Practical Privacy-Preserving Protocols for Criminal Investigations

Florian Kerschbaum, Andreas Schaad, Debmalya Biswas
SAP Research
Karlsruhe, Germany
Email: firstname.lastname@sap.com

***Abstract*—Social Network Analysis (SNA) is now a commonly used tool in criminal investigations, but evidence gathering and analysis is often restricted by data privacy laws. We consider the case where multiple investigators want to collaborate but do not yet have sufficient evidence that justifies a plaintext data exchange. We propose a practical solution that allows an investigator to expand his current view without actually exchanging sensitive private information. The investigator gets a partially anonymized view of the entire social network, while preserving his known view.**

I. Introduction

In federated states or organization of states, such as the European Union or the United States, a common approach to organized crime is necessary. For this purpose, federal law enforcement agencies, such as Europol or the FBI, have been established. Nevertheless, data privacy laws or simply data governance concerns restrict supplying institutions from sharing their data, unless there is a hard corroborating evidence on a case and subject under investigation. In particular, in the European Union [1], data privacy is regarded as a high social and political value and the dilemma on how to generate evidence without violating privacy laws is evident.

A common tool for the criminal investigator is social network analysis. It graphically depicts the suspects and their connections to other people or artifacts, such as telephone numbers or bank accounts, and allows the computation of certain metrics. Not all the facts composing the entire picture of a case may be known to one investigator. In particular, in pan-European organized crime, local police forces may only be aware of a partial view of the picture, as the case studies in the framework of the R4eGov project suggest [2].

This necessitates data exchange between the institutions, but European data privacy laws prohibits data exchange without reasonable cause and in excessive amounts. Therefore, we propose a solution where a local investigator can track his subjects or visualize the entire network, but without revealing sensitive or private details. This allows the investigator to still use SNA and profit from its achievements without breaking individual privacy rights or guidelines of other institutions.

Privacy-Preserving SNA has been suggested in the literature before, but we found the solutions to be insufficient for the requirements of our scenario. In [7] a fully anonymized version of the social network is computed. This does not allow the investigator to track his suspect anymore and he cannot gain additional information or collect evidence about him.

We propose the protocol "Compute Entire Network" that computes the entire social network from distributed sources. The protocol does not reveal any personally identifying information. However, the protocol does allow the investigators to keep track of their own inputs, as well as the sources of inputs shared by other investigators. Given this, the investigators can perform analyses on their subjects, requesting additional information from the other investigators as deemed necessary by their analyses.

The remainder of the paper is structured as follows: The next Section reviews related work. This is followed by description of the protocol in Section III. This section is divided into building blocks, protocol description and analysis. The final section presents the conclusions of this work.

II. Related Works

SNA has been used for criminal investigations for a long time [4], [5], [6]. Recent research [6] suggests using graphical tools and investigates the impact of SNA. We can conclude that SNA is a widely accepted tool in criminal investigations.

Privacy-Preserving SNA was first proposed by [7]. They compute an anonymized graph of the social network, such that no one should be able to track their position in the graph. They allow for certain modifications of the correctness of the anonymized graph in order to prevent tracking of one's position. E.g. they may bound the number of incoming connections or apply similar restrictions. While this provides strong privacy guarantees it does not match the requirements of our scenario. An investigator intends to gather *additional* information to his present view of the social network. It is therefore unacceptable to anonymize his view, but the goal is to augment it with additional information about the entire network.

In a previous paper, Kerschbaum and Schaad [14] had shown how certain association metrics, such as betweeness and closeness, can be computed without revealing personally identifying information and without revealing the entire social network. In this work, we show how to compute the entire social network in an anonymous fashion.

Privacy-Preserving SNA can be seen as a special case of secure multi-party computation (SMC) which can solve any distributed function privately. SMC has been suggested in [8] for the two-party case. The first multi-party solution has been suggested in [9] for the computational setting and in [12]

978-1-4244-4173-0/09/$25.00 ©2009 IEEE

for the information-theoretic setting. Efficient construction has been identified for different secret sharing schemes [10], [11]. Nevertheless, as in [7] stated, a straight-forward application of these techniques would result in an unpractical protocol.

III. Visualizing and Analyzing the Entire Network

A. Building Blocks

A social network can be represented as a graph $G = (V, E)$ consisting of a set V of (connected) vertices and a set E of edges between the vertices. Each vertex $v \in V$ represents a person or other artifact, such as a telephone number or company, and is associated with some personally identifying, unique information (such as the name or the telephone number). An edge $e \in E$ connects a pair of vertices v_1 and v_2 and is in correspondence with our application undirected.

