgeb, "Deep multi-finger fuzzy commitment," in *Proceedings of the European Signal Processing Conference*, Berlin, Germany, 2025, pp. 1342-1346.

[27] S. A. S. Almola, R. S. Khudeyer, and H. A. Younis,

"Biometric-based secure encryption key generation using convolutional neural networks and particle swarm optimization," *Informatica*, vol. 49, no. 16, pp. 1-18, 2025.

[28] P. A. Grassi, E. M. Newton, R. A. Perlner, A. R. Regenscheid, W. E. Burr, and J. P. Richer, et al., "Digital identity guidelines: Authentication and lifecycle management," *NIST Special Publication 800-63B*, 2017.

[29] L. Yan, F. Wang, L. Leng, and A. B. J. Teoh, "Toward comprehensive and effective palmprint reconstruction attack," *Pattern Recognition*, vol. 255, p. 110655, 2024.

[30] X. Wu, D. Zhang, and K. Q. Wang, "Fisher palms based palmprint recognition," *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2829-2838, 2003.

[31] L. Zhang, L. Li, A. Yang, Y. Shen, and M. Yang, "Towards contactless palmprint recognition: A novel device, a new benchmark and a collaborative representation-based identification approach," *Pattern Recognition*, vol. 69, pp. 199-212, 2017.

AUTHORS

Zhengrong Liao is pursuing his master degree in School of Software, Nanchang Hangkong University.
His main research interests focus on palmprint recognition and the field of palmprint feature cryptography.

Jiafeng Hu is pursuing his master degree in School of Software, Nanchang Hangkong University.
His main research interests focus on palmprint recognition and biometric template security.

Lu Leng received his Ph.D degree from Southwest Jiaotong University, Chengdu, P. R. China, in 2012. He performed his postdoctoral research at Yonsei University, Seoul, South Korea, and Nanjing University of Aeronautics and Astronautics, Nanjing, P. R. China. He was a visiting scholar at West Virginia University, USA, and Yonsei University, South Korea. Currently, he is a full professor, the dean of Institute of Computer Vision at Nanchang Hangkong University. Prof. Leng has published more than 150 international journal and conference papers, including more than 80 SCI papers. He has been granted several scholarships and funding projects, including six projects supported by National Natural Science Foundation of China (NSFC). He serves as a reviewer of more than 100 international journals and conferences. His research interests include computer vision, biometric template protection, biometric recognition, medical image processing, data hiding, etc.