This is a post-print of the paper:

Amna Qureshi, Helena Rifà-Pous, David Megías, “Enabling Collaborative Privacy in User-Generated Emergency Reports”, In: *Proceedings of Privacy in Statistical Databases* (PSD 2016), Lecture Notes in Computer Science (LNCS), *Springer*, pages 255-271. ISBN: 978-3-319-45381-1.

Enabling Collaborative Privacy in User-Generated Emergency Reports

Amna Qureshi, Helena Rifà-Pous, and David Megías

Internet Interdisciplinary Institute (IN3),
Estudis d'Informàtica, Multimèdia i Telecomunicació,
Universitat Oberta de Catalunya
{aqureshi,hrifa,dmegias}@uoc.edu

Abstract. *Witnesses are of the utmost importance in emergency systems since they can trigger timely location-based status alerts. However, their collaboration with the authorities can get impaired for the fear of the people of being involved with someone, some place, or even with the same cause of the emergency. Anonymous reporting solutions can encourage the witnesses, but they also pose a threat of system collapse if the authority receives many fake reports. In this paper, we propose an emergency reporting system that ensures the anonymity of honest witnesses but is able to disclose the identity and punish the malicious ones. The system is designed over an online social network that facilitates the indistinguishability of the witness among a group of users. We use a game-theoretic approach based on the co-privacy (co-utility) principles to encourage the users of the network to participate in the protocol. We also use discernible ring signatures to provide the property of conditional anonymity. In addition, the system is designed to provide rewards to a witness and his/her group members in a privacy-preserving manner.*

Keywords: Co-utility; Co-privacy; Revocation; Emergency management; Online social network

1 Introduction

An emergency is an unanticipated situation that may lead to the loss of lives (road accidents) or properties (collapsed buildings), to the harm of the physical integrity of human life (robberies), or to the damage of properties (fire) or the environment (wildfire). In such situations, the traditional way to report the incident and ask for help is to call an emergency service that allows the caller to contact local emergency operators for assistance. On average, it takes an emergency operator at least two to three minutes to collect the necessary information in order to respond to the caller [19]. At the time of emergencies, the loss of a few seconds can mean the difference between life and death. Therefore, emergency rescue systems should be fast and efficient in order to ensure a timely response to emergency situations. The recent advances in mobile communication and mobile information systems have made a significant impact on the development of emergency response systems. These systems or platforms allow citizens to communicate location-based emergency information to

emergency responders, who, in return, respond quickly to the situation. For example, Alpify [1] is a mobile application that uses a mobile phone's global positioning system (GPS) functionality so that citizens can locate, document and report emergencies (fire or road accidents) to 112/911 emergency services, quickly and effectively.

A main issue faced by the emergency service providers is fake or false emergency calling. A fake call is when a person deliberately calls the emergency service to falsely inform them that there is an emergency when in fact there is not, or when somebody contacts the emergency services for reasons not related to any emergency, or when the situation is not considered an emergency by the emergency services but it is for the caller (e.g. car keys are lost) [4]. The statistics in a recent study show that the emergency services across the United Kingdom (U.K.) receive over 5 million fake calls per year [13]. These fake calls are a misuse of the system and divert emergency services away from people who may be in life-threatening situations and need urgent help. Also, a fake call is an expensive problem because emergency service providers need to multiply their resources to assure that they are not being overloaded by false calls and, therefore, may not be able to respond to true emergencies. Thus, there is a need to figure out mechanisms to prevent people from making fake calls to emergency services so that the true emergencies that require immediate assistance always get a top priority. The communities across Europe and U.K. are trying to face false emergency calls by instituting ordinances and/or special measures by police departments [4]. For example, alternative three-digit numbers for non-emergency calls have been introduced in the recent years in the U.K.

Many systems for emergency management have been envisioned [10, 16, 14]. In [10], the authors proposed the use of social media in a collaborative effort to inform people about crime events that are not reported to the police. Their wiki website (WikiCrimes) allows users to register criminal events online in a specific geographic location represented by a map; hence, other users can use this information and keep track of the locations to make decisions. However, a limitation of this approach is that each crime registered in WikiCrimes requires confirmation from at least one another person (besides the reporting user) in order to be registered as a true event. In addition, WikiCrimes requires users to log into the system by means of a valid email address, and then tracking the reporting user is possible. In [16], Okolloh proposed Ushahidi, a map-based mash-up tool to visualize crowd-sourced information by allowing citizens to submit information by sending a text message (SMS), a tweet, or an email; or by inputting the information on a form available on Ushahidi's web portal. Though Ushahidi has proven to be a successful online platform for spreading awareness of critical situations worldwide, it faces some limitations, such as the requirement that the reports from incidents have to go through an approval process conducted by a group of volunteers, who publish an online interactive map of reports after successful verification. In [14], a location-aware Smart Phone Emergency and Accident Reporting System (SPEARS) is proposed that allows users of an online social network (Facebook or Twitter) to quickly report emergency situations to the agencies responsible for handling emergency situations. The agencies store their locations via SPEARS, so that users involved in an emergency situation can retrieve

the shortest path from the point of alert to the point of care. Though it is an efficient tool for emergency reporting on Android smartphones, it has a few limitations: (1) it can only be used in Thai language, which does not help much for most foreigners living in Thailand, and (2) users must be identified by phone numbers and names before reporting an emergency.

All the systems referred above pose at least one of the following drawbacks: (1) they allow user re-identification, and (2) they require manual sorting of legitimate/fake information. The re-identification of a user by means of his/her email address, phone number or location is a relevant issue, since in most emergency situations, the witnesses are reluctant to report an emergency because they do not want to be identified or reveal their specific location for personal reasons, or because they fear the possibility of being considered as suspects of a crime. Anonymity is, thus, a desired property from the witness point of view. However, total user anonymity is not feasible since it would encourage fake emergencies, which could make the authority collapse. Therefore, the challenge lies in providing anonymity to true reports, but which could be revoked in case of a false emergency report.

