Private Proximity Testing on Steroids: An NTRU-based Protocol

Constantinos Patsakis[1], Panayiotis Kotzanikolaou[1] and Mélanie Bouroche[2]

[1]Department of Informatics, University of Piraeus, Greece
[2]Distributed Systems Group, School of Computer Science and Statistics,
Trinity College, Dublin, Ireland
{kpatsak,pkotzani}@unipi.gr,melanie.bouroche@scss.tcd.ie

Abstract. Nowadays, most smartphones come pre-equipped with location (GPS) sensing capabilities, allowing developers to create a wide variety of location-aware applications and services. While location awareness provides novel features and functionality, it opens the door to many privacy nightmares. In many occasions, however, users do not need to share their actual location, but to determine whether they are in proximity to others, which is practically one bit of information. Private proximity protocols allow this functionality without any further information leakage. In this work we introduce a novel protocol which is far more efficient than the current state of the art and bases its security on lattice-based cryptography.

Keywords: Location privacy, Cryptographic Protocols, Private Equality Testing, Location Services

1 Introduction

Private equality testing is a very well-known problem in cryptography. In general, it involves two entities, Alice and Bob that want to reveal only a single bit of information: whether they have the same value or not. One solution to the problem is using Diffie-Hellman as proposed by Huberman, Franklin and Hogg [1]. A problem which is very close to private equality testing is *private proximity testing*. Again, we have Alice and Bob that want to reveal only a single bit of information, which now is whether they are in proximity or not. The twist here is that Alice and Bob may not have the same value (location), but they are "close". Notably, in this case we have an additional restriction: location is a low entropy source as the possible values are of the scale of 2^{32}, therefore one could easily brute force it. Narayanan et al. [2] with an ingenious encoding managed to reduce the problem of private proximity testing to private equality testing.

Lattices are being studied for decades and several problems in their theory, such as the shortest and closest lattice vector (SVP and CVP) have been proven to be extremely hard to solve. This has led to the development of several public key encryption schemes based on these problems. However, in the past few years the interest in these schemes has greatly increased as they 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.

Moreover, lattices have very interesting algebraic features that can be exploited to develop fully homomorphic encryption (FHE) [3, 4]. Nevertheless, most of the lattice-based encryption schemes provide only somewhat homomorphic encryption. While FHE supports arbitrary number of operations, somewhat homomorphic encryption support only a limited number of operations.

In this work we exploit the properties of NTRU, a well-known lattice-based algorithm to introduce a novel 2-party private protocol which is used for private equality matching and then tested for private proximity testing. The main advantages of the proposed protocol are the following:

1. It outperforms the current state of the art by a factor of around 20x, depending on the security level. The reason why the protocol is far more efficient than its peers is that it uses more lightweight computations, e.g. instead of performing calculations over large finite fields, the computations are performed over small polynomial rings.
2. In terms of security, NTRU is considered the best alternative for the post-quantum era [5] as its security does not seem to be significantly decreased by quantum algorithms [6].
3. Apart from private proximity testing, the protocol can be used for private equality testing.

The rest of this work is organized as follows. The next section provides an overview of the related work and in Section 3 we introduce the protocol and discuss its security. Section 4 presents some experimental results and a comparison of the proposed protocol with the one of Narayanan et al. Finally, the article concludes with some notes for future work.

2 Related work

2.1 NTRU

NTRU is a secure and extremely fast public key encryption algorithm developed in the mid 90s, and its security is based on lattices. In fact it is so efficient that it can be even compared with symmetric ciphers [7]. The original algorithm, introduced by Hoffstein, Pipher and Silverman [8] works as follows. Firstly, we select some parameters N, p and q which are publicly known and determine the security of the NTRU instance. N is a prime, used to determine the degree of the polynomials that we are going to use, so every polynomial is reduced modulo $x^N - 1$. In NTRU we use two moduli numbers one “large” (q); currently q is set to 2048, and one “small” (p), which typically is equal to 3. Generally, all NTRU operations are $\mathbb{Z}_q[x]/(x^N - 1)$, while some of them are made in $\mathbb{Z}_p[x]/(x^N - 1)$.

