An Opportunistic Authority Evaluation Scheme for Data Security in Crisis Management Scenarios

Enrico Scalavino
Department of Computing
Imperial College London
180 Queen's Gate
London, United Kingdom
SW72AZ
escalavino@imperial.ac.uk

Giovanni Russello
Create-Net
Via alla Cascata 56/D
Povo (TN), Italy 38123
giovanni.russello@create-net.org

Rudi Ball
Department of Computing
Imperial College London
180 Queen's Gate
London, United Kingdom
SW72AZ
rkb@imperial.ac.uk

Vaibhav Gowadia
Department of Computing
Imperial College London
180 Queen's Gate
London, United Kingdom
SW72AZ
vgowadia@imperial.ac.uk

Emil C. Lupu
Department of Computing
Imperial College London
180 Queen's Gate
London, United Kingdom
SW72AZ
e.c.lupu@imperial.ac.uk

ABSTRACT

We propose a novel version and implementation of the *Policy-based Authority Evaluation Scheme (PAES)* to protect data disseminated amongst the responders to an emergency situation when no network connectivity is available. In such situations Delay Tolerant Networks (DTN) are used to disseminate the data by exploiting the peers' mobility in the area. However, existing DTN protection models require recipients to be known in advance. In emergency situations the data may instead be received by unknown responders who might need it while carrying out their duties. Existing data dissemination solutions such Enterprise Rights Management (ERM) systems rely on centralized architectures where recipients must contact the authorities that can grant access to data. Such centralized solutions cannot be deployed when connectivity cannot be guaranteed. Our solution combines data protection schemes such as ERM systems with DTNs. The result allows us to implement a distributed policy evaluation procedure for DTNs. Simulations demonstrate that the approach permits recipients to obtain fast access to protected data even when no authority can be contacted. This is particularly important in crisis situations where timely access to data is necessary.

Categories and Subject Descriptors

D.4.6 [**Software**]: Operating Systems—*Security and Protection*; K.6.5 [**Computing Milieux**]: Management of Computing and Information Systems—*Security and Protection*

ASIACCS'10 April 13–16, 2010, Beijing, China.

General Terms

Security, Management

Keywords

Policies, ERM, Encryption, Crisis Management, Security, Delay Tolerant Networks

1. INTRODUCTION

Incidents, natural events and technical breakdowns lead to situations that threaten peoples lives, the environment, or human settlements. Crisis situations develop unpredictably and unexpectedly, and require rapid deployment of relief operations and short decision times. The timeliness and efficiency of decisions depend in turn on the availability of accurate information on the situation and its development. However, collecting and disseminating data during a crisis is often difficult as traditional communication infrastructures may be damaged, unusable or simply missing, e.g. in an underground tunnel. Moreover, the information required may be of a sensitive nature, especially as the crisis unfolds. Medical records, building or infrastructure plans, information on hazardous substances etc. are under normal circumstances protected by complex access control procedures; the number of victims and the nature of hazards are also considered sensitive as the crisis unfolds to avoid panic situations.

When network connectivity is absent, the movements of first responders at the scene can be leveraged to carry the information to whoever needs it. Such networks that leverage node mobility to disseminate messages are known as *opportunistic networks (Oppnets)* or *delay tolerant networks (DTNs)* [23, 30, 12]. Oppnets are ideal for first responders that need to exchange information in the area of an incident [17, 4, 3]. However, data protection schemes for ad-hoc networks [18, 28] are mostly intended for protecting communications between a specific sender-recipient pair, i.e. to protect data packets while they are stored and forwarded by intermediate nodes. This conflicts with the need to timely and widely disseminate information. Information must be shared among several responders and sending separate copies of the same data to each intended recipient is not a viable solution in such a low bandwidth network. Further-

more, not all recipients are known in advance. Timeliness of data is a crucial factor for the success of the operations and it is therefore important to balance the trade-off between data dissemination requirements and data protection requirements. Communication-only protection schemes are no longer sufficient.

Several rights management approaches have been proposed to protect disseminated content in both academia and industry (often referred to in industry as *Enterprise and Digital Rights Management (ERM and DRM)*). Although these solutions differ in their management of user authentication, policy and rights retrieval, audit of user actions and other tasks [6, 19, 20, 22, 27], their centralised architecture make them unusable in opportunistic and ad-hoc networks. Such solutions require clients to contact centralised trusted authorities that must issue access authorisations, usage rights and decryption keys. Intermittent connectivity which is often present in a crisis situation makes this impractical if not impossible.

As the scale of the incident grows larger, several agencies are called to intervene and share information. Although *Data Sharing Agreements (DSAs)* between the agencies would have been typically set up, it is not possible for them to agree which responders will be authorised to access which data (and pre-emptively give them access keys [24, 10, 5]), as crisis situations often evolve in sudden and unpredictable ways that do not conform to the rehearsed scenarios. The context in which the events unfold must thus be considered at each access request. Agencies with no pre-existing DSA may also be called upon to intervene. A re-negotiation of the agreements among the different agencies is not viable for two main reasons: first, the timescales of emergencies and the need for a quick intervention prohibit it, second rescuers already scattered in the area may not be reachable and thus unaware of the new policies.

In this paper we propose a solution to integrate traditional protection mechanisms for disseminated data in oppnets for crisis scenarios. Our proposal is based on the *Policy-based Authority Evaluation Scheme (PAES)* presented in [25]. The basic idea is that trusted authorities are not part of static sets but are defined by characterisation. As the context evolves, entities can gain new rights and become authorities for others. In this paper we extend the original PAES design to allow a more flexible policy evaluation and then adapt it for use in oppnets. The aim is to increase the probability that rescuers can meet and contact a trusted authority by means of an ad-hoc connection, and thus receive the needed authorisations.

The remainder of this paper is organised as follows: related work is presented in Section 2 whilst Section 3 describes a crisis scenario providing a context for simulations; Section 4 gives a general overview of the oppnet deployed at the crisis site while Section 5 describes how data is protected before being disseminated; PAES's principles are described in Section 6; the policy evaluation and key distribution processes realising the scheme are described in Section 7. In Section 8 we show the results of the simulations proving the validity of our approach. Finally, conclusions are drawn in Section 9, which also briefly discusses future work.

2. RELATED WORK

Protecting wireless data communications between first responders in crisis situations has been gaining increasing attention in recent years, especially from public safety agencies [1, 13]. The proposed solutions are based either on a static distribution of decryption keys, before any crisis occurs, or on centralised architectures where responders are assumed to have perfect connectivity. However this is often unrealistic, especially as crisis situations are often unpredictable. Moreover, statically loading decryption keys into responders' devices prevents any dynamic evaluation based on the actual situation and context.

The multi-hop ad-hoc network paradigm [12] has been introduced to address situations where no network connectivity is available. Intuitively, peers leverage the temporary ad-hoc connections they establish with each other when moving to deliver messages. Vahdat et al. [30] introduced *Epidemic Routing* that broadcasts messages to all nearby peers to guarantee their final delivery to an intended recipient. Lilien et al. [17] proposed the *Opportunistic Networks* (oppnets) paradigm, where a small set of initial peers (*seeds*) iteratively ask others to participate in the network, and become *Helper Nodes*. By including new peers the oppnet grows, thus increasing the likelihood that messages are delivered. Li et al. [16] proposed an algorithm to actively modify peers movement trajectories to increase message delivery. The *Haggle* project [26, 29] integrates the various approaches by introducing an architectural layer that automatically selects the best connection mode available when applications require network access.

Several proposals have applied ad-hoc networking to crisis management. Aschenbruck et al. [4, 3] present a mobility model for rescuers based on the established procedures that public agencies follow when facing a crisis. The simulation results show that ad-hoc networks benefit from the higher density of the peers and their constrained mobility. However, the security models proposed present some shortcomings. Lilien et al. [18] highlight the general security requirements for ad-hoc networks in terms of message privacy and integrity and of common attacks that can hinder the network operation (e.g. Denial of Service). However, the authors suggest that because of the impossibility to authenticate peers when they enter the network, it is necessary to rely on Intrusion Detection or Trust Management Systems to discover malicious nodes only after they have already misbehaved. Seth et al. [28] propose the use of Hierarchical Identity-Based Encryption (HIBE) to encrypt messages disseminated in an ad-hoc network. However, this solution is suitable only when the identity of the intended recipient is known in advance and when all the participants to the network know the central Private Key Generator (PKG).

