Privacy-Preserving Biometric Authentication and Matching via Lattice-Based Encryption

Constantinos Patsakis[1(✉)], Jeroen van Rest[2], Michał Choraś[3,4], and Mélanie Bouroche[5]

[1] Department of Informatics, University of Piraeus, Piraeus, Greece
kpatsak@unipi.gr
[2] TNO, The Hague, The Netherlands
jeroen.vanrest@tno.nl
[3] University of Science and Technology, UTP, Bydgoszcz, Poland
chorasm@utp.edu.pl
[4] ITTI Sp. z o.o., Poznan, Poland
mchoras@itti.com.pl
[5] Distributed Systems Group, School of Computer Science and Statistics, Trinity College, Dublin, Ireland
melanie.bouroche@scss.tcd.ie

Abstract. The continuous dependence on electronic media has radically changed our interactions, many of which are now performed online. In many occasions users need to authenticate to remote machines, but the hostile environment of the Internet may severely expose users and service providers. To counter these shortcomings, strong authentication is pushed forward. As a means to authenticate individuals, biometric authentication is gradually gaining more and more ground. While the use of biometric data enables many useful applications, these data are very sensitive. For this reason, it is essential to handle them with the least user exposure. In this work we propose a very efficient protocol for privacy-preserving biometric authentication using lattice-based encryption. More precisely, we exploit the homomorphic properties of NTRU to provide a robust and secure solution and provide experimental results which illustrate the efficacy of our proposal.

Keywords: Biometric authentication · Privacy-preserving authentication · Lattice-based encryption

1 Introduction

While we have transferred a wide variety of our social, economic and working interactions in the cyber world, one of the major challenges is to guarantee that all the entities involved are the ones they claim to be. To provide entity authentication most services depend on the secure exchange of credentials of the entities, which are assumed to be properly registered. In general, users are given a user name and they create a password which they use to access the services.

J. Garcia-Alfaro et al. (Eds.): DPM and QASA 2015, LNCS 9481, pp. 169–182, 2016.
DOI: 10.1007/978-3-319-29883-2_11

While theoretically this model works and current protocols can provide high security standards, the truth is that most users do not pick good passwords, enabling an adversary to easily gain access.

The general concept of passwords is to authenticate users by something that they *know* (password). Another paradigm is to authenticate users by something that they *are*, something that cannot be forgotten or forged. The past decade, the use of biometric authentication is gradually becoming more widespread since the cost of the devices has been drastically reduced. While there is a variety of biometric authentication methods, ranging from fingerprints and vein recognition, to retina and iris scanning, all these methods have two inherent drawbacks:

- They are not exact: Regardless of the underlying data, every measurement is not exactly the same as the one registered. For instance, a user scans her iris to register as a user. The system extracts the pattern and stores the feature vector in the system. However, the next time that she will scan her iris, it is highly improbable that the system will extract the exact same feature vector. This differentiation is subject to many factors. For instance, the alterations might be introduced due to angle, motion, imaging noise, reflection etc. Therefore, all biometric authentication methods have a threshold τ which denotes how many differences in two measurements can be tolerated in order to authenticate a user.
- They are permanent: While one could easily pick another password if a service has been compromised, she could not change her eyes or fingertips. If an adversary could acquire the biometric measurements of a user, then she could masquerade as her forever. Notably, depending on the method this data can be easily acquired and replicated[1].

Due to their nature and how they can be used, biometric data are very sensitive and should be dealt with much caution. Their fuzziness; the fact that two measurements of the same subject may differ, creates further problems. Implicit authentication is fairly easy when using passwords, a user may prove the knowledge of the password without actually revealing it. However, the fuzziness of biometric measurements renders such protocols useless.

The problem where two entities want to check whether the values that they hold are the same without presenting them to each other or to any other entity is widely known as private equality testing, and there are many solutions in the literature. However, if the underlying data are not equal, the case of biometric data, then most of these protocols cannot work as well, or they will be inefficient. For instance, if the two values may differ in τ bits, then one of the parties may need to present 2^{τ} candidate values for checking. Other approaches such as the scheme of Feigenbaum et al. [19] are far more efficient, but not efficient enough for such applications.