Contribution and plan of this paper: The contribution of this paper is to introduce a system for the notification of location-based emergency-related information so that the information is managed by an authorized entity that takes appropriate action. Our proposal stems from a game-theoretic design inspired by the co-privacy (co-utility) approach [5, 6], which leads to a mechanism of rewards and punishments to encourage legitimate information and discourage false reporting. Unlike existing emergency management systems in which witnesses are required to reveal their identities or personal information to the emergency service provider, our system protects the witness's identity by using the concept of groups, in such a way that the witness is indistinguishable from other members of the group. Here, we propose to use an online social network, whose scope is to facilitate the social interaction among an interconnected trusted network of people, for creating dynamic and location-based user groups. Anonymity is considered in the system by means of ring signatures, i.e. the emergency reports are linked to the groups instead of individual users (witnesses). However, to avoid complete anonymity in our scheme, an accountability property is provided in the sense that a malicious witness who sends false reports can eventually be identified by the collaborative effort of other members of the group. Furthermore, a source routing protocol (similar to onion routing [17]) is used to provide anonymous communication with the authority. The group members would help to run the appropriate protocol for the system to fulfil the requirements.

The rest of this paper is organized as follows. In Section 2, we overview the functionality of our system. In Section 3, the reward and punishment model of the proposal based on game theory is detailed. Section 4 presents the protocol for sending and managing anonymous emergency reports. In Section 5, we discuss the security and privacy aspects of the protocol. Finally, Section 6 concludes the paper.

2 Overview of the System

This section describes the architecture of the system proposed for the notification of location-based emergency-related information to a so-called Emergency Management System (EMS) that takes appropriate action to solve the emergency.

A. Requirements of the system: The design requirements of the system are as follows: **(1)** The system must be efficient to minimize the time taken by the emergency responders to reach the location of the emergency. **(2)** The amount of fake reports that are considered by the EMS needs to be limited since the management of a false emergency leads to a waste of the resources of the EMS. **(3)** The system must provide privacy guarantees so that the identities of the users reporting the emergencies remain hidden to everyone, i.e. to the EMS, the OSN and the users of the network. **(4)** The exact location of the incident must be reported to the EMS so that it can immediately respond to the emergency. **(5)** The users of the system are organized in groups, which are dynamically formed by the witness of an emergency. Group members must be active users (online contacts), who lie within the vicinity, i.e. within a pre-defined distance of the witness. **(6)** When the awardees redeem their rewards at City Council, it should not be able to link the recipient with any reward assigned previously.

B. Design assumptions: In our proposed system, the dynamic and location-based user groups are created by assuming users to be registered members of a popular OSN, Facebook [9]. There are mainly two reasons for selecting Facebook as a choice for the OSN: (1) it provides its data to external applications via application programming interfaces (APIs), and (2) it does not require an authorization before using an API.

In the following, the security and general assumptions related to the design of the emergency reporting system are defined: **(1)** Each user is a registered member of Facebook. Users can log in via a Facebook account to access and use the emergency reporting system on their smartphones. **(2)** A public key infrastructure (PKI) is considered for providing cryptographic keys in such a way that each entity of the system has a public and a private key. **(3)** A group created by a witness can contain up to $n \geq 3$ users. **(4)** In case of a false emergency report, a threshold of t users of the reporting group will be able to disclose the identity of the witness. This t is set to 60% of n. **(5)** The public keys and the parameters of the ring signature (of each group member) and the public key of the EMS are publicly available. (**6)** The threshold discernible ring signatures (TDS) [12] provides unforgeability and signer anonymity (details of TDS can be found in Appendix B). **(7)** The system proposes to leverage GPS and signal triangulation technologies to automatically sense device location. Triangulation is used only if a GPS signal is unavailable. **(8)** The system provides three user status modes: online (available or busy), idle (away) and offline. In online mode, the actual location is available to the users' friends, showing a person icon, his/her location coordinates and description of a distance on their map-based screens, whereas, in idle and offline modes, the last recorded distance interval of the user along with his/her last online visibility status are provided.

C. System entities: Fig. 1 illustrates the model of the proposed emergency reporting system that contains the following basic entities: **(1) The witness:** The user who witnesses an event and reports it. This user wants to safeguard his/her identity. **(2) The social group:** A group in which the witness is a member. **(3) The system manager:** A service provider who is responsible for executing the emergency reporting system via a Facebook API. It also manages the registration of the users and imports a list of users' friends from Facebook (who are already the members of the reporting system). Additionally, the system manager uses location information to calculate the distance between the users and display it on the Google Map along with a person icon. **(4) EMS:** An entity that receives and manages the emergency reports. On receiving the report, the EMS forwards it to emergency entities for validation. **(5) The reporter:** A friend of the witness (both are members of the same group). The reporter helps the witness to send an emergency report to the EMS. This user can be identified by the EMS. **(6) The intermediate hops**: The users (members of the same group) that serve as report forwarding agents. **(7) The City Council (CC):** A trusted entity from which the witness, the reporter and the group members can redeem their rewards in form of vouchers, one-time discount coupons or tax payments. Also, CC issues punishment to the witness for false reporting. **(8) The emergency entities (EE):** Entities such as police stations, hospitals, rescue units and fire stations. **(9) The Certification Authority (CA):** A trusted entity that has pre-generated key pairs and issues a key pair upon successful authentication. It is an offline process and thus does not affect the performance of the system.

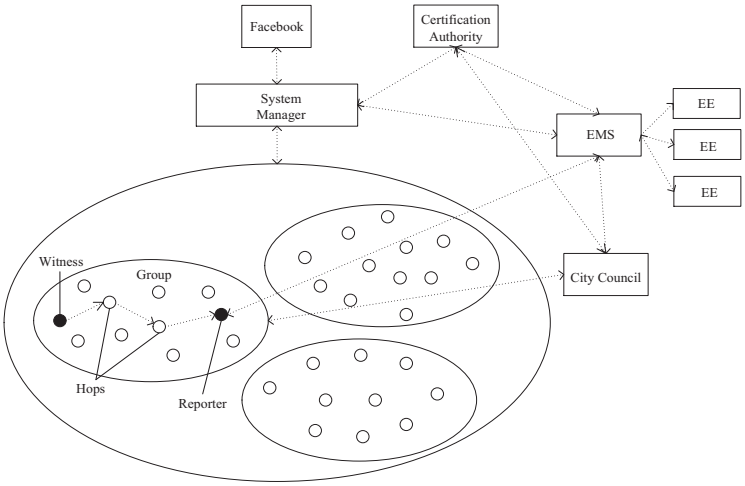

Fig. 1. Overview of the system

It can be seen, in Fig. 1, that the interaction between the witness and the EMS is carried out through multiple intermediary hops and a member of the group (i.e. the reporter), who assumes the responsibility of submitting the witness's report.