Different proposals for protection of disseminated data have been made in the ERM area. Most ERM products, such as the Authentica [6], Liquid Machines [2] or Microsoft RMS (Rights Management System) [19] originate in industry and present similar characteristics. MS RMS is perhaps representative of their common design. Its architecture is centralised and based on the deployment of publishing servers (or trusted authorities) that issue encryption keys to users authorised to disseminate data and decryption keys to users authorised to access it. Before disseminating data, users *publish* it on a *publishing server*. The server then creates a *publishing licence* that contains the key used to encrypt the data (protected for the server itself). The user can freely distribute the licence to recipients that use it to contact the publishing server and obtain, if authorised, the decryption keys. A software running on the recipients device will then enforce locally the access and usage control policies for the data. Other systems differ mainly in the expressiveness of the authorisation policies, the authentication mechanisms employed, the policy deployment methods and the techniques used to store credentials. General guidelines for ERM architecture were also given by Park et al. in [22]. As noted earlier, the main shortcoming of these architectures is their centralised nature.

Our approach aims to apply the evaluation mechanism used in ERM systems to ad-hoc networks. To do so, we leverage the Policy-based Authority Evaluation Scheme [25] where authority over a policy evaluation is granted by the positive evaluation of another policy itself evaluated by an authorised entity. Intuitively, this recursive process generates an evaluation/authority chain that can also be realised between peers moving in a crisis scenario.

3. A CRISIS MANAGEMENT SCENARIO

The main difficulty when dealing with events that escalate to full blown disasters is the unpredictability of their occurrence, of their evolution over time and of the context in which they occur. Civil protection agencies such as Police, Fire Brigade and paramedic teams all intervene according to pre-existing guidelines and strategies that are prepared only for the most general situations. It is in fact almost impossible to foresee all possible cases and difficulties that can arise during an evolving crisis. In the following we introduce a crisis management scenario where different organisations, including some companies not directly involved with the rescue operations (e.g. utilities and transport providers), are called to gather and share information. Despite being only an hypothetic scenario, similar situations have happened in the past. For example in 1999 the tragedy of the Mont Blanc Tunnel caused the death of 39 people when a camion caught fire in the tunnel. The rescue operations lasted for 53 hours during which cooperation amongst the different agencies, belonging to different countries, was fundamental.

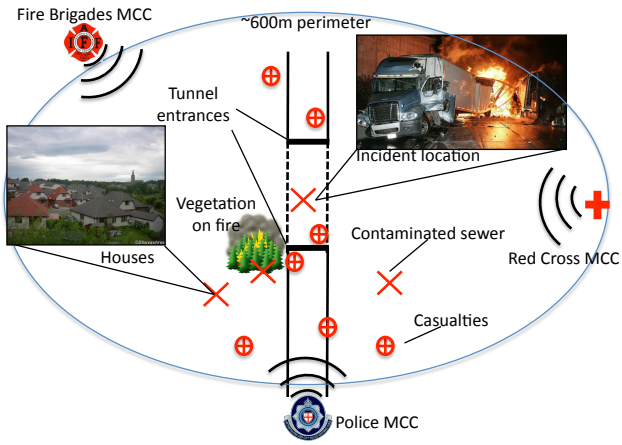

Figure 1: An example scenario. Several vehicles crash in a tunnel. Different rescue agencies arrive in the area for support.

The scenario describes a traffic accident that rapidly escalates into a threat to a larger surrounding area. Figure 1 depicts the scene. A car has been bashed between a petrol tanker and a chemical tanker in the middle of a tunnel. An explosion at the time of the impact caused a thick black smoke to fill the tunnel, while escaped witnesses state that one of the tankers is leaking a "suspicious" liquid. From the street it is possible to notice a chemical spilling out of the tunnel seeping into a nearby sewer, while several injured persons come out of the tunnel with breathing problems. Nearby vegetation, close to a small residential district, also catches fire.

Police, Fire Brigade and Red Cross immediately intervene on the scene with both vehicles and mobile stations and deploy in the area *Mobile Command Centres (MCCs)*. The "threat area" they consider is a perimeter of 100m radius centred on the tunnel. Examples of information the rescuers need to exchange and that must be protected are: 1) personal and medical information of victims; 2) the tunnel and sewer plants and information on nearby gas pipelines, gas storages and electrical facilities; 3) information on the tankers' content; 4) the map of the surrounding area with information on the houses to be evacuated; 5) information on the state of the accident and of the rescue operations. In addition to the initial rescue teams, many other organisations are called to share data on the accident: the service providers, the local council, an environmental agency, the tankers' transport companies.

In this context, rescuers' timely access to data represents the basis for a prompt and effective rescue plan, while failures to share information are a source of later embarrassment and can have a high cost even in terms of human life. Data confidentiality must nevertheless be protected for several reasons. First, victims' privacy is governed by legislation such as HIPAA in US. Second, information on the accident may cause panic while burglars may target houses that have been evacuated and criminals may use information on the local facilities and utilities for future crimes. Finally, leaked information revealing a possible mismanagement of the operations may be used to embarrass the involved agencies and individuals.

In this scenario, the lack of long-range connectivity is a very likely circumstance. The communication infrastructure could have been disrupted by the incident itself (e.g. if the explosion damaged a local antenna) or could have never been present in the tunnel. In this sense, incidents in particular environments such as underground tunnels or caves are generally very difficult to deal with. Also, the communication network may have been shut down for security reasons, or public panic may overload the existing infrastructure making it unusable for emergency services. A more detailed description of the rescue teams, vehicles and stations deployment and of their behaviour will be given in section 8.

4. NETWORK OVERVIEW

We assume that responders are equipped with mobile devices such as PDAs, smartphones and netbooks capable of short-range wireless communication and that can establish temporary ad-hoc connections when responders meet. The devices can also store small amounts of information and can deal with encryption keys and algorithms. Modern smartphones providing a Java Virtual Machine and equipped with an SD card or other memory are typical of the devices we consider. The emergency opportunistic network established in our case comprises three types of peers: rescuers, data aggregators and roots of authority. Rescuers move in the disaster area, gather context information and store and forward messages while moving. Note that they may carry messages they cannot themselves access, which are protected as described in the following sections. Data aggregators are usually located on the perimeter of the disaster area. Although it is not a strict requirement, we consider data aggregators to have special connection capabilities that allow them to send/receive information to/from Remote Command Centres (RCC) located in the organisations' headquarters. They thus receive context information both from rescuers and RCCs. They may also receive strategic information from RCCs (e.g. maps of the area) that are used to take management decisions and coordinate the operations. Intuitively, they aggregate information received from both outside and inside the disaster area and redisseminate derived information of limited size. Roots of authorities are the peers entrusted by the respective organisations to start a sequence of policy evaluations as described in the next sections. They basically keep the data access and usage policies and deploy them in the crisis area. Each individual rescuer is assumed to store in his/her device a copy of the identity (public key) certificate of the roots of authority and of the data aggregators belonging to his agency and of others well-known trusted third parties. In our scenario, we consider MCCs to work as both roots of authorities and data aggregators. Figure 2 shows a snapshot of the the network.

5. DATA GATHERING AND PROTECTION

Whenever new data is gathered by a rescuer or created by a data

Data Category	Usage Control Policies	EAGs	Node Type
Toxic threat	accessType="read" ∧ emergencyCode="red" signed by (FF_MCC, RC_MCC)	fire_fig param	loose
Terrorist threat	accessType="read"	pol_off	
Evacuation info	accessType="read" ∧ spaceInclusion(gpsLocation,data.area)	...	...
Building plans	accessType="read" ∧ emergencyCode="red" signed by (FF_MCC)	...	...
Facilities and services	...	...	...
Casualties	...	...	...

Table 1: Example data category table for the Police Force.

