Location Privacy Preserving of Mobile Users using Secured Homomorphism

Nitin Kamble
SESGOIFOE, Diksal
Raigad 410102

Narendra Shekokar , Ph. D
DJSCOE, Vile Parle,
Mumbai 400056

ABSTRACT

In today's highly interconnected world users are increasingly dependent on high end smartphones and mobile devices. Users arrange and plan their daily routines using such high end devices. These applications often rely on current locations of individual users or a group of users to provide the desired service. By means of such applications and services, majority of such user population reveal their current location details to the third party service providers. Knowingly or unknowingly mobile users compromise their privacy. Without efficient protection, even sharing location information has been shown to provide reliable information about a users' private globe, which could have severe consequences on the users' personal, social, and financial life

Users, who are cautious about their whereabouts, do not necessarily want to reveal their current locations to the service provider or to untrusted users. This paper, proposes algorithms for location privacy preserving of mobile users. This is to provide practical privacy-preserving techniques to solve this problem, such that neither an untrusted user, nor participating users, can learn other users' locations, legitimate users only learn the optimal location.

Keywords

Location determination server, homomorphism, location privacy.

1. INTRODUCTION

The fast abundance of smartphone technology in urban communities has enabled mobile users to utilize context aware services on their devices. Service providers take benefit of this dynamic and ever-growing technology landscape by proposing innovative context-dependent services for mobile subscribers. Location based Services (LBS), for instance, are employed by several mobile subscribers each day to get location-specific information [1].

Location privacy preservation in mobile surroundings is difficult for two reasons. First off wireless communications are simple to intercept e.g. eavesdropper can collect transmitted data of mobile users at certain public place. Besides, since individuals are in public discernible, context data will simply be obtained from their conversation or behaviors. As a result, partial flight related to user's real identity is inevitably exposed to the eavesdropper. Second, the limited resources of mobile devices greatly limit Privacy Enhancing Technologies one may apply and deploy in wireless network. Current solutions rely on simple schemes to hide the real identity of a mobile user from a passive adversary, rather than complex cryptographic technologies.

Two popular features of location-based services are *location check-ins* and *location sharing*. By checking into a locality, users can disclose their current location with family and friends or obtain location-specific services from third-party providers. The obtained service doesn't rely on locations of different user. The other types of location-based services, that have confidence on sharing of locations by a cluster of users so as to get some service for the whole group, are also becoming popular. Privacy of a user's location or location preferences, with relevance different users and therefore the third-party service provider, may be an essential concern in such location-sharing-based applications [2]. For instance, such information can be used to de-anonymize users and their availabilities [3], to track their preferences [4] or to identify their social networks [5]. In the taxi-sharing application, a third-party supplier could easily deduce home/work location pairs of users who regularly use their service. Without effective protection, if the collected data is leaked in an unauthorized fashion or improperly shared with corporate partners, which could have severe consequences on the users' social, financial and private life [6], [7].

Service providers who legitimately track users' location information in order to improve the offered service can unintentionally harm users' privacy. Recent user studies [8] show that end-users are extremely sensitive about sharing their location information. Thus, the disclosure of private location in any Location-Sharing-Based Service (LSBS) is a major concern and must be addressed.

The problem of privacy preserving location has received little or no attention in the literature. Although considering aspects such as user preferences and constraints, their work does not address any security or privacy issues. All private information about users is public. Privacy of a user's location or location preferences, with connectedness totally different users and thus third-party service provider, could also be a vital concern in such location-sharing-based applications. As an example, such data can be used to de-anonymize users and their availabilities, to trace their preferences or to discover their communal networks.

The problem of finding a rendezvous point among a set of user-proposed locations, such that (i) The Rendez-Vous point is *fair* with respect to the given input locations, (ii) each user learns only the final Rendez-Vous location and (iii) no participating user or third-party server learns private location preference of any other user involved in the computation. The algorithm termed as *Privacy-Preserving Fair Rendez-Vous Point (PPFRVP)* algorithm.

2. ANALYSIS OF LITERATURE SURVEY

In 2004, Frikken and Atallah proposed Secure Multiparty Computation (SMC) protocols for securely computing the distance between a point and a line segment, the distance between two moving points and the distance between two line segments. One difficulty with route planning protocols is the requirement that the device know where it is at, which would seem to require some form of query to a GPS system, but this would reveal the location of the device [9].