1.1 Contribution of This Work

In our work, we use the well-known NTRU [24] public encryption algorithm and exploit its efficiency and additive homomorphic property to enable

[1] http://www.ccc.de/en/updates/2014/ursel.

privacy-preserving biometric authentication and matching. An overview of the proposed protocol is the following. Assume that Alice and Bob hold a biometric measurement and Alice wants to know whether Bob's measurement differs from her measurements by less than a threshold value. First, both of them split the biometric measurement into blocks and Alice encrypts them with her NTRU public key, blinding them from every other entity. However, Bob is still able to perform some operations on the encrypted data, which in our case is to subtract the according block value from his biometric measurements. To obfuscate the results, Bob randomly permutes the results and returns them to Alice. While Alice can decrypt each block, she cannot recover the order of the blocks to find Bob's measurements. Thus, she can compute whether their measurements are below the required threshold without further information leakage. For the sake of simplicity and performance, we will present the protocol for the standard NTRU algorithm, nevertheless, adapting it to the more secure variant of Stehlé and Steinfeld [34] is straightforward and does not imply further changes than the obvious ones. In this paper the considered biometric modality is the iris. However, our privacy-preserving methodology can be applied to any modality which can be represented as sequence of bits such as faces, DNA etc.

1.2 Organization of This Work

The rest of this work is organized as follows. In Sect. 2 we provide a small overview of the NTRU algorithm and then present the state of the art in privacy-preserving biometric authentication. Section 3 introduces our protocol and discusses its security, mostly focusing on the semi-honest model. In Sect. 4 we provide some experimental results and compare its performance with current state of the art. Then in Sect. 6 we present some application scenarios were our protocol could be applied. Finally the article concludes with some remarks and ideas for future work.

2 Related Work

2.1 NTRU and Its Variants

Lattices are being studied for decades and several problems in their theory, such as the shortest and closest lattice vector have been proven to be extremely hard to solve, leading to the development of several public key encryption schemes. However, in the past few years the interest in these schemes has been greatly increased as these schemes provide many interesting features in terms of security and applications. For instance, while the widely used public key algorithms such as RSA and ElGamal could be broken with quantum algorithms, lattice-based encryption algorithms seem to be immune to such attacks making them a good candidate for the post-quantum era of cryptography [6]. Moreover, lattices have very interesting algebraic features that can be exploited to develop fully homomorphic encryption.

Table 1. NTRU parameters for different security levels

Level(bits)	p	q	n	D_1	D_2	D_3	D_g	D_m
128	3	2048	439	9	8	5	146	112
192	3	2048	593	10	10	8	197	158
256	3	2048	743	11	11	15	247	204

One of the most well known lattice based algorithms is NTRU [24]. The algorithm was developed in the mid 90 s and it is an extremely fast public key encryption algorithm. In fact it so efficient that its performance can be compared to symmetric ciphers [22]. Currently there are many variants, however in this work we will work with the original algorithm of Hoffstein, Pipher and Silverman. To generate the public/private key pair, we firstly, select some parameters N, p and q which are publicly known and determine the security of the NTRU instance. N is a prime number, denoting the degree of the polynomials that are going to be used. In what follows, every polynomial is reduced modulo the polynomial $x^N - 1$. The other two parameters, p and q are the two moduli numbers, the "large" (q); current standards set q equal to 2048, and one "small" (p) typically equal to 3. All NTRU operations are either performed in $\mathbb{Z}_q[x]/(x^N - 1)$ or in $\mathbb{Z}_p[x]/(x^N - 1)$. We then select two random polynomials f and g with small coefficients, that is -1, 0 and 1. We also require f to be invertible in $\mathbb{Z}_q[x]/(x^N - 1)$ and $\mathbb{Z}_p[x]/(x^N - 1)$, and we denote these inverses f_q and f_p respectively. The public key h is defined as $h = pgf_q$, while f and f_p are the private key. The most common parameters for NTRU are shown in Table 2.

To encrypt a message we map it to a polynomial m with small coefficients and pick a random "small" polynomial r, and send the message $c = hr + m \in \mathbb{Z}_q[x]/(x^N - 1)$. To decrypt c, the recipient multiplies it with f and rearranges the coefficients to reside within $[-q/2, q/2]$ and reduces it modulo p. Finally, she multiplies the result with f_p.

The amount of 1s, 0s and -1s in f, g, m and r are very important for NTRU. More precisely, a message can be decrypted only if the following inequality holds:

$$\|f * m + p * r * g\|_\infty \leq q$$

Otherwise the result will be a random polynomial. The randomness of r may introduce some problems in the decryption of the ciphertext, that is some ciphertext might not be decrypted. However proper parameter selection can bound this probability so that this event can be considered improbable.

