Privacy-Preserving Emotion Detection for Crowd Management

Zeki Erkin[1], Jie Li[2], Arnold P.O.S. Vermeeren[2], and Huib de Ridder[2]

[1] Cyber Security Group, Department of Intelligent Systems, Delft University of Technology, 2628 CD, Delft, The Netherlands
[2] Persuasive Experience Research, Industrial Design Engineering, Delft University of Technology, 2628 CD, Delft, The Netherlands
{z.erkin,j.li-2,a.p.o.s.vermeeren,h.deridder}@tudelft.nl

Abstract. Emotion detection plays a vital role in crowd management as it enables social event organizers to detect the actions of masses and react accordingly. There are several approaches to detect emotions in a crowd, including surveillance cameras, human observers and sensors. One other approach to gather emotion data is self-reporting. A recent study showed that self-reporting is feasible, reliable and efficient. However, there is a strong privacy concern among people that risks the use of such self-reporting mechanisms in wide use. In this work, we address the privacy aspect of self-reporting mechanism and propose a cryptographic approach that hides the sensitive data from the organizers but permits to compute statistical data for crowd management. The feasibility of using cryptography in real life for privacy protection is also investigated in terms of complexity.

Keywords: Emotion detection, crowd management, privacy, homomorphic encryption.

1 Introduction

Crowd management has become an important aspect of today's social life. Particularly in events with massive attendance, safety, security and guidance of the attendants are very important aspects. There are three phases of crowd management, before the event, during the event and after the event. Different parties like the event organizers, medical health support, fire department and the security team plan each step of the event carefully to mitigate possible problems that may arise.

Among many aspects of this procedure, measuring crowd emotion during the event has many valuable insights for the management of the crowd as well as assessing the success of the event afterwards. Emotions influence and serve as a predictor for human behavior [1]. As noted by Arnold in [2], emotions can be essentially characterized as "felt action tendencies", which could be understood as impulses that motivate people to move towards the stimulus appraised as beneficial and avoid the one appraised as harmful. That is to say, through

D. Ślęzak et al. (Eds.): AMT 2014, LNCS 8610, pp. 359–370, 2014.

understanding crowd emotions, crowd managers can judge crowd members intentions and predict their behaviors so as to act accordingly.

A recent study in this field presented a self-reporting mechanism to collect data from the attendants of an event [3]. In that work, the authors designed a non-intrusive software application for mobile phones to gather data that consist of a unique identifier for the device, location of the device, the emotion of the attendant and the perceived emotion of the crowd in proximity of the attendant, and drew interesting conclusions in terms of real-time crowd emotion maps based on self-report emotions from different areas of the event. The collected data reflected the real situation as we observed. In general, the amount of emotion reports in an area reflected the crowdedness of that area. Attendants' movement and emotional changes were consistent with the activities at the event. For example, when the performance of a stage stopped, it is observed that the amount of emotion reports declined in that area. More negative emotions popped up in an area on the emotion map when we received some spontaneous complaints from the attendants about the unsatisfactory performance at that area. Real-time crowd emotion maps do not only provide information about crowd size and density, but also rich emotion information for crowd managers to predict crowd behavior and prepare for the possible incidents. In addition, attendants' emotion reaction to the activities in the event can be used as an indicator for evaluating the event.

Among many aspects of crowd emotion detection, the authors point out issues related to trust and privacy as a serious concern. It is reported that many attendants were concerned about a number of possible misuse cases. Two most relevant concerns for this work are as follows:

- *Tracking.* Individuals can be tracked during and after the event using the software.
- *Identity linking.* Individuals and their emotion feedback can be linked and used against them.

Particularly, the identity linking problem observed to be a serious concern that demotivates the use of the software. This is a valid concern, considering that the emotion detection can be performed for any kind of events like political congresses, protests, uprisings, and riots as seen in many countries recently. This serious concern on privacy also clearly indicates that without proper security and privacy mechanisms employed, emotion detection for crowd management using software and even sensor technology can be perceived as a privacy invasion and not be accepted by the individuals. Obviously, for the better management of the crowd and for reducing the security and safety risks, a proper privacy protection mechanism should be provided to establish trust among people.