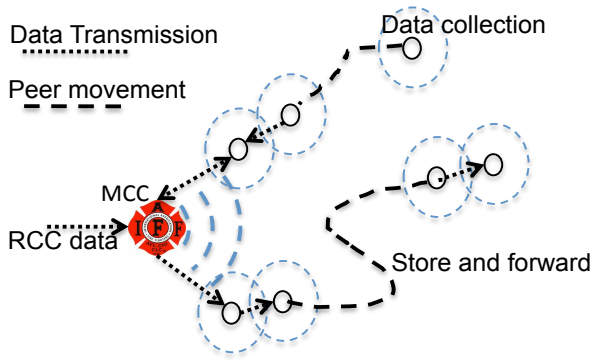

Figure 2: MCCs aggregate data from individual responders and RCCs and further disseminate it in the disaster area.

aggregator it must be encrypted before dissemination. A new symmetric key k_r is thus generated by the peer and used to protect the data. k_r is then encrypted with the public key of a trusted evaluation authority (as in ERM systems) and attached to the data itself. The difficulty of applying traditional protection schemes to oppnets is the lack of reachable Trusted Evaluation Authorities (EA) that can evaluate policies and issue decryption keys. To address this issue we need to distribute the concept of a Trusted Authority in an ad-hoc network but first we have to distinguish between the different *data categories*.

Rescue organisations typically have pre-defined intervention strategies that their personnel follow. Similarly, we assume that pre-defined policies are defined for different categories of data most likely to be disseminated during a specific type of emergency. In our approach, rescue organisations are required to specify such data categories and the corresponding usage control policies that must be enforced on the recipient device for the whole duration of the data access. Moreover, each data category-usage control policy pair must be associated with a characterisation of the group of peers that are trusted to access the data and to enforce the usage control policy. In other words, which peers are evaluation authorities for the usage control policy. This information is specified in a *data category table* that is statically defined before any crisis happens. Each group of authorities is associated with a pair of public and private keys whose generation and distribution we discuss later.

Table 1 shows an example of data categorisation for the Police force. The table is stored in every policeman's personal device. Each row defines for a category of data its corresponding usage control policy (in the **Usage Control Policies** column) and who is authorised to evaluate it (in the **Evaluation Authority Groups**, EAGs column). In our example, only members of groups *fire_fighter* and *paramedics* are authorised to evaluate the usage control policy for data belonging to the "Toxic threat" category and thus to access it. Both fire fighters and paramedics must not be able to access the document once the crisis is over (e.g. when the alert level disseminated by any of the three MCCs is not "red"). The column **Node Type** specifies whether authorised peers must be members of only one (*loose*) or all (*strict*) of the groups specified in the EAGs field. We explain the terminology and how users become members of the different groups of authorities in the next sections. Note that any policy language could be used to specify policies, provided that its conditions can be verified. The conditions are in most cases application specific and our system is agnostic to the language employed or the conditions specified. As an example, we consider here conditions based on recipients' attributes (credentials) and context attributes (e.g. time, location, state of the operations etc, see section 7.2.1 for details on their verification).

Whenever new data is gathered, it is associated with a category (automatically or by a user). For simplicity we do not consider here data belonging to more than one category. If the chosen data category corresponds to a loose evaluation node, then for each group listed in the corresponding EAGs field a different copy of k_r is encrypted with that group's public key. If the chosen data category corresponds instead to a strict evaluation node, then k_r is sequentially encrypted with all the public keys of the groups listed in the EAGs field. The encrypted k_r or its multiple copies are then attached to the new data package that can be now disseminated. Only members of the authorised groups can decrypt it.

6. POLICY-BASED AUTHORITY EVALUATION SCHEME

The Policy-based Authority Evaluation Scheme (PAES) was introduced in [25] to simultaneously permit distributing policy evaluation over a flexible set of authorities and increasing the resilience of policy enforcement. PAES is based on the consideration that authority to evaluate policies can be granted by other associated policies in the same manner as access to data is granted by data access policies. Users can thus be evaluated by any reachable authority satisfying the data originator requirements. In this paper we extend the original scheme to deal with cases where different authorities have to evaluate an access request, and then describe how the scheme is applied to emergency oppnets for allowing peers to become members of different evaluation authority groups. We define a policy as:

DEFINITION 1. *A policy p is a tuple (t,a,c) where t is a protected target object, a is an operation that can be executed on the object and c denotes a set of conditions constraining the operation execution. A usage control policy indicates the conditions that must*

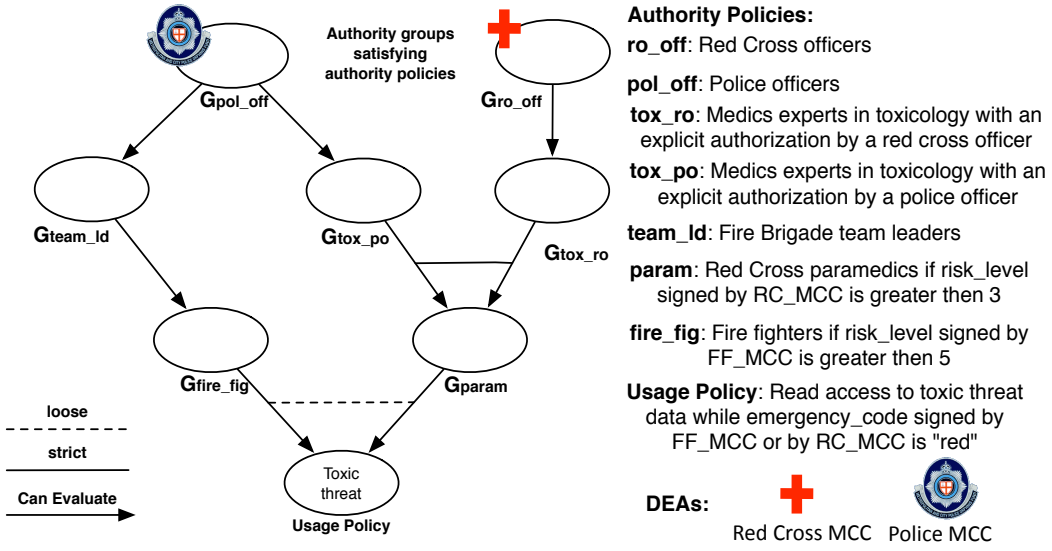

Figure 3: An example policy graph. Each group of authorities satisfying specific requirements has right to evaluate the satisfaction of the requirements for the underlying groups.

be verified while sensitive data is being used. An authority policy p_i is a policy (P_i, eval, c), describing the conditions that must be satisfied for an entity to be authorised to evaluate any of the policies $p \in P_i$.

In the following G_p indicates the group of entities satisfying policy p, i.e. that can perform action a on object t. For an entity to be included in G_p, an *evaluation authority* trusted by the data originator must vouch that the conditions specified in p are satisfied. The core idea of PAES lies in the distinction between directly trusted evaluation authorities (i.e. authorities whose identity is well known) and authorities defined by characterisation, i.e. defined by a policy that must be in turn evaluated by other authorities. Traditional ERM systems only use evaluation authorities directly trusted by the data provider, i.e. data recipients are forced to request access to only one specific authority. PAES aims instead at allowing recipients to request access to any entity satisfying certain requirements. In the following we use DEA_p and EA_p to designate the set of directly trusted evaluation authorities and the whole set of evaluation authorities (therefore, $DEA_p \subseteq EA_p$) for policy p (i.e. that can evaluate p).