To generate the secret/public key pair, we select two random polynomials f and g with small coefficients, that is -1, 0 and 1. However, for f we additionally require that it is invertible in $\mathbb{Z}_q[x]/(x^N - 1)$ and in $\mathbb{Z}_p[x]/(x^N - 1)$, so 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 consist the private key. To encrypt a message m, we map m to a polynomial 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, we multiply the result with f_p.

To make NTRU work, the amount of 1s, 0s and -1s in f, g, m and r need to be specific to allow message decryption. A message can be decrypted only if the following inequality holds:

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

If this is not the case, then the result will be a random polynomial.

Due to a number of attacks, the original parameters of NTRU have been updated [9] and the algorithm and its parameters have been standardized in both IEEE 1363.1 and X9.98. A comparison of NTRU parameters with RSA and elliptic curves is illustrated in Table 1. While there are many variants of the algorithm such as [10–12], of specific interest are the recent variant of Stehlé and Steinfeld [13] which makes it even more secure[1], using Regev's learning with error approach [14], and the variant of Lopez et al. [4] which builds on top of NTRU to create a FHE scheme.

Security	**RSA**	**Elliptic**				**NTRU**
Level		**Curves**	**p**	**q**	**n**	**Public key (bits)**
128	3072	256	3	2048	439	4829
192	7680	384	3	2048	593	6523
256	15360	521	3	2048	743	8173

Table 1. Parameters for the most popular security levels (in bits). For RSA and elliptic curves, the numbers denote the length (in bits) of the underlying modulo field according to NIST (`https://www.nsa.gov/business/programs/elliptic_curve.shtml`). For NTRU, the numbers are precise and recommended by SecurityInnovation [15].

2.2 Private proximity testing

In private proximity testing, Alice and Bob want to check whether they are in proximity, without disclosing their whereabouts. These protocols are gaining

[1] In this variant, NTRU becomes CPA-secure in the standard model, under the assumed quantum hardness of standard worst-case problems over ideal lattices.

more importance due to the wide adoption of location awareness from Online Social Networks which notify users of friends that are close. The feeling of closeness and the hope that one could find the other half just around the corner is also exploited by mobile dating applications. However, as it has been shown, this exposes users to many threats [16–18].

The protocols in the literature can be categorized in three overlapping categories. The first categorization is made according to who makes the testing. For instance, if Alice and Bob outsource the testing to Trudy, a trusted third party, then we have the so called *asynchronous* protocols. Note that in these protocols while Alice and Bob trust Trudy in that she will make the proper computations and that she will not collude with either, they are not willing to provide her with their locations. On the contrary, Alice and Bob will only provide Trudy with an encrypted version of their locations. However, if Alice and Bob want to perform the tests on their own without another entity, we have the so-called *synchronous*. Clearly, in the first case only the initiator needs to be online, while in the latter both need to be online. We consider privacy preserving data dissemination techniques beyond the scope of this work, nevertheless, the interested reader may refer to [19] for an overview of such methods related to location privacy.

Depending on the nature of the exchanged data, we can have further categorization. Most protocols will use the GPS location of the users, or more precisely their position on a grid, allowing users to report fake locations. To counter this issue, Zheng et al. [20] as well as Lin et al. [21] have recently provided efficient solutions. Both these protocols gather "environmental" data such as GPS and WiFi signals which are known only to users who are in a specific area at a given time to derive some "fingerprints". These fingerprints are then sent to the other user to perform a private check to determine whether they are within proximity.

Finally, one could categorize the private proximity protocols depending on the underlying cryptographic primitive. For instance, there are protocols which are based on symmetric algorithms, grid transformation keys, while others are based on homomorphic encryption or specific hard mathematical problems.

An overview of the categorization of current state of the art algorithms in private proximity testing is illustrated in Figure 1.

2.3 The protocol of Narayanan et al.

Narayanan et al. in [2] make a significant contribution in private proximity testing. As already discussed, they introduced a new grid system with three overlapping grids which reduces the problem of private proximity testing to private equality testing. From the protocols that they introduced in their work, of specific interest is the synchronous protocol which is based on an elliptic curve variant of ElGamal.

