Secure & Trusted Communication in Emergency Situations

Antonis Michalas, Menelaos Bakopoulos and Nikos Komninos
Athens Information Technology,
Algorithms & Security Group,
Athens, Greece
Email: amic,mbak,nkom@ait.edu.gr

Neeli R. Prasad
University of Aalborg,
Department of Electronic Systems,
Aalborg, Denmark
Email: np@aau.dk

***Abstract*—In this paper we propose SETS, a protocol with main aim to provide secure and private communication during emergency situations. SETS achieves security of the exchanged information, attack resilience and user's privacy. In addition, SETS can be easily adapted for mobile devices, since field experimental results show the effectiveness of the protocol on actual smart-phone platforms.**

***Index Terms*—Security; Privacy; Trust; Emergency Handling; Crisis Management;**

I. Introduction

In practice, in emergency situations such as earthquakes, wildfires and floods, all privacy and security concerns fade into the background. There are many emergency situations, however, where these aspects are necessary. For example, the medical history of patients should be available in comprehensible format exclusively to medical first responders in order to preserve the privacy of the patients.

The use of the Internet and specifically the use of Ad Hoc Networks can be an indispensable tool for crisis management and emergency response teams. Furthermore, there is a lack of research regarding the security and privacy of the data sent between first responders and the command post. The fact that sensitive information is made available without adequate protection from malicious users can be potentially very damaging.

For the purpose of establishing a secure and trusted communication protocol, this paper focuses on three aspects. Firstly, the confidentiality of the exchanged information. Secondly, the resilience of the system to malicious attacks and thirdly, on the privacy of the users.

For the first aspect, we propose a protocol that can be used for authenticating the user who sends a request while at the same time keeping message contents secret to everybody but the receiver. Authentication is needed in order to prevent fake requests from unauthorized users. This way we can ensure that the resources of a team will only be spent on legitimate requests - key to effective treatment of emergency situations and crisis management.

For the second aspect, we analyze our protocol against a multitude of attacks that is important in designing new protocols regarding emergency handling.

Regarding the third aspect, each user that informs an authority about an emergency, needs to send data that contain sensitive information, such as the location and the time of the event. Thus, any authority that receives these data will have access to this specific information. For this reason, we manage to protect the privacy of the sender of a message by not revealing her real identity.

Furthermore, the analysis is backed by extensive experimental results demonstrating the protocol's validity and efficiency.

In Section II, we review schemes that provide secure communication in emergency situations. In Section III, we present the problem statement and we define the basic terms used throughout the paper. In Section IV, we present SETS, while in Section V we provide a security discussion. In Section VI, we present experimental evidence that shows the effectiveness of our protocol and Section VII concludes this paper.

II. Related Work

Although there are many emergency handling protocols, few of them deal with the problem of securing the communication between users. Furthermore, there is a stark absence of schemes that provide (partial) privacy in such environments.

T. Levin *et al.* in [1] proposed a system for secure distribution and control of sensitive information during crisis. Authors, suggested two significant enhancements to emergency information management: transient-memory encryption for secure data storage and new hardware instructions to support distributed emergency state management. Even though the proposed protocol is considered as secure, it has a great disadvantage, it is not easily applicable since, the existence of a specific device called *E-device* is mandatory.

In [2] F. Dozer *et al.* focus on vehicle-to-infrastructure communication for active safety, particularly between vehicles and traffic lights. Their protocol assumes that emergency vehicles send control messages to the traffic lights to actively influence their current state. To reduce the number of vehicles potentially hindering a free way for the emergency vehicle, this communication allows each intersection to optimize the traffic flow. Consequently, emergency vehicles can reach their destination quicker and safer. Furthermore, they argue that most of the services offered by road side units require a secure authentication mechanism in order to trust the information and prevent unauthorized use. This authentication process does not reveal the identity of the vehicles involved.