PAES originally defines *policy chains* where authorisation to evaluate a policy is granted by the satisfaction of a subsequent authority policy. In this paper we extend this to permit the definition of *policy graphs*. This allows a more flexible authority evaluation since authorisation to evaluate a policy is granted by satisfaction of one of several or all of the subsequent policies in the graph, as in turn evaluated by different authorities. In other words, this allows the definition of looser or stricter requirements for the granting of evaluation rights. In the following, we will use the terms $in(n)$ to refer to the sets of nodes with an outgoing arc to node n and $out(n)$ to refer to those with an incoming node from node n in the policy graph:

DEFINITION 2. *A policy graph is a directed acyclic graph where:*

- *each node n_i such that $out(n_i) \neq \emptyset$ is a tuple (p_i, DEA_i) where p_i is an authority policy (P_i, eval, c), DEA_i is the set of authorities directly trusted to evaluate it and such that $\forall n_j | n_j \in out(n_i) : p_j \in P_i$;*
- *each node n_i such that $in(n_i) = \emptyset$ is a tuple (p_i, DEA_i) where $DEA_i \neq \emptyset$ (i.e. root nodes always have at least one directly trusted evaluation authority);*
- *each node n_j such that $out(n_j) = \emptyset$ is a tuple (p_j, DEA_j) where p_j is a usage control policy (i.e. leaf nodes always contain a usage control policy).*

Note that policies contained in root nodes are always evaluated by an authority directly trusted by the policy writer (i.e. $EA_{p_i} = DEA_i$). Nodes in a policy graph can be classified as either *strict-evaluation nodes* or *loose-evaluation nodes*. Intuitively, strict and loose evaluation nodes represent respectively boolean conjunction and disjunction of conditions that must be evaluated by different authorities. More formally, the difference among them resides in the way authorities for the contained policies are identified:

DEFINITION 3. *Let $n_i = (p_i, DEA_i)$ be a strict-evaluation node in a policy graph and ea be an entity acting in the system where the policy graph is defined. Then $ea \in EA_{p_i} \iff ea \in DEA_i \vee \forall n_j \in in(n_i) : ea \in G_{p_j}$.*

DEFINITION 4. *Let $n_i = (p_i, DEA_i)$ be a loose-evaluation node in a policy graph and ea be an entity acting in the system where the policy graph is defined. Then $ea \in EA_{p_i} \iff ea \in DEA_i \vee \exists n_j | n_j \in in(n_i) : ea \in G_{p_j}$.*

In other words, an entity e is considered an evaluation authority for a policy contained in a strict-evaluation node $n_i = (p_i, DEA_i)$ (i.e. $e \in EA_{p_i}$), if either a) it satisfies all the policies contained in nodes $n_j = (p_j, DEA_j) \in in(n_i)$ as evaluated by all evaluation authorities $ea \in EA_{p_j}$ (possibly satisfying different requirements) or b) the policy writer entitled it, i.e. $e \in DEA_i$. Similarly, an entity e is considered an evaluation authority for a policy contained in a loose-evaluation node $n_i = (p_i, DEA_i)$ if either a) it satisfies

any of the policies contained in nodes $n_j = (p_j, DEA_j) \in in(n_i)$ as evaluated by evaluation authorities $ea \in EA_{p_j}$ (possibly satisfying different requirements) or b) the policy writer entitled it, i.e. $e \in DEA_i$. PAES was directly inspired by trust management systems and credentials-based models such as the SPKI/SDSI [11], the RT [15] family of languages and SecPAL [9]. However, when used in the context of ERM, these systems have a number of shortcomings. First, their evaluation model relies on a central evaluator to which all the credentials must be presented. Second, the policies used after the first level in the chain are not decided by the data originator who thus loses control over the delegation sequence. In contrast, in PAES entities are only delegated the right to evaluate pre-defined policies and authority over a policy evaluation cannot be further delegated.

6.1 Using PAES in Opportunistic networks

Figure 3 shows the example graph of authorities described in the following. Consider the police MCC receives from one of the tanker companies a short document with details about the liquid leaking into the sewer. The information is immediately disseminated in the area via opportunistic forwarding. The recipients authorised to access the document are: 1) fire fighters (satisfying the *fire_fig* policy), if the general emergency level estimated by the Fire Brigade MCC is evaluated to 5 or more (e.g. if toxic smokes are in the area); 2) Red Cross paramedics (satisfying the *param* policy) if the general emergency level estimated by the Red Cross MCC is evaluated to 3 or more (e.g. if victims' conditions may be worsened by contact with the liquid). Specifying who estimates the emergency levels is important since while organisations can be considered truthful, individuals cannot. Before recipients can access the document, satisfaction of such requirements must be verified. It is therefore necessary that evaluation authorities trusted by the data originator and located close to the recipients evaluate the policy. Such authorities are considered to be Fire Brigade team leaders (satisfying the *team_ld* policy) and medics expert in toxicology (satisfying the *tox* policies), i.e. the individuals most suitable for the evaluation of their subordinates. This is necessary since despite the fact that fire fighters must keep victims or curious people gathering around far from the toxic threat and paramedics need to properly treat patients, information on a possible contamination of the environment must be kept secret at least initially to avoid panic. Similarly, even team leaders and toxicologists must be evaluated to verify that they can be trusted for policy evaluation, i.e. that they satisfy the specified criteria. It is thus decided that team leaders are evaluated by police officers (satisfying the *pol_off* policy), while medics must receive an explicit authorisation by both a Red Cross officer (satisfying the *ro_off* policy) and a police officer. These new authorities can be directly evaluated by the respective MCCs.

The solution we propose to allow policy evaluation in ad-hoc networks is to preemptively evaluate policies and authorise (even unknown) peers when they are met in the crisis area, possibly before they receive any data. Since it is not likely that all rescuers come into communication range with a root of authority (e.g. an MCC) to be evaluated, root of authorities also give authorisation to act as evaluation authorities to peers that satisfy certain characteristics, as specified in a policy graph. In our example, fire fighters may have no contact with the Police MCC. However, the police MCC gives to Police officers the right to evaluate Fire Brigade team leaders. Both officers and team leaders are likely to move in the area and meet more peers. They can then evaluate them and give them either access rights or evaluation rights, according to the defined policy graph. In our example Police officers may meet team leaders and give them evaluation rights over fire fighters. Team leaders are then likely to move and meet fire fighters, to whom they can assign the corresponding access rights. Evaluation or access rights are therefore passed from an authority to a peer whenever they meet for the first time, opportunistically. One of the most important aspects of crisis management is the timeliness of the information. Delays in accessing data caused by the absence of connectivity to an evaluation authority may cause severe problems during the operations. The preemptive policy evaluation we propose is therefore a suitable solution, since its purpose is to allow rescuers to immediately access data with no need for further evaluations (except the usage control policies evaluated locally). Moreover, it does not hinder the evaluation of conditions based on context, as it is described in the following. We explain how rescuers get policies from different agencies in section 7.2.

7. AUTHORITY SPECIFICATION AND DEPLOYMENT

Intuitively, the data categories introduced in section 5 can be associated with multiple usage control policies and evaluation authorities, i.e. to different leaf nodes in a policy graph. An *authority table*, defined statically with the *data category* table, defines the criteria peers must satisfy to become members of authority groups. In other words, the combined information contained in the two tables allows the construction of a policy graph as shown in figure 3. Table 2 shows an example authority table for the Police force in our example. Each row associates an authority group (in the **Target Authority Group**, TAG column) to the requirements (in the **Requirements** columns) peers must fulfill to be assigned to it, and to the authority group(s) authorised to evaluate whether the requirements are satisfied (in the **Evaluation Authority Groups**, EAGs column). It is also possible to specify directly trusted authorities for the evaluation of the policies in the **DEAs** column, i.e. well-known entities that can act as evaluation authorities without being further evaluated. The **Node Type** column specifies whether the evaluation policies are contained either in a loose-evaluation node (i.e. can be evaluated by entities satisfying just one authority policy) or in a strict-evaluation node (i.e. can be evaluated only by authorities satisfying several authority policies). Intuitively, each row represents a non-leaf node of a policy graph. Authority policies are split in two different fields to distinguish between requirements whose evaluation results can be cached for a longer period of time (e.g., credentials with longer expiry deadlines) and context-dependent requirements that are transient and need to be frequently re-evaluated, periodically during data accesses (for usage control policies [21]) and at each meeting . When a peer becomes member of an authority group (see later), it is trusted to periodically verify the satisfaction of the corresponding context-dependent policy. Thus, while a condition for a group G is temporarily not satisfied, the peer does not use the corresponding authority, i.e. it does not evaluate peers for groups G_i for which G is an evaluation authority group.

