BibTeX:

Privacy Preserving Aggregation of Distributed Mobility Data Streams

Thomas Liebig

TU Dortmund University, Germany

Abstract. Proliferation of pervasive devices capturing sensible data streams, e.g. mobility records, raise concerns on individual privacy. Even if the data is aggregated at a central server, location data may identify a particular person. Thus, the transmitted data must be guarded against re-identification and an un-trusted server. This paper overcomes limitations of previous works and provides a privacy preserving aggregation framework for distributed data streams. Individual location data is obfuscated to the server and just aggregates of k persons can be processed. This is ensured by use of Pailler's homomorphic encryption framework and Shamir's secret sharing procedure. In result we obtain anonymous unification of the data streams in an un-trusted environment.

Keywords. Privacy Preserving Big Data Collection, Mobility Analysis, Distributed Monitoring, Stream Data

1. Introduction

Smartphones became a convenient way to communicate and access information. With the integration of GPS sensors mobility mining was pushed forward (Giannotti & Pedreschi, 2008). The mobility information of multiple devices is usually stored on a server which performs analysis in order to extract knowledge on the movement behavior. In the easiest case this is the number of visitors to dedicated places, compare Figure 1.

Published in “Proceedings of the 11th International Symposium on Location-Based Services”, edited by Georg Gartner and Haosheng Huang, LBS 2014, 26–28 November 2014, Vienna, Austria.

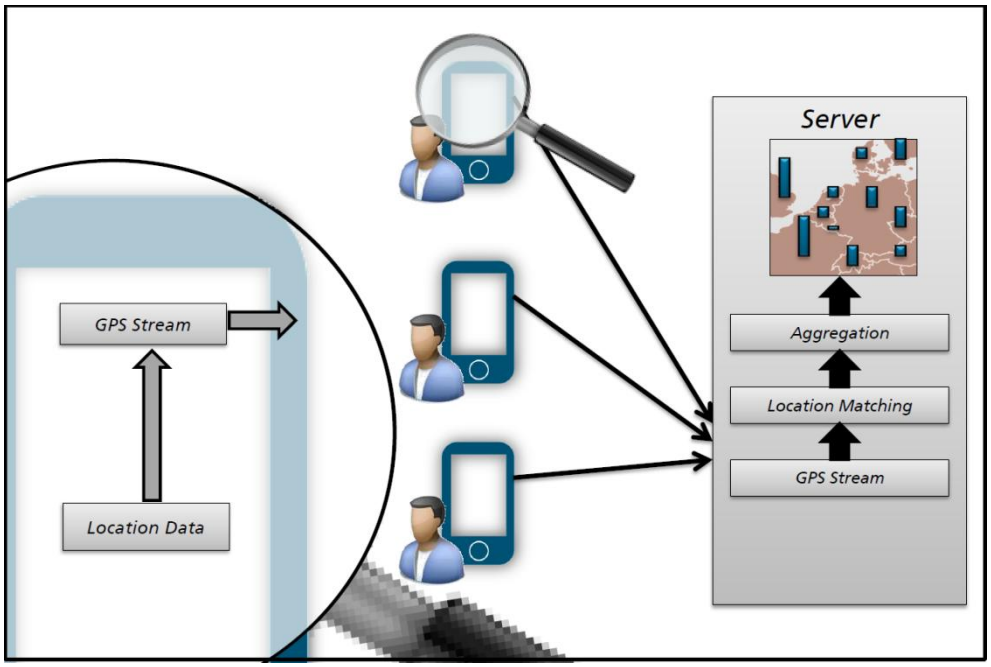

Figure 1: Centralized Mobility Data Analysis

The processing of the data streams became infeasible for large use cases, where millions of people are monitored, and massive data streams have to be processed. In this Big Data scenarios, the expensive computation (matching and counting in individual, continuous GPS streams) is split among the parties and just the aggregation step remains in the server. Thus, the continuous movement records (GPS) are reduced to episodic movement data (Andrienko et. al. 2012) consisting of geo-referenced events and their aggregates: number of people visiting a certain location, number of people moving from one location to another one, and so on. The preprocessing of the GPS data streams is then locally embedded in the location based devices and the aggregation is subject to crowd sourcing. Recent work focuses on in-situ analysis to monitor location based events (*visits* (Kopp et. al. 2012), *moves* (Hoh et. al. 2012)) or even more complex *movement patterns* (Florescu et. al. 2012) in GPS streams. In all cases a database with the locations or patterns of interest is provided in advance, and the mobile device computes event-histograms for succeeding time-slices. These histograms are much smaller and may be aggregated by the server in order to achieve knowledge on current movement behavior, compare Figure 2.