3 Co-utility Model for the Proposed Solution

The proposed emergency reporting system uses a "reward and punishment" mechanism to reward legitimate reports and punish fake ones. We use a co-utility model based on game theory (see Appendix A) to examine the implications of the witness and the members of his/her social group. We assume that users are interested in two aspects: (1) obtaining rewards, and (2) keeping their anonymity. The co-utility model presented below considers these two aspects. We borrow from [8] the following definition of co-utility:

Definition 1 (Co-utility). *Let Π be a game with self-interested, rational players $P^1, \cdots, P^N$, with $N > 1$. The game Π is said to be* co-utile *with respect to the vector $U = (u_1, \cdots, u_N)$ of utility functions if there exist at least three players P^i, P^j and P^k having strategies s^i, s^j and s^k, such that: (i) s^i involves P^i expecting co-operation from P^j and P^k; (ii) s^j involves P^j co-operating with P^i and P^k; (iii) s^k involves P^k co-operating with P^i and P^j; and (iv) (s^i, s^j, s^k) is an equilibrium for P^i, P^j and P^k in terms of u_i, u_j and u_k, respectively. In other words, there is co-utility between P^i, P^j and P^k, for some $1 \leq i, j, k \leq N$ with $i \neq j \neq k$, if the best strategy for P^i involves expecting co-operation from P^j and P^k, the best strategy for P^j is to co-operate with P^i, and the best strategy for P^k is to co-operate with P^i and P^j.*

If the equilibrium in Definition 1 is a Nash equilibrium, we have *Nash co-utility*. If the utility functions U in Definition 1 only considers privacy, co-utility becomes the co-privacy notion introduced in [5, 6]; if utilities only consider security, we could speak of co-security; if they only consider functionality, co-utility becomes co-functionality. We can use these definitions to obtain a game-theoretic model for the emergency reporting protocol with the following notations: (1) P^i is the witness of the emergency or wants to attack the system; (2) P^j is a hop (another member of the group) contacted by the witness to forward the emergency report to the reporter. For simplicity, we present the model with only one hop, but the it can be easily extended to multiple hops; and (3) P^k is a reporter who submits the emergency report (received from the witness through P^j) to the EMS. The possible strategies for player P^i, P^j and P^k are shown in Table 1. The utility model for the game is the following:

- $-c$: Negative payoff for forwarding/submitting an emergency report.
- d_i: Payoff (reward) that P^i obtains from the EMS for reporting a true emergency.
- d_j ($d_j < d_i$): Payoff (reward) that P^j obtains from the EMS for assisting in the submission of a true emergency report.
- d_k ($d_k > d_i > d_j$): Payoff (reward) that P^k obtains from the EMS for submitting a true emergency report.
- $-v_i$: Negative payoff (punishment) that P^i obtains from the EMS for reporting a false emergency report.
- $-v_j$ ($v_j < v_i$): Negative payoff (punishment) that P^j, P^k and all the other group members obtain from the EMS for forwarding a false emergency report.
- r_j ($r_j < r_k$): Reward that P^j and the remaining group members obtain after revealing the source of a false emergency report to the EMS.

Table 1. Possible Strategies of Players

No.	Possible Strategies of Players		
	P^i	P^j	P^k
1.	S_0^{ii}: Reports a true emergency directly to the EMS.	W_0^{jk}: Forwards the emergency report to P^k.	T_0^k: Submits the emergency report to the EMS.
2.	S_1^{ii}: Reports a false emergency directly to the EMS.	W_1^j: Ignores the emergency report.	T_1^k: Ignores the emergency report.
3.	S_0^{ij}: Forwards a true emergency report to P^j.	W_2^{jl}: Deviates from its pre-defined routing path and does not deliver the report to P^k.	T_2^k: Joins other players that may include P^j to reveal the source P^i to the EMS after being accused of sending a false emergency report.
4.	S_1^{ij}: Forwards a false emergency report to P^j.	W_3^{jk}: Joins other players that may include P^k to reveal the source P^i to the EMS after being accused of sending a false emergency report.	
5.	S_0^{ik}: Forwards a true emergency report to P^k.		
6.	S_1^{ik}: Forwards a false emergency report to P^k.		
7.	S_2^i: Ignores a true emergency and does not report it.		

- r_k: Reward that P^k obtains after revealing the source of a false emergency report to the EMS.
- $-w_j$: negative payoff that P^j incurs from not following the fixed routing path.
- $-w_k$: negative payoff that P^k obtains due to a loss of privacy w.r.t the EMS.
- $-z_k$: negative payoff that P^k incurs due to a false accusation by the EMS. Typically, $z_k = 0$ if the protocol guarantees that P^k is not the creator of the report.

The values of the utility functions for P^i, P^j and P^k are presented in Table 2.

We can have two possibilities in this situation: P^i either witnesses a true emergency or generates a fake emergency report. In the former case, the witness P^i can decide either to ignore the emergency and obtain a neutral (0) payoff, or to report the emergency an obtain a maximum payoff $d_i - c > 0$ if he/she decides to use the hop P^j and the reporter P^k. In this case, the maximum payoff that P^j can obtain from the EMS is $d_j - c > 0$ for relaying the emergency report from P^i to P^k. Also, P^k obtains a maximum payoff $d_k - c - w_k > 0$ by reporting the emergency to the EMS. The Nash equilibrium $(S_0^{ij}, W_0^{jk}, T_0^k)$ for P^i is to report the emergency using P^j and P^k, for P^j is to forward the report to P^k and for P^k to submit the report to the EMS. In the latter case, if P^i reports a fake report either directly or through P^j and P^k, group members will obtain positive payoff by revealing the source P^i of the message, who would then be punished by getting a negative payoff $-c - v_i$. P^j will obtain a smaller payoff $r_j - 2c - v_j > 0$, P^k will obtain a major payoff $r_k - 2c - v_j > 0$, and the remaining group members of the group a smaller payoff $r_j - c - v_j > 0$. Hence, there is no profit in generating a fake emergency report, unless some (small) probability may exist that a fake emergency report is taken to be valid by the EMS. In any case, the risk of receiving a punishment should be enough to discourage users from generating false emergency reports.