In 2007, Santos and Vaughn presented a survey of existing literature on meeting-location algorithms and propose a more comprehensive solution for such a problem. The list of participants, the proposed meeting time, likely start locations and possible travel methods are known. The "cost" function (time, distance, social constraints, etc.) for each person to travel to locations are calculated. Although considering aspects such as user preferences and constraints, their work does not address any security or privacy issues. The system, while useful, may be complicated for some users. Automating system defaults when users provide insufficient data from calendars or start points can help, but preferences about times, venues, and travel methods can be complicated even when known. An organizer, or participants who vote, need to evaluate choices and fine-tune results to suit group criteria [10].

Zhong design and implement three distributed privacy-preserving protocols for nearby friend discovery, and they show how to cryptographically compute the distance between a pair of users. However, due to the fully distributed nature of the above mentioned approaches, the computational and communication complexities increase significantly with the size of the participants and inputs. Moreover, all parties involved in the computations need to be online and synchronized [11].

In 2009, Berger proposed an efficient meeting-location algorithm that considers the time in-between two consecutive meetings. However, all private information about users is public [12].

In 2010, Jaiswal and Nandi suggest a novel approach to deploy location-based services in which user privacy is guaranteed without any entity having knowledge of both pieces of sensitive user information i) location ii) queries (interests + social relationships). Inspite of operating on encoded information got from the operator and the LBS; it is able to trigger updates to the mobile user whenever the user is in the same location of its interested services [13].

In 2012, Guha proposed a privacy-preserving location based matching as a fundamental platform primitive and as an alternative to exposing low-level, latitude-longitude coordinates to applications. Applications set rich location-based triggers and have these be fired based on location updates either from the local device or from a remote device. But issue pertains to malicious applications registering a large number of triggers at sensitive locations, and reverse-engineering a victim user's location based on triggers matched. A weak defense against this attack would be rate-limit to the number of trigger registrations from an application [14].

Carbunar also propose a set of privacy-preserving protocols, using well-known cryptographic constructs, which anonymously proves to a venue that a user checked-in a certain number of times [15].

In 2013, in the direction of anonymous location sharing, Pidcock proposed to disassociate user identity information from user location information in our privacy-friendly location hub. No entity should know both a user's identity and user's location. The foundation of location hub, ZeroSquare, is two noncolluding entities, one that stores information about users and another that stores information about locations. ZeroSquare also provides a callback framework to support scenarios where a user wishes to be notified when a condition is met. However, by having only users (but not locations) become first-class citizens in the architecture, the applicability of these architectures to geosocial applications remains limited because storing or retrieving information about locations is difficult [16].

Importantly, the principle of "Location Privacy Preserving" suggests to proactively embed privacy into the design of any service. We want to demonstrate that location-based services can be built in more privacy-friendly ways; this in turn may shift people's thinking about and expectations of the inner workings of location-based services

3. PROPOSED SYSTEM

Proposed system have two algorithms for solving the Fair Rendez-Vous Point (FRVP) problem in a privacy-preserving way, wherever every user participates by providing solely one location preference to the FRVP solver or the service provider. It has multi-preference cases, where every user might have multiple prioritized location preferences.

Goal is to offer realistic privacy preserving techniques to resolve the FRVP hitch, specified neither a third-party, nor participating users will learn other users' locations; participating users only learn the optimal location. The privacy issue within the FRVP problem is representative of the relevant privacy threats in LSBSs.

4. PROPOSED TECHNIQUE

It addresses the privacy issue in Location-Sharing-Based Services (LSBS) by focusing on a specific problem called the *Fair Rendez-Vous Point (FRVP)* problem. Given a group of user location preferences, the FRVP problem is to settle on a location among the proposed ones such that the maximum distance between this location and all other users' locations is minimized, i.e. it is *fair* to all users. We first formulate the FRVP problem as *k-center* optimization problem, and then analytically outline the privacy requirements of the participants with respect to each other and with respect to the third-party service provider. We have two algorithms for solving the above formulation of the FRVP problem in a privacy-preserving way, where each user participates by providing only a sole location preference to the FRVP service provider. The proposed algorithms take advantage of the homomorphic properties of well-known cryptosystems, like BGN, ElGamal and Paillier, so as to secretly compute an optimally fair Rendez-Vous point from a set of user location preferences

Apart from that, the multi-preference case, where each user may have multiple prioritized location preferences can be considered.