However, the transmission of these individual movement behaviors still poses privacy risks. The devices monitor daily behavior and thus reveal working place and hours, the place where we spent the night and other loca-

tions indicating information on sensitive subjects as health, religion, political opinions, sexual orientation, etc. Thus, the transferred episodic movement data may even lead to re-identifications.

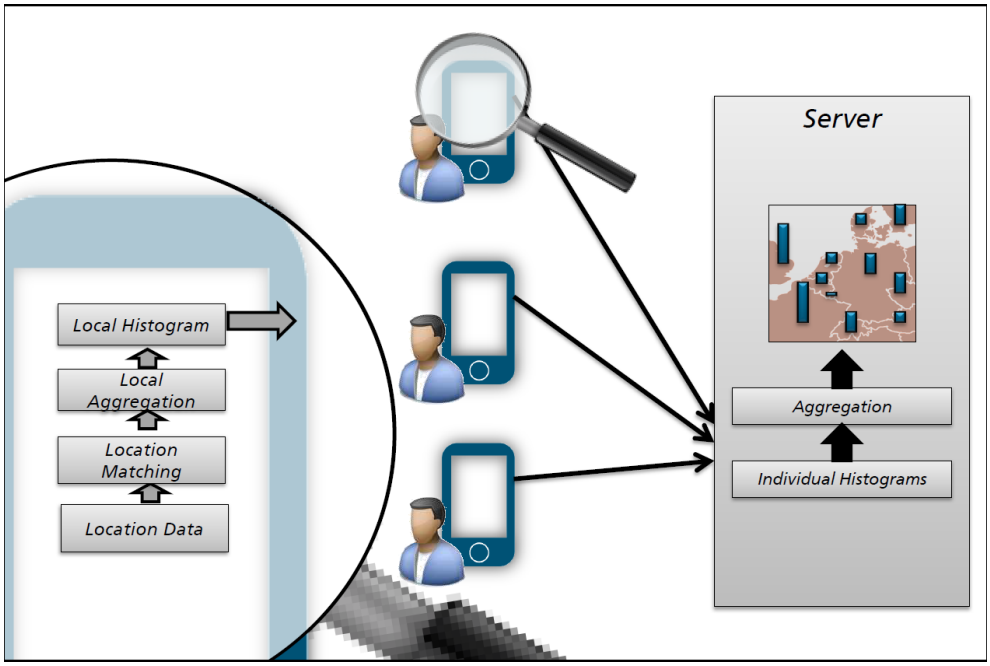

Figure 2: Aggregation of Distributed mobility Data Streams

The problem we thus focus is the protection of the individual histogram in such a data stream of locally aggregated mobility events. The adversary model is a corrupted server that utilizes the received individual histogram for inferences on the identities and other sensitive data.

Existing methods either act on the network layer (Kopp et. al. 2012) or inspired by the differential privacy paradigm they add random noise (Monreale et. al. 2013). The work in (Clifton et. al . 2004) denotes a protocol for secure aggregation among multiple parties, but their algorithm requires extensive communication among the parties and is infeasible in the considered crowd sourcing (i.e. single server) scenario, also their encryption can be broken after several computation cycles.

In contrast, our approach bases on homomorphic crypto systems (Paillier, 1999). These are systems where the decryption of several multiplied encrypted values reveals the sum of the original messages. Similarly to the RSA algorithm (Rivest et. al. 1983), the system, based on (Damgard & Jurik, 2001), uses one-way encryption functions to protect the messages. Thus a public key is used for encryption and a secret private key will be used for

decryption. We share the secret key among the clients in the network using hamir's secret sharing scheme (Shamir, 1979). The temporal entanglement of the messages is prevented using a one-way hash as in (Lamport, 1981).

The paper proceeds with a detailed discussion of latest work that tackle the described problem. Afterwards our approach is presented in conjunction with preliminaries on crypto systems. However, our approach poses new requirements to the architecture from Figure 2, which are briefly discussed afterwards. We conclude with a discussion of our achievements and an outlook on future research.

2. Related Work

The problem to protect individual privacy in a distributed scenario with an untrusted server receives increasing importance with the spread of Big Data architectures and the wide availability of massive mobility data streams. Thus, the problem is subject of many recent publications.