Let g a generator of the group G, x Alice's private key and let $h = g^x$. For efficiency, we may use as G the additive group of an elliptic curve over a finite field. Moreover, we assume that Alice is located at ℓ_A, Bob at ℓ_B. Alice's public key is (E, g, h); where E denotes the elliptic curve she uses, and x is her private key. The steps of the protocol are the following:

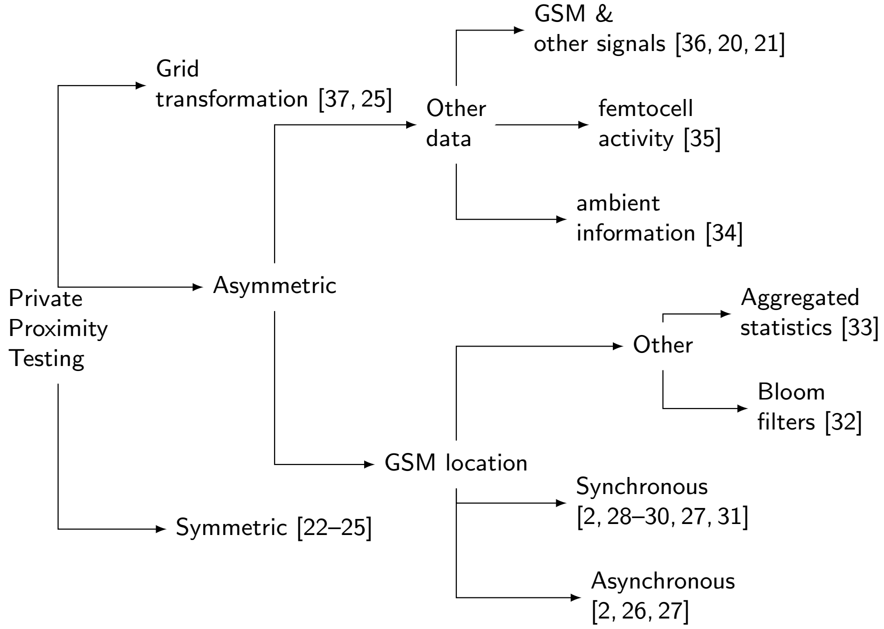

Fig. 1. Categorization of current state of the art protocols in private proximity testing.

Firstly, Alice encrypts ℓ_A with her public key and sends Bob $C_A = (g^r, h^{r+\ell_A})$, where r is a random integer. On receiving $C_A = (c_1, c_2)$, Bob picks two random integers s, t and sends Alice: $C_B = (c_1^s g^t, c_2^s h^{(t-s\ell_B)})$. Finally, when Alice receives: $C_B = (u_1, u_2)$, she computes $R = u_2 u_1^{-x}$. If $R = 1$, then she deduces that $\ell_A = \ell_B$, otherwise R will be a random point of E.

3 The proposed protocol

3.1 Threat model

Like most privacy-preserving techniques, we assume that users have a *honest but curious* (HBC) behavior. According to the HBC model, also known as *semi-honest*, users will follow the rules of the protocol (honesty), they will not act maliciously, nevertheless, they will try to extract as much information as possible from the other users. This threat model can be considered realistic as in most social LBS services, the users have some form of acquaintance (e.g. friends, colleagues) or want to have (case of mobile dating applications). Thus, users have no incentive to behave maliciously.

We assume probabilistic polynomial time (PPT) passive adversaries that are polynomially bounded and do not have the ability to break the underlying cryptographic primitives. Moreover, we assume that an adversary may monitor all the exchanged traffic of the users. Nevertheless, we do not consider active attacks; the exchanged messages in a execution of the protocol are authenticated and integrity protected, therefore an adversary cannot modify or inject fake messages making them seem legitimate.

3.2 Main actors and desiderata

In what follows, we use the grids of Narayanan et al [2], to reduce private proximity testing to private equality testing. In this regard, we assume that we have divided the earth with a grid, where each square is marked as $\mathcal{L}_i$. Therefore, the scope of the protocol will be to determine whether two users, Alice and Bob are in the same square. The set of all possible squares is denoted as $\mathcal{L}$, so $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k\}$ where $|\mathcal{L}| = \mathcal{O}(2^{32})$.