5. ARCHITECTURAL DESIGN

It addresses the privacy issue in Location-Sharing-Based Services (LSBS) by focusing on a specific problem called the *Fair Rendez-Vous Point (FRVP)* problem. For a set of user location preferences, the FRVP problem is to settle on a location amongst the proposed ones such that the maximum distance between this location and all other users' locations is minimum, i.e. *fair* to all users.

The mobile devices are able to perform public-key cryptographic operations. Each of the N users has his own public/private key pair ($K_P^{u_i}, K_S^{u_i}$), certified by a trusted CA, which is used to digitally sign/verify the messages that are sent to the LDS. Furthermore, N users share a common secret that is utilized to generate a shared public/private key pair $\left(K_P^{I_n}, K_S^{I_n} \right)$ in an online fashion for each meeting setup instance n. The private key $K_S^{I_n}$ generated in this way is known only to all meeting participants, whereas the public key $K_P^{I_n}$ is known to everyone including the LDS. The LDS executes the FRVP algorithm on the inputs it receives from the users in order to compute the FRV point. The LDS is also proficient to do public-key cryptographic functions. A common public-key infrastructure using the RSA cryptosystem [17] could be employed. Let K_P^{LDS} be the public key, certified by a trusted CA, and K_S^{LDS} the corresponding private key of the LDS. K_P^{LDS} is publicly known and users encrypt their input to the FRVP algorithm using this key; the encrypted input can be decrypted by the LDS using its private key K_S^{LDS}. This ensures message confidentiality and integrity.

5.1.1 Location Determination Server

The primary type of LDS adversarial behavior that we want to protect against is an honest-but-curious or semi-honest adversary, where LDS is assumed to execute the algorithms correctly. It may try to learn information about the users' location preferences from the usual inputs, the intermediary results and the produced outputs. Service providers have a commercial interest in providing a faithful service to their customers, the assumption of a semi-honest LDS is generally sufficient. We will later also analyze how our proposed solutions fair against certain active attacks, including collusion with users and fake user generation.

5.1.2 Users

Similar to the LDS assumption, our main goal is to protect against semi-honest participating users who may want to learn the private location preferences of other users from the intermediate results and the output of the FRVP algorithm, referred as passive attacks. As user inputs are encrypted with the LDS's public key K_P^{LDS}, there is a confidentiality guarantee against basic eavesdropping by participants and non-participants. The goal is of protecting against semi-honest users.

6. IMPLEMENTATION

6.1 Algorithms

Proposed algorithms receive advantage of the homomorphic properties of well-known cryptosystems, like BGN, ElGamal and Paillier, in order to privately compute an optimally fair Rendez-Vous point from a set of user location preferences.

6.2 Implementation Modules

In this section, we outline the details of proposed protocol for solving the PPFRVP problem. In order to separate the optimization aspect from the implementation, we first formally outline the fairness and transformation functions and then discuss the construction of the PPFRVP protocol. In general any PPFRVP algorithm should accept the inputs and generate the outputs, as described below.

Input: Transformation function f of private locations L_i. Where f is a secret-key based encryption function which determines the input L_i without knowing the secret key.

Output: An output is $f\ (L_{fair})$, where g is a fairness function and $L_{fair} = (x_l\ , y_l\ )$ is the fair rendez-vous location such that it is hard for the LDS to determine L_{fair} by just observing $f(L_{fair})$.

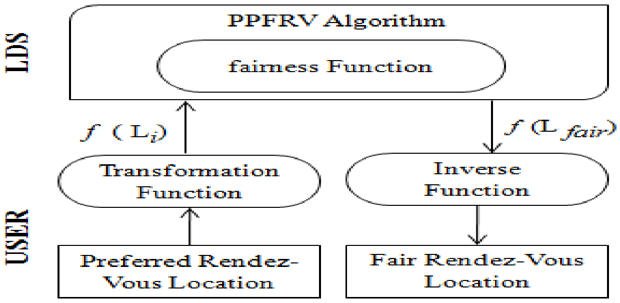

Fig. 1 Functional diagram of the PPFRVP protocol

Fig. 1 shows a functional diagram of the PPFRVP protocol, wherein the PPFRVP algorithm A is executed by an LDS. The fairness function g can be defined in several ways, depending on the preferences of users or policies.