The work in (Abul et. Al. 2008) computes k-anonymity and assumes a trusted server. The work from (Kopp et. al. 2012) tries to solve the untrusted server problem by introduction of an obfuscation layer in the network communication, see Figure 3. But individual location data is identifying, even if it is aggregated in space-time compounds (Monreale et. al. 2010). Therefore, this work still delivers the vulnerable data to the server. Recently, differential privacy was applied to the problem in (Monreale et. al. 2013). Originated in database theory, differential privacy implies that adding or deleting a single record to a database does not significantly affect the answer to a query (Dwork et. al. 2006). The work in (Monreale et. al. 2013) follows the common method to achieve differential privacy by adding Laplace noise (with the probability density function $(\mu, \lambda) = p(x|\mu, \lambda) = \frac{1}{2\lambda} e^{-\frac{|x-\mu|}{\lambda}}$, where μ is set to zero and λ=1/ε) to every flow value in the vector, as proposed in (Dwork et. al. 2006), compare Figure 4.However, for cell counts differential privacy is known to provide strange behavior, especially if a large number of cells are zero (Muralidhar & Sarathy, 2011).

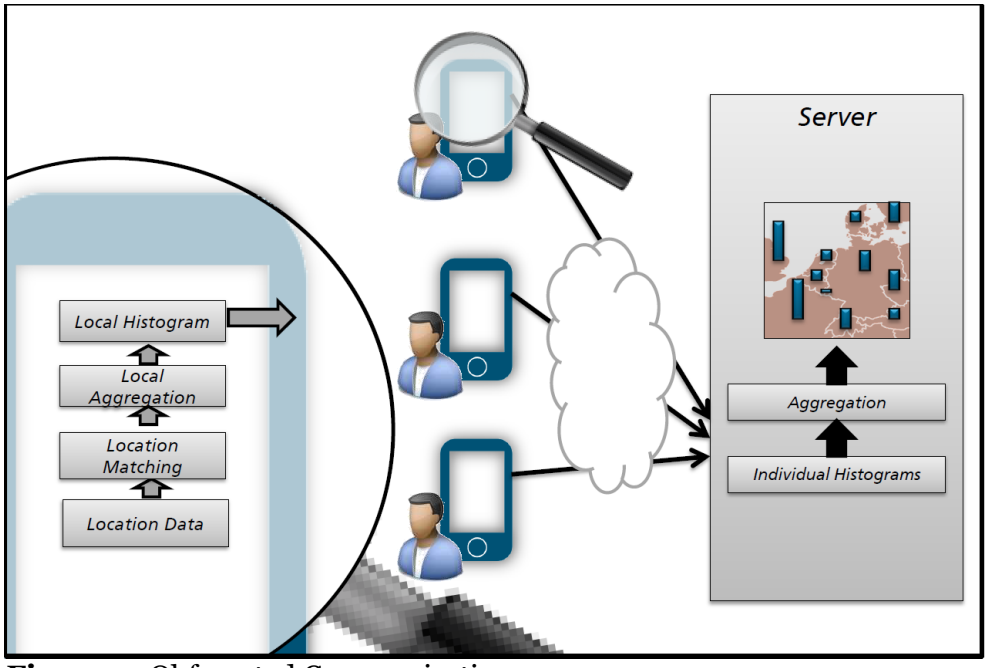

Figure 3: Obfuscated Communication

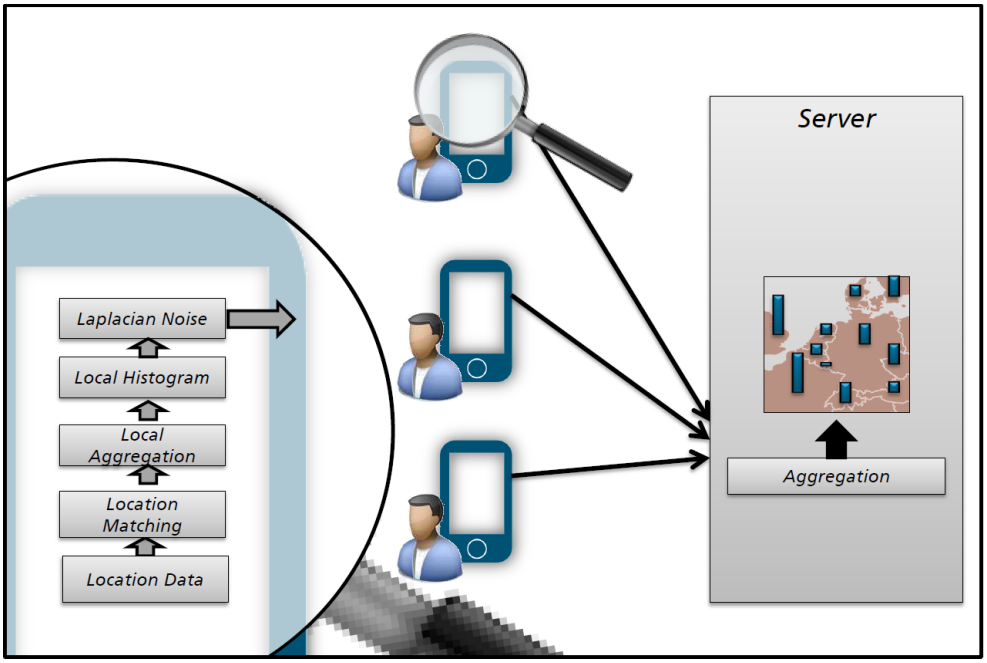

Figure 4: Differential Privacy

