A Quantum-Resistant Face Template Protection Scheme using Kyber and Saber Public Key Encryption Algorithms

Roberto Román, Rosario Arjona, Paula López-González, Iluminada Baturone
Instituto de Microelectrónica de Sevilla (IMSE-CNM)
University of Seville, CSIC
Seville, Spain
{roman, arjona, paula, lumi}@imse-cnm.csic.es

***Abstract*— Considered sensitive information by the ISO/IEC 24745, biometric data should be stored and used in a protected way. If not, privacy and security of end-users can be compromised. Also, the advent of quantum computers demands quantum-resistant solutions. This work proposes the use of Kyber and Saber public key encryption (PKE) algorithms together with homomorphic encryption (HE) in a face recognition system. Kyber and Saber, both based on lattice cryptography, were two finalists of the third round of NIST post-quantum cryptography standardization process. After the third round was completed, Kyber was selected as the PKE algorithm to be standardized. Experimental results show that recognition performance of the non-protected face recognition system is preserved with the protection, achieving smaller sizes of protected templates and keys, and shorter execution times than other HE schemes reported in literature that employ lattices. The parameter sets considered achieve security levels of 128, 192 and 256 bits.**

Keywords*— *Homomorphic Encryption, Kyber, Saber, Post-Quantum Cryptography, Biometric Template Protection, Face Recognition.

I. INTRODUCTION

Face recognition systems are popular for being easy to deploy and implement. However, as other biometric systems, they need to be implemented carefully since biometric data are considered sensitive by the European Union in the General Data Protection Regulation (GDPR) 2016/679 [1]. Therefore, biometric template protection schemes should be applied to handle the security and privacy of templates, as defined by the ISO/IEC IS 24745 standard [2].

Protection of biometric templates can be achieved by systems based on Homomorphic Encryption (HE). Biometric references and queries are compared in the encrypted domain and, thus, sensitive data are not revealed. To gain widespread, these systems need to be efficient. Also, quantum-resistant solutions need to be explored to have long-term security. The objective of post-quantum cryptography, also known as quantum-resistant cryptography, is to develop cryptographic solutions resistant to attacks executed from both quantum and classical computers, and able to operate with existing communications protocols and networks.

The National Institute of Standards and Technology (NIST) initiated at the end of 2016 a contest to standardize quantum-resistant cryptographic solutions. At time of writing this paper, the third round of the standardization process has been completed and one public-key encryption and key-establishment algorithm has been selected for its standardization: Crystals-Kyber, which will be referred to herein as Kyber [3]. This scheme is based on lattice cryptography and, concretely, the Module Learning-With-Errors (M-LWE) problem and has interesting homomorphic properties. Classic McEliece, whose security has remained stable over 40 years, BIKE, HQC and SIKE are the remaining schemes that are in the fourth round of the contest, and can be standardized in the future, but they must be better checked. Classic McEliece, BIKE and HQC are based on code-based cryptography, and SIKE is based on pseudo-random walks in supersingular isogeny graphs. An efficient key recovery attack on SIKE-protected transactions has been published recently [4]. Saber, a lattice-based cryptographic scheme whose security is based on the Module Learning-With-Rounding (M-LWR) problem, has been a finalist in the third round of the contest, and has interesting homomorphic properties like Kyber.

Regarding proposals based on homomorphic encryption and lattice cryptography in the literature, [5] presented biometric template protection schemes based on ideal lattices and Ring Learning-With-Errors (R-LWE) for binary biometric data, [6] used R-LWE for floating-point biometric data with CKKS and integer biometric data with BFV in an identification scenario, [7] used R-LWE for binary biometric data with NTRU, [8] studied the efficiency of CKKS, BFV and NTRU, [9] also used BFV, and [10] presented a variant to improve the efficiency of one of the proposals presented in [8]. In [11], it is studied a coefficient packing technique with CKKS, BFV and NTRU to reduce workload in a face identification scenario. Concerning other quantum-resistant solutions, in [12] it is reported the first biometric template protection scheme using the Learning Parity with Noise (LPN) problem. Although most of these proposals are based on lattice cryptography, none of them employ the finalists of the NIST contest. Note that a submission called "NTRU" was in

This research was conducted thanks to Grant PDC2021-121589-I00 funded by MCIN/AEI/10.13039/501100011033 and the "European Union NextGenerationEU/PRTR", and Grant PID2020-119397RB-I00 funded by MCIN/AEI/ 10.13039/501100011033. The work of Roberto Román was supported by VI Plan Propio de Investigación y Transferencia through the University of Seville.

the third round of the NIST's competition for standardizing post-quantum resistant cryptographic schemes, but this scheme is not the same as the original NTRU used in [7], [9] or [11], which has homomorphic properties.