Moreover, we assume that there is a bijection $\chi : \mathcal{L} \to L(x)$, where $L(x)$ is a set of polynomials in $\mathbb{Z}_q/(x^N - 1)$. The role of χ is to encode a square $\mathcal{L}_i$ to a polynomial ℓ_i in order to use it in the NTRU-based protocol. Therefore, for simplicity from now on when we refer to a location of an entity, we will represent it as ℓ_i. Clearly, this encoding is known to everyone.

Finally, we assume that each user constructs a NTRU key pair. Note that users do not need to share their public keys with others, but only n, p, q and the "noise" parameters. For the sake of simplicity, we assume that users have already agreed on the above, e.g. they use NTRU_EES439EP1 for 128 bits of security, and we call them *public parameters*. Clearly, this feature drastically decreases the communication cost as users do not need to store any additional information about their "friends". As it will become apparent, the knowledge of the actual public key does not add any additional value, since the operations that have to be made by the recipient are depend solely on the n, p, q and the "noise" parameters. Moreover, since NTRU is considered secure, the publication of the public key h does not jeopardize the security of the scheme.

3.3 The protocol

Let Alice be located in ℓ_A and Bob in ℓ_B. Even if Bob does not know Alice's public key, he may perform some operations on Alice's encrypted location using the public parameters, to allow Alice determine whether he is within her proximity, that is $\ell_A = \ell_B$.

Initially, Alice sends the message $c_A = rh + \ell_A$ to Bob, where r is a random invertible polynomial in $\mathbb{Z}_q[x]/(x^N - 1)$. Then Bob picks a random polynomial ρ with small coefficients and sends Alice $c_B = \rho(c_A - \ell_B)$. Alice receives it and checks whether $r^{-1}c_B$ decrypts to zero.

3.4 Protocol correctness

Let us assume that $l_A = l_B$. Then, in step 2, Bob computes:

$$c_B \equiv \rho(c_A - \ell_B) \equiv rh\rho$$

that he will sent to Alice. Thus, when Alice in step 3 decrypts:

$$r^{-1}c_B = r^{-1}rh\rho \equiv h\rho$$

the result will be 0, otherwise it will be a random polynomial.

3.5 Security Analysis

We consider both external and internal adversaries. An external adversary represents all entities other than the users running the protocol, while an internal adversary represents an honest-but-curious user running the protocol. In any case, the goal of the adversary is on input the messages exchanged during a protocol run and (in case of internal adversaries the private keys of the adversary), to learn the private input of the honest user(s) running the protocol. In the following analysis we will first examine internal adversaries (either a curious Bob against Alice or a curious Alice against Bob). Obviously, the security arguments also hold for external adversaries.

Definition 1. *A function $\nu(\cdot)$ is negligible in x, or just negligible, if for every positive polynomial $p(\cdot)$ and any sufficiently large x it holds that:*

$$\nu(x) \leq \frac{1}{p(x)}$$

Private input indistinguishability We formalize private input indistinguishability by a security experiment $DistExp$ in which the adversary $\mathcal{A}$ has access to an oracle $\mathcal{O}$ that on input: the low-entropy set of all possible private input $\mathcal{L}$, the public parameters of two users Alice and Bob n_A, n_B and a protocol run $[c_A, c_B, y]$, is used to extract information about the private input of Alice and/or Bob. In case on an internal adversary, then the oracle is also given the private keys $\mathcal{K}$ of the compromised user. If $\mathcal{O}^{dist}$ is able to distinguish the private input of a target user (l_A and/or l_B) from the set $\mathcal{L}$ using the given input (where $|\mathcal{L}|$ is the security parameter), then the output of the experiment is 1, else the output is 0.