Each authority group G is also associated to a pair of public and private keys (APK_G and APK_G^{-1}) contained in the **Keydata** column. Possession of a group private key is synonymous with membership in that group. Keys are periodically refreshed (as explained in the next section) before any new crisis event, to avoid reuse of keys distributed during a previous events that may have been compromised. Initially, rescuers' devices contain only group public keys as their membership in a specific authority group must be assigned by a trusted evaluation authority. No one is part of any group a-priori, as keys are distributed during the crisis. Peers are in fact members of different organisations that cannot agree a-priori on any set of

TAG	Semi-permanent Authorisation Requirements	Context-Dependent Requirements	EAGs	DEAs	Node Type	Keydata
fire_fig	role="fire fighter" signed by FireBrigade	riskLevel ≥ 5 signed by (FF_MCC)	team_ld			APK_{fire_fig}
param	role="paramedic" signed by RedCross	riskLevel ≥ 3 signed by (RC_MCC)	tox_po tox_ro		strict	APK_{param}
team_ld	role="team leader" signed by FireBrigade		pol_off	P_MCC		APK_{team_ld}
tox_po	role="toxicologist" signed by NHS $\wedge$ explicitAuthorisation(role="lieutenant")		pol_off			APK_{tox_po}
tox_ro	role="toxicologist" signed by NHS $\wedge$ explicitAuthorisation(role="red cross officer")		ro_off			APK_{tox_ro}
pol_off	role="lieutenant" signed by MetPolice			P_MCC		APK_{pol_off}
ro_off	role="red cross officer" signed by RedCross			RC_MCC		APK_{ro_off}

Table 2: Example of an authority table for the Police Force.

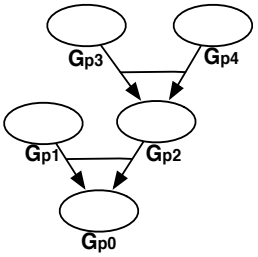

Figure 4: Policy graph with multiple strict evaluation nodes.

keys and policy tree. Moreover, new keys must be generated for each crisis and thus re-distributed, as described in the next section.

Group private keys are initially kept only by roots of authority in special data structures also described in the next section. We discuss how keys are distributed in section 7.2.

7.1 Key Generation

The authority key generation process described is a function of the policy graph defined by an organisation. It can be run either by roots of authority or higher level entities managing them, e.g. the central department for the police forces. For each row in the authority table a new key pair is generated and the corresponding public key is stored in the *Keydata* field. Consider an authority *ea* and a peer *e* being evaluated by *ea* to be assigned to group G_p. If the evaluation is positive, *ea* must be able to give $APK_{G_p}^{-1}$ to *e*. Therefore, every evaluation authority must be given the group private keys of all the authority groups it can evaluate. However, if policy *p* is part of a strict evaluation node, then *ea* must be member of different authority groups to evaluate *p*, e.g. ea_i and ea_j. Therefore *ea* should be able to obtain $APK_{G_p}^{-1}$ and give it to *e* only if authorised by two *different* authorities. This is why for strict evaluation nodes private keys are split in several shares (e.g., with an XOR function) and given to different evaluation authorities specified in the table. However, not only $APK_{G_p}^{-1}$ must be fragmented, but also all the keys or fragments of keys of groups for which G_p is an evaluation authority group. Consider the example depicted in figure 4. To be authorised to evaluate p_0 (i.e. to have $APK_{G_{p0}}^{-1}$), a peer must satisfy p_1 and p_2. Satisfaction of each policy should provide a key fragment of $APK_{G_{p0}}^{-1}$ ($frag_1$ and $frag_2$ respectively) to the peer. To be authorised to evaluate p_2 and provide $frag_2$, a peer must satisfy p_3 and p_4. Also $frag_2$ should then be fragmented, so that each evaluation provides a fragment of $frag_2$, i.e. $frag_3$ and $frag_4$ respectively. Key $APK_{G_{p0}}^{-1}$ should thus be actually split in three fragments $frag_1$, $frag_3$ and $frag_4$. We assume peers do not collude to obtain fragments they do not possess to reconstruct whole keys and impersonate authorities they are not authorised for. We also assume they do not evaluate policies before being members of all the groups they need (e.g. a member of G_{p1} does not give $frag_1$ if not also possessing $frag_2$). This assumption is reasonable since each fragment is issued only to peers that satisfy specific trust requirements (policies). Private keys and their fragments are stored in special data structures of the form $(keyFragment, evalChain)$, where *keyFragment* is a key or fragment of key and *evalChain* is the sequence of authorities that can have the key and pass it to others after a positive policy evaluation. Keeping information on such sequences is necessary as authorities store keys for their direct children, but also for all their descendants. Furthermore, the key fragmentation procedures are also necessary to actually enforce strict-node policies. Algorithm 1 describes the recursive procedure to generate the key pairs and key data structures initially stored by roots of authority. $keyset_n$ is a set of key data structures. The result obtained for the policy graph of our crisis example would then be:

$$
\begin{aligned}
keyset_{pol\_off} = \{&(KP_{pol\_off}^{-1}, pol\_off),\\
&(KP_{fire\_fig}^{-1}, fire\_fig/team\_ld/pol\_off),\\
&(KP_{team\_ld}^{-1}, team\_ld/pol\_off),\\
&(KP_{tox\_po}^{-1}, tox\_po/pol\_off),\\
&(frag_{param,1}, param/tox\_po/pol\_off)\}\\
keyset_{ro\_off} = \{&(KP_{ro\_off}^{-1}, ro\_off),\\
&(KP_{tox\_ro}^{-1}, tox\_ro/ro\_off),\\
&(frag_{param,2}, param/tox\_ro/ro\_off)\}
\end{aligned}
$$

Note that only the key sets for root nodes are generated, since any other information would be redundant. Keys and key fragments belonging to each group can be easily recognized as the group's id is included at the first position in the evaluation chain.

Input: An authority table
forall *TAG elements ea* **do**
  generate random key pair $(APK_{ea}, APK_{ea}^{-1})$;
  AddKey(ea, $(APK_{ea}^{-1}$, ea));
end
Procedure AddKey:
Input: An authority ea, a data structure $KD = (frag, chain)$
if $EAGs(ea) = \emptyset$ **then**
  $keyset_{ea} = keyset_{ea} \cup \{KD\}$;
end
else if *Node Type(ea)* $\neq$ *"strict"* **then**
  forall $a \in EAGs(ea)$ **do**
    $AddKey(a, (frag, chain/a))$;
  end
end
else if *Node Type(ea)* = *"strict"* **then**
  $num = \#EAGs(ea)$;
  /* break the fragment in num new fragments */
  $\{frag_1 \ldots frag_{num}\} = break(frag, num)$;
  $i = 1$;
  forall $a \in EAGs(ea))$ **do**
    AddKey(a, ($frag_i$, chain/a));
    i++;
  end
end

Algorithm 1: Authority keys generation procedure

7.2 Authority Evaluation and Key Distribution

When a crisis situation occurs the agencies involved must first of all ensure that responders (including those from other agencies) are aware of their data protection policies. When operative in the area, roots of authority (in our example police MCCs) first try to localise nearby directly trusted authorities (in our example MCCs from different agencies) and contact them. If communication is possible as we generally assume, they exchange the respective data category tables and authority tables. Even if direct communication is not possible, officials of the different agencies would typically have a face to face meetings (e.g., briefing) to organise the operations. Data could then be exchanged. Thus, initially all MCCs are considered to know the policy graphs of all the involved agencies. From here, the tables can be disseminated to all responders.

At this point the root of authorities begin sending to each nearby directly trusted authority the key sets for the groups they are authorised to evaluate. In our example the police MCC is both a root of authority and a directly trusted authority for the group *pol_off*. Thus it inserts $keyset_{pol\_off}$ in the *keydata* column of the authority table at row *pol_off*. Since the Red Cross MCC is a directly trusted authority for the group *ro_off*, the police MCC sends it $keyset_{ro\_off}$ (via the same channel used to exchange the tables). The generation of key sets to pass to directly trusted authorities of not-root nodes is the same as for authorities defined by characterisation and is described in the following.

Starting from directly trusted authorities, group membership is opportunistically evaluated as follows (figure 5). Whenever two peers e_i and e_j come into communication range, unless they have previously met during the current crisis and have not received new keys in the mean time, exchange the list of authorities they are authorised to evaluate (according to the authority groups they are currently members of). Amongst these, each peer chooses the authority groups they want to become members of. The choice can be made either by the human user or automatically, according to a pre-defined strategy. The two peers exchange the list of chosen authorities along with the certificates needed for the evaluation of the corresponding policies. Each peer evaluates the policies (both semi-permanent and context-dependent policies) with the received certificates and available context parameters. For each satisfied policy, i.e. for each new group the evaluated peer must become member of, a key set is created containing all the key data structures available whose evaluation chain contains that group. The key sets are then exchanged and used to update the authority table. Key fragments are also merged whenever possible to obtain a new fragment or a whole key.