Moreover, movement often is a routine behavior and within their considered time interval most likely similar counts are produced for every person, this offers a chance to extract the mean and thus the correct value of the distribution within a stream environment (Duan, 2009) as the noise is sampled from $Lap(0, 1/\varepsilon)$ instead of sampling from $Lap(0, m/\varepsilon)$, where m denotes the expected number of queries. Additionally, movement is not random, and thus the frequencies in the vector are not independent, but correlate (Liebig et. al. 2008, Liebig et. al. 2009). Thus, combination of various noisy replies may be utilized to reveal the true distributions.

In contrast, our approach based on homomorphic cryptology in conjunction with a shared key ensures that individual data may not be accessed by the server but only aggregates of at least k people can be used, Since k may equal the number of clients, no data on the individual persons need to be revealed.

3. Proposed Cryptographic Approach

In contrast to previously described approaches our method (1) encrypts the values of the histogram, (2) communicates these ciphertexts to the server, (3) aggregates the ciphertexts and finally (4) decrypts the result, see an overview in Figure 5. The process utilizes asymmetric cryptography methods using two separate keys: one for encryption and another one for decryption. The utilization of a homomorphic crypto system in conjunction with Shamir's secret sharing guarantees that the individual messages cannot be restored, but their sum.

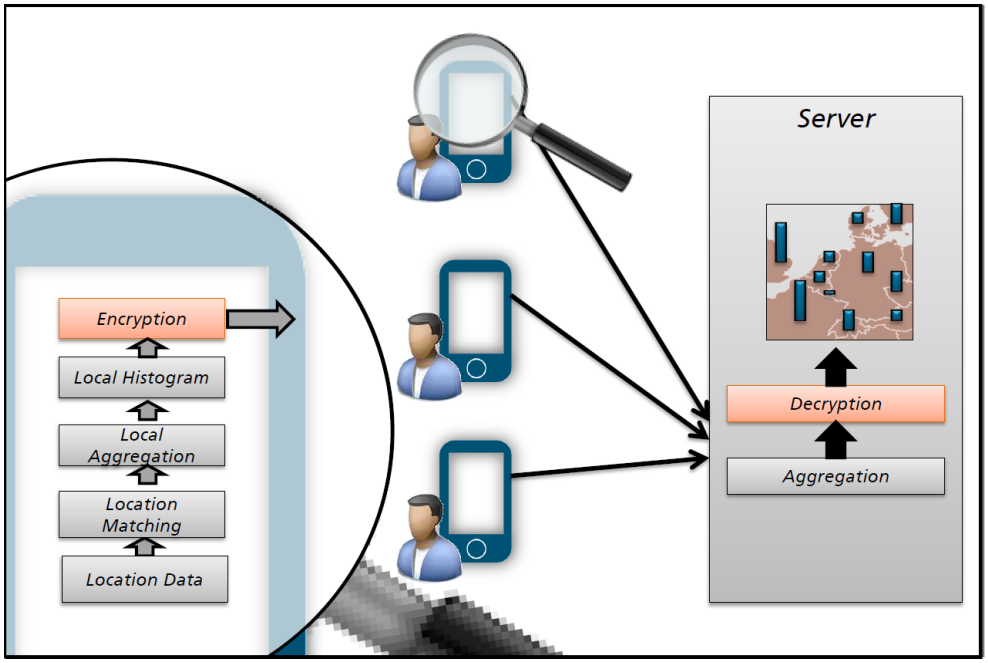

Figure 5: Proposed Cryptographic Approach for Privacy Preserving Aggregation of Distributed Mobility Data Streams