III. Problem Statement & Definitions

In an emergency handling situation, we distinguish three different roles, each of which is instrumental in the smooth and efficient intervention and termination of a situation.

Definition 1. *An authority*[1] *(a_i) is an organization that coordinates emergency response in a given context, such as the Department of Homeland Security, non-governmental organization, or a selected enterprise department. The set of authorities is denoted as $A = \{a_1, a_2, \ldots, a_n\}$.*

Definition 2. *First responders (FR) are members of an agency/organization and asked to provide their services at a scene of a disaster with main goal the successful resolution of a situation. The set of first responders for the authority a_i is denoted as $FR_i = \{r_1^{a_i}, r_2^{a_i}, \ldots, r_m^{a_i}\}$.*

Definition 3. *Third Party (TP) is one or more data providers that supply emergency information. Emergency information is information designated to be available to emergency authorities and first responders, which they may not have been vetted or cleared to see. The set of third parties is denoted as $TP = \{p_1, p_2, \ldots, p_l\}$.*

Definition 4. *A certification authority (CA) is an organization that is responsible for issuing a credential to a new user p_t. Apart from that, CA validates to an authority a_i if p_t is legitimate or not.*

In this work we are concerned with the following problem:

Problem Statement: *An authority $a_i \in A$, receives a request from a third party p_t. First, a_i has to check that p_t is valid. In order to do so, a_i asks from p_t to prove that she is indeed legitimate. If the validation fails, a_i does not proceed with the request and thus she manages to protect her resources from unauthorized users/fake requests. After the successful validation, p_t needs to send back to a_i the emergency information in a secure way and then a_i will forward the request to the appropriate FR's*[2]*. The problem is to create a protocol that will guarantee the following: i)The identity of p_t will not be revealed to anyone, ii)Authority a_i will not be deceived by malicious users who sends fake requests and iii)Emergency information that p_t will send to a_i will be protected from unauthorized users and only a_i will be in position to disclose the information to the appropriate first responders.*

Anonymous credentials systems, as described in [3], are widely used and allows users to prove certain statements about themselves by making use of certain attributes. This allows them to inquire valuable services, while at the same time hiding the user's identity.

For the following sections, we assume that we are working in a group G_q of prime order q, such that computing discrete logarithms in G_q is infeasible. In addition to that, each authority (a_i, $i \in [1, n]$) has generated a public/private key pair (k_{a_i}/K_{a_i}). The private key is kept secret, while the public key is shared with the rest of the participants.

Based on [4] we give some basic definitions regarding nyms generation, issuing and validation of credentials.

Definition 5 (Pseudonym Generation)**.** *Nym generation (NG) is a secure interactive procedure between a user p_t and an authority a_i. The private input of p_t and a_i is K_{p_t} and K_{a_i} respectively, while their common input is k_{a_i}. The execution of NG has three outputs. First, a nym ($N_{(p_t,a_i)}$) which is the pseudonym of p_t with a_i and is common for both parties, a private output $N_{s,a_i}^{p_t}$ for p_t and a private output $N_{s,p_t}^{a_i}$ for a_i.*

Informally, in a protocol for signing a value α, there are two parties. A user who queries for a signature and a signer with a pair of public/private keys.

Definition 6 (Signatures)**.** *Lets suppose that p_t queries a signature from a_i. The signature scheme $Sign(k_{a_i}, K_{a_i}, m)$ is the procedure that on input of a key pair k_{a_i}/K_{a_i} and a message m, outputs a signature $s \in \sigma_{k_{a_i}}(m)$. Additionally, $Verify(k_{a_i}, m, s)$ is the verification algorithm.*

The common input to such a protocol is a commitment C which is a basic component of many cryptographic protocols. The main idea behind commitment schemes is that one party p_t commits a value β to another party a_i in such a way that a_i has no idea what β is. Then, p_t can reveal β and a_i can verify that this is indeed the value to which a_i committed [5].