Table 2. Utility functions of P^i, P^j and P^k

Players' Strategies	**Utilities**		
	$\boldsymbol{u_i}$	$\boldsymbol{u_j}$	$\boldsymbol{u_k}$
S_0^{ii}, $\emptyset$, $\emptyset$	$d_i - c - w_k$ (1)	$\times$	$\times$
S_1^{ii}, $\emptyset$, $\emptyset$	$-c - v_i - w_k < 0$	$\times$	$\times$
S_2^{ii}, $\emptyset$, $\emptyset$	0	$\times$	$\times$
S_0^{ij}, W_0^{jk}, T_0^k	$d_i - c$ (2)	$d_j - c$ (3)	$d_k - c - w_k^{(4)}$
S_0^{ij}, W_0^{jk}, T_1^k	$-c < 0$	$-c < 0$	0
S_0^{ij}, W_1^j, $\emptyset$	$-c < 0$	0	$\times$
S_0^{ij}, W_2^{jl}, $\emptyset$	$-c < 0$	$-c - w_j < 0$	$\times$
S_0^{ik}, $\emptyset$, T_0^k	$d_i - c$ (2)	$\times$	$d_k - c - w_k^{(4)}$
S_0^{ik}, $\emptyset$, T_1^k	$-c < 0$	$\times$	0
S_1^{ik}, $\emptyset$, T_0^k	$-c - v_i < 0$	$\times$	$-c - v_j - z_k < 0$
S_1^{ik}, $\emptyset$, $T_0^k + T_2^k$	$-c - v_i < 0$	$\times$	$-2c - v_j + r_k^{(5)}$
S_1^{ik}, $\emptyset$, T_1^k	$-c < 0$	$\times$	0
S_1^{ij}, W_0^{jk}, T_0^k	$-c - v_i < 0$	$-c - v_j < 0$	$-c - v_j - z_k < 0$
S_1^{ij}, W_0^{jk}, T_1^k	$-c < 0$	$-c < 0$	0
S_1^{ij}, W_1^j, $\emptyset$	$-c < 0$	0	$\times$
S_1^{ij}, W_2^{jl}, $\emptyset$	$-c < 0$	$-c - w_j < 0$	$\times$
S_1^{ij}, $W_0^{jk} + W_3^j$, $T_0^k + T_2^k$	$-c - v_i < 0$	$-2c - v_j + r_j^{(6)}$	$-2c - v_j + r_k^{(5)}$

Comments: (1) $c + w_k$ must be smaller than d_i to be positive; (2) c must be smaller than d_i to be positive; (3) c must be smaller than d_j to be positive; (4) $c + w_k$ must be smaller than d_k to be positive; (5) positive if $r_k > v_j + 2c$; and (6) positive if $r_j > v_j + 2c$.

Note that, in both cases, the best strategy for P^j and P^k is to co-operate with the witness P^i, since they can obtain a positive payoff either by forwarding a true emergency report or by accusing P^i as the source of a fake emergency report. P^k will only succeed in accusing P^i if P^j and other group members collaborate, but since this is also the best strategy for group members, the dominant strategy $(S_1^{ij}, W_0^{jk} + W_3^j, T_0^k + T_2^k)$ for P^k is to forward emergency reports always. Of course, there are several possible attacks in this scheme to try to obtain a positive payoff. For example, a player P^i may cause an emergency and forward it to P^j for submission to the EMS in order to obtain a positive payoff. This is not exactly an attack to the system, since that would be a real emergency after all (and there is a risk of being traced by the authorities anyway). Another possibility is to try to impersonate another user to generate a fake report, forward it to the EMS as P^k, and obtain a positive payoff by revealing the impersonated source. This is not possible since the signature algorithm of TDS (Appendix B.1) used in the protocol provides unforgeability.

4 Proposed Protocol

In this section, we present the protocol for sending and managing anonymous emergency reports to the EMS. The protocol mainly consists of three phases: witnessing

an emergency, managing and processing the emergency report, and the witness distinguisher.