In this paper, we address the identity linking problem of the emotion detection and propose a scientific solution to protect the privacy of individuals. We define two entities: 1) a server, which collects data or the crowd management, and 2) users, whose data are collected using the software in [3]. Our goal is to protect the privacy of the users by hiding the sensitive emotional data from the server. While the server cannot observe the emotions of any user, it can still draw

conclusions based on statistical data such as the histogram of the emotional data for a specific location. We achieve our goal by deploying techniques known from cryptography. More precisely, the privacy sensitive data of the users are only given to the server in the encrypted form. Without having the decryption key, the server can still process the encrypted data for crowd management.

Our contributions are as follows:

1. We present a self-reporting mechanism for emotion detection that is privacy-preserving in a server-client model.
2. We propose a cryptographic protocol based on existing tools and optimized in terms of bandwidth using data packing [4].
3. We provide a complexity analysis to show that the proposed cryptographic protocol is feasible to be deployed in real life.

The rest of the paper is organized as follows: Section 2 presents the related work in this field. Section 3 provides background information for emotion detection software from [3], as well as the cryptographic tools that are used in this paper. Section 4 describes the privacy-preserving emotion detection protocol. Section 5 presents security and complexity analyzes. Section 6 provides a number of open questions for further research. Finally, Section 7 draws conclusions.

2 Related Work

To the best of our knowledge, privacy for emotion detection has not been addressed before in literature. Protecting privacy sensitive data, on the other hand, is a well-known topic and studied in different perspectives. Anonymization and perturbation techniques have been used in data mining [5,6]. Unfortunately, anonymization is not a possible solution for our case since the mobile devices can easily be identified by the server. Data perturbation techniques are not suitable either as they provide privacy at the cost of adding noise to the original data, which is not desirable in our scenario. Another approach in literature provides a way to perform the desired service based on privacy-sensitive data using cryptography [7]. This approach is based on cryptographic tools like homomorphic encryption [8] and multi-party computation techniques [9].

The main idea in cryptographic approach is to hide the sensitive data using encryption. The encrypted data is given to the server, which does not have the decryption key, and yet it can process the encrypted data using the homomorphic properties of the encryption scheme. This line of research has been applied effectively in many different domains, including but not limited to e-voting [10], biometric-medical data processing [11,12], recommender systems [13,14] and data clustering [15,16].

3 Preliminaries

In this section, we discuss the emotion detection in [3], briefly introduce the cryptographic tools we use, and present our security and privacy requirements.

3.1 Emotion Detection

We assume that user i sends the tuple T to the server using the software in [3]:

$$T =< ID, t, \ell, e_i, c_i > , \tag{1}$$

where ID is the unique mobile phone identifier, t is the time of the report, ℓ is the code for the location, e_i is the user emotion, and c_i is the user's observation on the crowd. The mobile software works as follows. After installing the application, the software determines the location of the user. However, the measurement is not precise due to technological challenges and thus, the user is asked to state their location –one of the six stages– as a precaution. The software, then, asks every half an hour about the emotional state of the user and the perceived emotional state of the crowd around that user. The reporting mechanism is designed carefully with an circular emotion detector, corresponding colors for emotions and cartoon characters. To motivate self-reporting, a game component is also added to the application where a number of reports are rewarded later. The collected data is then processed for analysis; an example is given in Figure 1. We refer readers to the original work [3] on the design of user interface and the details about the data collection.