Definition 7 (Commitment Scheme)**.** *Suppose we are given a commitment scheme $(Commit, Check)$ where $Commit(\beta, r)$ is the commitment algorithm that produces a commitment to β with randomness r. If $c = Commit(\beta, r)$ then $Check(c, \beta, r)$ is the verification algorithm for the commitment to β.*

IV. SETS Protocol

In this section we present our main protocol (SETS) which is based on [4]. In our model each user p_t must first register with the *CA* by revealing her true identity (public key k_{p_t}), as well as demonstrating the possession of a valid private key (K_{p_t}). Then, CA issues a credential to p_t which can be used for further communication with an authority $a_i \in A$. Upon a request, a_i checks the validity of p_t by communicating with CA. If p_t is successfully authenticated, she sends the emergency information in an encrypted form in order to protect it from unauthorized access. Upon reception, a_i is responsible for forwarding the request to the appropriate FR's.

The rest of the section is divided into the following parts: i) Registration - Issue of a credential for a new user, ii) Validation of the user who sends a request to an authority and iii) Securing the messages that contains emergency information.

A. Registration (Fig. 1)

A user p_t will have to first install the application on her mobile device. During installation, p_t communicates with the CA where she is prompted to enter a username and a password ($pass_{p_t}$). Then, automatically and on behalf of p_t, a public/private key pair ($k_{p_t} = g^x$/$K_{p_t} = x$) is generated and saved on the mobile device of p_t. In order to protect this information, the private key is encrypted with $pass_{p_t}$. So, instead of keeping k_{p_t}/K_{p_t} on client's device, $k_{p_t}/\{K_{p_t}\}_{pass_{p_t}}$ is calculated and

[1] An authority is also known as *Emergency Operation Center (EOC)*.

[2] The problem of finding which element(s) of FR_i are the most appropriate for handling a specific event is beyond the scope of this paper.

saved for later use. For the registration, p_t reveals k_{p_t} to CA which uses K_{CA} to sign k_{p_t} and outputs $s_{p_t,CA} \in \sigma_{CA}(k_{p_t})$.

At this point, p_t needs to establish a nym with CA, so she generates a random string r that corresponds to her private output $N^{p_t}_{s,CA}$, and computes $N_{(p_t,CA)} = Commit((k_{p_t}, K_{p_t}), r)$. A user's nym is formed by taking a random base a, such that the user does not know $log_g a$ and raising it to the power of x. This means that nyms are tied to K_{p_t} and thus we can verify that when a credential is issued it will not be valid for any other secret than x. More precisely, the following steps are taking place:

1) $p_t \xrightarrow{(g,g^x)} CA$
2) $CA \xrightarrow{a} p_t$: CA picks $\tilde{r} \in_R \mathbb{Z}_q^*$, sets $a = g^{\tilde{r}}$ and send it to p_t.
3) p_t compute $b = a^x = g^{\tilde{r}^x} = g^{\tilde{r}x}$.
4) $p_t \xrightarrow{(A,B)} CA$: p_t choose $r \in_R \mathbb{Z}_q^*$, and sets $A = g^r, B = \tilde{g}^{r}$[3].
5) $CA \xrightarrow{c} p_t$: CA picks $c \in_R \mathbb{Z}_q^*$ and sends it to p_t.
6) $p_t \xrightarrow{y=r+cx \ mod \ q} CA$
7) CA checks that $g^y = g^{(r+cx)} = g^r g^{cx} = Ag^{x^c} = Ah^c$ and $\tilde{g}^y = B\tilde{h}^c$.

At the end of step 7, p_t has established a nym $N_{(p_t,CA)}$ with CA.