Definition 2. [Private input indistinguishability] *A protocol provides private input indistinguishability if $\forall$ PPT adversary $\mathcal{A}$, $\exists$ a negligible function ν such that:*

$$Advantage(\mathcal{A}) = \mid Pr[DistExp(|\mathcal{L}|) = 1] - \frac{1}{|\mathcal{L}|} \mid \; = \; \nu(|\mathcal{L}|)$$

Theorem 1. *The proposed PET protocol provides private input indistinguishability for Alice against a curious Bob, provided that the NTRU encryption algorithm is secure.*

Proof. Since Bob only learns the public key h of Alice and c_A which is the NTRU encryption of l_A with the key h, clearly Bob cannot learn the private input of Alice assuming that the NTRU cryptosystem is secure. □

Theorem 2. *The proposed PET protocol provides private input indistinguishability for Bob against a curious Alice, provided that the NTRU encryption algorithm is secure.*

Proof. Let us assume that Alice wants to find Bob's location when $\ell_A \neq \ell_B$. Alice has c_A, c_B, as well as to her private keying material f and f_p and to the polynomial r. Since Bob is assumed honest, the structure of c_B will be of the form $c_B = \rho(rh + \ell_A - \ell_B)$.

Since the value ρ is only known to Bob, the only possible way for Alice to reveal ℓ_B is through brute forcing. Alice may attempt to calculate all $\delta_i = \ell_A - \ell_i$, for each possible $\ell_i \in \mathcal{L}$ (recall that $\mathcal{L}$ is a low entropy set). Then Alice would decrypt c_B in order to find which δ_i corresponds to the actual decrypted value and thus learn ℓ_i.

While the values $rh + \delta_i$ are known to Alice, trying to solve these equations in $\mathbb{Z}_q[x]/(x^N - 1)$ would not give her an actual advantage. We consider two cases: In the first case, if $rh + \delta_i$ in not an invertible polynomial, then Alice will not be able to recover ρ from c_B and thus she will not be able to test these values, without the knowledge of ρ.

In the second case, if $rh + \delta_i$ is invertible, then for each such ℓ_i, Alice would get $|K|$ additional possible values for ρ, without being able to distinguish the correct one. Therefore, Alice cannot distinguish the private input of Bob in case of private input inequality.

We should note that in either case; $rh + \delta_i$ being or not being invertible, Alice would have to brute force the polynomial which would requires $\mathcal{O}(c^N)$ attempts. For more details on the latter, the interested reader may refer to [15]. □

Note that while the original NTRU is not IND-CPA secure, like RSA without padding, the variant of Stehlé and Steinfeld [13] provides this feature and the adaption of the latter scheme is straightforward. Moreover, the paddings proposed in [38–40] make NTRU IND-CCA2-secure, with the latter making it IND-CCA2-secure in the random oracle model.

Theorem 3. *The proposed PET protocol provides private input indistinguishability against external adversaries, under the NTRU assumption.*

Proof. The proof is an immediate result of the previous proofs. Notice that external adversaries have no access to any keying material (e.g. of a curious party). □

4 Comparison/Experimental Results

We compare the NTRU with the Narayanan et al. protocol in a machine with an Intel Core i3-2100 CPU at 3.1 GHz with 6GB of RAM, running on Ubuntu 15.04 64 bit. The implementation in both cases is made in Sage[2]. For NTRU we have used the latest parameters proposed by SecurityInnovation [15]. The parameters

[2] `sagemath.org`

are illustrated in Table 2. 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 code to perform the experiments is publicly available on Github[3].

Table 2. 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

The protocol of Narayanan et al. has been implemented over elliptic curves, as the original. To provide 128-bits of security, we used Curve25519 [41] well-known for its security and performance. Furthermore, to provide 192 and 256 bits security we used the curves M-383 and M-511 respectively, both described in [42]. Note that all these curves are renowned for their security and performance, so they were selected instead of random elliptic curves. The experiments report the averages of 1,000 executions of the protocol in a single thread.

Table 3 clearly illustrates that the proposed protocol is far more efficient than the protocol of Narayanan et al. More precisely, the protocol is approximately 20 times faster. The result can be considered expected, as the protocol of Narayanan et al. has to perform more and heavier computations. In Narayanan et al. Alice (the initiator of the protocol) has to perform 3 exponentiations and Bob 4 exponentiations. On the contrary, in the proposed protocol Alice has to perform 1 encryption and 1 decryption with NTRU, while Bob has to perform one polynomial addition and one polynomial multiplication. Therefore, in all security levels Bob's cost is below 2ms. Further comparison to other schemes is illustrated in Table 5.