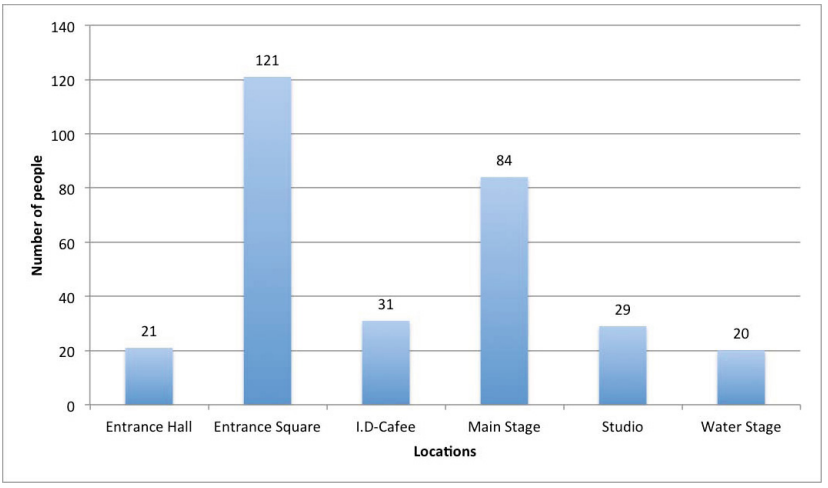

Fig. 1. One of the analysis from [3]: Amount of reports in six areas

3.2 Homomorphic Encryption

Encryption schemes are designed to make messages unreadable for anyone who does not have the decryption key. The output of an encryption function, called cipher text, looks completely random and thus, it is not possible to deduce meaningful information from it. Public key encryption schemes also output cipher text

that requires the decryption key of the recipient to obtain the plain text message. Even though it is not feasible to decipher the cipher text without the decrytion key, public key encryption schemes do preserve some structure after encryption that we can exploit. For example, Paillier encryption scheme [17] is additively homomorphic, meaning that if two cipher text are given, it is possible to obtain an encryption of the sum of original messages by simply multiplying their cipher texts. Given two messages, m_1 and m_2,

$$\mathcal{D}_{sk}(\mathcal{E}_{pk}(m_1) \cdot \mathcal{E}_{pk}(m_2) = m_1 + m_2) \ , \tag{2}$$

where $\mathcal{E}_{pk}(\cdot)$ and $\mathcal{D}_{sk}(\cdot)$ are encryption and decryption functions with public key and decryption key, respectively. Consequently, any plain text m can also be scaled with a public constant c due to the homomorphic property:

$$\mathcal{D}_{sk}(\mathcal{E}_{pk}(m)^c) = mc \ . \tag{3}$$

By using the additively homomorphic property, it is possible to realize linear functions using only encrypted inputs. In addition to homomorphism, the Paillier encryption scheme is also semantically secure. That is there is a random factor in the encryption function, which causes different cipher texts even the same message is encrypted. We refer interested user to [17] for more information on the encryption scheme.

In this paper, we use the threshold version of the Paillier encryption scheme [18]. The threshold version of the scheme is similar to the original one however, in order to decrypt a cipher text, a pre-defined number of users should participate. This means that each user in the system has a share of the decryption key and only with the contribution of any of the pre-defined number of users, the cipher text can be decrypted.

3.3 Privacy Requirements and Security Assumptions

In this work, we assume that the software transmits a unique identifier for each mobile device, along with the location and time information, user's emotion data, and the crowd emotion. Our aim is to hide the last three from the server, while it is still possible for the server to compute a number of functions on the encrypted versions of the data:

1. Total number of users for a specific location,
2. Emotion distribution of users in each location.

We assume that both the server and the users are honest-but-curious, a notion used in cryptography, meaning that they perform tasks according to the protocol description but they are curious at the same time and they might try to extract more information than they are entitled to from the communication.

4 Private Emotion Detection

In this section we describe the cryptographic protocol for privacy-preserving emotion detection. We assume that every mobile device has a unique identifier,

assigned by the software, and a Paillier public key pk and a corresponding decryption key share sk_i. We design our system for an event with N participants, where there are L different locations and E number emotions defined.

In our proposal, the server is not trusted with the raw data of the users. Furthermore, the mobile devices are considered to be limited, and thus the number of operations in terms of encryption and decryption should be kept minimum. In the following, we describe data formatting and collection, both of which are handled locally by the mobile devices, and data analysis, which is done by the server.