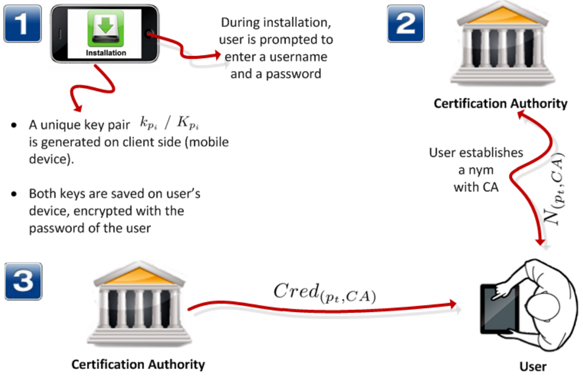

1
Installation
During installation, user is prompted to enter a username and a password
A unique key pair k_{p_i} / K_{p_i} is generated on client side (mobile device).
Both keys are saved on user's device, encrypted with the password of the user
2
Certification Authority
User establishes a nym with CA
$N_{(p_t,CA)}$
3
Certification Authority
$Cred_{(p_t,CA)}$
User

Fig. 1. Generation of public/private keys and Acquisition of a Credential

The final step of the registration procedure is the generation and issuing of a credential from CA to p_t. For this purpose, we will use the protocol Γ for producing a blind transcript and the protocol for issuing a credential that is described in [4].

Protocol Γ:

1) $p_t \xrightarrow{(A,B)} CA$: p_t picks $\tilde{r} \in_R \mathbb{Z}_q^*$, sets $A = g^r, B = \tilde{g}^r$ and send it to CA.
2) $CA \xrightarrow{c} p_t$: CA picks $\alpha, \beta \in_R \mathbb{Z}_q^*$, sets $A' = Ag^\alpha h^\beta, B' = (B\tilde{g}^\alpha \tilde{h}^\beta)^\gamma$, calculates $c = H(A' + B') + \beta \ mod \ q$ and send it to p_t.
3) $p_t \xrightarrow{y=r+cx \ mod \ q} CA$
4) CA checks that $g^y = Ah^c$ and $\tilde{g}^y = B\tilde{h}^c$.

$$\begin{aligned} g^{(y+\alpha)} &= g^r g^{cx} g^a \\ \Leftrightarrow g^{(y+\alpha)} &= Ag^{cx} A' A^{-1} h^{-\beta} \\ \Leftrightarrow g^{(y+\alpha)} &= A' h^c h^{-\beta} \\ \Leftrightarrow g^{(y+\alpha)} &= A' h^{c-\beta} \end{aligned} \qquad (1)$$

We know that:

$$\begin{aligned} g^y &= g^y g^\alpha g^{-\alpha} \\ \overset{(1)}{\Leftrightarrow} g^y &= A' h^{c-\beta} g^{-\alpha} \\ \Leftrightarrow g^y &= A' h^c h^{-\beta} A'^{-1} A h^\beta \\ \Leftrightarrow g^y &= Ah^c \end{aligned} \qquad (2)$$

5) $CA \xrightarrow{T} p_t$: $T = ((A', B'), H(A', B'), y + \alpha)$.

CA generates a private credential key $(s_1, s_2) : s_1, s_2 \in \mathbb{Z}_q^*$ and publishes the public credential key $(g, h_1, h_2) : h_1 = g^{s_1} \, mod \, p$ and $h_2 = g^{s_2} \, mod \, p$. The steps for issuing a credential are described below:

1) $CA \xrightarrow{(A,B)} p_t$: CA calculates $A = b^{s_2}$ and $B = (ab^{s_2})^{s_1}$.[4]
2) p_t picks $\gamma \in \mathbb{Z}_q^*$.
3) Obtain transcript T_1 by running Γ.
4) Obtain transcript T_2 by running Γ.
5) Credential $Cred_{(p_t,CA)} = (a^\gamma, b^\gamma, A^\gamma, B^\gamma, T_1, T_2)$ is issued.