As our method bases on the RSA-method (Rivest et. al. 1983), homomorphic crypto systems (Paillier, 1999), (Damgaard & Jurik, 2001), Shamir's secret sharing (Shamir, 1979) and the work on hash chains, described in (Lamport, 1981), we proceed with a brief primer and describe our method afterwards.

The RSA-algorithm (Rivest et. al. 1983) is an asymmetric crypto system. The system bases on two keys, a *private key* which is used for decryption and a *public key* used for encryption. Whilst the public key can be shared with multiple parties, the private key is the secret of the receiver, and may hardly be computed from the public key.

3.1. RSA Algorithm

The RSA method uses one-way functions. These are functions which are easy to compute in one direction but difficult to reverse. A simple metaphor of this function is a phone book: While it is easy to derive the call number of a particular person, it is hard to look up the name given a phone number.

Preliminary for understanding is the notion of multiplicative inverse b of a number a, which is defined as $a \cdot b = 1 \bmod m$. This inverse just exists, if m and a are co-prime, i.e. $gcd(m, a) = 1$.

Consider a communication among the client who wants to send a message to the server. In this case, the system works as follows. In a key generation process, the server chooses two different primes p and q and computes $n = pq$ and $m = (p-1)(q-1)$. Furthermore, the server chooses a number a which is co-prime to m. The public key, created by the server, then denotes as $pk = (n, a)$. The server computes the multiplicative inverse $b = a^{-1} \bmod m$ of a, which is the secret private key.

Encryption

The client has a message x, with $x < m$. He sends the ciphertext c, computed as $E(x, pk) = x\hat{}a \bmod n$.

Decryption:

The server decrypts the message and restores the plaintext by computing

$x = D(c) = c\hat{}b \bmod n$.

The system is secure, as knowledge of n does not reveal p and q, since factorization is in NP (Johnson, 1984).

3.2. Homomorphic Crypto Systems

A public key encryption scheme (E, D), where E and D are algorithms for encryption and decryption, is homomorphic when it meets the condition $D(E(m_1) \cdot E(m_2)) = m_1 + m_2$.

Our approach bases on the generalisation of Paillier's public-key system (Paillier, 1999), introduced in (Damgaard & Jurik, 2001). Their crypto system uses computations modulo $n\hat{}(s+1)$, with n being the RSA modulus and s a natural number. By setting $s = 1$ Paillier's scheme is a special case (Paillier, 1999). If $n = pq$ with p and q being odd primes, then the multiplicative group $\mathbb{Z}^*_{n^{(s+1)}}$ is a direct product of $\mathcal{G} \times \mathcal{H}$, where $\mathcal{G}$ is of cyclic order n^s and $\mathcal{H}$ is isomorphic to $\mathbb{Z}^*_n$. Thus, $\overline{\mathcal{G}}$=$\mathbb{Z}^*_{n^{(s+1)}}/\mathcal{H}$ is cyclic of order n^s.

For an arbitrary element $a \in \mathbb{Z}^*_{n^{(s+1)}}$, a=$\overline{a}$=a$\mathcal{H}$ denotes the element represented by a in the factor group $\overline{\mathcal{G}}$.

Choose $g \in \mathbb{Z}^*_{n^{(s+1)}}$ such that $g = (1+n)^{jx} \bmod n^{s+1}$ for known j relatively prime to n and $x \in \mathcal{H}$. Let λ be the least common multiplier of $p-1$ and $q-1$, $\lambda := lcm(p-1, q-1)$. Choose d by the Chinese Remainder Theorem, such that $d \bmod n \in \mathbb{Z}^*_n$ and $d = 0 \bmod \lambda$. The public key then is n, g whilst the secret key is d.

Encryption:

The plaintext m is element of $\mathbb{Z}^*_{n^s}$. With a plaintext m we choose at random r in $\mathbb{Z}^*_{n^{(s+1)}}$. The ciphertext $E(m, r)$ computes as:

$E(m,r) = g\hat{}m\, r^{n^s} \bmod n\hat{}(s+1)$.