4.1 Data Formatting

The software formats the data before sending it to the server as follows: the location data is represented as a binary value which is constructed with L compartments of size $\log_2(N)$-bits:

$$\ell_i = (\ell_{i,1}||\ell_{i,2}||\dots||\ell_{i,L}) \ , \tag{4}$$

where $||$ is the concatenation operator. The software sets the value to 1 for the location of the user, and all others to 0. In a similar fashion, it also creates the following values:

$$\begin{aligned} e_i &= (e_{i,1}||e_{i,2}||\dots||e_{i,E}) \ , \\ c_i &= (c_{i,1}||c_{i,2}||\dots||c_{i,E}) \ , \end{aligned} \tag{5}$$

where e_i is the emotion indicator for user i and c_i is the emotion indicator of the crowd around user i. Only the value $e_{i,j}$ is set to 1 if the user's emotion code is j. This is also true for c_i. These two values are consisting of E compartments of size $\log_2(E)$ bits.

In addition to above values, the software also computes the following:

$$\begin{aligned} \tilde{e}_i &= \ell_i \times e_i \ , \\ \tilde{c}_i &= \ell_i \times c_i \ . \end{aligned} \tag{6}$$

Notice that in addition to the information sent in the original work, we now have two new values, $\tilde{e}_i$ and $\tilde{c}_i$. The former indicates the emotion of the user i at his/her current location, the later indicates the crowd emotion at that location as perceived by user i. These two values are also consist of L compartments; $L-1$ of them having $\log_2(N)$ zeros.

4.2 Data Collection

The software sends the tuple T' to the server at time slot t:

$$T' =< ID_i, t, \mathcal{E}_{pk}(\ell_i), \mathcal{E}_{pk}(e_i), \mathcal{E}_{pk}(c_i), \mathcal{E}_{pk}(\tilde{e}), \mathcal{E}_{pk}(\tilde{c}_i) > \ . \tag{7}$$

While the first 5 elements of the tuple are same as it is defined in the original work, we send the encrypted versions. We also send two additional piece of

information for the ease of computation on the server side: $\mathcal{E}_{pk}(\tilde{e}), \mathcal{E}_{pk}(\tilde{c_i})$. These terms actually define the user and crowd emotion per location, respectively. Note that each encryption is indistinguishable than the others as the encryption scheme is semantically secure as explained in Section 3.2.

4.3 Data Analysis

Upon receiving the tuple, the server can perform the following functions on the encrypted data. We assume that $M < N$ users sent their data.

1. The total number of people in each location:

$$\mathcal{E}_{pk}(\ell) := \mathcal{E}_{pk}(\sum_{i=1}^{M} \ell_i) = \prod_{i=1}^{M} \mathcal{E}_{pk}(\ell_i) \; . \tag{8}$$

   The resulting ℓ is in the form of integer values for each location in a concatenated form:

$$\ell = (\sum_{i=1}^{M} \ell_{i,1} || \sum_{i=1}^{M} \ell_{i,2} || \ldots || \sum_{i=1}^{M} \ell_{i,L}) \; . \tag{9}$$

   Recall that the server has ℓ in the encrypted form, thus he cannot access its content.
2. The total number of users per each emotion:

$$\mathcal{E}_{pk}(e) := \mathcal{E}_{pk}(\sum_{i=1}^{M} e_i) = \prod_{i=1}^{M} \mathcal{E}_{pk}(e_i) \; , \tag{10}$$

   where e is

$$e = (\sum_{i=1}^{M} e_{i,1} || \sum_{i=1}^{M} e_{i,2} || \ldots || \sum_{i=1}^{M} e_{i,E}) \; .$$

3. The emotion distribution per location for the users:

$$\mathcal{E}_{pk}(\tilde{e}) := \mathcal{E}_{pk}(\sum_{i=1}^{M} \tilde{e}_i) = \prod_{i=1}^{M} \mathcal{E}_{pk}(\tilde{e}_i) \; . \tag{11}$$