NTRU has been extensively studied and after many attacks, the original parameters have been updated [23]. Currently, the algorithm is considered highly secure and has been standardized in both IEEE 1363.1 and X9.98. Moreover, NTRU has triggered the introduction of many variants such as [3,15,29], however of specific interest are the recent variants of Stehlé and Steinfeld [34] and the variant of Lopez et al. [27]. The first variant is CPA-secure in the standard model under the assumed quantum hardness of standard worst-case problems

over ideal lattices, using Regev's learning with error approach [32]. The latter exploits the homomorphic properties of NTRU to create a fully homomorphic encryption scheme.

Generally, most lattice-based encryption schemes have homomorphic properties, however, there are specific constraints. In (partial) homomorphic encryption, the cryptographic primitives can transfer only one operation from the plaintext to one operation of the ciphertext, while recently introduced fully homomorphic encryption can transfer two operations. Nevertheless, in both cases the operations can be applied arbitrary amount of times. However, somewhat homomorphic or leveled encryption cannot support arbitrary homomorphic operations. For instance, in the case of NTRU, with each operation the amount of "noise" that is added is increased. Therefore, at one point the added noise is so high that the message cannot be recovered. Therefore, NTRU can support only a limited amount of additions and multiplications. Note that the homomorphic properties of NTRU hold over $\mathbb{Z}_p[x]/(x^N - 1)$, so for instance the additive property is applied over polynomials which is very important in our protocol.

2.2 Privacy-Preserving Biometric Authentication

Nowadays, biometric human identification is widely used in many large-scale security applications such as border crossings, visa/passports etc. Also law enforcement agencies use biometrics in order to search for criminals and terrorists. Several modalities, including iris, face or fingerprint, are very mature and the discussion now is not about the performance rates (FAR/FRR), but rather about the scalability and throughput of the system as well as on assuring privacy and fundamental human rights.

Iris is the part of the eye bounded by the pupil and sclera and it consists of muscle tissue [17]. Nowadays, iris acquisition devices are gaining momentum and can acquire high-quality images even of the walking subjects in operational environment (e.g. airport) [31]. Typically, the iris recognition system consists of the following steps: image acquisition using iris acquisition device(s), iris segmentation, extraction of iris features (such as eg. iris codes, Gabor filters or wavelets), and iris pattern matching. Hereby, in order to assure privacy and template security, especially in realistic systems used by law enforcement agencies, we also propose to add privacy-preserving methodology. The latter is considered a basic ingredient in building cyber-physical systems which are compliant with the "privacy-by-design" concept [11].

The first privacy-preserving identification protocol for iris was introduced by Blanton et al. [7] which exploit the homomorphic properties of the encryption method of Damgard et al. [16]. Based on the Paillier homomorphic scheme, Shahandashti et al. [33] propose a method for private fingerprint matching. Other approaches include the use of oblivious RAM from Bringer et al. [12] or garbled circuits from Luo et al. [28] and Bringer et al. [14]. Kulkarni and Namboodiri [26] use the somewhat homomorphic scheme of Boneh et al. [9] to privately compute the hamming distance of two sequences. Another approach, more focused on

faces, would be to divide the biometric into smaller pieces, store them in independent compartments and use methods such as the one of Forczmański and Łabędź to identify them [20].

Similar methods have also been used in private DNA sequence matchmaking, as genetic information is also very sensitive [2], however, the size of the data render most of these methods inefficient. Therefore, many researchers have resulted to the use of a semi-trusted third party which can significantly improve computational and bandwidth requirements [25].

Recently, Blundo et al. [8] proposed a probabilistic protocol for the privacy-preserving evaluation of sample set similarity. Based on the MinHash approach, they sample each set, and perform the protocol of De Cristofaro et al. [18] to determine the cardinality of the common elements of both sets. More precisely, we assume that we have Alice and Bob, holding sets A and B respectively and that each one selects k values $(r_1, r_2, \ldots, r_k)$ for the sample of their set, that is $a_{r_1}, a_{r_2}, \ldots, a_{r_k}$ and $b_{r_1}, b_{r_2}, \ldots, b_{r_k}$. Furthermore, we assume that Bob has published a prime p. Alice picks a random $\alpha, gcd(\alpha, p-1) = 1$ and sends Bob the message:

$$m_A = \{h(a_{r_i})^{\alpha} \mod p\}, i \in [1, k]$$

On receiving this message, Bob picks a random β and computes:

$$m'_A = \{m_{A_i}^{\beta} \mod p\}, i \in [1, k]$$

Then, Bob computes:

$$m_B = \{h(h(b_{r_i})^{\beta} \mod p)\}, i \in [1, k]$$

and sends Alice the message: $A' = \pi(m'_A), B' = m_B$ where π is a random permutation. Finally, Alice computes:

$$C = \{h(c^{\alpha^{-1} \mod p-1} \mod p)\}, \forall c \in A'$$

and checks how many elements in common does C have with B'. If there are ν, then Alice assumes that the Jaccard similarity of the two sets is approximately ν/k, subject to $\mathcal{O}(1/\sqrt{k})$ error.