A. Witnessing an emergency: When a user wants to report an emergency, he/she proceeds as follows. **(1)** The witness logins to the system, using his/her Facebook account details, and looks for nearby online contacts in the system. **(2)** The witness creates a dynamic and covert group of $n \geq 3$ nearby users. Since the users share location information with each other, the witness does not require any assistance of the system manager or the users to form a group. **(3)** An online group member (reporter) is selected by the witness to assist him/her in reporting the emergency to the EMS. **(4)** Multi-hop routes are computed at the witness's end to forward the emergency report to the reporter. The report is propagated along a selected route from hop to hop until it reaches the reporter. The hops simply forward the report without checking its content, which is encrypted and unreadable for them. **(5)** The witness prepares a report message r, which is a tuple $r = \{R_{id}, STdata, Content, k_m\}$: R_{id} is a report identifier; $STdata$ is a spatio-temporal tag; the $Content$ is the information of the emergency; and k_m is a random symmetric key that the user generates to establish an anonymous confidential channel between himself and the EMS. **(6)** The witness ciphers the report r with the public key of the EMS: $m = E_{Kp_{EMS}}(r)$, where $E()$ is a public-key cipher. **(7)** The witness signs the ciphered report m applying the signing procedure of the TDS scheme (Appendix B.1). With his/her private key x_i and the public keys of the group members, he/she generates the signature: $\sigma = S_{TDS}(g, x_i, y_1, \cdots, y_n, \alpha_1, \cdots, \alpha_n, t, m)$. **(8)** The witness sends the signed and ciphered report request (m, σ) to the reporter through a pre-defined routing path. Assuming that the path consists of two hops (P^{j_1}, P^{j_2}). The first hop P^{j_1} receives the packet: $(Sign_\sigma(ID_{witness}), \{((m,\sigma)_{y_k}, P^k)_{y_{j_2}}, P^{j_2}\}_{y_{j_1}})$. It decrypts the destination field to check whether it is the destination or not. If not, it generates a session key K_{m_1}, encrypts it with the public key of EMS $(K_{p_{EMS}})$, adds it into the packet and sends $(Sign_\sigma(ID_{witness}), E_{Kp_{EMS}}(K_{m_1}), \{(m,\sigma)_{y_k}, P^k\}_{y_{j_2}})$ to P^{j_2}. P^{j_2} would do the same thing to execute the similar operation and forward the packet $(Sign_\sigma(ID_{witness}), E_{Kp_{EMS}}(K_{m_1}), E_{Kp_{EMS}}(K_{m_2}), (m,\sigma)_{y_k})$ to the reporter P^k. **(9)** On receiving the packet from P^{j_2}, P^k checks the destination field of the packet. If no further hop is present, P^k decrypts the payload to obtain (m, σ). Then, P^k verifies whether the signature is discernible, authentic and integral by applying the verifying procedure of the TDS scheme (Appendix B.2). If the signature is verified, he/she submits $(P^k, y_k, (m,\sigma), E_{Kp_{EMS}}(K_{m_1}), E_{Kp_{EMS}}(K_{m_2}))$ to the EMS in accordance with the strategies explained in Section 3.

B. Managing and processing the emergency report: The EMS receives a signed and ciphered report request from a reporter. The EMS obtains the identity data of the reporter; the reporter is responsible for the information in front of the EMS, although the EMS knows that the reporter is not the witness of the event but a proxy chosen by the actual witness. The EMS also receives the session keys K_{m_1} and K_{m_2} of the intermediary hops.

Following are the steps that EMS follows to process the emergency report. **(1)** The EMS verifies the TDS signature generated by the witness. (**2**) The EMS deciphers the report using its private key: $r = D_{KS_{EMS}}(m)$, with $D()$ a public-key decipher; **(3)** The EMS obtains the public keys of n group members from the TDS signature and the report identifier from the report. It signs a group acknowledgement of emergency receipt $Ack = \{R_{id}, Group_{info}\}$, where $Group_{info}$ contains the public keys of n group members. It sends this acknowledgment Ack to the system manager, who sends it to all the group members in such a way that the witness knows about the report reception. If the witness does not receive Ack in a timeout t_0, he/she will try to send the report through another route or reporter; **(4)** Then, after verifying the correctness of the reported information (i.e. the emergency was true), the EMS prepares a reward or a punishment response. This response will be signed using the private key of the EMS. If the report is correct, the EMS first generates a hash value, $H_{EC} = H(ID_{EMS}\ ||Date||Time||STdata)||R_{id}||nonce_{R_{id}}||y_i)$ (where $H()$ is a collusion-resistant hash function, $nonce_{R_{id}}$ is a fixed value assigned to all the group members that have submitted the emergency report (R_{id}) and y_i is a public key of a group member), signs it and then generates the following rewards: (1) for the reporter P^k, which consists of the payoff ciphered with the reporter's public key y_k and a signed H_{EC}: $Reward_R = \{P^k, E_{y_k}(payoff),\ Sign_{KS_{EMS}}(H_{EC})\}$, (2) for the intermediary hops with the payoffs ciphered with the received symmetric keys K_{m_1} and K_{m_2} and a signed hash value: $Reward_H = \{Group_{info},\ C_{k_{m_1}}(payoff),\ C_{k_{m_2}}(payoff),\ Sign_{KS_{EMS}}(H_{EC})\}$, and (3) for the witness, ciphered with the symmetric key received from the witness k_m and signed hash value: $Reward_W = \{Group_{info}, C_{k_m}(payoff),\ Sign_{KS_{EMS}}(H_{EC})\}$ (with $C()$ a symmetric key cipher). The EMS sends these rewards to the system manager, who forwards the first reward $Reward_R$ to P^k and broadcasts the remaining two rewards $Reward_H$ and $Reward_W$ to all group members. Only P^{j_1}, P^{j_2} and the original witness P^i will be able to decipher $Reward_H$ and $Reward_W$, respectively, in order to redeem them from the CC. Also, the EMS sends a signed H_{EC} to the CC for later use in the reward redemption phase (see Appendix C). If the report is false, the EMS prepares punishments $Punishment_k = \{P^k, R_{id},\ E_{y_k}(payoff)\}$ and $Punishment_x = \{y_x,\ R_{id},\ E_{y_x}(payoff)\}$ (where $x = 1, \ldots, n-1$) for P^k and the remaining group members, respectively. Then, EMS sends $Punishment_k$ and $Punishment_x$ to the system manager, who retransmits them among the respective users. Also, the EMS requests the system manager to forward the identities of the group members ($Group_{info}$) to the CC, so that they get punished for reporting a false emergency; and **(5)** If the EMS repeatedly receives false information from the users of $Group_{info}$, the EMS puts them on a black list and no longer pays attention to the reports coming from them.

C. The witness distinguisher: If the group has been punished for a false emergency report, a subgroup of t users can join to reveal the identity of the malicious witness in order to obtain compensation (in terms of rewards) for the punishments inflicted on them by the EMS. The steps of the process are as follows. **(1)** A user P_u that participates in the disclosure process deciphers his/her share V_u of the request secret parameter and obtains ρ_u. He/She enciphers this information for the EMS, makes a

personal signature, and sends the result to the EMS. **(2)** The EMS deciphers and verifies the secret shares it receives. It also checks that the secret shares ρ_u received indeed correspond with the encrypted secret shares V_u. **(3)** When the EMS has the secret shares ρ_u of t users, it triggers the distinguisher algorithm of TDS (see Appendix B.3). It reconstructs the secret f_0 using the public parameters $(\alpha_1, \cdots, \alpha_n)$ and the secret shares $(\rho_1, \cdots, \rho_n)$ of the t participating users. Using f_0, the EMS can recover the identity of the original signer. **(4)** Then the EMS generates a nominal punishment for the malicious witness $Punishment_W = \{P^i, R_{id}, payoff\}$ and, at least, t rewards (one for each participant in the distinguisher process). It ciphers each payoff using the recipient's public key and sends $Reward_{P_u} = \{y_{p_u}, Sign_{KS_{EMS}}(H_{EC}), E_{y_{p_u}}(payoff)\}$ to the system manager, which distributes it to the respective members. The members can then redeem their rewards from the CC through by executing reward redemption protocol (Appendix C). The punishment for the malicious witness is sent to the witness as well as the CC, who will issue a penalty (fee) to the witness.