The graph G may be distributed among n data sources, such that each party X_i holds a part of the graph $G_i = (V_i, E_i)$. The combination of the data sources result in the entire graph

$$V = \bigcup_{i=1}^{n} V_i \quad E = \bigcup_{i=1}^{n} E_i$$

The parts may be overlapping, such that the intersection may not be the empty set.

$$\bigcap_{i=1}^{n} V_i \neq \emptyset \quad \bigcap_{i=1}^{n} E_i \neq \emptyset$$

In the protocols, we use a commutative encryption scheme. In a commutative encryption scheme the order of encryption (with different keys) does not matter. We denote the encryption with Alice's key as $E_A()$ and with Bob's key as $E_B()$. Then, in a commutative encryption scheme, it holds that

$$E_A(E_B(x)) = E_B(E_A(x))$$

As we compare ciphertext, the encryption system cannot be semantically secure, but may be secret key. A candidate encryption system with all these properties is Pohlig-Hellman encryption [13].

B. Protocol to Compute the Entire Network

The goal of the protocol is to compute the entire graph G, but without revealing the identifiers of the vertices. The algorithm must maintain the partial view of the graph, such that each investigator can track his information in the assembled graph. The algorithm may reveal the source of each vertex or edge, the size of each part of the graph and the overlapping sets, such as the vertices or edges already known to the local investigator.

The source information can be used by the investigator to selectively request additional information from other institutions that can enhance the case. Therefore, this protocol can be used to combine data sources selectively, such that personally identifying information is only revealed when there is a reasonable cause.

We refer to the collaborating investigators as participants in the protocol. Before the protocol begins, each participant X_i holds a key for encryption $E_i()$ in the commutative, secret key encryption scheme and another key $E'_i()$. $E'_i()$ should be in the same (commutative) encryption scheme.

The "Compute Entire Network" protocol proceeds as follows:

1) Each participant X_i prepares for each of his edges $e_j = (v, v') \in E_i$ a tuple
$$\langle i, E'_i(v), E'_i(v'), E_i(v), E_i(v') \rangle$$
2) Each participant X_i sends his tuples to the next participant X_{i+1}. Participant X_n sends to participant X_1.
3) Each participant X_{i+1} encrypts the last two fields of each received tuple.
$$\langle i, E'_i(v), E'_i(v'), E_{i+1}(E_i(v)), E_{i+1}(E_i(v')) \rangle$$
4) All participants repeat steps 2 and 3 n times.
5) Each participant X_i keeps a copy of the received tuples and forwards them to participant X_{i+1}. They repeat this step $n - 1$ times.

After the "Compute Entire Network" protocol, each participant X_i holds a set T of tuples representing all edges E plus some potential duplicates. Let $E^n(v)$ denote the encryption with all keys $E_i()$ $i = 1, \ldots, n$. Note that due to the commutative encryption the order of encryption does not matter. A tuple after the protocol therefore looks like this:

$$\langle i, E'_i(v), E'_i(v'), E^n(v), E^n(v') \rangle$$

The local investigator can now

- *remove duplicates*, since duplicates have the same two last fields, resulting in the set E.
- *build an anonymized graph* G from E with the pseudonyms $E^n(v)$ for v and e.g. visualize it.
- *track his input* by identifying the tuples with (his) i in the first field and deanonymizing those pseudonyms by decrypting the second and third field.

The investigator ends up with an anonymous view of the entire social network with a partial non-anonymous view of his present knowledge. He can then perform SNA (e.g. metrics computation) on his suspects without having learnt personally identifiable from his collaborators. He may request additional information from some collaborators, since he can now target individual collaborators that are likely to have useful information. This protects the privacy of suspects, since it reduces the overall data exchange to reasonable amounts that are likely to contribute positively to the case. The information about likely innocent people remains protected.

Note that the protocol actually allows all the participants to compute their own local anonymized graphs of the entire network *simultaneously*. Thus, the protocol can be regularly scheduled as a batch process by all the participants to maintain their own local copies of G. To further improve efficiency, the protocol can be be run in a change-oriented fashion. That is, after an initial run of the protocol, for any subsequent runs, it is sufficient if each participant only processes any newly added edges (after the last run) at its site. Of course, this holds only as long as the set of participants does not change.

C. Analysis

The protocol operates in the semi-honest model [3]. We strongly argue that this is appropriate for our application, since we are concerned with cooperating police organizations and officers whose main concern is protecting the privacy of the suspects and keeping practical data governance. That is, the organizations are inclined to follow the protocol, since their objective is not only the outcome of the collaborations, but also the process of data privacy protection. Since interest in collaboration can be assumed, the organizations could simply exchange data by bypassing the protocol, if they were not interested in data protection.