It has to be noted that implementing the protocols in another language like C would make the implementations far more efficient, mostly in favor of NTRU as its structure is rather lightweight and can receive many improvements, as highlighted in other works e.g. [7]. Nevertheless, the result can be considered in accordance with the reported results of other implementations e.g. NTRU project[4].

[3] `https://github.com/kpatsakis/NTRU_Private_Proximity_Testing`
[4] `http://tbuktu.github.io/ntru/`

	Narayanan et al.			Proposed			Ratio
Security	Alice	Bob	Total	Alice	Bob	Total	
128	80.718	99.194	179.912	7.362	1.051	8.413	21.385
192	102.267	133.873	236.140	10.527	1.518	12.045	19.605
256	155.329	193.887	349.216	12.733	1.745	14.478	24.120

Table 3. Comparison of the Narayanan et al. protocol with the proposed. Time in ms and Security in bits. Ratio denotes the ratio of the total time of the Narayanan et al. protocol over the total time of the proposed protocol.

While Sage is based on Python, and there is already a Python implementation of Curve25519 available[5], the Sage implementation was far more efficient so it was used it for the experiments.

Security	Narayanan et al.	Proposed
128	128	1208
192	192	1630
256	256	2044

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

Table 5 provides an overview of the comparison of the proposed protocol with its peers, highlighting the "heavy" computations that each party needs to perform.

The major disadvantage of the protocol is that it has a significant bandwidth overhead, see Table 4. Since NTRU has far bigger keys and messages compared to elliptic curves, the exchanged messages are far bigger than the ones in Narayanan et al. so performance boost is balanced by the communication cost.

Protocol		Efficiency
Pierre	[27]	6exp+3 DL Bob: 6exp
NFP	[28]	2 exp/user
EG-PET	[2]	Alice: 3exp Bob: 4exp
DH-PET	[30]	2 exp/user
Proposed		Alice: 3 mult. Bob: 1 mult. 1 add.

Table 5. Comparison of our protocol with its peers.

[5] `http://ed25519.cr.yp.to/software.html`

5 Conclusions

The continuous development of location-aware applications and services might provide users novel features and functionality, nevertheless, it implies serious privacy exposure. This exposure can be significantly reduced in many occasions, since users do not need to share their actual location, but their proximity to other entities, which is a single bit of information. Current state of the art contains several private proximity protocols to enable this functionality with the least, if any, user exposure as they diminish information leakage.

In this work we introduced a novel protocol which is far more efficient than its peers basing its security on lattice-based cryptography, and more precisely the well-known NTRU algorithm. To the best of our knowledge, this is the first private proximity testing protocol, and probably the first for private equality testing, using lattice-based cryptography. In the future, we plan to make a more optimized implementation, using a low level programming language to further examine the efficiency of the protocol. Furthermore, we plan to study the cost of converting the protocol according to the variant of Stehlé and Steinfeld [13] to provide CPA-security, as theoretically, the changes in the protocol can be easily made.

Acknowledgments

This work was supported by the European Commission under the Horizon 2020 Programme (H2020), as part of the *OPERANDO* and *TACTICS* projects (Grant Agreements no. 653704 and no. 285533 respectively) 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. B. A. Huberman, M. Franklin, T. Hogg, Enhancing privacy and trust in electronic communities, in: Proceedings of the 1st ACM conference on Electronic commerce, ACM, 1999, pp. 78–86.
2. A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, D. Boneh, Location privacy via private proximity testing, in: NDSS, The Internet Society, 2011. URL `http://www.isoc.org/isoc/conferences/ndss/11/`
3. C. Gentry, A fully homomorphic encryption scheme, Ph.D. thesis, Stanford University (2009).
4. A. López-Alt, E. Tromer, V. Vaikuntanathan, 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, ACM, 2012, pp. 1219–1234.
5. R. A. Perlner, D. A. Cooper, Quantum resistant public key cryptography: a survey, in: Proceedings of the 8th Symposium on Identity and Trust on the Internet, ACM, 2009, pp. 85–93.