4. The emotion distribution per location for the crowd:

$$\mathcal{E}_{pk}(\tilde{c}) := \mathcal{E}_{pk}(\sum_{i=1}^{M} \tilde{c}_i) = \prod_{i=1}^{M} \mathcal{E}_{pk}(\tilde{c}_i) \; . \tag{12}$$

Figure 2 illustrates the main idea of computing $\tilde{e}$ with an example, ignoring the encryption. In this example, we assume that there are 2 locations and 3 emotions. Clearly, ℓ gives the number of people in each location, e gives the total number of people for each emotion. However, $\tilde{e}$, provides more detailed

$$
\begin{array}{l}
\ell_1 = (1||0) \quad \ell_2 = (1||0) \quad \ell_3 = (0||1) \\
e_1 = (1||0||0)\ e_2 = (0||0||1)\ e_3 = (0||0||1) \\
\hline
\ell = \ell_1 + \ell_2 + \ell_3 = (2||1) \\
e = e_1 + e_2 + e_3 = (1||0||2) \\
\hline
\tilde{e}_1 = \ell_1 \times e_1 = (1|0|0||0|0|0) \\
\tilde{e}_2 = \ell_2 \times e_2 = (0|0|1||0|0|0) \\
\tilde{e}_3 = \ell_3 \times e_3 = (0|0|0||0|0|1) \\
\hline
\tilde{e} = \tilde{e}_1 + \tilde{e}_2 + \tilde{e}_3 = (\ \underbrace{1|0|1}_{location\ 1} \ ||\ \underbrace{0|0|1}_{location\ 2}\ )
\end{array}
$$

Fig. 2. Illustration of the output of data analysis

information on emotions per location: there are 2 people in the first location with emotions 1 and 3, and only one person in location 2 with emotion 2. $\tilde{c}$ can be visualized in the same manner.

Recall that none of the above values can be obtained in plain text by the server as it does not have the decryption key. In order to decrypt the cipher texts, the users run a joint decryption protocol. This means that users also receive messages from the server. At the end of the decryption protocol, the server obtains all the statistical data for that specific time interval.

5 Analysis

In this section, we provide an informal discussion on our proposed method for emotion detection and give complexity analysis, which is important for the deployment of the protocol for real life.

5.1 Security

In this section, we provide an informal discussion on the security of our proposal. Recall that we assume all involved parties, the server and the users, are honest-but-curious. Our goal is to hide the individual emotion data from the server. It is clear that as long as the underlying cryptographic primitive, that is the threshold encryption scheme, is secure, the server cannot learn the content of the encrypted messages from the users. As the encryption scheme is also semantically secure, the server cannot distinguish cipher text, even if the same value is encrypted by different users. The users, on the other hand, do not obtain additional information, for example about other users' emotions, other than the aggregated data that may be broadcasted by the server after the joint decryption protocol.

Even though, our protocol is conceptually simple, and provably secure, there are a number of possibilities for the server to extract more information. These actions are not supposed to be performed under our security assumption, nevertheless we would like to address them here.

In the first type of attack, the server can present a specific encrypted message of a user as the aggregated data and ask users to decrypt it. These users can avoid

this attack by performing only one decryption per time slot. However, remember that the joint decryption protocol is performed by a subset of users and thus, the server can ask users from the remaining users to decrypt the encrypted message. This can be avoided by setting the number of users required in the threshold decryption to $N/2 + 1$ users. By this way, there will be overlapping users for the decryption so that these users can check whether there are more than one decryption per time slot or not.

If such malicious acts are expected from the server, a better approach is to use zero-knowledge proofs (ZKP) for the verification of the performed actions of the server. However, using ZKPs are costly in terms of computation and communication, and can be overwhelming for the mobile device owners.

5.2 Complexity

Our proposal is designed to have as minimum number of operations as possible on the user mobile device. We assume that the operations on the plain text data is negligible compared to the operations on the encrypted data. Therefore, in Table 1, we only provide the complexity of operations in the encrypted domain and the amount of encrypted data transmission. Note that multiplication and exponentiation are over modulo n^2, where n is a very large number.