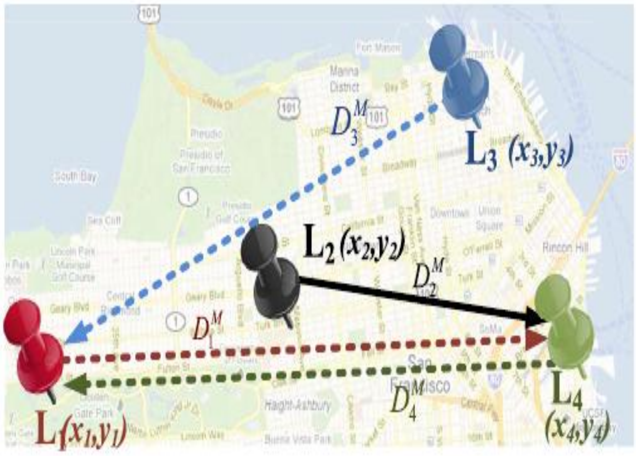

Fig. 2 PPFRVP scenario

Fig. 2 shows one such fairness function that minimizes the maximum displacement of any user to all other locations. This function is globally fair and can be easily extended to include additional constraints and parameters.

6.2.1 Fairness Function g

To determine a Rendez-Vous location that is *fair* to each and every user, the fairness function needs to optimize based on the spatial constraints set by the users' preferred locations. A Rendez-Vous location $L_{fair} = (x_l\ , y_l\ )$ among N users $\mathbb{U} = \{u_i\}_{i=1}^{N}$ will be fair to all users if everyone can reach L_{fair} in a "reasonable" amount of time. Another criterion is to minimize the total displacement of all users in order to reach L_{fair} , or making sure that no user is "too far" from L_{fair} as compared to further users. We model the fairness criteria of the PPFRVP problem by using a formulation of the *k-center* problem. In the k-center problem, the goal is to determine k locations $(L_1, \ldots, L_k\ )$ for placing facilities, among N possible candidates, such that the maximum distance from any place to its closest facility is minimized. For a two dimensional coordinate scheme, the Euclidean distance metric is usually employed. Fig. 2 shows a PPFRVP scenario modeled as a k-center problem. It should be noted that the current k-center formulation does not encompass other fairness parameters, for instance accessibility of a place and the means of transportation. Let $d_{ij} \geq 0$ be the Euclidean distance between two points L_i , L_j and $D_i^M = max_{j\neq i}\, d_{ij}$ be the maximum distance from L_i to any other point L_j. LDS privately compute the fair Rendez-Vous location; the fairness function g would be required to operate without having access to the location preferences L_i. This can be accomplished using cryptographic techniques with homomorphic encryption properties.

6.2.2 Transformation Function f

We are interested in using cryptographic schemes that allow us to compute the Euclidean distance between two points and the maximization/minimization functions. We utilize cryptographic schemes with homomorphic properties, specifically, *Boneh-Goh-Nissim* (BGN) [18], *ElGamal* [19] and *Paillier* [20] cryptosystems, as the transformation function f in our PPFRVP protocol. Given two plain text m_1, m_2 with their respective encryptions $E(\,m_1)$, $E(\,m_2)$, the multiplicative homomorphic property (ElGamal and partial BGN ciphers) states that $E(\,m_1) \odot E(\,m_2) = E(\,m_1\ .\ m_2)$ where $\odot$ is an arithmetic operation in the encrypted domain that is equivalent to the usual multiplication operation in the plain text. The additive homomorphic property (BGN and Paillier schemes) states that $(\,m_1)\ \oplus\ E(\,m_2) = E(\,m_1\ +\ m_2)$, where $\oplus$ is an arithmetic operation in the encrypted domain which is equivalent to the usual sum operation in the plain text domain.

6.2.3 Distance Computations

The fair Rendez-Vous point L_{fair} is the location preference that minimizes the maximum distance between any other location preference and L_{fair} . In these algorithms, we minimize with respect to the *square* of the distances, because distance squares are much easier to compute in an oblivious fashion with the help of homomorphic encryptions than simple distances.