In this work, we focus on exploiting the advantages of Kyber and Saber to further improve the efficiency of biometric template protection schemes. Kyber is particularly fast since it allows using fast multiplication based on the negacyclic number-theoretic transform (NTT). Saber uses a combination of Toom-Cook and Karatsuba polynomial multiplication algorithms. This paper is structured as follows: Section 2 presents the proposed biometric template protection scheme based on Kyber and Saber, Section 3 shows the experimental results that support the suitability of the proposal, and Section 4 concludes the work. Face recognition using FaceNet embeddings is selected since the deployment of these systems is easy among smartphone users [13].

II. Proposed Biometric Template Protection Scheme

A. Kyber and Saber public key encryption algorithms

First Kyber [14] and Saber [15] are public-key encryption schemes defined by key generation, encryption, and decryption algorithms. These algorithms are given in Tables I, II, and III, respectively, for Kyber and Saber (using a similar notation for both of them). The key-generation algorithm returns a pair (pk, sk) consisting of a public key and a secret key, respectively. The encryption algorithm takes a public key pk and a message $m \in \mathcal{M}$ (with $\mathcal{M} \in \{0,1\}^n$) to produce a ciphertext c. Finally, the decryption algorithm takes a secret key sk and a ciphertext c, and outputs either a message $m \in \mathcal{M}$ or a special symbol $\perp$ to indicate rejection.

Let $\mathbb{Z}_q$ denote the ring of integers modulo an integer q, and R and R_q denote the rings $\mathbb{Z}[X]/(X^n+1)$ and $\mathbb{Z}_q[X]/(X^n+1)$, respectively, where $n = 2^{n'-1}$ such that X^n+1 is the $2^{n'}$-th cyclotomic polynomial. The notation employed is that regular font letters denote elements in R or R_q (which includes elements in $\mathbb{Z}$ and $\mathbb{Z}_q$), bold lower-case letters represent vectors with coefficients in R or R_q, and bold upper-case letters are matrices. For a vector $\mathbf{t}$ (or matrix $\mathbf{A}$), we denote by $\mathbf{t}^T$ (or $\mathbf{A}^T$) its transpose. β_η is a centered binomial distribution with parameter η, where η is even, and the samples are in the interval $[-\eta/2, \eta/2)$. Sam is an extendable output function XOF (its output can be extended to any desired length). $y \backsim S := \mathrm{Sam}(x)$ means that Sam takes as input x and produces a value y that is distributed according to distribution S. If the distribution is uniform for a value r, then the notation is $r \longleftarrow \{0,1\}^n$. The operator $\lceil x \rfloor$ denotes rounding to the x nearest integer.

In Kyber, $\mathrm{Compress}_q(x,d)$ takes an input $x \in \mathbb{Z}_q$ and outputs an integer in $\{0, \dots, 2^d-1\}$, where $d < \lceil log_2(q) \rceil$; if $x \in R_q^k$, the procedure is applied to each coefficient individually. $\mathrm{Decompress}_q(x,d)$ takes an input $x \in \mathbb{Z}_q$ and outputs $\lceil (q/2^d) \cdot x \rfloor$; if $x \in R_q^k$, the procedure is applied to each coefficient individually. $\mathrm{Compress}_q(v - \mathbf{s}^T \cdot \mathbf{u}, 1)$ outputs 1 if $v - \mathbf{s}^T \cdot \mathbf{u}$ is closer to $\lceil q/2 \rfloor$, and 0 otherwise.