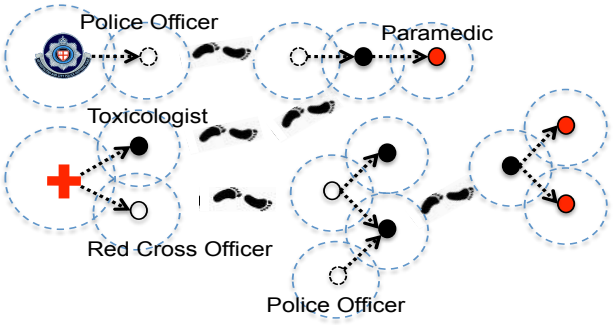

Figure 5: The opportunistic dissemination of authority keys.

Consider a police officer member of group *pol_off* meeting a fire brigade team leader. After evaluating the policy for group *team_ld*, the police officer's device will generate $keyset_{team\_ld} = \{(KP_{fire\_fig}^{-1}, fire\_fig\ /team\_ld\ /pol\_off), (KP_{team\_ld}^{-1}, team\_ld\ /pol\_off)\}$ collecting the key data structures containing *team_ld* in their chains. The fire brigade team leader will then receive the key set and use it to fill row *team_ld* in its authority table. With this mechanisms, authority keys are distributed in the crisis area only according to the requirements specified in the policy graph. When data is received and must be decrypted, if the usage control policy for the data category is part of a loose-evaluation node, then the private key of the corresponding evaluation authority can be used to decrypt the symmetric key k_r. If the usage control policy is part of a strict-evaluation node, then multiple decryptions with the private keys of the different authority groups are performed to decrypt k_r (with the encryption order being the same as the order of the authorities in the EAGs field in the authority table). Note that using a mechanism such as attribute-based encryption (ABE) [14] is not a viable solution as it would require all the responders to receive credentials and keys from the same authority, even if members of different agencies. Moreover, ABE would prevent context-dependent conditions to be evaluated, since it provides decryption keys only on the base of statically defined credentials.

7.2.1 *Context Information and Revocation Lists*

Both authority and usage control policies specify conditions over user and context attributes. We distinguish between attributes that can be checked locally such as location or time and attributes that need to be checked remotely such as the the general emergency level estimated by an external entity. In a fully connected sce-

nario, the latter would be verified by connecting to trusted third parties to obtain up-to-date context information (or to check revocation). In our scenario this is not possible. We therefore propose two solutions to partially address this problem. First, revocation lists and up-to-date signed context information can be disseminated from the MCCs via opportunistic forwarding. Each organisation may also assign well-known rescuers deployed in the crisis area with the right/duty to make decisions upon specific context values (e.g. again, the emergency level). Other rescuers may then update their local knowledge with the most recent received information. Second, MCCs may perform re-keying operations at fixed time intervals or whenever important context information change. This would result in the dissemination of a new authority table and of new key sets together with the updated context information. Note that re-keying does not correspond to a revocation of the already obtained access rights, but would only prevent recipients from obtaining new data without a new evaluation.

8. SIMULATION

The PAES-driven key distribution creates an overhead on data distribution and data access. Responders acting in the crisis area may receive useful data that they cannot access because no valid authority has evaluated them yet. It is therefore important to verify that such an overhead does not create undue burden and does not significantly hinder rescue operations. We have simulated the movements and communications of rescuers in the crisis scenario described in section 3 using a derivative of the GUS (Geographic Urban Simulator) [7], which is capable of simulating large systems of interacting peers within urban settings. The GUS is built atop of the Java in Simulation Time (JiST) framework [8], which provides a discrete event simulation base. Applications can be written in Java and subsequently simulated in the environment, merging mobility traces with application logic. We performed tests that considered the dissemination of a toxic threat data originally sent from the police MCC, using the example policy introduced in section 6. The aim of the simulation was to evaluate the rates at which keys and data are distributed, i.e. to verify that delay between the receipt of the data and the receipt of the key is negligible.

8.1 Mobility Model

The simulation area is divided into five different sub-areas:

- The accident area, i.e. the 100m radius perimeter centred on the car crash location;
- The evacuation area, i.e. the area external to the incident perimeter and limited by a 400m radius perimeter with the same centre. This is the area used by rescuers to evacuate victims and to move towards the respective MCCs;
- The Police Mobile Command Centre area, the Fire Brigade Mobile Command Centre area and the Red Cross Mobile Command Centre area, placed respectively north-west, north-east and south on the evacuation area perimeter.

Rescuers move according to the following patterns. Fire fighters move in teams of three persons, including a team leader. They first move from the Fire Brigade MCC area towards the accident area. We simulate the search for victims, fires or collapsing structures inside the area with a Random WayPoint mobility model. For each reached destination they wait 120 seconds and then keep moving to a new one. Policemen follow the same pattern as fire fighters. Police officers move back and forth from the Police MCC area to a random point in the accident area, to simulate a control-and-report activity. Paramedics and toxicologists follow the same pattern to simulate rescue and evacuation of victims. All the rescuers move at 1.1 m/s and broadcast messages every second. Although not ideal and realistic, a partially random way point model is actually the most conservative choice for peer mobility. Obstacles, forced routes (e.g. roads) and attraction points in the area constrain in fact the rescuers to move on the same paths, increasing the number of times they meet and thus actually the performance of the protocol. Our simulation is thus based on a situation that is more demanding with respect to the protocol than a real world setting.

Rescuers	10m	20m	30m
30	1386	979	887
75	1040	943	883
150	869	804	756

Table 3: Mean key receipt times (seconds).

Rescuers	10m	20m	30m
30	1333	891	759
75	1197	1117	900
150	1031	864	760

Table 4: Mean data delivery times (seconds).

The simulation was executed multiple times, each time varying parameters, including the number of rescuers involved and the communication range of their devices. In particular, we considered 30, 75 and 150 rescuers (evenly divided between the three participating organisations) with 10m, 20m and 30m communication ranges. Rescuers were divided as follows: Police officers represent 20% of the Police force; toxicologists and Red Cross officers represent 9% and 20% of the Red Cross force respectively; fire fighters team leaders represent 33% of the Fire Brigade force.

8.2 Results and Evaluation

To measure the effectiveness of PAES-driven key distribution, we compare message delivery with the key receipt time for all rescuers in the system. In the opportunistic networking context, message delivery time refers to the elapsed time taken for a message to be received by all peers within the system. The delivery ratio represents the total fraction of rescuers which have successfully received a message at a specific instance in time. In contrast, key receipt time does not require a message to be received, rather it measures how quickly peers receive authority group keys based on the policy graph available.

The results show a logistic function similar to that found in closed systems. If we refer to the elapsed experiment time as t, then we see a logistic shaped delivery function where the function tends towards 1.0 for increasing t ($\lim_{t\to\infty} f(t) = 1.0$). Increasing the communication range was seen to improve the derivatives of both message delivery and key receipt, as well as increasing the peer population, as expected.