6. S. Fluhrer, Quantum cryptanalysis of ntru, Cryptology ePrint Archive, Report 2015/676, `http://eprint.iacr.org/` (2015).
7. J. Hermans, F. Vercauteren, B. Preneel, Speed records for ntru, in: Topics in Cryptology-CT-RSA 2010, Springer, 2010, pp. 73–88.
8. J. Hoffstein, J. Pipher, J. H. Silverman, Ntru: A ring-based public key cryptosystem, in: Algorithmic number theory, Springer, 1998, pp. 267–288.
9. P. S. Hirschhorn, J. Hoffstein, N. Howgrave-Graham, W. Whyte, Choosing ntruencrypt parameters in light of combined lattice reduction and mitm approaches, in: Applied cryptography and network security, Springer, 2009, pp. 437–455.
10. W. D. Banks, I. E. Shparlinski, A variant of ntru with non-invertible polynomials, in: Progress in Cryptology-INDOCRYPT 2002, Springer, 2002, pp. 62–70.
11. M. Coglianese, B.-M. Goi, Matru: A new ntru-based cryptosystem, in: Progress in Cryptology-INDOCRYPT 2005, Springer, 2005, pp. 232–243.
12. M. Nevins, C. Karimianpour, A. Miri, Ntru over rings beyond {\ mathbb {Z}}, Designs, Codes and Cryptography 56 (1) (2010) 65–78.
13. D. Stehlé, R. Steinfeld, Making ntru as secure as worst-case problems over ideal lattices, in: Advances in Cryptology–EUROCRYPT 2011, Springer, 2011, pp. 27–47.
14. O. Regev, On lattices, learning with errors, random linear codes, and cryptography, Journal of the ACM (JACM) 56 (6) (2009) 34.
15. J. Hoffstein, J. Pipher, J. M. Schanck, J. H. Silverman, W. Whyte, Z. Zhang, Choosing parameters for ntruencrypt, Cryptology ePrint Archive, Report 2015/708, `http://eprint.iacr.org/` (2015).
16. G. Qin, C. Patsakis, M. Bouroche, Playing hide and seek with mobile dating applications, in: N. Cuppens-Boulahia, F. Cuppens, S. Jajodia, A. Abou El Kalam, T. Sans (Eds.), ICT Systems Security and Privacy Protection, Vol. 428 of IFIP Advances in Information and Communication Technology, Springer Berlin Heidelberg, 2014, pp. 185–196.
17. C. Patsakis, A. Zigomitros, A. Papageorgiou, A. Solanas, Privacy and security for multimedia content shared on osns: Issues and countermeasures, The Computer Journal (2014) bxu066.
18. C. Patsakis, A. Zigomitros, A. Solanas, Analysis of privacy and security exposure in mobile dating applications, in: S. Boumerdassi, S. Bouzefrane, E. Renault (Eds.), Proceedings of the International Conference on Mobile, Secure and Programmable Networking (MSPN'2015), Springer, 2015.
19. M. Terrovitis, Privacy preservation in the dissemination of location data, ACM SIGKDD Explorations Newsletter 13 (1) (2011) 6–18.
20. Y. Zheng, M. Li, W. Lou, Y. T. Hou, Sharp: Private proximity test and secure handshake with cheat-proof location tags, in: Computer Security–ESORICS 2012, Springer, 2012, pp. 361–378.
21. Z. Lin, D. F. Kune, N. Hopper, Efficient private proximity testing with gsm location sketches, in: Financial Cryptography and Data Security, Springer, 2012, pp. 73–88.
22. L. Šikšnys, J. R. Thomsen, S. Šaltenis, M. L. Yiu, O. Andersen, A location privacy aware friend locator, in: Advances in Spatial and Temporal Databases, Springer, 2009, pp. 405–410.
23. L. Siksnys, J. R. Thomsen, S. Saltenis, M. L. Yiu, Private and flexible proximity detection in mobile social networks, in: Mobile Data Management (MDM), 2010 Eleventh International Conference on, IEEE, 2010, pp. 75–84.
24. S. Mascetti, D. Freni, C. Bettini, X. S. Wang, S. Jajodia, Privacy in geo-social networks: proximity notification with untrusted service providers and curious buddies, The VLDB journal 20 (4) (2011) 541–566.