In Saber, $q = 2^{\epsilon_q}$, $p = 2^{\epsilon_p}$, and $T = 2^{\epsilon_T}$. Higher values for parameters p and T result in lower security, but higher correctness. Saber uses the constant polynomial $h_1 \in R_q$, which has all its coefficients equal to $2^{\epsilon_q - \epsilon_p - 1}$, a constant vector $\boldsymbol{h} \in R_q^k$ where each polynomial is equal to h_1 and a

TABLE I. Steps of Kyber and Saber Key-Generation Algorithms

Kyber KeyGen()		**Saber KeyGen**()	
1.	$\rho, \sigma \longleftarrow \{0,1\}^n$	1.	$\rho, \sigma \longleftarrow \{0,1\}^n$
2.	$\mathbf{A} \backsim R_q^{kxk} := \mathrm{Sam}(\rho)$	2.	$\mathbf{A} \backsim R_q^{kxk} := \mathrm{Sam}(\rho)$
3.	$(\mathbf{s}, \mathbf{e}) \backsim \beta_\eta^k \times \beta_\eta^k := \mathrm{Sam}(\sigma)$	3.	$\mathbf{s} \backsim \beta_\eta^k := \mathrm{Sam}(\sigma)$
4.	$\mathbf{t} := \mathrm{Compress}_q(\mathbf{A} \cdot \mathbf{s} + \mathbf{e}, d_t)$	4.	$\mathbf{t} := (\mathbf{A}^T \cdot \mathbf{s} + \boldsymbol{h})_q \gg (\epsilon_q - \epsilon_p) \in R_p^k$
5.	**return** pk $:= (\mathbf{t}, \rho)$, sk $:= \mathbf{s}$	5.	**return** pk $:= (\mathbf{t}, \rho)$, sk $:= \mathbf{s}$

TABLE II. Steps of Kyber and Saber Encryption Algorithms

Kyber Enc(pk $= (\mathbf{t}, \rho), m \in \mathcal{M}$)		**Saber Enc**(pk $= (\mathbf{t}, \rho), m \in \mathcal{M}$)	
1.	$r \longleftarrow \{0,1\}^n$	1.	if r is not specified then $r \longleftarrow \{0,1\}^n$
2.	$\mathbf{t} := \mathrm{Decompress}_{\boldsymbol{q}}(\mathbf{t}, d_t)$	2.	$\mathbf{A} \backsim R_q^{kxk} := \mathrm{Sam}(\rho)$
3.	$\mathbf{A} \backsim R_q^{kxk} := \mathrm{Sam}(\rho)$	3.	$\mathbf{r} \backsim \beta_\eta^k := \mathrm{Sam}(r)$
4.	$(\mathbf{r}, \boldsymbol{e_1}, e_2) \backsim \beta_\eta^k \times \beta_\eta^k \times \beta_\eta := \mathrm{Sam}(r)$	4.	$\mathbf{u} := (\mathbf{A} \cdot \mathbf{r} + \boldsymbol{h})_q \gg (\epsilon_q - \epsilon_p) \in R_p^k$
5.	$\mathbf{u} := \mathrm{Compress}_q(\mathbf{A}^T \cdot \mathbf{r} + \boldsymbol{e_1}, d_u)$	5.	$v' := \mathbf{t}^T \cdot \mathbf{r}_p \in R_p$
6.	$v := \mathrm{Compress}_q(\mathbf{t}^T \cdot \mathbf{r} + e_2 + \lfloor q/2 \rfloor \cdot m, d_v)$	6.	$v := (v' + h_1 - (2^{\epsilon_p - 1} \cdot m)_p) \gg (\epsilon_p - \epsilon_T) \in R_T$
7.	**return** c $:= (\mathbf{u}, v)$	5.	**return** c $:= (\mathbf{u}, v)$

TABLE III. Steps of Kyber and Saber Decryption Algorithms

Kyber Dec(sk $= \mathbf{s}$, c $= (\mathbf{u}, v)$)		**Saber Dec**(sk $= \mathbf{s}$, c $= (\mathbf{u}, v)$)	
1.	$\mathbf{u} := \mathrm{Decompress}_q(\mathbf{u}, d_u)$	1.	$v' := \mathbf{u}^T \cdot \mathbf{s}_p \in R_p$
2.	$v := \mathrm{Decompress}_q(v, d_v)$	2.	$m' := (v' + h_2 - 2^{\epsilon_p - \epsilon_T} \cdot v)_p \gg (\epsilon_p - 1) \in R_2$
3.	$m' := \mathrm{Compress}_q(v - \mathbf{s}^T \cdot \mathbf{u}, 1)$		
4.	**return** m'	3.	**return** m'

constant polynomial $h_2 \in R_q$ with all its coefficients equal to $2^{\epsilon_p-2} + 2^{\epsilon_p-\epsilon_T-1} + 2^{\epsilon_q-\epsilon_p-1}$. These constants are used to replace rounding operations by simple shift operations >> (to the right) or << (to the left), which can be bitwise and extended to polynomials and matrices by applying it coefficient-wise.