5 Discussion

The proposed protocol encourages users to send anonymous reports regarding some witnessed emergency. Anonymity is provided in two ways: (1) in the network layer using multi-hop report retransmissions, and (2) in the application layer using strong cryptography. Regarding multi-hop retransmissions, a witness forwards the emergency report through a fixed routing path (nearby online friends) to another online friend (within his/her vicinity), who in turn sends it to the EMS. This scenario, together with co-privacy, is analogous to the problem of user-private information retrieval [7]. If a witness sent his/her emergency report directly to the EMS, the EMS would know the IP address of this user and get his/her location, so his/her privacy would be surrendered. With this information and the emergency location (this data is always present in the report), the EMS could require more information of the reporter and the intermediary hops and involve them in the investigation of the events. Thus, users are always advocated to select user proxies for sending emergency reports.

When an emergency report is sent to the EMS, all group users are responsible for that report, although the main responsible entity is the reporter. If the report is true, the reporter receives a major payoff and the hops receive nominal payoffs, but if it is false, all group users are punished with the aim that they collaborate to find out the true witness. If the true witness can be discovered, the group members that participated in the witness distinguisher protocol, share some stipulated payoff and the reporter receives a major reward. The witnesses who sent false reports are never rewarded with a payoff even if they participated in the distinguisher protocol. The entire payoff that the EMS pays to the hops, the reporter and the users involved in the distinguisher protocol, is always smaller than the punishment for the malicious witness. This discourages Sybil attacks, where a user generates multiple accounts in order to gain a disproportionately large influence in the group and eventually obtain a global benefit although one of his/her identities (the witness) is severely punished.

In the protocol, the group is created dynamically based on the users' locations to avoid re-identification by strong adversaries. Thus, we propose to use a group consisting of users who are all in the partition where the emergency is located. This reduces the risk of re-identification of the witness even if the system manager and the EMS collude. However, there is a possibility that a witness finds only one user within a pre-defined distance to forward the report to the EMS. This implies that the identification of the witness would be immediate. A possibility to solve this problem is to step-wise increase the distance threshold (in meters). Since the reporting system is proposed for smart cities, it is highly likely that the witness could find at least three users within his/her close vicinity to form a group.

The proposed protocol uses cryptography to provide anonymity and authenticity in the application layer. Our proposal to protect users' identities is to work with TDS that authenticate a group of users (friends) instead of individual users. If a witness sends a report on the group's behalf, it should be impossible to identify which user is the originator. The security of TDS holds in the random oracle model [2], similar to the majority of the ring signature schemes. The security of these signatures has two aspects: unforgeability and signer anonymity. Unforgeability means that an external member of a group cannot create a ring signature with non-negligible advantage in polynomial time. Anonymity entails that at least t ring members of the group are required to discover the original signer of the t-threshold ring signature (with non-negligible advantage in polynomial time). It is worth noting that, in the presented protocol, anonymity is provided to the users without the presence of trusted third parties. The system manager and the EMS do not know the identity nor the IP address of the witness. However, two trusted parties (CA and CC) are required in the reward redemption protocol (Appendix C) so that the users can redeem their rewards in a privacy-preserving manner.

6 Conclusions and Future Work

We have presented an emergency reporting system that ensures the anonymity of honest witnesses but is able to disclose the identity and punish the malicious ones. The system is designed using the Facebook API that facilitates the creation of a group of users among which a witness can become indistinguishable. For a group formation or submission of the report, the witness does not need the assistance of the system manager and hence, it could not figure out the group's location. A game-theoretic approach based on the co-privacy principles is used to encourage the users to participate in the protocol. The conditional anonymity property is provided through threshold discernible ring signatures.

Future research should be directed: (1) To make the emergency information public and show it on a map (a feature which entails privacy risks that shall be examined and prevented); (2) to extend the co-utility model using multiple hops; and (3) to address the possibility of collusion between ring members such that each member gets a reward for reporting.

Acknowledgment

This work was partly funded by the Spanish Government through grants TIN2011-27076-C03-02 "CO-PRIVACY" and TIN2014-57364-C2-2-R "SMARTGLACIS".