Decryption:

For the ciphertext c compute $c\hat{}d \bmod n\hat{}(s+1)$. If $c = E(m,r)$ this results in

$$\begin{aligned} c^d &= \left(g^m r^{n^s}\right)^d = E(m,r) \\ &= \left((1+n)^{\{jm\}} x^i r^{\{n^s\}}\right)^d \\ &= (1+n)^{jmd \bmod n^s} \left(x^m r^{\{n^s\}}\right)^{\{d \bmod \lambda\}} \\ &= (1+n)\hat{}(jmd \bmod n^s)\,. \end{aligned}$$

In (Damgaard & Jurik, 2001) an algorithm is proposed to compute $jmd \bmod n^s$. Their method bases on a function $L(b) = (b-1)/n$ which ensures that

$$L(1+n)^i \bmod n^{\{s+1\}} = \left(i + \binom{i}{2} n + \cdots + \binom{i}{s} n^{\{s+1\}}\right) \bmod n^s\,.$$

The basic idea of their algorithm is to compute the value iteratively in a loop For convenience, their algorithm is cited in Algorithm 1.

With the same method computed for g instead of c the value $jd \bmod n^s$ is computed. The plaintext then is:

$$(jmd) \cdot (jd)^{-1} = m \bmod n^s\,.$$

$i := 0$

For $j := 1$ to s

$$t_1 := L(a \bmod n\hat{}(j+1))$$

$t_2 = i$

For $k := 2$ to j

$i := i - 1$

$t_2 := t_2 \cdot i \bmod n^j$

$$t_1 := t_1 - \frac{\{t_2 \cdot n^{\{k-1\}}\}}{k!} \bmod n^j$$

ENDFOR

$i := t_1$

ENDFOR

Algorithm 1: Damgard Jurik Algorithm (Damgaard & Jurik, 2001)

The crypto system is additively homomorphic. As example consider two

messages m_1 and m_2 which are encrypted using the same public key pk such that $c_1 = E(s, pk)(m_1, r_1)$ and $c_2 = E(s, pk)(m_2, r_2)$ then $c_1 c_2 = g^{m_1} g^{m_2} r_1^{n^s} r_2^{n^s} = g^{(m_1+m_2)} r^{n^s}$ so $c_1 c_2 = E(s, pk)(m_1 + m_2, r)$.

3.3. Shamir's Secret Sharing

The work presented in (Shamir, 1979) discusses how to distribute a secret value d among n parties, such that at least k parties are required for restoring the secret.

The idea utilizes a polynomial function $f(x) = \sum_{i=0}^{k-1} a_i x^i$,with $a_0 = d$, and distributes the values $f(i)$ to the parties.

In case k of these values are commonly known, the polynomial $f(0)$ can be restored.

The advantage of this method is that the shared parts are not larger than the original data. By some deploying strategies of the parts hierarchical encryption protocols are also possible.

3.4. Hash Chain

The work in (Lamport, 1981) describes a method for authentication with temporally changing password messages. The passwords series are created in advance using a cryptographic hash function which is a one-way function $F(x)$. They are created as follows $F^n(x) = F\left(F^{(n-1)}(x)\right)$, where x is a password seed. The passwords are used in reversed order. Thus, the server stores the last value that the client sent, $F^n(x)$, and proves correctness of the new value $F^{n-1}(x)$ by verification of $F^n(x) = F(F^{n-1}(x))$. Afterwards the server stores the latest received value for the next check. As $F(\cdot)$ is a one-way function, the server may not pre-compute next password.

3.5. Putting Things Together

Our cryptographic system follows the protocol of the homomorphic crypto system in (Damgaard & Jurik, 2001). Consider communication among w clients with a single server. Similar to (Damgaard & Jurik, 2001) key generation starts with two primes p and q which are composed as $p = 2p' + 1$ and $q = 2q' + 1$, where p' and q' are also primes but different from p and q. The RSA modulus n is set to $n = pq$ and $m = p'q'$. With some decision for $s > 0$ the plaintext space becomes $\mathbb{Z}_{n^s}$. Next, d is chosen such that $d = 0 \bmod m$ and $d = 1 \bmod n^s$. Now, we use Shamir's secret sharing scheme (Shamir, 1979) to generate the private key shares of d to be divided among the clients. Thus, we apply the polynomial $f(X) = \sum_{i=0}^{w} a_i X^i \bmod l$, by picking a_i for $0 < i \leq w$ as random values from $0, \ldots, l$ and $a_0 = d$, l is a prime with $n^{s+1} < l$. We choose g as $g = n + 1$. The secret share of d for the i'th client will be $s_i = f(i)$. A verification key $v_i = v^{\{\Delta s_i\} \bmod} n^{s+1}$ is associated with

each client i. The public key then becomes (n, s, l) and $(s_1, \ldots, s_w)$ is a set of private key shares.