B. Template protection scheme using Kyber and Saber

The Kyber and Saber encryption algorithms encrypt binary messages of length n. If biometric data $\boldsymbol{b}$ have length m (with $m > n$), the data must be divided in $\boldsymbol{b^i}$ blocks with i=1,…, $\lceil m/n \rceil$. Note that if n does not divide m or m < n, a padding must be appended to the data. Then, the protected biometric data can be represented as follows:

$$Enc(pk, \boldsymbol{b}) = \{Enc(pk, \boldsymbol{b^i}) \mid 1 \leq i \leq \lceil m/n \rceil\} \quad (1)$$

The comparison in the protected domain is based on that Kyber and Saber public key encryption schemes accomplish the following homomorphic property:

$$Enc(pk, \boldsymbol{b_1^i}) - Enc(pk, \boldsymbol{b_2^i}) = (\boldsymbol{u_1^i} - \boldsymbol{u_2^i}, \mathrm{v}_1^i - \mathrm{v}_2^i) = \\ = Enc(pk, \boldsymbol{b_1^i} \oplus \boldsymbol{b_2^i}) \quad (2)$$

where $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ are biometric data, '$-$' is the coefficient-wise subtraction operation applied to each polynomial that conforms the ciphertexts, and $\oplus$ is the XOR operation applied to the biometric data. Then, the Hamming distance of $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ can be computed in the protected domain as follows:

$$HD(\boldsymbol{b_1}, \boldsymbol{b_2}) = \sum_{i=1}^{\lceil m/n \rceil} HW(Dec(sk, Enc(pk, \boldsymbol{b_1^i}) - \\ - Enc(pk, \boldsymbol{b_2^i}))) \quad (3)$$

Where HW(**x**) is the Hamming weight (the number of logic 1's) of the vector **x**.

This template protection scheme can be employed in a distributed biometric recognition system. In this work, we consider a system with three parties involved: 1) the Client Device (CD), 2) the Database Server (DB), and 3) the Authentication Server (AS). Also, three phases are considered: 1) setup phase, 2) enrollment phase, and 3) verification phase. In a setup phase, AS generates the key pair pk and sk (to be used with Kyber or Saber), and distributes pk to CD. In an enrollment phase, CD acquires a capture of the biometric characteristic (faces, in this work) from which biometric features are extracted. If biometric features are not binary, they are binarized with some procedure which preserves distances as in [16]. Then, CD encrypts them using pk, and sends the result to DB, which stores it. In a verification phase (depicted in Fig. 1), CD acquires a biometric probe and extracts the associated binary features b using *acquire()*, encrypts them with *Kyber/Saber.Enc*, generating the protected biometric probe eb, and sends them to DB. Later, DB performs the coefficient-wise difference $eb_{diff} = eb - eb_{ref}$ and sends it to AS. Finally, AS decrypts eb_{diff} using sk with *Kyber/Saber.Dec*, computes its Hamming weight, compares the result with a threshold *Th*, and provides the binary verification decision, *res*.

C. Security analysis

The proposal assumes an honest-but-curious model in which the Database Server and the Authentication Server behave according to their roles, but try to learn information about the biometric characteristics of the subjects. It is assumed that the Database Server and the Authentication Server cannot collude, i.e., they cannot interchange information about the protected template. If this is not accomplished, the privacy of the subjects enrolled in the system could be compromised. If biometric data $\boldsymbol{b}$ have length m greater than n, the data are divided in $\boldsymbol{b^i}$ blocks. Hence, the Authentication Server knows the partial distances between the probe and the reference blocks. Other template protection schemes reported in literature also split the biometric data into blocks prior to encrypting them [7]. To the best of authors' knowledge, no biometric information of the subjects can be obtained from these partial distances.