6.2.3.1 BGN-Distance

First distance computation algorithm is based on the BGN encryption technique. This novel protocol requires only one round of communication between each user and the LDS, and it efficiently uses both the multiplicative and additive homomorphic properties of the BGN scheme. The BGN-distance protocol works as follows.

i. Every user u_i creates the vectors $E_i(a)$ and $E_i(b)$, where the encryption is done using the BGN scheme with the fresh session key $K_P^{I_n}$, $L_i = (x_i\ ,y_i\ )$ is the desired Rendez-Vous location of user u_i.

ii. Each user sends the two vectors $E_i(a)$, $E_i(b)$ over a secure channel to the LDS.

iii. LDS computes the scalar product $E_i(a)\,.\,E_i(b)$ of the received vectors, which produces the encrypted pairwise distances $E(d_{ij}^2)$ by first applying the multiplicative and then the additive homomorphic property of BGN.

6.2.3.2 Paillier-ElGamal-Distance

In addition to the multiplicative homomorphic property of ElGamal, we rely on the two following properties of the Paillier encryption

$$E(m_1)\cdot E(m_2)\ =\ E(\,m_1 + m_2\ \ mod\,n)$$

$$E(m_1)^r\ =\ E(\,r\ .\ m_1\ \ \ mod\,n)$$

As neither Paillier nor ElGamal possess both multiplicative and additive properties, the resulting algorithm requires one extra step in order to obliviously compute the pairwise squared distances d_{ij}^2 . In this scheme the participating users derive two pairs of public/private session keys $\{(K_P^{I_{n1}}, K_S^{I_{n1}}), (K_P^{I_{n2}}, K_S^{I_{n2}})\}$ from the shared secret, where the pair n_1 is used with the ElGamal encryption scheme and n_2 with the Paillier one. The distances are computed as follows.

i. Each user u_i creates the vectors $E_i(a)$.

ii. Each user u_i sends the vector $E_i(a)$ to the LDS, encrypted with LDS's public key.

iii. LDS computes the scalar product of the second and fourth element. To hide result from the users, the LDS obliviously randomizes these results with random values.

iv. After choosing random values, the LDS computes their inverses.

v. LDS permutes randomized scalar product element with its private element-permutation function and sends N such distinct elements to each user u_i.

vi. Each user simply decrypts the received elements with the ElGamal private key $K_S^{I_{n1}}$ and re-encrypts them with the Paillier public key $K_P^{I_{n2}}$. Then, each user sends the re-encrypted elements to the LDS in the same order as he received it.

vii. LDS reverts the element permutation function.

viii. Finally LDS computes the d_{ij}^2 for all i, j, after having removed the randomizing factors with their inverses

At this point, the LDS compute $E(\,d_{ij}^2\,)$, the encrypted square of the pairwise distances between all pairs of user-desired locations $L_i \neq L_j$.

6.2.4 PPFRVP Protocol

The PPFRVP protocol has three main modules

6.2.4.1 Distance Computation

The distance computation module uses either the BGN-distance or the Paillier-ElGamal distance protocols. $E(.)$ refer to encryption using either the BGN or the Paillier encryption scheme.

6.2.4.2 MAX Computation

LDS hide the values within the encrypted elements before sending them to the users. This is done to avoid disclosing private information, such as the pairwise distances or location preferences to users and carried out as:

i. For each index i, LDS generates two random values; those are used to scale and shift the encrypted square distance between L_i and L_j, obtaining. This is done in order to (i) ensure privacy of real pairwise distances, (ii) preserve the internal order among the pairwise distance from each user to all other users.

ii. LDS chooses two private element-permutation functions one for i and another for j and permutes d_{ij}^* . LDS sends N such distinct elements to each user.

iii. Each user decrypts the received values, determines their maximum and sends the index of the maximum value to the LDS.

iv. LDS reverses the permutation functions and removes the masking from the received indexes corresponding to the maximum distance values.

6.2.4.3 ARGMIN MAX Computation

The ARGMIN MAX computation carried out as:

i. LDS masks the true maximum distances by scaling and shifting them by the same random amount such that their order is preserved. Then, the LDS sends to each user all the masked maximum distances.

ii. Each user decrypts the received masked (scaled and shifted) maximum values, and determines the minimum among all.

iii. Each user knows which identifier corresponds to him and the user whose preferred location has the minimum distance sends to all other users the fair rendezvous location in an anonymous way.

After the last step, every user receives the final fair Rendez-Vous location, but no other information regarding non-fair locations or distances is leaked.