References

1. Alpify: An app that can save your life. `http://www.alpify.com` (2014), (Accessed on June 9, 2018)
2. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM conference on Computer and communications security. pp. 62–73. CCS '93, ACM, New York, NY, USA (1993)
3. Camenisch, J.: Efficient and generalized group signatures. In: Fumy, W. (ed.) Advances in Cryptology EUROCRYPT 97, Lecture Notes in Computer Science, vol. 1233, pp. 465–479. Springer Berlin Heidelberg (1997)
4. Committe, E.: False emergency calls. Operations Document 3.1.2, European Emergency Number Association (EENA) (2011)
5. Domingo-Ferrer, J.: Coprivacy: an introduction to the theory and applications of cooperative privacy. SORT-Statistics and Operations Research Transactions 35(special issue: Privacy in statistical databases), 25–40 (September 2011)
6. Domingo-Ferrer, J.: Coprivacy: Towards a theory of sustainable privacy. In: Domingo-Ferrer, J., Magkos, E. (eds.) Privacy in Statistical Databases, Lecture Notes in Computer Science, vol. 6344, pp. 258–268. Springer Berlin Heidelberg (2011)
7. Domingo-Ferrer, J., Bras-Amorós, M., Wu, Q., Manjón, J.: User-private information retrieval based on a peer-to-peer community. Data & Knowledge Engineering 68(11), 1237 – 1252 (2009)
8. Domingo-Ferrer, J., Megías, D.: Distributed multicast of fingerprinted content based on a rational peer-to-peer community. Computer Communications 36(5), 542 – 550 (2013)
9. Facebook. `http://www.facebook.com/` (2004), last accessed on May 06, 2016
10. Furtado, V., Ayres, L., de Oliveira, M., Vasconcelos, E., Caminha, C., D'Orleans, J., Belchior, M.: Collective intelligence in law enforcement - the wikicrimes system. Inf. Sci. 180, 4–17 (January 2010)
11. Klonowski, M., Krzywiecki, ., Kutyowski, M., Lauks, A.: Step-out ring signatures. In: Ochmaski, E., Tyszkiewicz, J. (eds.) Mathematical Foundations of Computer Science 2008, Lecture Notes in Computer Science, vol. 5162, pp. 431–442. Springer Berlin Heidelberg (2008)
12. Kumar, S., Agrawal, S., Venkatesan, R., Lokam, S., Rangan, C.: Threshold discernible ring signatures. In: Obaidat, M., Tsihrintzis, G., Filipe, J. (eds.) e-Business and Telecommunications, Communications in Computer and Information Science, vol. 222, pp. 259–273. Springer Berlin Heidelberg (2012)
13. Meier, P.: Digital Humanitarians: How Big Data is changing the face of humanitarian response, chap. 2. CRC Press (2015)
14. Namahoot, C., Bruckner, M.: Spears: Smart phone emergency and accident reporting system using social network service and dijkstra's algorithm on android. In: Mobile and Wireless Technology 2015. Lecture Notes in Electrical Engineering, vol. 310, pp. 173–182. Springer Berlin Heidelberg (2015)
15. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press, New York, NY, USA (2007)
16. Okolloh, O.: Ushahidi or 'testimony': Web 2.0 tools for crowdsourcing crisis information. Participatory Learning and Action 59, 65–70 (2009)
17. Reed, M., Syverson, P., Goldschlag, D.: Anonymous connections and onion routing. IEEE Journal on Selected Areas in Communications 16(4), 482–494 (1998)

18. Rivest, R., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) Advances in Cryptology ASIACRYPT 2001, Lecture Notes in Computer Science, vol. 2248, pp. 552–565. Springer Berlin Heidelberg (2001)
19. Tapshield: How mobile and cloud technologies are reducing emergency response times. White paper, Tapshield (2014), last accessed on May 06, 2016

Appendix A Basics of Game Theory

As detailed in [15], a game is a protocol between a set of N *players*, $\{P^1, \cdots, P^N\}$ who must choose among a *set S_i of possible strategies*. Let $s_i \in S_i$ be the strategy played by player P^i and $S = \Pi_i S_i$ the set of all possible strategies for all players.

The vector of strategies $s \in S$ chosen by all players determines the outcome of the game for each player which can be thought of as a payoff or a cost. For all players, a preference ordering of these outcomes should be given in the form of a complete, transitive and reflexive relation on the set S. A simple and effective way of achieving this goal is by defining a scalar value for each outcome and each player. This value may represent a payoff (if positive) or a cost (if negative). A function that assigns a payoff to each outcome and each player is called a utility function: $u_i : S \longrightarrow \mathbb{R}$.

Given a strategy vector $s \in S$, s_i denotes the strategy chosen by P^i, and let s_{-i} denote the $(N-1)$-dimensional vector of the strategies chosen by all other players. With this notation, the utility $u_i(s)$ can also be expressed as $u_i(s_i, s_{-i})$. A strategy vector $s \in S$ is a *dominant strategy solution* if it yields the maximum utility for a player irrespective of the strategy played by all other players, i.e. if

$$u_i(s_i, s'_{-i}) \geq u_i(s'_i, s'_{-i}),$$

for each alternate strategy vector $s' \in S$.

In addition, a strategy vector $s \in S$ is said to be a *Nash equilibrium* if it provides the largest utility for all players, larger than any other alternate strategy $s'_i \in S_i$ or

$$u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i}).$$

This mean that, in a Nash equilibrium, no player will be able to change his/her strategy from s_i and achieve a better payoff when all the other players have chosen their strategies in s. Note that Nash equilibria are self-enforcing if players behave rationally, since it is in all players' best interest to stick to such a strategy. Obviously, if all players are in a dominant strategy solution at the same time, this is a Nash equilibrium. Further background on game theory can be found in [15].

Appendix B Threshold Discernible Ring Signatures

We base our system on threshold discernible ring signatures (TDS), which were introduced by Kumar et al. [12]. In a t-threshold discernible ring signature, a user in the system can generate a signature using his/her own private key and the public keys of the other n ring members (with $n > t$). A verifier is convinced that someone in the

ring is responsible for the signature, but he/she cannot identify the real signer. The identity of the signer can only be revealed if a coalition of at least t members of the group cooperates to open the secret identity. In the following, three TDS operations that were used in the proposed protocol are outlined.

B.1 Signature

The signing algorithm $S_{TDS}(g, x_i, y_1, \cdots, y_n, \alpha_1, \cdots, \alpha_n, t, m)$ generates a ring signature of a message m and a set of verifiable encrypted shares of a secret that allows disclosing the identity of the original signer. The secret, which we call f_0, can only be revealed when a group of t ring members brings together some information. For signing a message m, the user first generates t random numbers $f_j \in Z_q^*$ and computes $F_j = g^{f_j}$ for each of them. The first random number f_0 is used as a trapdoor to hide the real signer of m, and hence, this f_0 is partitioned using the Shamir's secret sharing scheme [18] and verifiably encrypted (VE) in n shares V_k, one for each user of the group, using the public parameters of all the group members $\{(y_1, \alpha_1), (y_2, \alpha_2), \cdots, (y_n, \alpha_n)\}$.

$$s_k \leftarrow f_0 + \sum_{j=1}^{t-1} f_j \alpha_k^j, k = 1, \cdots, n,$$

$$\hat{g} \leftarrow g^l,$$

$$V_k \leftarrow VE_{y_k}(s_k : g^{s_k} = \hat{g} \prod_{j=1}^{t-1} F_j^{\alpha_k^j}), k = 1, \cdots, n.$$