B. Contacting an Authority

After p_t has established a credential with CA, she can contact any authority $a_i \in A$ in order to make a request. As a first step, p_t will have to prove to a_i that she is a legitimate user. To do so, she picks a random number r_{a_i} calculates $f_{r_{a_i}}(Cred_{(p_t,CA)}) = (a^{r_{a_i}}, b^{r_{a_i}}, A^{r_{a_i}}, B^{r_{a_i}})$ and generates a session public/private key pair k'_{p_t}/K'_{p_t}. Then calculates $\left\langle \left\{ \{f_{r_{a_i}}(Cred_{(p_t,CA)}), timestamp\}_{k_{CA}}, k'_{p_t} \right\}_{k_{a_i}} \right\rangle$ and sends it to a_i . Upon reception, a_i decrypts the received message with K_{a_i}, keeps k'_{p_t} and sends $\langle \{ f_{r_{a_i}}(Cred_{(p_t,CA)}, timestamp) \ \rangle_{k_{CA}}$ to CA. CA now decrypts the received message with K_{CA} and checks whether $f_{r_{a_i}}(Cred_{(p_t,CA)})$ corresponds to a legitimate user or not.

C. Protecting Emergency Information

If CA validates that p_t is a legitimate user, it sends to a_i a confirmation message along with the received time stamp increased by one and encrypted with k_{a_i}. Upon reception, a_i decrypts the message with K_{a_i}, finds the time stamp, generates a one time public/private key pair (k'_{a_i}/K'_{a_i}) and sends to p_t the following $\left\langle \{k'_{a_i}, timestamp + 1\}_{k'_{p_t}} \right\rangle$. Upon reception, p_t finds k'_{a_i} by decrypting with K'_{p_t} and checks the value of the time stamp. If the time stamp is correct, she encrypts the emergency information with k'_{a_i} and sends back to a_i $\left\langle \{EmergencyInfo, timestamp + 2\}_{k'_{a_i}} \right\rangle$. Now, a_i decrypts the received message with K'_{a_i} and sends the emergency information to the appropriate FR_i's encrypted with her public key ($\{EmergencyInfo\}_{k^{a_i}_{r_t}}$). FR_i's decrypt the received message and finds the specific information about the event. At this point, FR_i's is in position to find the best possible way to effectively handle the situation.

Furthermore, FR_i's sends back to a_i a message $\left\langle \{m\}_{k_{a_i}} \right\rangle$ confirming that they indeed received the emergency information correctly. Upon reception, a_i decrypts $\left\langle \{m\}_{k_{a_i}} \right\rangle$ with

[3] p_t knows $x \in \mathbb{Z}_q^* : h = g^x$ and $\tilde{h} = \tilde{g}^x$

[4] a, b are known since CA and p_t have already established $N_{(p_t,CA)}$.