7. CONCLUSION

This paper proposes a new method for providing the users of LBS with location privacy. Our method is based on the homomorphic properties of well-known cryptosystems to privately compute an optimally fair Rendez-Vous point from a set of user location preferences. The scheme relies on a basic method, consists of calculations of the average location of a set of users, and improves it in the sense that it guarantees the location privacy of the users and the location exchange among them by using a public-key privacy homomorphism. Proposed solutions will preserve user preference privacy and have acceptable performance in a real implementation. Moreover, in proposed algorithms, users will have several prioritized locations preferences. In particular, it may encourage users to stop revealing sensitive information to third-parties and untrusted users, such as their home and work locations, and agree to privacy-preserving mechanisms.

8. REFERENCES

[1] Igor Bilogrevic, Murtuza Jadliwala, Vishal Joneja, Kübra Kalkan, Jean-Pierre Hubaux, and Imad Aad, “Privacy-Preserving Optimal Meeting Location Determination on Mobile Devices” IEEE Transactions on Information Forensics and Security, vol. 9, no. 7, pp. 1141-1156, JULY 2014.

[2] (2011, Nov.). Facebook Statistics [Online]. http://www.facebook.com/press/info.php?statistics

[3] P. Golle and K. Partridge, “On the anonymity of home/work location pairs,” in Proc. 7th Int. Conf. Pervasive Computing, pp. 390–397, 2009.

[4] J. Freudiger, R. Shokri, and J.-P. Hubaux, “Evaluating the privacy risk of location-based services,” in Proc. 15th Int. Conf. Financial, pp. 31–46, 2011.

[5] J. Freudiger, M. Jadliwala, J.-P. Hubaux, V. Niemi, P. Ginzboorg, and I. Aad, “Privacy of community pseudonyms in wireless peer-to-peer networks,” Mobile Netw. Appl., vol. 18, no. 3, pp. 413–428, 2012.

[6] (2011, Nov.). Please Rob Me [Online]. Available: http://pleaserobme.com/

[7] J. Krumm, “A survey of computational location privacy,” Personal Ubiquitous Computing, vol. 13, no. 6, pp. 391–399, 2009.

[8] (2011). Microsoft Survey on LBS [Online]. Available: http://go.microsoft.com/?linkid=9758039

[9] K. B. Frikken and M. J. Atallah, “Privacy preserving route planning,” in Proc. ACM WPES, pp. 8–15, 2004.

[10] P. Santos and H. Vaughn, “Where shall we meet? Proposing optimal locations for meetings,” in Proc. MapISNet, 2007.

[11] G. Zhong, I. Goldberg, and U. Hengartner, “Louis, Lester and Pierre: Three protocols for location privacy,” in Proc. 7th Int. Conf. PrivacyEnhancing Technologies, pp. 62–76, 2007.

[12] F. Berger, R. Klein, D. Nussbaum, J.-R. Sack, and J. Yi, “A meeting scheduling problem respecting time and space,” GeoInformatica, vol. 13, no. 4, pp. 453–481, 2009.

[13] S. Jaiswal and A. Nandi, “Trust no one: A decentralized matching service for privacy in location based services,” Proc. ACM MobiHeld, 2010.

[14] S. Guha, M. Jain, and V. Padmanabhan, “Koi: A location-privacy platform for smartphone apps,” Proc. 9th USENIX Conf. NSDI, 2012.

[15] B. Carbunar, R. Sion, R. Potharaju, and M. Ehsan, “The shy mayor: Private badges in geosocial networks,” in Proc. 10th Int. Conf. ACNS, pp. 436–454, 2012.

[16] S.Pidcock and U. Hengartner, “Zerosquare: A privacy-friendly location hub for geosocial applications,” Proc. 2nd ACM SIGCOMM Workshop Networking, Systems, and Applications Mobile Handhelds, 2013.

[17] R. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” Commun. ACM, vol. 21, no. 2, pp. 120–126, 1978.

[18] D. Boneh, E.-J. Goh, and K. Nissim, “Evaluating 2-DNF formulas on ciphertexts,” in Proc. TCC, pp. 325–341, 2005.

[19] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” IEEE Trans. Inf. Theory, vol. 31, no. 4, pp. 473–481, Jul. 1985.

[20] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in Proc. 17th Int. Conf. Theory Application Cryptographic Techniques, pp. 223–238, 1999.