It is also assumed that communication channels between the Client and the Servers are secure to avoid that an attacker could steal a protected biometric probe *eb* with a successful verification response *res*. Otherwise, the attacker could impersonate a verified subject.

Hill-climbing attacks are prevented since the Authentication Server does not provide the similarity between biometric data but only the verification decision [17]. In order to mitigate brute-force attacks, which typically require a large number of iterations, additional solutions such as rate limiting could be considered.

Irreversibility of protected templates is achieved since it is used a public key encryption algorithm. Note that this is achieved even if the Database Server or an attacker has a

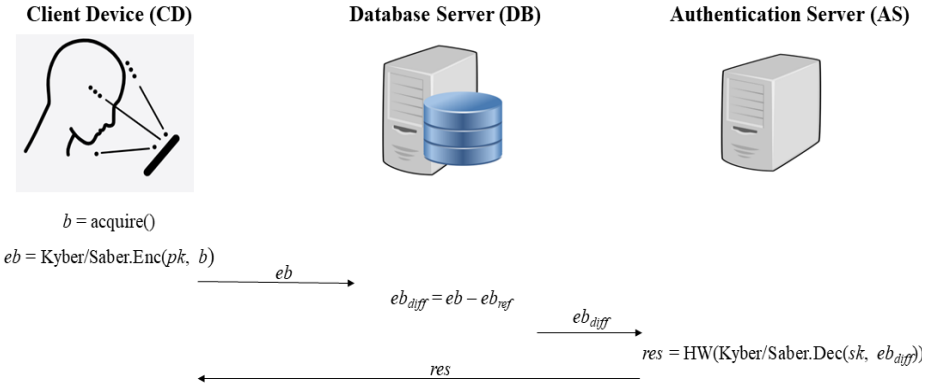

Fig. 1: Verification phase of the proposed scheme in a distributed recognition system.

quantum computer available. As Kyber and Saber use a random seed each time that a template is encrypted, a subject that is enrolled two times in the systems has two different protected templates from the same or different samples. So, unlinkability and renewability are achieved.

Note that if an attacker external to the system wants to gain information about the biometric characteristics of the enrolled subjects, s/he needs to attack two parties instead of one.

III. Experimental Results

All the experiments were carried out in a laptop with an Intel Core i7-1165G7 at 2.80 GHz. To take measurements of execution times, a virtual machine with 1 core processor and 4 GB of main memory running Ubuntu 20.04.4 was used. To implement Kyber and Saber public key encryption algorithms, the reference implementations found in the GitHub repositories published by its authors were used [18][19]. Both implementations are written in C. The parameter sets used were the ones specified by the authors of Kyber and Saber in the NIST Post-Quantum Cryptography contest, which are Kyber512, Kyber768 and Kyber1024, and LightSaber, Saber and FireSaber.

The FERET and LFW databases [20][21] were used to evaluate the proposal. Facenet [13] was used to extract floating-point embeddings as biometric features. Since at the face detection and crop process some samples were lost, 8,160 embeddings were extracted from 994 individuals in FERET database and 12,770 embeddings were extracted from 5,566 individuals in LFW database. Each embedding was binarized using linearly separable subcode (LSSC) [16] with the codes 000, 001, 011 and 111 and a segmentation of the feature space with the intervals (-∞,-0.1), [-0.1,0.0), [0.0,0.1) and [0.1,+∞). The resulting binary embeddings were composed of 384 bits. In the FERET database, they achieved an accuracy of 98.9%, with FMR and FNMR of 1.69%, at the selected threshold. In the LFW database, they achieved an accuracy of 99.2%, with FMR and FNMR of 1.18%, at the selected threshold. DET curves for the proposal without the protection are shown in Fig. 2.