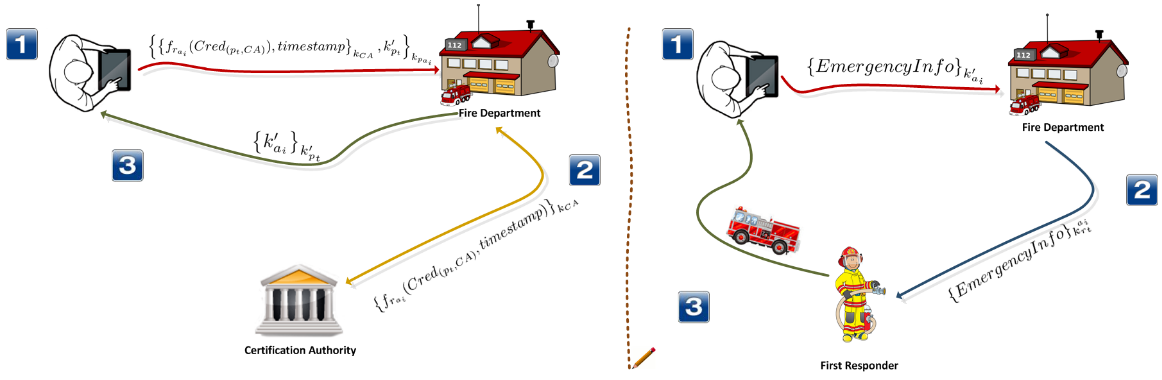

1
$\{\{f_{r_{a_i}}(Cred_{(p_t,CA)}), timestamp\}_{k_{CA}}, k'_{p_t}\}_{k_{pa_i}}$
Fire Department
$\{k'_{a_i}\}_{k'_{p_t}}$
3
2
$\{f_{r_{a_i}}(Cred_{(p_t,CA)}, timestamp)\}_{k_{CA}}$
Certification Authority
1
$\{EmergencyInfo\}_{k'_{a_i}}$
Fire Department
2
$\{EmergencyInfo\}_{k_{rt}}^{a_i}$
3
First Responder

Fig. 2. User Authentication with an Authority & Sharing Emergency Information

K_{a_i} and sends back to p_t an acknowledgment message m' encrypted with k_{p_t}. By doing so, we protect users from situations where either a CA, a_i or FR_i's are not available.

V. Security Discussion

In this section we analyze the behavior of SETS regarding different kinds of attacks.

Theorem 1 (Man-in-the-Middle)**.** *Assume an adversary $\mathcal{ADV}$ who intrudes into an existing connection (either between p_t and a_i or between a_i and CA) with main aim to intercept the exchange of data and inject false information into it. Then, $\mathcal{ADV}$ will not gain any useful information from the interception.*

Proof. Since all the communication between the participants is encrypted and $\mathcal{ADV}$ does not know the corresponding private keys he cannot decrypt the messages. Lets assume that $\mathcal{ADV}$ is intercepting the transmitted messages in order to make an offline attack. In this case, $\mathcal{ADV}$ can theoretically compromise the keys. Even if $\mathcal{ADV}$ manage to retrieve the keys of all the participants (p_t, CA and a_i) he will not be in a position of retrieving crucial information from the data exchange. This is achieved by the use of session keys that p_t and a_i generate for their communication. Apart from that, since $\mathcal{ADV}$ will be in position to successfully decrypt the messages that are encrypted with the public keys of the participants (and thus partially breaking the security of SETS), we propose that at least CA and a_i should change their keys frequently. By doing this, $\mathcal{ADV}$ cannot glean any useful information from the communication since all the messages between a registered user p_t, CA and a_i will be encrypted either with the fresh keys of CA or a_i, or with the session keys k'_{p_t} or k'_{a_i}. □

Theorem 2 (Retransmission of Messages)**.** *Assume an adversary $\mathcal{ADV}$ who eavesdrops on the communication between p_t and a_i and retransmits a message that p_t sent to a_i. Then, $\mathcal{ADV}$ cannot be validated as a legitimate user from CA.*

Proof. Lets assume that an adversary $\mathcal{ADV}$ eavesdrops on the first message m that p_t sends to a_i. Thus, ADV knows $\left\langle \left\{ \{f_{r_{a_i}}(Cred_{(p_t,CA)}), timestamp\}_{k_{CA}}, k'_{p_t} \right\}_{k_{a_i}} \right\rangle$. Then, since $\mathcal{ADV}$ is not aware of K_{a_i} sends m to a_i, as it is. Upon reception, a_i cannot realize that this is a replay message so, she decrypts m in order to find k'_{p_t} and forwards the rest of the message to CA. At this point, CA checks the time stamp that is embedded in m, finds that it is not a fresh one and sends a message to a_i to drop the connection. □

From Theorems (1) and (2) we conclude that SETS provides robustness also to an Interleaving Attack where a selective combination of information from one or more previous protocol executions is made by an adversary $\mathcal{ADV}$.