The main security goal of the protocol is to not leak identifiable information as in the labels of the vertices. We state the following theorem:

Theorem 1. *The identifier of any vertex v known only to participants $X_i \in \mathbb{X} | v \in V_i$ is inaccessible to any participant $X_j \notin \mathbb{X}$.*

Proof Sketch: The identifier of v only leaves the systems of a participant X_i encrypted under $E_i()$. Therefore, the security of the identifier is based on the security of the encryption. ■

Privacy Model Comparison: The paper by Frikken and Golle [7] presents a privacy definition for SNA. Our paper takes a different stance on privacy than their paper. Their main concern is that an attacker is able to track his position in the network and thereby possibly identify some other vertices in the network. They assume that one may have partial information about the network and exploit this knowledge to gain additional information. On the contrary, we assume that one completely divulges one's partial information. The threat is that an attacker might learn additional identifying information accidentally revealed during the protocol. In contrast to [7] where no quantification of the previous knowledge was given, we prove that absolutely no private (that is, identifying) information is being leaked. This implies that the attack with previous knowledge does not apply in our framework, since the attacker would simply reveal that information and then the revealed information would be part of the result of the computation. Obviously, the result is not protected by our (or for that matter any) protocol.

Computational Complexity Comparison: The paper [14] by Kerschbaum and Schaad gives a protocol to compute the SNA metrics (betweeness and closeness) in the same setting as the current paper. Let the number of participants be p and $|V|$ be the number of vertices in the entire network G. Given this, the computational complexity of their protocol is $O(p|V|^3)$. In contrast, our protocol has a computational complexity of $O(p^2|V|)$. As the number of vertices is expected to be much more than the number of participants, the "Compute Entire Network" protocol is much more suitable for practical implementations.

IV. Conclusion

Social Network Analysis is becoming an important tool for investigators, but all the necessary information is often distributed over a number of sites. Privacy legislation and data governance concerns prohibit freely sharing the information. We have presented a practical protocol that allows selective disclosure of information for Social Network Analysis. An investigator can use the protocol to compute the entire network in an anonymized fashion, while preserving his local information. The protocol has low computational complexity and allows the investigators to perform many analyses. It preserves the privacy of personally identifiable information of suspects and limits the data exchange to prevent misuse.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful suggestions. The developments presented in this paper were partly funded by the European Commission through the ICT program under Framework 7 grant 213531 to the *SecureSCM* project.

References

[1] *Directive 95-46-EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data*, Available at http://ec.europa.eu/justice_home/fsj/privacy, 1995.

[2] T. V. Cangh and A. Boujraf, *The Eurojust-Europol Case Study*, Available at http://www.r4egov.eu, 2007.

[3] O. Goldreich, *Secure Multi-party Computation*, Available at *http://www.wisdom.weizmann.ac.il/˜oded/pp.html*, 2002.

[4] W. R. Harper and D. H. Harris, *The application of link analysis to police intelligence*, Human Factors, 17(2): 157-164, 1975.

[5] M. K. Sparrow, *The application of network analysis to criminal intelligence: An assessment of the prospects*, Social Networks, 13: 251-274, 1991.

[6] J. Xu and H. Chen, *Criminal Network Analysis and Visualization*, Communications of the ACM, 48(6): 100-107, 2005.

[7] K. Frikken and P. Golle, *Private Social Network Analysis: How to Assemble Pieces of a Graph Privately*, in proceedings of the 5th ACM Workshop on Privacy in the Electronic Society (WPES), pp. 89-98, 2006.

[8] A. C. Yao, *Protocols for secure computations*, in proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 160-164, 1984.

[9] O. Goldreich, S. Micali and A. Wigderson, *How to play any mental game*, in proceedings of the 19th Annual ACM conference on Theory of Computing (TOC), pp. 218-229, 1987.

[10] R. Cramer, I. Damgard and U. Maurer, *General Secure Multi-party Computation from any Linear Secret-Sharing Scheme*, in Advances in Cryptology - EUROCRYPT, Springer LNCS vol. 1807, pp. 316-334, 2000.

[11] R. Cramer, I. Damgard and J. Nielsen, *Multiparty Computation from Threshold Homomorphic Encryption*, in Advances in Cryptology - EUROCRYPT, Springer LNCS vol. 2045, pp. 280-300, 2001.

[12] M. Ben-Or and A. Wigderson, *Completeness Theorems for Non-cryptographic Fault-tolerant Distributed Computation*, in proceedings of the 20th Annual ACM symposium on Theory of computing (TOC), pp. 1-10, 1988.

[13] S. Pohlig and M. Hellman, *An improved algorithm for computing logarithms over GF(p) and its cryptographic significance*, IEEE Transactions on Information Theory, 24(1): 106-110, 1978.

[14] F. Kerschbaum and A. Schaad, *Privacy-preserving Social Network Analysis for Criminal Investigations*, in proceedings of the 7th ACM Workshop on Privacy in the Electronic Society (WPES), pp. 9-14, 2008.