Tables 3 and 4 show the mean key receipt times and mean data delivery times (in seconds) obtained varying the devices' communication range and number of rescuers. Figure 7, 8 and 9 depict the logistic functions for three different configurations. Only with a communication range set at 10m is data delivery faster than key receipt for most of the simulations, as confirmed by the mean results plotted in Figure 6. The results indicate that keys can be received

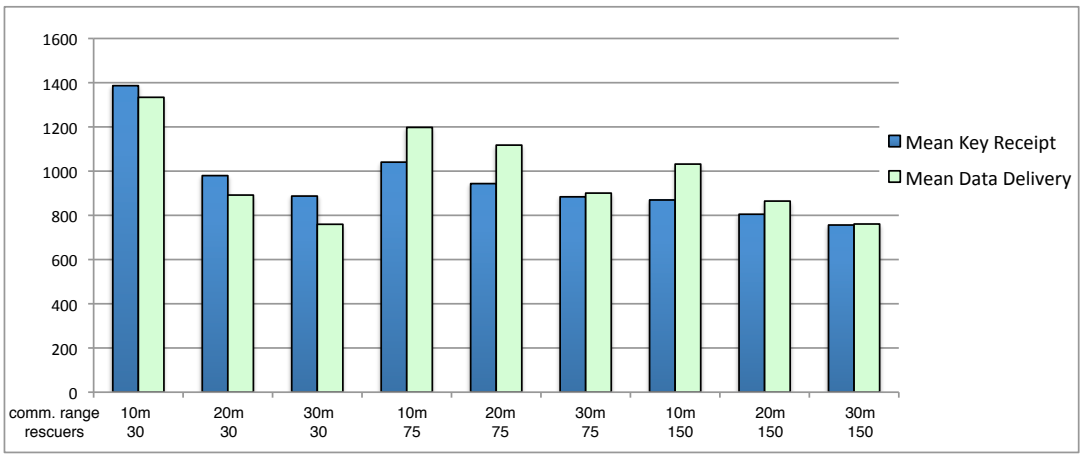

Figure 6: Mean key receipt times and data delivery times with different parameters.

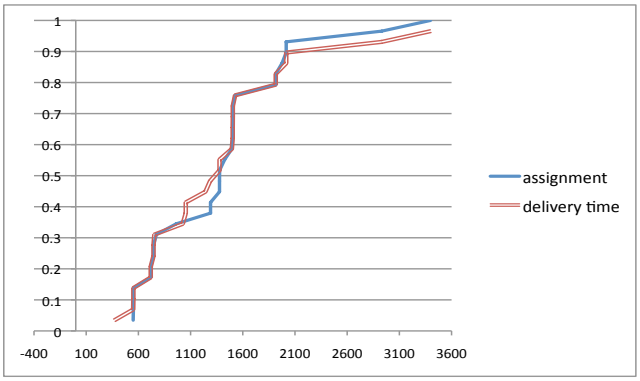

Figure 7: Key receipt and data delivery ratio (30 rescuers, 10m communication range).

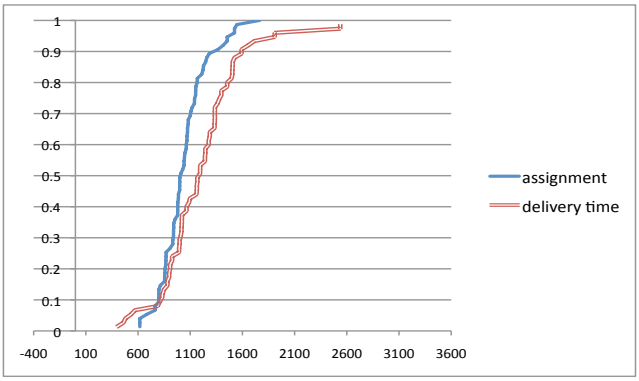

Figure 8: Key receipt and data delivery ratio (75 rescuers, 10m communication range).

before accessing the data, so the overhead posed by the protocol is negligible. We can explain such results as follows. As the rate of increase in performance fundamentally depends on the multiplicity of broadcast messages to neighbouring rescuers (i.e. the number of peers and the communication range), the rate of data delivery and the rate of key receipt are affected by mobility and by the complexity of the policy graph (see later). The mobility model we proposed actually favours key distribution w.r.t. data dissemination. Policies are in fact evaluated starting from all MCC areas (i.e. wherever directly trusted authorities are) and thus keys are distributed from three different locations (and by three different organisations) towards the centre of the crisis area. Also, having rescuers from the same organisation moving in teams eases the evaluation process, as keys are directly passed from team leaders to their subordinates. The data considered for distribution is instead initially disseminated only from the Police MCC area and only by police units. To be received by all peers, the data must not only reach the centre of the area, but also the more distant cooperating MCCs.

We define the complexity of a policy graph as the inverse of the average probability for peers satisfying certain requirements to meet authorities that can evaluate them, i.e. the probability to actually obtain membership to the corresponding authority groups. Several factors contribute to the complexity of a policy graph: 1)

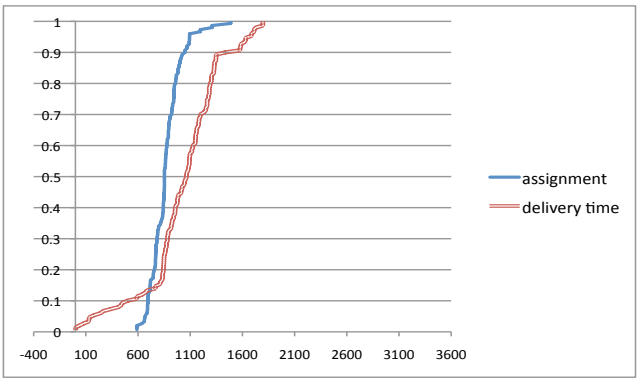

Figure 9: Key receipt and data delivery ratio (150 rescuers, 10m communication range).

the depth of the policy graph, i.e. the minimum number of evaluations that must be performed before a peer can actually become an authority; 2) the width of the policy graph, i.e. the number of different disjoint groups peers can be members of; 3) the number

of strict-evaluation nodes, i.e. the number of peers that must meet more than one authority to be evaluated. The complexity of the policy graph also depends on the deployment scenario. A high probability that context-dependent conditions are verified during the crisis and an high number of peers satisfying policies at high level-nodes speeds up the evaluation process. This is why an important aspect of the mobility model is the percentages with which rescuers are divided among the several groups. Intuitively, weakening the requirements to be an evaluation authority increases the number of policy evaluations and thus speeds up the key distribution process. Similarly, stricter requirements mean fewer authorities are present in the area, thus resulting in a higher key distribution delay.

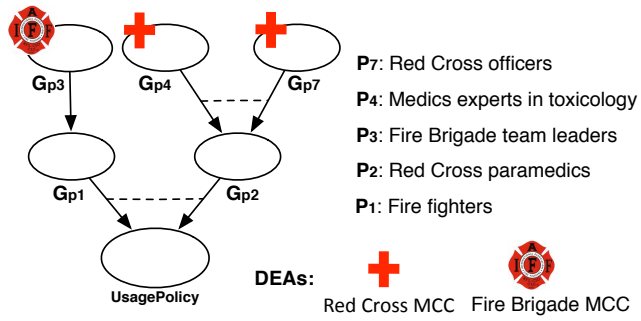

Figure 10: A simpler policy graph.

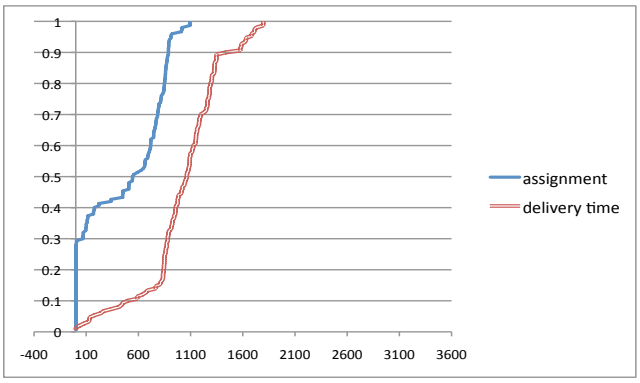

Figure 11: Key receipt and data delivery ratio (150 rescuers, 10m communication range) with a simpler policy graph.

Building a more complex policy graphs for a specific scenario impacts negatively on the performance of the system (shifting the resultant logistic functions) and thereby influences the capacity of the system to deliver keys. Simulations run with the simpler policy graph depicted in figure 10 confirm this hypothesis. The results in figure 11 clearly show a shift to left of the logistic function for the key receipt rate. An optimal pattern may match an optimal scenario and hence a balance of mobility and policy graph is considered as a question for future work, where simpler policy graphs may favour more random-like mobility patterns and vice versa. The policy graph we used for our experiments is not an optimal one and is probably more complex than required in most emergency scenarios. Nevertheless, the protocol overhead shown in the simulations was negligible.