Theorem 3 (Compromised CA and a_i)**.** *Assume that CA colludes with a_i in order to break the privacy of p_t. Then, no information can be inferred about the identity of p_t.*

Proof. Let CA and a_i act as adversaries in the sense that they collude to break the privacy of a legitimate user p_t. First, p_t sends $\left\langle \left\{ \{f_{r_{a_i}}(Cred_{(p_t,CA)}), timestamp\}_{k_{CA}}, k'_{p_t} \right\}_{k_{a_i}} \right\rangle$ to a_i. From this message, a_i can only find k'_{p_t}. So, after decrypting, she sends to CA the first part of the message which is encrypted with the public key of CA. Then, CA can validate that p_t is valid without knowing who exactly she is. Additionally, if CA and a_i collude to find the real identity of p_t they will fail. As we can see, p_t does not send to a_i her real public key. Instead she generates and sends a session key k'_{p_t} that will be used for further communication between them. So, since none of them knows the public key of p_t they are unable to find her real identity. □

VI. Experimental Results

To prove the effectiveness of SETS, we experiment on a wi-fi network with mobile devices (PDAs). Our PDAs use a 624MHz processor, 128MB RAM and operates running Windows Mobile 6. The capabilities of the specific device compared to conventional commercial Tablet PC's and devices is limited and thus the measurements we present in this section will be more efficient in devices with better hardware.

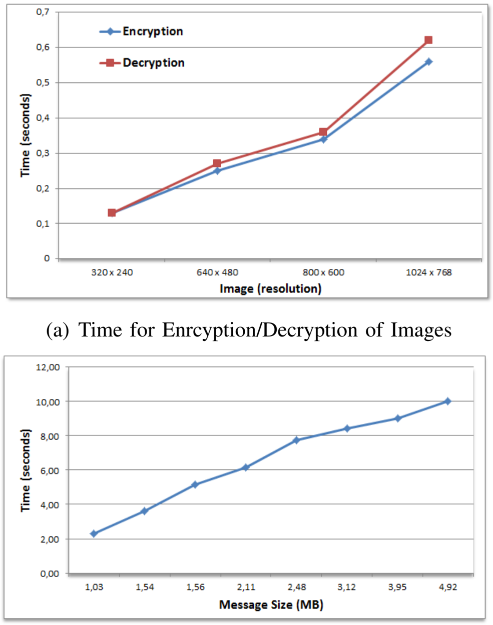
Our experiments have aimed at analyzing two main performance metrics; processing time and communication overhead. Each of the experiments was run through 1000 iterations.

A. Processing Time

The first phase of our experiments involved measuring the processing time of SETS. To this end, we measured the completion time for the encryption decryption of different kinds of messages. For the encryption and decryption we used the RSA cryptosystem with keys of 256bits length.

As we described in Section IV a user p_t will have to encrypt on her mobile device the emergency information which apart from the location of the event can also contain a media file.

Figure 3(a), illustrates the time in seconds that the PDA needed in order to encrypt and decrypt the same image in the following resolutions: i) 320X240 (32.3 kb), ii) 640X480 (73.5 kb), iii) 800X600 (100kb) and iv) 1024X768 (163kb).