Table 1. Complexity analysis per operation

	Server	**User**
Encryption	-	$\mathcal{O}(1)$
Decryption	-	$\mathcal{O}(1)$
Multiplication	$\mathcal{O}(M)$	-
Exponentiation	-	-
Data	$\mathcal{O}(1)$	$\mathcal{O}(1)$

As seen in Table 1, the complexity is reasonable low. Each user encrypt 5 messages if s/he reports emotion, and participates in 1 decryption per time slot. The server, on the other hand, computes the aggregated data by performing $5M$ multiplications in each time slot. Note that $M = N$ in the worst-case scenario, since there will be less people reporting their emotions per time slot. As for the data transmission, each user sends 5 encryption to the server and server broadcasts 5 cipher texts for decryption. In summary, the overall protocol is quite efficient in terms of computation and communication.

6 Discussion

In this section, we address a number of open issues in terms of privacy preserving emotion detection.

6.1 Application Scenario

The protocol described previously protects the private data of the participants, while it is still feasible for the server to analyze the data for crowd management. However, there is a strong dependency on the involvement of people in the protocol, particularly for the decryption of the encrypted values. This fact introduces two challenges:

1. **On-line processing only.** It is not reasonable to expect people to use the software after the event. Therefore, all measurements should be performed in real time. This means that the duration of collecting data has to be determined carefully. Short intervals can be overwhelming for the users in terms of processing, and long intervals cannot provide useful information.
2. **Time constraint.** Consequently, it is also essential to associate each encryption with the corresponding time-slot so that the server cannot combine encrypted messages from different time slots and ask users to decrypt for deducing more information than it should have.

In the following, we address these two challenges and also provide a direction for off-line processing.

On-line Processing with Time Constraint. Our proposal requires 2 round of communication between the server and the users: 1) the users sends their data to the server, and 2) the server runs the joint decryption protocol to obtain the aggregated data. As noted before, a user can send emotion data at any time during the event. However, it is essential to have the state of the crowd for a given time interval. Therefore, we assume that the event is divided into certain time slots and for each time interval a different generator for the encryption is used. Using different generators for each time slot guarantees that only the data provided for that time slot are aggregated by the server.

Off-line Processing. In a server-client model, it is not possible to process the collected data without the help of the users. Unfortunately, in our application scenario, users go off-line after the event. To be able to process data after the event, a third semi-trusted entity (STE) is required. Assuming that such an entity exists, we have two options to be able to process data off-line:

1. The users send their partial keys to the STE at the end of the event so that for any computations can be performed with its help afterwards.
2. The protocol can be changed in such a way that users submit their data encrypted using the public key of the STE. In that approach, the users are not required to participate in any computations. The server and the STE, on the other hand, run a similar protocol to obtain the aggregated emotional data.

In either case above, there is a strong assumption that the server and the STE are not colluding, that is they act according to the protocol and they do not co-operate to reveal the privacy-sensitive data of the users.

6.2 Location Privacy

Even though our protocol is secure and privacy-preserving, it only allows the server to deduce statistical data. However, an essential component in crowd management is emotion maps, where users are monitored in space and time. This is very important especially for emergency procedures and security countermeasures. Unfortunately, determining the exact location of every single individual along with their emotion state for crowd management creates a trade-off in terms of privacy protection. Although there are ways to hide the location of users in a crowd, for example using Mixnets [19], it becomes impossible to create emotion maps, which require the exact location of the users. Therefore, we envision that it is not possible at the moment to hide both the emotion state and the location of the users for crowd management.