25. K. P. Puttaswamy, S. Wang, T. Steinbauer, D. Agrawal, A. El Abbadi, C. Kruegel, B. Y. Zhao, Preserving location privacy in geosocial applications, Mobile Computing, IEEE Transactions on 13 (1) (2014) 159–173.
26. E. Novak, Q. Li, Near-pri: Private, proximity based location sharing, in: INFOCOM, 2014 Proceedings IEEE, IEEE, 2014, pp. 37–45.
27. G. Zhong, I. Goldberg, U. Hengartner, Louis, lester and pierre: Three protocols for location privacy, in: Privacy Enhancing Technologies, Springer, 2007, pp. 62–76.
28. S. Chatterjee, K. Karabina, A. Menezes, A new protocol for the nearby friend problem, in: Proceedings of the 12th IMA International Conference on Cryptography and Coding, Springer-Verlag, 2009, pp. 236–251.
29. J. D. Nielsen, J. I. Pagter, M. B. Stausholm, Location privacy via actively secure private proximity testing, in: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on, IEEE, 2012, pp. 381–386.
30. E. Magkos, P. Kotzanikolaou, M. Magioladitis, S. Sioutas, V. S. Verykios, Towards secure and practical location privacy through private equality testing, in: Privacy in Statistical Databases, Springer, 2014, pp. 312–325.
31. P. Kotzanikolaou, C. Patsakis, E. Magkos, M. Korakakis, Lightweight private proximity testing for geospatial social networks, Computer Communications(Accepted for publication).
32. P. Palmieri, L. Calderoni, D. Maio, Spatial bloom filters: Enabling privacy in location-aware applications, in: D. Lin, M. Yung, J. Zhou (Eds.), Information Security and Cryptology - 10th International Conference, Inscrypt 2014, Beijing, China, December 13-15, 2014, Revised Selected Papers, Vol. 8957 of Lecture Notes in Computer Science, Springer, 2014, pp. 16–36.
33. R. A. Popa, A. J. Blumberg, H. Balakrishnan, F. H. Li, Privacy and accountability for location-based aggregate statistics, in: Proceedings of the 18th ACM conference on Computer and communications security, ACM, 2011, pp. 653–666.
34. T. Halevi, D. Ma, N. Saxena, T. Xiang, Secure proximity detection for nfc devices based on ambient sensor data, in: Computer Security–ESORICS 2012, Springer, 2012, pp. 379–396.
35. J. Brassil, R. Netravali, S. Haber, P. Manadhata, P. Rao, Authenticating a mobile device's location using voice signatures, in: Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on, IEEE, 2012, pp. 458–465.
36. G. Saldamli, R. Chow, H. Jin, B. Knijnenburg, Private proximity testing with an untrusted server, in: Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks, ACM, 2013, pp. 113–118.
37. M. Li, H. Zhu, Z. Gao, S. Chen, K. Ren, L. Yu, S. Hu, All your location are belong to us: Breaking mobile social networks for automated user location tracking, arXiv preprint arXiv:1310.2547.
38. J. Hojfstein, J. Silverman, Protecting NTRU against chosen ciphertext and reaction attacks, Tech. Rep. 16 (2000).
39. J. Hojfstein, J. Silverman, Optimizations for ntru, in: Public-Key Cryptography and Computational Number Theory: Proceedings of the International Conference organized by the Stefan Banach International Mathematical Center Warsaw, Poland, September 11-15, 2000, Walter de Gruyter, 2001, p. 77.
40. P. Q. Nguyen, D. Pointcheval, Analysis and improvements of ntru encryption paddings, in: Advances in CryptologyCRYPTO 2002, Springer, 2002, pp. 210–225.
41. D. J. Bernstein, Curve25519: new diffie-hellman speed records, in: Public Key Cryptography-PKC 2006, Springer, 2006, pp. 207–228.

42. D. F. Aranha, P. S. L. M. Barreto, G. C. C. F. Pereira, J. E. Ricardini, A note on high-security general-purpose elliptic curves, Cryptology ePrint Archive, Report 2013/647, `http://eprint.iacr.org/` (2013).