(a) Time for Enrcyption/Decryption of Images

(b) Communication Delay

Fig. 3. Image Encryption/Decryption & Communiction Delay

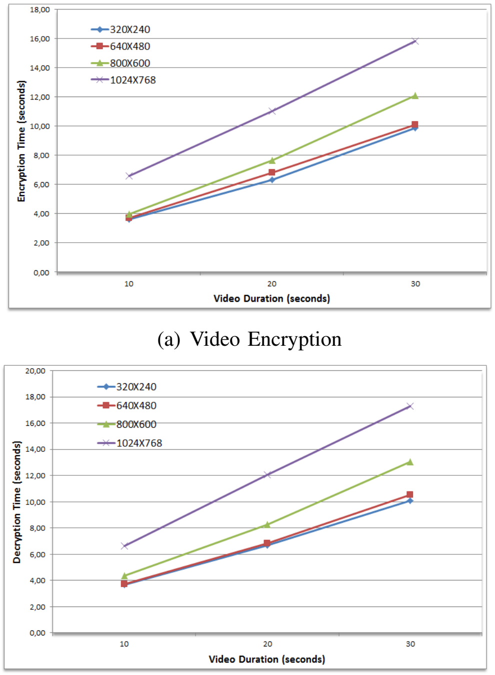
As we can see from the graph, for an image of 320X240 resolution the encryption time is $0.13sec$ while for the case of a more detailed image (1024X768) the time still remains less than one second ($0.56sec$). Moreover, since an FR will receive from a_i the emergency information in a encrypted form, she will have to decrypt it. So, we also measured the time that is needed for the decryption of the above encrypted images by the PDA. From figure 3(a) we obtain that the decryption of an image (1024X768) needs $0.62sec$.

Furthermore, we implemented the same process for video files with different resolutions and durations. Figure 4(a) illustrates the time needed for the encryption of a video while figure 4(b) illustrates the times for the decryption process. As we can see, the average time for a $10sec$ video of 320X240 resolution is $3.64sec$ while for the decryption of the same video $3.68sec$ needed. Additionally, for the encryption of a $30sec$ video of 1024X768 resolution the average time was $15.84sec$ while for the decryption $17.30sec$.

From the above results we can observe that the encryption/decryption process, even for video files with a satisfactory resolution that could point out the details of an event, is efficient even for mobile devices with limited resources.

B. Communication Dealy

In the second phase of our experiments, we measured the communication delay of encrypted packets over the network. For that purpose, we established a network between a laptop computer and two PDAs that were exchanging packets of different sizes through wireless communication. Figure 3(b) illustrates the time that needed for messages from $1.03MB$ to $4.92MB$ to transmitted from the PDA to the laptop and vice versa. For a message with size between $1MB$ and $3MB$ the time is less than $6sec$ while for a file with a size near to $5MB$ the average time is slightly more than $10sec$.



(a) Video Encryption

(b) Video Decryption

Fig. 4. Video Encryption/Decryption on a PDA

VII. Conclusion

In this paper we presented SETS, a protocol with main aim to provide secure communication during emergency situations. The effectiveness of our protocol is proved in our security discussion where SETS' appears to be resistant to Replay, Forced Delay and Man-in-the-Middle attacks. SETS manages to protect the user privacy through anonymity. Finally, we proved that SETS can be easily adapted for mobile devices, since field experimental results showed the effectiveness of the protocol on actual smart-phone platforms.

References

[1] Timothy E. Levin, Cynthia E. Irvine, Terry V. Benzel, Thuy D. Nguyen, Paul C. Clark and G. Bhaskara. *Idea: Trusted Emergency Management*. In Engineering Secure Software and Systems, First International Symposium ESSoS 2009, Leuven, Belgium, February 4-6, 2009. Proceedings. Springer, 2009.

[2] F. Dotzer, F. Kohlmayer, T. Kosch and M. Strassberger. *Secure Communication for Intersection Assistance*. WIT 2005: 2nd International Workshop on Intelligent Transportation, 2006.

[3] Xiangxi Li, Yu Zhang and Yuxin Deng. *Verifying Anonymous Credential Systems in Applied Pi Calculus*. In Proceedings of the 8th International Conference on Cryptology and Network Security (CANS '09), Springer-Verlag, Berlin, Heidelberg, pp. 209–225, 2009.

[4] A. Lysyanskaya, R. L. Rivest and A. Sahai. *Pseudonym Systems (Extended Abstract)*. In Selected Areas in Cryptography, pp. 184–199, 1999.

[5] I. B. Damgard. *Payment Systems and Credential Mechanisms with Provable Security Against Abuse by Individuals (Extended Abstract)*. In Advances in Cryptology–Crypto' 88, pp. 328–355, Springer-Verlag, 1988.