9. CONCLUSIONS

The solution proposed aims to simultaneously satisfy conflicting requirements: 1) data dissemination in disconnected networks, 2) timely data access and 3) data protection. We addressed all of them by integrating the Policy-based Authority Evaluation Scheme with oppnets. The results of the simulations showed that the overhead imposed by PAES broadly varies with the defined policy graph. Stricter policies come at the cost of a performance decrease. However, we expect policies to be reasonably simple and thus to not impede the rescue operations. The example scenario showed how keys for data access are mostly received before the data itself, thus granting data availability and timely accesses. This means that while responding to a crisis, responders may immediately use the data received as they already possess the necessary decryption keys. Moreover, since crisis management is a delicate activity that must often rely on human judgement rather then automated procedures, users entrusted with specific authorities may also be allowed to manually give users memberships to certain authority groups, effectively providing a *break-the-glass* override. Consider for example the case where a seriously wounded victim is being assisted by a fire fighter, but no paramedic is available nearby. Assume that the fire fighter has access to the victim's information and that an uninjured victim of the accident is willing to lend assistance, attesting he is a medic. In this situation, despite the fact that the medic does not satisfy any authority policy, the fire fighter might decide to give him access to the victim's data. Break-the-glass policies would probably also increase the performances of the key dissemination process, as meetings between peers may result in a key exchange by means of a manual overriding of the normal authority policies that would not permit it. Future work may investigate this aspect of the system. We further plan to integrate the proposed solution with a framework for hybrid networks such as Haggle.

10. ACKNOWLEDGMENTS

We acknowledge financial support from the EC Consequence project (Grant Agreement 214859). We are grateful to the reviewers for their detailed and constructive comments that helped improve this paper.

11. REFERENCES

[1] Introduction to encryption key management for public safety radio systems, 2001. Public Safety Wireless Network Program. Security Issues Report: Encryption Key. Accessed December 2009. URL: http://www.safecomprogram.gov/NR/rdonlyres/7C31664D-5B0B-4128-A85B-DC79B1D734ED/0/Security_Issues_Analysis_Report.pdf.

[2] Liquid machines and microsoft windows rights management services (rms): End-to-end rights management for the enterprise, 2006. Accessed September 2009. URL: http://www.cmdsolutions.com/pdfs/LiquidMachines%20Windows%20RMS%20Business%20White%20Paper%20FINAL%20060213.pdf.

[3] N. Aschenbruck, M. Frank, P. Martini, and J. Tölle. Human mobility in manet disaster area simulation - a realistic approach. In *29th Annual IEEE Conf. on Local Computer Networks (LCN)*, pages 668–675, Tampa, Florida, U.S.A., November 2004.

[4] N. Aschenbruck, E. Gerhards-Padilla, M. Gerharz, M. Frank, and P.Martini. Modelling mobility in disaster area scenarios. In *Proc. of the 10th Int. Symp. on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 4–12, Chania, Crete Island, Greece, October 2007.

[5] M. J. Atallah, K. B. Frikken, and M. Blanton. Dynamic and efficient key management for access hierarchies. In *12th ACM Conf. on Computer and Communications Security (CCS)*, pages 190–202, New York, USA, November 2005. ACM.

[6] Authentica. Enterprise rights management for document protection, 2005. White Paper.

[7] R. Ball and N. Dulay. Approximating travel times using opportunistic networking. In *2nd IEEE Int. Workshop on Opportunistic Networking*, pages 844–849, May 2009.

[8] R. Barr, Z. J. Haas, and R. van Renesse. Jist: Embedding simulation time into a virtual machine. In *8th Int. Symp. on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, page 16, Montreal, Quebec, Canada, 2003.

[9] M. Becker, Fournet C., and A. Gordon. Secpal: Design and semantics of a decentralized authorization language. Technical Report MSR-TR-2006-120, Microsoft, 2006.

[10] E. Bertino, E. Ferrari, F. Paci, and L. Parasiliti Provenza. A system for securing push-based distribution of xml documents. *Int. J. Inf. Secur.*, 6(4):255–284, 2007.

[11] D. Clarke, J. Elien, M. Ellison, C.and Fredette, A. Morcos, and R. L. Rivest. Certificate chain discovery in spki/sdsi. *J. of Computer Security*, 9(4):285 – 322, 2001.

[12] M. Conti and S. Giordano. Multihop ad hoc networking: The reality. *IEEE Communications Magazine*, 45(4):88–95, April 2007.

[13] Office for Interoperability and Compatibility Department of Homeland Security. Public safety statement of requirements for communications & interoperability, 2006. Accessed December 2009. URL: http://www.safecomprogram.gov/SAFECOM/.

[14] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proc. 13th ACM Conf. on Computer and Communications Security*, pages 89–98, New York, NY, USA, October 2006. ACM.

[15] N. Li, J.C. Mitchell, and W. H. Winsborough. Design of a role-based trust management framework. In *Proc. of the 2002 IEEE Symp. on Security and Privacy*, pages 114–130. IEEE Computer Society Press, May 2002.

[16] Q. Li and D. Rus. Sending messages to mobile users in disconnected ad-hoc wireless networks. In *6th ACM/IEEE Int. Conf. on Mobile Computing and Networking (MOBICOM)*, pages 44–55, Boston, Massachusetts, USA, August 2000.

[17] L. Lilien, A. K. Gupta, and Z. Yang. Opportunistic networks for emergency applications and their standard implementation framework. In *26th IEEE Int. Performance Computing and Communications Conf. (IPCCC)*, pages 588–593, New Orleans, Louisiana, USA, April 2007.

[18] L. Lilien, Z. Huma Kamal, V. Bhuse, A. Gupta, and Wise (wireless Sensornet Lab. Opportunistic networks: The concept and research challenges in privacy and security. In *Proc. NSF Int. Workshop on Research Challenges in Security and Privacy for Mobile and Wireless Networks (WSPWN)*, Miami, Florida, U.S.A., March 2006.

[19] Microsoft. Technical overview of windows rights management services for windows server 2003, 2005. White Paper. Accessed December 2009. URL: http://www.safecomprogram.gov.

[20] M. Casassa Mont, S Pearson, and P. Bramhall. Towards accountable management of identity and privacy: Sticky policies and enforceable tracing services. In *14th Int. Workshop on Database and Expert Systems Applications (DEXA)*, pages 377–382, Prague, Czech Republic, September 2003.

[21] J. Park and R. S. Sandhu. The $UCON_{ABC}$ usage control model. *ACM Trans. Inf. Syst. Secur.*, 7(1):128–174, 2004.

[22] J. Park, R. S. Sandhu, and J. Schifalacqua. Security architectures for controlled digital information dissemination. In *16th Computer Security Applications Conf. (ACSAC)*, pages 224–, New Orleans, USA, December 2000. IEEE Computer Society.

[23] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan. Prioritized epidemic routing for opportunistic networks. In *Proc. of the 1st int. MobiSys Workshop on Mobile Opportunistic Networking (MobiOpp)*, pages 62–66, New York, USA, March 2007. ACM.

[24] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *24th Int. Conf. on the Theory and Applications of Cryptographic (EUROCRYPT)*, volume 3494 of *LNCS*, pages 457–473, Aarhus, Denmark, May 2005. Springer.

[25] E. Scalavino, V. Gowadia, and E. C. Lupu. Paes: Policy-based authority evaluation scheme. In *DBSec*, pages 268–282, Montreal, Quebec, Canada, July 2009.

[26] J. Scott, P. Hui, J. Crowcroft, and C. Diot. Haggle: A Networking Architecture Designed Around Mobile Users. In *IFIP Conference on Wireless On demand Network Systems (WONS)*, pages 78–86, Les Menuires, France, January 2006.

[27] Avoco Secure. Choosing an enterprise rights management system: Architectural approach, 2007. Accessed December 2009. URL:www.windowsecurity.com/uplarticle/Authentication _and_ Access_Control/ERM-architectural-approaches.pdf.

[28] A. Seth and S. Keshav. Practical security for disconnected nodes, 2005. Centre for Applied Cryptographic Research (CACR): Technical Report.

[29] J. Su, J. Scott, P. Hui, J. Crowcroft, E. de Lara, C. Diot, A. Goel, M. Lim, and E. Upton. Haggle: Seamless networking for mobile applications. In *Ubiquitous Computing, 9th Int. Conf. (Ubicomp)*, pages 391–408, Innsbruck, Austria, September 2007.

[30] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. 2000. Duke Technical Report CS-2000-06.