Encryption:

The plaintext of the ith client m'_i is multiplied with the one-way hash function $F^n = F(F^{n-1}(a))$ of a commonly known seed a. Thus the plaintext for the encryption results as $m_i := m'_i F^n$. Given this plaintext m_i we choose at random $r \in \mathbb{Z}^*_{n^{s+1}}$. The ciphertext $E(m_i, r)$ computes as:

$E(m_i, r) = g^{m_i}\ r\hat{}(n^s) \bmod n^{s+1}$.

The client i then communicates $c_i^{2\Delta s_i}$, with $\Delta = l!$ (Damgaard & Jurik, 2001).

Decryption:

The server can verify that the client raised s_i in the encryption step by testing for $log_{c_i^4}(c_i^2) = log_v(v_i)$. After the required k number of shares S arrived. They can be combined to (Damgaard & Jurik, 2001):

$c' = \prod_{i\ in\ S} c_i^{2\lambda^S_{0,i}} \bmod n^{s+1}$, where

$\lambda^S_{0,i} = \Delta \prod_{i' in\ S \setminus i} \frac{-i}{i-i'}$.

Thus, the value of c' has the form $c' = (\prod_{i\ in\ S} c_i)^{4\Delta^2 f(0)} = (\prod_{i\ in\ S} c_i)^{4\Delta^2 d}$. As $4\Delta^2 d = 0 \bmod \lambda$ and $4\Delta^2 d = 4\Delta^2 \bmod n^s$, $c' = (1+n)^{4\Delta^2 \sum_{i\ in\ S} m_i} \bmod n^{s+1}$. The desired plaintext $\sum_{i\ in\ S} m_i$ can be obtained by previously introduced algorithm and succeeding multiplication with $(4\Delta^2)^{-1}\ mod\ n^s$. The original plaintext can be computed by dividing the resulting sum by F^n. This ensures that previous messages may not be used for analysis of current messages. The homomorphic property of the system is directly used, and bases on the work presented in (Damgaard & Jurik, 2001).

Security:

The security of the crypto system is based on the *decisional composite residuosity assumption* already used by (Paillier, 1999). The assumption states that given a composite n and an integer z it is hard to decide whether z is a n-residue (i.e. a n-th power) modulo n^2, i.e. whether it exists an y with $z = y^n \bmod n^2$.

4. Consequences for the Architecture

As a consequence of our method the keys need to be distributed among the communicating parties: the clients and the server. This may not be done by the server, but has to be performed by a (commonly) trusted authority (TA).

Once the keys are distributed, the communication channel to this TA can be closed. Thus, no vulnerable data reaches this third party.

5. Discussion

The hereby presented method overcomes limitations of related work. In addition, our approach may be combined with the methods presented in (Monreale et. al. 2013). Thus, the transmitted histograms can be obfuscated by Laplacian noise (Monreale et. al. 2013). On the other hand transmission may not be obscured by anonymous messages (Kopp et. al. 2012) since the identifier of the clients is required for verification of the transmitted messages and reconstruction of the aggregated plaintext.

However, our method assumes that the space covered by individual movements overlaps. If this assumption does not hold, e.g. with persons from different cities, the privacy of each individual is not guaranteed (Abul et. al. 2008). An approach to overcome this limitation is by sending messages to the server just if the according entry in the histogram is at least once (i.e. the person was at least once at this location or used at least once the movement pattern). This ensures that the server may just decode the aggregated histogram if a sufficient number of people sent their messages and thus have been there. On the other hand, then the transmission of the message itself contains information on a person's movement behaviour. Thus, future studies should find a message encoding of a zero which does not allow to compute the aggregated sum but passes all verification steps of the server.

References

Abul, O., Bonchi, F., Nanni, M.:Never walk alone: Uncertainty for anonymity in moving objects databases, In Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, ser. ICDE ’08. Washington, DC, USA: IEEE Computer Society, (2008), 376–385.