Yasuda et al. [35,36] exploit the properties of the somewhat homomorphic scheme of Brakerski and Vaikuntanathan [10] by packing the feature vectors of the biometrics, however, their method was proven to be insecure [1].

An overview of these methods can also be found in [5,13].

3 The Proposed Protocol

3.1 Main Actors and Desiderata

Let us assume two entities, Alice, the initiator of the protocol and Bob, the responder. Both Alice and Bob hold a sequence of bits $\mathcal{A} = a_1, a_2, \ldots, a_k$ and $\mathcal{B} = b_1, b_2, \ldots, b_k$ respectively. The goal of Alice is to determine whether

$d_H(\mathcal{A}, \mathcal{B}) < \tau, \tau \in \mathbb{N}$; and d_H denotes the Hamming distance, without disclosing any information to Bob or anyone else. On the other hand, Bob is willing to allow this computation, nevertheless, he does not want to leak any information regarding $\mathcal{B}$ to Alice or another entity.

In what follows, we work in the *honest-but-curious/semi-honest model*. Therefore, while each party is assumed to follow each step of the protocol correctly (honest), they may try to analyze any received information or messages to extract information about their peers (curious). Therefore, if the protocol dictates that a participant should send a message of a specific form, we assume that the participant will conform, and will not send a tampered version.

3.2 The Protocol

We assume that Alice has created an NTRU key pair, so h is her public key and f, f_p her private. Both parties split their sequences in blocks of length λ, creating k blocks. Moreover, we assume that both of them know a function $\chi : \{0,1\}^\lambda \rightarrow \mathbb{D}$, where $\mathbb{D}$ contains the polynomials of $\mathbb{Z}_q[x]/(x^N - 1)$ with coefficients -1, 0 and 1. For the sake of simplicity instead of $\chi(m)$ we will write m. Additionally, we denote α_i and $\beta_i, i \in [1, k]$ the blocks of Alice and Bob respectively.

The steps of the protocol are as follows. Initially, Alice sends Bob the message

$$M_A = \{hs_i + \alpha_i\}, \forall i \in [1, k]$$

where s_i are random polynomials in $\mathbb{D}$. On receiving the vector m_A, Bob computes the vector

$$M_B = \{M_{A_i} - (hs'_i + \beta_i)\}, \forall i \in [1, k]$$

where s'_i are random polynomials in $\mathbb{D}$. That is the encryption of her blocks with NTRU. Then, Bob picks a random permutation π and sends Alice $M'_B = \pi(M_B)$. So Bob encrypts his blocks with NTRU, subtracts them from Alice's; he exploits the additive homomorphic property of NTRU, and rearranges them.

On receiving this message, Alice can decrypt each $M_{B'_i}$ and compute the weight w_i of each recovered message. If $\sum_{i=1}^{k} w_i < \tau$ then Alice deduces that $d_H(\mathcal{A}, \mathcal{B}) < \tau$. Figure 1 illustrates the proposed protocol.

Initially, Alice and Bob extract the templates of their biometrics and encrypt them in blocks using the NTRU encryption algorithm using Alice's public key. Alice sends her encrypted data to Bob who subtracts them in the according order and then permutes the results. Alice decrypts the messages to recover the Hamming weight and compare it against the threshold τ.

3.3 Protocol Correctness

In the first step, the protocol splits $\mathcal{A}$ into blocks and encrypts them to hide them from Bob. In the second step, Bob subtracts his values from the encrypted ones. If two values are the same, then they will cancel each other out, otherwise,

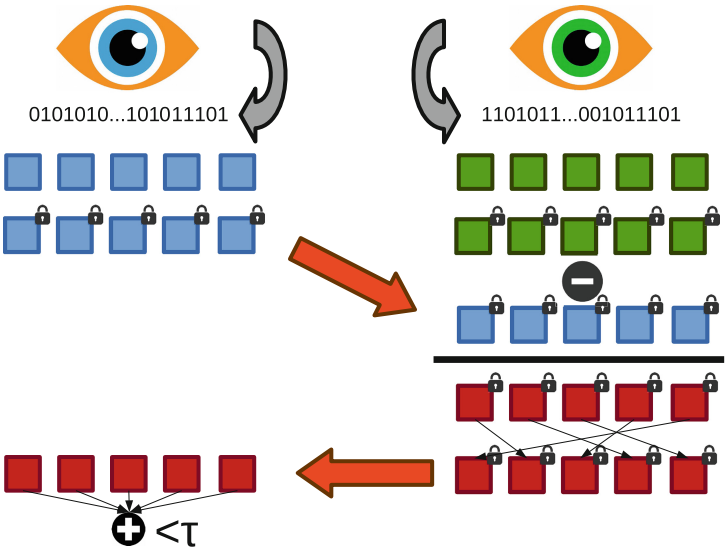

Fig. 1. The proposed protocol.