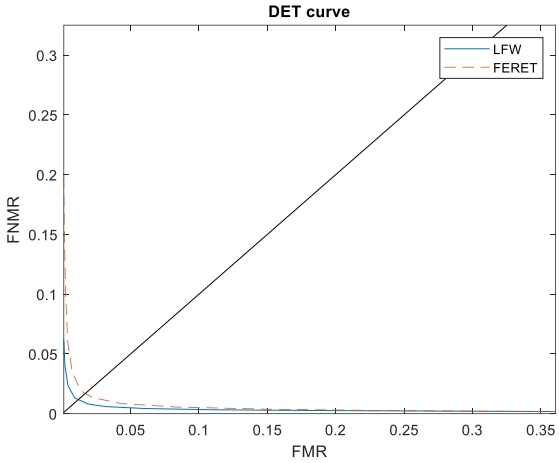

Fig. 2: DET curves for the results using the FERET database (discontinuous) and the LFW database (continuous).

Since Kyber and Saber parameter sets are selected to provide a small probability of decryption failures, all the binary embeddings were compared in the protected domains to check if the probabilities remained small with the homomorphic encryption. The results showed that all the Hamming distances were preserved for the parameter sets Kyber512, Kyber768, Kyber1024, Saber and FireSaber. For LightSaber, the matching failure was approximately of one bit every 4,000 comparisons. This was proven to be a negligible probability since it did not affect the biometric performance.

To check the viability and the performance of the proposal, the sizes of the keys, the sizes of the protected templates and the execution times of the KeyGen, Encryption and Comparison algorithms were obtained. Comparison execution time includes Hamming weight computation, decryption and comparison with a threshold value. These results, together with the results of other solutions based on lattice and homomorphic encryption reported in the literature, are shown in Tab. IV. The solutions in [5] employed features of 2,048 bits and were executed in an Intel Xeon X3480 at 3.07 GHz

TABLE IV. Lattice-Based Proposals in Terms of Security, Sizes and Times.

	Security (in bits)	**Size (KB) of keys**	**Size (KB) of templates**	**Time (ms) for KeyGen**	**Time (ms) for Encrypt**	**Time (ms) for Comp.**
Ideal lat. [Ya17]	80	-	19	870	19.89	18.13
R-LWE [Ya17]	80	47	31	1.89	3.65	8.78
CKKS [Ko20]	128	$99 \cdot 10^3$	516	779	6	3391
BFV [Ko20]	128	$12 \cdot 10^3$	132	255	76	618
NTRU [Ko20]	128	6	5.5	362	27	23
Ideal lat. [Ta21]	128	-	10.3	-	4	7.73
Kyber512 (This)	128	1.53	1.5	0.10	0.26	0.11
Kyber768 (This)	192	2.28	2.13	0.17	0.42	0.16
Kyber1024 (This)	256	3.03	3.06	0.26	0.61	0.18
LightSaber (This)	128	1.47	1.44	0.57	1.54	0.50
Saber (This)	192	2.19	2.13	1.27	3.04	0.88
FireSaber (This)	256	2.91	2.87	4.17	5.05	0.92

with 16 GB of memory. The solutions in [8] employed features of 384 bits and were exectued in an Intel Core i7 at 2.7 GHz CPU with 16 GB of memory. The solutions in [10] employed features of 384 bits and were executed in an Intel Core i7-8700 at 3.2 GHz CPU with 16 GB of memory. In terms of sizes, the solutions using Kyber and Saber have the smallest keys and protected templates. In fact, the solution using Kyber has protected templates 3.67 times smaller than the solution using NTRU (which also employs FaceNet embeddings of 384 bits). Moreover, the solution using Saber has protected templates 3.82 times smaller than the solution using NTRU. Tab. 4 also shows that the achieved execution times using Kyber and Saber are very small in comparison with the other solutions. Note that the great difference of our proposal and [8] can be explained partially because we use C implementations while in [8] it is reported a Python implementation speeded-up using PyPy3. Until now, the smallest comparison time reported in literature is that of [10]. Regarding that, the solution employing Kyber is 70.27 times faster and the one employing Saber is 15.46 times faster. It is worth mentioning that C implementations are also employed in [10].

As an example, let us consider a queue of 1,000 individuals that had to be authenticated at a checkpoint. The database server needs from around 1.4 MB of storage for the weakest security to 3.0 MB for the strongest one. The time needed to check all the individuals is always less than a second for Kyber and ranges from 2 to 6 seconds for Saber. These results show the viability of our proposal.