Andrienko, N., Andrienko, G., Stange, H., Liebig, T., Hecker, D.: Visual Analytics for Understanding Spatial Situations from Episodic Movement Data. KI - Künstliche Intelligenz (2012) 241-251

Clifton, C., Kantarcioglu, M., Doan, A., Schadow, G., Vaidya, J., Elmagarmid, A.K., Suciu, D.: Privacy-preserving data integration and sharing. In: DMKD. (2004) 19-26

Damgard, I., Jurik, M.: A generalisation, a simplification and some applications of paillier’s probabilistic public-key system, In Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography, ser. PKC ’01. London, UK, UK: Springer-Verlag, (2001), pp. 119–136.

Duan, Y.: Privacy without noise, In Proceedings of the 18th ACM conference on Information and knowledge management, CIKM '09. New York, NY, USA: ACM, (2009) 1517–1520.

Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis, In Proceedings of the Third conference on Theory of Cryptography, ser. TCC'06. Berlin, Heidelberg: Springer-Verlag, (2006) 265–284.

Florescu, S.C., Mock, M., K□orner, C., May, M.: Efficient Mobility Pattern Detection on Mobile Devices. In: Proceedings of the ECAI'12 Workshop on Ubiquitous Data Mining. (2012) 23-27

Giannotti, F., Pedreschi, D.: Mobility, Data Mining and Privacy - Geographic Knowledge Discovery. Springer (2008)

Hoh, B., Iwuchukwu, T., Jacobson, Q., Work, D.B., Bayen, A.M., Herring, R., Herrera, J.C., Gruteser, M., Annavaram, M., Ban, J.: Enhancing Privacy and Accuracy in Probe Vehicle-Based Traffic Monitoring via Virtual Trip Lines. IEEE Trans. Mob. Comput. 11(5) (2012) 849-864

Johnson, D. S.: The NP-completeness column: An ongoing guide, Journal of Algorithms, vol. 5, no. 3, (1984) 433–447.

Kopp, C., Mock, M., May, M.: Privacy-preserving distributed monitoring of visit quantities. In: Proceedings of the 20th International Conference on Advances in Geographic Information Systems. SIGSPATIAL '12, New York, NY, USA, ACM (2012) 438-441

Lamport, L.: Password authentication with insecure communication, Commun. ACM, vol. 24, no. 11, Nov. (1981) 770–772.

Liebig, T., Körner, C., May, M.: Scalable sparse bayesian network learning for spatial applications. Proceedings of the Workshops of the IEEE International Conference on Data Mining. ICDMW'08, Pisa, Italy, IEEE Press (2008) 420-425

Liebig, T., Körner, C., May, M.: Fast visual trajectory analysis using spatial bayesian networks. Proceedings of the Workshops of the IEEE International Conference on Data Mining. ICDMW'09, Miami, USA, IEEE Press (2009) 668-673

Monreale, A., Wang, W., Pratesi, F., Rinzivillo, S., Pedreschi, D., Andrienko, G., Andrienko, N.: Privacy-preserving distributed movement data aggregation. In: Geographic Information Science at the Heart of Europe. Lecture Notes in Geoinformation and Cartography. Springer International Publishing (2013) 225-245

Monreale, A., Andrienko, G., Andrienko, N., Giannotti, F., Pedreschi, D., Rinzivillo, S., Wrobel, S.: Movement data anonymity through generalization, Journal of Transactions on Data Privacy, vol. 3, no. 2, (2010) 91–121.

Muralidhar K., Sarathy, R.:Does differential privacy protect terry gross privacy? In Privacy in Statistical Databases, ser. Lecture Notes in Computer Science, J. Domingo-Ferrer and E. Magkos, Eds. Springer Berlin Heidelberg, vol. 6344, (2011) 200–209.

Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of the 17th international conference on Theory and application of cryptographic techniques. EUROCRYPT'99, Berlin, Heidelberg, Springer-Verlag (1999) 223-238

Prism, NSA slides explain the PRISM data-collection program. The Washington Post, Available: http://www.washingtonpost.com/wp-srv/special/politics/prismcollection-documents/ [Last accessed: 23 June 2013] (June 06, 2013)

Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 26(1) (January 1983) 96-99

Shamir, A.:How to share a secret, Communications of the ACM, vol. 22, no. 22, (1979) 612–613.