one coefficient (-1 or 1) is left in the encrypted block. Clearly, Bob's permutation does not alter the weight of the encrypted messages, but hides their order from Alice, who cannot recover Bob's sequence. Nevertheless, each block can be decrypted and the non-zero coefficients denote where each block differs with the others. Thus, Alice can easily find $d_H(\mathcal{A}, \mathcal{B})$.

3.4 Security of the Protocol

We do not consider active attacks; we assume that the messages exchanged in a protocol run are authenticated and integrity protected, thus the adversary is not able to modify or inject fake messages pretending to originate from another legitimate user.

Alice's input remains secret from Bob and any other active of passive adversary. Throughout the protocol, Alice sends a single message to Bob which contains her encrypted blocks. Therefore, anyone who wishes to recover Alice's input must break NTRU encryption which is considered infeasible. Note that NTRU is considered secure even from quantum algorithms.

While Alice can decrypt the encrypted blocks to compute their differences, in order to recover Bob's private input she has to find the proper order of κ blocks. This means $\kappa!$ arrangements, so finding the right order is infeasible. Clearly, an external adversary will not be able to recover any information about Bob's input, since it is encrypted with NTRU. Note that Bob does not simply subtract his input but he subtracts his encrypted input further confusing his output.

However, if Alice were malicious she could try to trick Bob and recover his input. For instance, instead of sending her input, she could mark each block

and then put them in the right order. Since the information in each block is not going to fill it up to capacity, e.g. for 128 bits of security, NTRU can accept messages up to 439 bits but it will take only a fraction such as 32 or 64 bits, Alice could hide additional information in the unused bits. To counter such attack Bob could simply use a random padding for each message and alert Alice about its existence so that Alice would correctly calculate the weight of each block.

4 Experimental Results

We chose to compare our algorithm against the algorithm of Blundo et al. as it is the most efficient one in current state of the art, even though it samples the retinas and does not return exact results. The computer where the experiments were made has an Intel Core i3-2100 CPU at 3.1 GHz with 6 GB of RAM, running on Ubuntu 15.04 64 bit. The implementation in both cases is made in Sage 6.5[2]. For NTRU we have used the parameters proposed by SecurityInnovation[3], illustrated in Table 1. According to their recommendations, to generate f, we compute a polynomial $P(x)$ which is of the form $A_1(x)A_2(x) + A_3(x)$, where polynomial $A_i, i \in \{1, 2, 3\}$ have D_i coefficients set to 1 and D_i coefficients set to -1. Similarly, to construct polynomial g, we select a polynomial having D_g coefficients set to 1 and $D_g - 1$ coefficients set to -1. Finally, each message, when converted to polynomial must have at most D_m coefficients set to 1 and $D_m - 1$ coefficients set to -1. The set of parameters used for RSA and NTRU is shown in Table 2. The role of D_1, D_2 and D_3 is going to be discussed in Sect. 4.

Table 2. Parameters for the most popular security levels (in bits). For RSA the numbers denote the length (in bits) of the underlying modulo field according to NIST [4]. For NTRU, the numbers are precise and recommended by SecurityInnovation (https://github.com/NTRUOpenSourceProject/ntru-crypto/blob/master/doc/NewParameters.pdf).

Security level	RSA	NTRU			
		p	q	n	Public key (bits)
128	3072	3	2048	439	4829
192	7680	3	2048	593	6523
256	15360	3	2048	743	8173

The experimental results in Table 3 clearly indicate the performance gains of our protocol. It should be highlighted that Alice in the Blundo et al. protocol has to perform light calculations as the exponentiations are "soft", the exponent is 2^{16}, however the RSA decryptions of Bob are very intensive. Note

[2] sagemath.org.

[3] https://github.com/NTRUOpenSourceProject/ntru-crypto/blob/master/doc/NewParameters.pdf.

Table 3. Comparison of the Blundo et al. protocol with the proposed. Time in seconds and security in bits.