IV. Conclusions

In this paper, homomorphic encryption schemes using Kyber and Saber public encryption algorithms are proposed to protect biometric data, maintaining recognition performance. Considering the sizes of the protected templates and the keys, and the execution times of the algorithms, these proposals outperform other solutions based on lattices and homomorphic encryption reported in the literature. Implementation results in a laptop show that less than a second (in the case of Kyber) or from 2 to 6 seconds (in the case of Saber) are needed for authenticating a queue of 1,000 individuals in a checkpoint. As the proposals are based on two finalists of the third round of the NIST Post-Quantum Cryptography contest, one of them selected to be standardized, the appearance of optimizations for different types of platforms, such as high-performance servers, are expected to be near, which would offer better results than the ones shown in this paper.

References

[1] European Parliament, *EU Regulation 2016/679 (General Data Protection Regulation)*, 2016.

[2] Information security, cybersecurity and privacy protection — Biometric information protection, document ISO/IEC 24745:2022, 2022.

[3] NIST, Post-Quantum Cryptography. Accessed: Aug. 27, 2022. [Online]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography.

[4] W. Castryck and T. Decru, "An efficient key recovery attack on SIDH (preliminary version)," *Cryptology ePrint Archive*, Report 2022/975, 2022.

[5] M. Yasuda, "Secure Hamming distance computation for biometrics using ideal-lattice and ring-LWE homomorphic encryption," *Inf. Sec. J., A Global Perspective*, vol. 26, pp. 85-103, Mar. 2017.

[6] P. Drozdowski, N. Buchmann, C. Rathgeb, M. Margraf and C. Busch, "On the Application of Homomorphic Encryption to Face Identification," *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2019, pp. 1-5.

[7] J. Kolberg, P. Bauspieß, M. Gomez-Barrero, C. Rathgeb, M. Dürmuth and C. Busch, "Template Protection based on Homomorphic Encryption: Computationally Efficient Application to Iris-Biometric Verification and Identification," *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2019, pp. 1-6.

[8] J. Kolberg, P. Drozdowski, M. Gomez-Barrero, C. Rathgeb and C. Busch, "Efficiency Analysis of Post-quantum-secure Face Template Protection Schemes based on Homomorphic Encryption," *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2020, pp. 1-4.

[9] D. Osorio-Roig, C. Rathgeb, P. Drozdowski and C. Busch, "Stable Hash Generation for Efficient Privacy-Preserving Face Identification," in *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 3, pp. 333-348, July 2022.

[10] H. Tamiya, T. Isshiki, K. Mori, S. Obana and T. Ohki, "Improved Post-quantum-secure Face Template Protection System Based on Packed Homomorphic Encryption," *2021 International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2021, pp. 1-5.

[11] P. Bauspieß, J. Olafsson, J. Kolberg, P. Drozdowski, C. Rathgeb and C. Busch, "Improved Homomorphically Encrypted Biometric Identification Using Coefficient Packing," *2022 International Workshop on Biometrics and Forensics (IWBF)*, 2022, pp. 1-6.

[12] R. Arjona and I. Baturone, "A Post-Quantum Biometric Template Protection Scheme Based on Learning Parity With Noise (LPN) Commitments," in *IEEE Access*, vol. 8, pp. 182355-182365, 2020.

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

[14] J. Bos et al., "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM," *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2018.

[15] J.-. D'Anvers, A. Karmakar, S. Sinha Roy and Frederik Vercauteren, "Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM," *Cryptology ePrint Archive*, Report 2018/230, 2018.

[16] M. -H. Lim and A. B. J. Teoh, "A Novel Encoding Scheme for Effective Biometric Discretization: Linearly Separable Subcode," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 300-313, Feb. 2013.

[17] C. Rathgeb and A. Uhl, "Attacking Iris Recognition: An Efficient Hill-Climbing Technique," *2010 20th International Conference on Pattern Recognition*, 2010, pp. 1217-1220.

[18] Kyber. Accessed: Aug. 27, 2022. [Online]. Available: https://github.com/pq-crystals/kyber.

[19] Saber. Accessed: Aug. 27, 2022. [Online]. Available: https://github.com/KULeuven-COSIC/SABER.

[20] P. J. Phillips, Hyeonjoon Moon, S. A. Rizvi and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090-1104, Oct. 2000.

[21] G. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," in *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, Marseille, France, pp. 1-11, Oct. 2008.