7 Conclusions

Emotion detection for crowd management presents itself as a powerful tool to understand the state of the people. Based on the information gathered in real time, authorities can have a clear idea about the event and react fast to sudden changes in the crowd. In an ideal case, emotion detection should be transparent to the people. However, due to technological challenges, it is not feasible to collect reliable emotion data without user's participation. Therefore, self-reporting tools have been developed for creating emotion maps with the help of people participating in an event. Unfortunately, without any privacy protection mechanism, it is not desirable for people to use such self-reporting tools. In this paper, we present a way to protect the privacy-sensitive data, in this case emotions, from the server, which would like to process for crowd management. Our proposal is to use cryptography to hide the private data and enable the server only obtain aggregated data. We achieve this goal by using cryptographic tools such as homomorphic encryption and increase the efficiency of the system by employing data packing technique. The resulting protocol is quite efficient to be used in real systems as it is shown in the complexity analysis.

References

1. Levenson, R.W.: The intrapersonal functions of emotion. Cognition & Emotion 13(5), 481–504 (1999)
2. Arnold, M.B.: Emotion and Personality: Psychological Aspects, vol. 1. Colombia University Press, New York (1960)
3. Li, J., Erkin, Z., de Ridder, H., Vermeeren, A.: A field study on real-time self-reported emotions in crowds. In: Proceedings of ICT OPEN 2013, Eindhoven, The Netherlands (2013)
4. Bianchi, T., Piva, A., Barni, M.: Composite signal representation for fast and storage-efficient processing of encrypted signals. IEEE Transactions on Signal Processing (2009)

5. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: SIGMOD 2000: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, vol. 29(2), pp. 439–450. ACM Press, New York (2000)
6. Lindell, Y., Pinkas, B.: Privacy preserving data mining. Journal of Cryptology, pp. 36–54 (2000)
7. Lagendijk, R.L., Erkin, Z., Barni, M.: Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation. IEEE Signal Process. Mag. 30(1), 82–105 (2013)
8. Melchor, C.A., Fau, S., Fontaine, C., Gogniat, G., Sirdey, R.: Recent advances in homomorphic encryption: A possible future for signal processing in the encrypted domain. IEEE Signal Process. Mag. 30(1), 108–117 (2013)
9. Goldreich, O.: Foundations of Cryptography II. Cambridge University Press (2004)
10. Hirt, M.: Receipt-free k-out-of-l voting based on elgamal encryption. In: Chaum, D., Jakobsson, M., Rivest, R.L., Ryan, P.Y.A., Benaloh, J., Kutylowski, M., Adida, B. (eds.) Towards Trustworthy Elections. LNCS, vol. 6000, pp. 64–82. Springer, Heidelberg (2010)
11. Barni, M., Failla, P., Lazzeretti, R., Sadeghi, A.R., Schneider, T.: Privacy-preserving ecg classification with branching programs and neural networks. IEEE Transactions on Information Forensics and Security 6(2), 452–468 (2011)
12. Erkin, Z., Franz, M., Guajardo, J., Katzenbeisser, S., Lagendijk, I., Toft, T.: Privacy-preserving face recognition. In: Goldberg, I., Atallah, M.J. (eds.) PETS 2009. LNCS, vol. 5672, pp. 235–253. Springer, Heidelberg (2009)
13. Erkin, Z., Veugen, T., Toft, T., Lagendijk, R.L.: Generating private recommendations efficiently using homomorphic encryption and data packing. IEEE Transactions on Information Forensics and Security 7(3), 1053–1066 (2012)
14. Kononchuk, D., Erkin, Z., van der Lubbe, J.C.A., Lagendijk, R.L.: Privacy-preserving user data oriented services for groups with dynamic participation. In: Crampton, J., Jajodia, S., Mayes, K. (eds.) ESORICS 2013. LNCS, vol. 8134, pp. 418–442. Springer, Heidelberg (2013)
15. Jagannathan, G., Wright, R.N.: Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In: KDD, pp. 593–599 (2005)
16. Beye, M., Erkin, Z., Lagendijk, R.L.: Efficient privacy preserving k-means clustering in a three-party setting. In: IEEE Workshop on Information Forensics and Security, pp. 1–6 (2011)
17. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
18. Damgård, I.B., Nielsen, J.B.: Universally composable efficient multiparty computation from threshold homomorphic encryption. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 247–264. Springer, Heidelberg (2003)
19. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM 24(2), 84–88 (1981)