Security	Blundo et al.			Proposed		
	Alice	Bob	Total	Alice	Bob	Total
128	0.024	2.227	2.251	0.187	0.115	0.302
192	0.066	12.352	12.418	0.250	0.153	0.403
256	0.183	59.421	59.605	0.299	0.220	0.519

Table 4. Approximate communication cost in KB. Security in bits.

Security	Blundo et al.	Proposed
128	78.125	75.453
192	190.625	101.922
256	378.125	127.703

that the reported times account for a single thread in both cases, therefore, by multithreading these timings will be significantly reduced.

In our tests, we used random feature vectors of 2048 bits, such as iris. The sample for the Blundo et al. protocol was 100 bits, which accounts for an error of 10 %. Practically, this means that the "hard" computations for Alice and Bob are 100 RSA encryptions and 200 decryptions respectively.

In our protocol, we split retinas in blocks of 32 bits; that is 64 blocks, so Alice had to perform 64 encryptions and decryptions, while Bob had to perform 64 encryptions. The comparison of the communication cost for different security levels is shown in Table 4. Again, our proposed algorithm introduces lower communication costs compared to the protocol of Blundo et al. In fact, the higher the security level, the better our protocol performs. Note that the increase in the key length of NTRU is lower than RSA when the security level increases.

5 Discussion

Our proposed protocol has many benefits compared to its peers. The one that is most obvious is its performance, however, the there are other important aspects as well. For instance, the protocol manages to pack far more information than other protocols without reducing its security. Therefore, not only the bandwidth is reduced, but the protocol is secure in the post-quantum era. Undoubtly, one could use the Paillier [30] or the Goldwasser-Micali [21] cryptosystems to perform the XOR of the bits of the templates. However, to achieve the same level of security the bandwith overhead is considerably higher as only one bit would be processed at a time. Moreover, NTRU is far more efficient in terms of performance than any of these algorithms. One could argue that Alice could potentially find patterns regarding Bob's biometrics, with the risk being subject to the block

size, the bigger the block, the higher the probability. While this is true, in the next paragraphs we provide a countermeasure for such attacks.

One generic attack of all these privacy preserving schemes is the following. Alice performs one execution of the protocol with Bob using firstly the sequence $00\ldots000$ and then $10\ldots000$. Clearly, comparing the output values Alice can determine whether the value of the first bit is 0 or 1. Having found the value of the first bit, Alice can proceed to the second bit etc. The main problem is that Bob uses the same template for each comparison and Alice can manipulate her own to find a better match at each execution. To counter this problem we propose the following method.

Let $\mathcal{F}(k, x)$ denote a Pseudo Random Function (PRF), where k is the PRF key and x is the point at which the function is evaluated. Bob proposes a random seed s so Alice and Bob compute the following for their sequences: $\mathcal{F}(s, m_i||i)$ mod $2, i \in \{1, 2, \ldots, k\}$. Clearly, for each position where the bits of Alice and Bob are the same, the result is also going to be the same. However, when they differ, the result is going to be equal 50 % of the times. By processing their sequences like this, Bob's input is always randomized so Alice cannot perform this attack or find patterns in our scheme. Nevertheless, one should note that the threshold should now be close to half.

6 Application Scenarios

The presented methodology can be applied in several scenarios and it is valid for various biometric modalities. Herein, we are concerned with security scenarios, especially those interesting for law enforcement agencies, where preserving the privacy of citizens is challenging. On the one hand, the methodology can be applied for access control, where a person (the subject) wants to get access to a certain asset (e.g. terrain, building, room, laptop, service etc.), including critical infrastructures and high-risk assets with high accuracy biometrics such as iris. Such scenario can be realized in a verification mode (1:1 matching) or in the identification mode (1:many matching). In the latter case, so called white-listing is used, since the data (biometric feature vector) of the subject is matched versus those who can enter/gain access to the asset.

The second scenario where the proposed methodology is useful, is the matching of the subject biometric pattern versus templates from the law enforcement, or vice versa from private organisations. It can be realized as the typical 1:many identification or as the blacklisting. In such a case, e.g. the template of the subject (we can even imagine a wanted terrorist) is compared to the database of the people that agencies search for or those who are not allowed cross borders etc. The proposed methodology is useful because the law enforcement agency can query the database without disclosing who is the terrorist, and without learning anything about the other templates. Vice versa, private organisations can query law enforcement databases without disclosing any information about their customers.