Then, the user generates another tuple of n random numbers $r_j \in Z_q^*$ and computes $w_j = g^{r_j}$ for each of them. He/She also calculates $\hat{y}_w \leftarrow \hat{g}^{x_i + r_i}$. Finally, he/she computes an equality signature [11] $(EC, ES) \leftarrow S_{SEQDL}(\hat{g}, g, x_i, r_i, \hat{y}_w, Y, W, m)$ and n knowledge signatures $\{(kc_k, ks_k) \leftarrow S_{SKDL}(g, w_k, m),\ k = 1, \cdots, n\}$ (with $Y \leftarrow y_1, \cdots, y_n$,$W \leftarrow w_1, \cdots, w_n$, $KC \leftarrow kc_1, \cdots, kc_n$,$KS \leftarrow ks_1, \cdots, ks_n$) that allow the signer to prove in zero-knowledge the integrity of the signed report and its group authenticity. The output of the signature algorithm is a threshold discernible ring signature $\sigma = (\sigma_1, \sigma_2)$ where, $\sigma_1 \leftarrow (\hat{g}, \hat{y_w}, Y, W, EC, ES, KC, KS)$ and $\sigma_2 \leftarrow (V, F)$ with $V \leftarrow V_1, \cdots, V_n$, and $F \leftarrow F_1, \cdots, F_t$.

B.2 Verification

The verification algorithm $V_{TDS}(m, \sigma)$ contains two actions: (1) checking the origin discernibility of the signature, i.e. the encrypted shares of the secret f_0 are verifiable and thus, a coalition of t users could reveal the identity of the signer,

$$Verify(VE_{y_k}(s_k : g^{s_k} = \hat{g} \textstyle\prod_{j=1}^{t-1} F_j^{\alpha_k^j}) = 0, \text{ for any } i = 1, \cdots, n).$$

and (2) verifying the ring signature, i.e. checking that some member of the group with a valid private key has signed m and, thus, that m is authentic and integral.

For this, a user first executes a proof of knowledge procedure [3] $V_{SKDL}(g, w_k, m)$ for any $i = 1, \cdots, n$, to check that the signer knows the n random numbers $r_j \in Z_q^*$ used in the signature. Then, it executes the verification algorithm of the signature of knowledge of equality of discrete logarithms $V_{SEQDL}(\hat{g}, g, \hat{y}_w, Y, W, EC, ES, m)$.

B.3 Threshold Distinguisher

The threshold distinguisher algorithm requires that at least t members of the ring decrypt their secret share V_i with their private key x_i to obtain ρ_i. Then, these users have to share their respective ρ_i's to disclose the secret element of the signature f_0. This can be computed using Lagrange's interpolation formula. After obtaining f_0, the users will be able to discover the signer of the message yielding the user P^i that matches the following equation: $(y_i w_i)_0^f = \hat{y}_w$.

Appendix C The Reward Redemption Protocol

The EMS responds the witness, the hops and the reporter (immediately or after some days) with a reward for reporting a true emergency. The witness receives a reward encrypted with k_m, which is only known to the witness. The hops and the reporter receive the rewards encrypted with their corresponding public keys.

In order to redeem the rewards from the CC, the awardees proceed as follows. **(1)** Each awardee A_i generates a pseudo-identity (PI) with the help of a CA. This PI is used by A_i for redeeming a reward at the CC anonymously. **(2)** On receiving a request from A_i for generation of PI, the CA selects a secret random number $b \in Z_p^*$, encrypts it with A_i's public key and sends it to A_i. Thus, CA and all the awardees share a secret number b. A_i deciphers b, selects a random number $a \in Z_p^*$ and uses his/her secret key to sign $\{ID_{A_i}, \text{Cert}_{CA}(A_i), b, a\}$. A_i computes his/her PI by using a hash function: $PI_{A_i} = H(ID_{A_i}, \text{Cert}_{CA}(A_i), b, a, \text{Sign}_{A_i}(\text{Cert}_{CA}(A_i), b, a))$. **(3)** A_i generates a key pair $(y_{A_i}^*, x_{A_i}^*)$, signs the public key with his/her private key, and sends $\text{Sign}_{A_i}(y_{A_i}^*, PI_{A_i})$ to CA. CA verifies the signature using the public key of A_i. If valid, CA generates an anonymous certificate $\text{Cert}_{CA}(PI_{A_i}, y_{A_i}^*)$ and sends it to A_i. **(4)** A_i sends a payoff redeem request, $payoff_{Req} = \{PI_{A_i}, \text{Cert}_{CA}(PI_{A_i})\}$, to the CC. **(5)** CC verifies the received certificate from the CA of the system. If verified, CC generates a session key k_{A_i}, encrypts it with A_i's public key and sends it to A_i. Otherwise, CC aborts the redemption process. **(6)** A_i encrypts the received *payoff* and the signed hash using k_{A_i} and sends $payoff_{Req} = \{C_{k_{A_i}}(payoff, Sign_{KS_{EMS}}(H_{EC})), \text{Cert}_{CA}(PI_{A_i}), PI_{A_i}\}$ to CC. **(7)** CC performs decryption with k_{A_i} and obtains the clear text $Sign_{KS_{EMS}}(H_{EC})$ and *payoff*. CC first checks if PI_{A_i} has already redeemed the *payoff* by looking up $\{Sign_{KS_{EMS}}(H_{EC}), payoff, PI_{A_i}\}$ in its database. If no such entry exists, CC sends $Sign_{KS_{EMS}}(H_{EC})$ to the EMS for validation. If the *payoff* has already been redeemed by PI_{A_i}, CC aborts the redemption process. **(8)** If the received H_{EC} is equal to the stored H_{EC}, the EMS sends *accept* notification to the CC. On receiving *accept*, CC sends rewards to A_i. CC then sets a redemption flag to 1 and stores $\{FL = 1, \text{Cert}_{CA}(A_i), PI_{A_i}, payoff, Sign_{KS_{EMS}}(H_{EC})\}$ in its database.