7 Conclusions

The continuous use of biometrics might strengthen user authentication, however, it implies serious privacy risks. It should be understood that unlike passwords which can be easily generated, a user cannot generate a new body part, such as an iris or face. Addressing this challenge, privacy-preserving biometric authentication methods were recently introduced. These methods provide the needed functionality: biometric authentication, while simultaneously minimizing user's privacy exposure using state of the art cryptographic primitives. Clearly, this introduces a computational and communication overhead which might not be considered important in one-to-one scenarios - a user wants to authenticate to his device, but in one-to-many scenarios - a user authenticates to a server, the overhead might be substantial and decrease the quality of the provided service.

Based on the above, we introduced a novel protocol that takes advantage of the additive homomorphic property of NTRU to enable secure and exact privacy-preserving biometric authentication. Even if our implementation is not optimized, it is rather efficient, enabling it to be faster even than the "sampling" method of Blundo et al. In the future, we plan to explore the possibility of packing more data in each package with other algorithms and/or encodings to further decrease the computational and communication cost.

Acknowledgments. The research leading to these results has received funding by the European Commission under the Horizon 2020 Programme (H2020), as part of the *OPERANDO* project (Grant Agreement no. 653704) and the FP7 *TACTICS* project (Grant Agreement no. 285533) and is based upon work from COST Action *CRYPTACUS*, supported by COST (European Cooperation in Science and Technology).

The publication of this paper has been partly supported by the University of Piraeus Research Center.

References

1. Abidin, A., Mitrokotsa, A.: Security aspects of privacy-preserving biometric authentication based on ideal lattices and ring-lwe. In: IEEE International Workshop on Information Forensics and Security (WIFS), pp. 60–65. IEEE (2014)
2. Ayday, E., De Cristofaro, E., Hubaux, J.-P., Tsudik, G.: Whole genome sequencing: revolutionary medicine or privacy nightmare? Computer **2**, 58–66 (2015)
3. Banks, W.D., Shparlinski, I.E.: A variant of NTRU with non-invertible polynomials. In: Menezes, A., Sarkar, P. (eds.) INDOCRYPT 2002. LNCS, vol. 2551, pp. 62–70. Springer, Heidelberg (2002)
4. Barker, E., Dang, Q.: NIST special publication 800–57 part 3: Application-specific key management guidance. NIST Special Publication **800**(57) (2015)
5. Belguechi, R., Alimi, V., Cherrier, E., Lacharme, P., Rosenberger, C.: An overview on privacy preserving biometrics. In: Recent Application in Biometric, pp. 65–84. INTECH (2011). https://halv3-preprod.archives-ouvertes.fr/hal-00992461
6. Bernstein, D.J., Buchmann, J., Dahmen, E.: Post-Quantum Cryptography. Springer Science & Business Media, Berlin (2009)

7. Blanton, M., Gasti, P.: Secure and efficient protocols for iris and fingerprint identification. In: Atluri, V., Diaz, C. (eds.) ESORICS 2011. LNCS, vol. 6879, pp. 190–209. Springer, Heidelberg (2011)
8. Blundo, C., De Cristofaro, E., Gasti, P.: EsPRESSo: efficient privacy-preserving evaluation of sample set similarity. In: Di Pietro, R., Herranz, J., Damiani, E., State, R. (eds.) DPM 2012 and SETOP 2012. LNCS, vol. 7731, pp. 89–103. Springer, Heidelberg (2013)
9. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
10. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) lwe. SIAM J. Comput. **43**(2), 831–871 (2014)
11. Bringer, J., Chabanne, H., Le Métayer, D., Lescuyer, R.: Privacy by design in practice: reasoning about privacy properties of biometric system architectures. In: Bjørner, N., de Boer, F. (eds.) FM : Formal Methods. LNCS, vol. 9109, pp. 90–107. Springer, Switzerland (2015)
12. Bringer, J., Chabanne, H., Patey, A.: Practical identification with encrypted biometric data using oblivious ram. In: International Conference on Biometrics (ICB), pp. 1–8. IEEE (2013)
13. Bringer, J., Chabanne, H., Patey, A.: Privacy-preserving biometric identification using secure multiparty computation: an overview and recent trends. IEEE Signal Process. Mag. **30**(2), 42–52 (2013)
14. Bringer, J., Favre, M., Chabanne, H., Patey, A.: Faster secure computation for biometric identification using filtering. In: 5th IAPR International Conference on Biometrics (ICB), pp. 257–264. IEEE (2012)
15. Coglianese, M., Goi, B.-M.: MaTRU: a new NTRU-based cryptosystem. In: Maitra, S., Veni Madhavan, C.E., Venkatesan, R. (eds.) INDOCRYPT 2005. LNCS, vol. 3797, pp. 232–243. Springer, Heidelberg (2005)
16. Damgard, I., Geisler, M., Kroigard, M.: Homomorphic encryption and secure comparison. Int. J. Appl. Crypt. **1**(1), 22–31 (2008)
17. Daugman, J.: How iris recognition works. IEEE Trans. Circuits Syst. Video Technol. **14**(1), 21–30 (2004)
18. De Cristofaro, E., Gasti, P., Tsudik, G.: Fast and private computation of cardinality of set intersection and union. In: Pieprzyk, J., Sadeghi, A.-R., Manulis, M. (eds.) CANS 2012. LNCS, vol. 7712, pp. 218–231. Springer, Heidelberg (2012)
19. Feigenbaum, J., Ishai, Y., Malkin, T., Nissim, K., Strauss, M.J., Wright, R.N.: Secure multiparty computation of approximations. ACM Trans. Algorithms **2**(3), 435–472 (2006)
20. Forczmański, P., Łabędź, P.: Recognition of occluded faces based on multi-subspace classification. In: Saeed, K., Chaki, R., Cortesi, A., Wierzchoń, S. (eds.) CISIM 2013. LNCS, vol. 8104, pp. 148–157. Springer, Heidelberg (2013)
21. Goldwasser, S., Micali, S.: Probabilistic encryption & how to play mental poker keeping secret all partial information. In: Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing, pp. 365–377. ACM (1982)
22. Hermans, J., Vercauteren, F., Preneel, B.: Speed records for NTRU. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 73–88. Springer, Heidelberg (2010)
23. Hirschhorn, P.S., Hoffstein, J., Howgrave-Graham, N., Whyte, W.: Choosing NTRUEncrypt parameters in light of combined lattice reduction and MITM approaches. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 437–455. Springer, Heidelberg (2009)

24. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: a ring-based public key cryptosystem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 267–288. Springer, Heidelberg (1998)
25. Kamara, S., Mohassel, P., Raykova, M., Sadeghian, S.: Scaling private set intersection to billion-element sets. In: Christin, N., Safavi-Naini, R. (eds.) Financial Cryptography and Data Security. LNCS, vol. 8437, pp. 195–215. Springer, Heidelberg (2014)
26. Kulkarni, R., Namboodiri, A.: Secure hamming distance based biometric authentication. In: International Conference on Biometrics (ICB), pp. 1–6. IEEE (2013)
27. López-Alt, A., Tromer, E., Vaikuntanathan, V.: On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing, pp. 1219–1234. ACM (2012)
28. Ying Luo, S., Cheung, T.P., Lazzeretti, R., Barni, M.: An efficient protocol for private iris-code matching by means of garbled circuits. In: 19th IEEE International Conference on Image Processing (ICIP), pp. 2653–2656. IEEE (2012)
29. Nevins, M., Karimianpour, C., Miri, A.: NTRU over rings beyond $\mathbb{Z}$. Des. Codes Crypt. **56**(1), 65–78 (2010)
30. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, p. 223. Springer, Heidelberg (1999)
31. Rakvic, R.N., Broussard, R.P., Kennell, L.R., Ives, R.W., Bell, R.: Iris acquisition device. In: Li, S.Z., Jain, A.K. (eds.) Encyclopedia of Biometrics, pp. 761–769. Springer, US (2009)
32. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM (JACM) **56**(6), 34 (2009)
33. Shahandashti, S.F., Safavi-Naini, R., Ogunbona, P.: Private fingerprint matching. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP 2012. LNCS, vol. 7372, pp. 426–433. Springer, Heidelberg (2012)
34. Stehlé, D., Steinfeld, R.: Making NTRU as secure as worst-case problems over ideal lattices. In: Paterson, K.G. (ed.) Advances in Cryptology - EUROCRYPT. LNCS, vol. 6632, pp. 27–47. Springer, Heidelberg (2011)
35. Yasuda, M., Shimoyama, T., Kogure, J., Yokoyama, K., Koshiba, T.: Packed homomorphic encryption based on ideal lattices and its application to biometrics. In: Cuzzocrea, A., Kittl, C., Simos, D.E., Weippl, E., Xu, L. (eds.) CD-ARES Workshops 2013. LNCS, vol. 8128, pp. 55–74. Springer, Heidelberg (2013)
36. Yasuda, M., Shimoyama, T., Kogure, J., Yokoyama, K., Koshiba, T.: Practical packing method in somewhat homomorphic encryption. In: Garcia-Alfaro, J., Lioudakis, G., Cuppens-Boulahia, N., Foley, S., Fitzgerald, W.M. (eds.) DPM 2013 and SETOP 2013. LNCS, vol. 8247, pp. 34–50. Springer, Heidelberg (2014)