TRUSTED SMS
A Novel Framework for Non-repudiable SMS-based Processes

Antonio Grillo, Alessandro Lentini, Gianlugi Me
Department of Computer Science and Engineering, University of Rome Tor Vergata, Rome, Italy

Giuliano Rulli
Spike Reply s.r.l., Reply S.p.A., Rome, Italy

Keywords: Confidentiality, data security, sms, ECDSA, elliptic curves.

Abstract: The exponential growth of the Short Message Service (SMS) use has led this service to an indispensable tool for social, marketing and advertising messaging. Moreover, mobile devices such as smartphones, handsets and PDAs represent an enabling factor for distributing digital content. Mobile devices are quickly becoming Personal Trust Device (PTD); mobile devices embed personal data, which allow sending/receiving confidential information from/to the PTD. This paper aims to introduce Trusted-SMS, a novel framework to exchange secure SMS. This system is composed by three main entities: the Service Supplier, which publishes and delivers services; the End User, which choices and eventually pays for a specific service, that belongs to the service-set offered by a Service Supplier; the Certification Authority (CA) which represents the trusted entity shared by the Service Supplier and the End User. The CA plays the role of the Certification Authority. The main requirements of the overall system are strictly non-repudiability, user friendliness and platform portability. The security requirement includes customer transaction authentication, confidentiality, integrity and non repudiation, in an environment composed of heterogeneous networks and devices, with different security weaknesses. Trusted-SMS allows exchanging SMS digitally signed with Elliptic Curve Digital Signature Algorithm. SMS digitally signed are useful in many scenarios, such as commercial transaction, production of delegation from a remote site and provisioning of e-healthcare services. The signature is fully contained in a single SMS; the size of a digital signature amount to fifty bytes leaving more than one hundred bytes (110 bytes) for the SMS payload. Moreover the application of Elliptic Curve Integrated Encryption Schema cryptographic algorithm, which is based on the same credentials needed by the digital signature algorithm, allows protecting the payload from intrusions.

1 INTRODUCTION

Short message services (SMS), thanks to its relative simplicity and ease of use, similar to the e-mail, keeps on growing in penetrating the market throughout the world. In fact, during the 2000, just 17 billion SMS messages were sent; in 2001, the number was up to 250 billion, and 500 billion SMS messages were sent in 2004 (Kivimaki and Fomin, 2001). At an average cost of USD 0.10 per message, this generates revenues in excess of $50 billion for mobile telephone operators and represents close to 100 text messages for every person in the world.
Text messaging has become so popular that advertising agencies are now considering it as an irrinuciable vehicle to keep the customer up to dated about the new offers and products (Dickinger et al., 2004). The services providing bulk text messages are also becoming a popular way for clubs, associations, and advertisers to quickly reach a group of opt-in subscribers. This advertising has proven to be extremely effective, but some insiders worry that advertisers may abuse the power of mobile marketing and it will be considered spam.
Moreover, the proliferation of SMS attracts malware writers that adapt phishing and other activities based on social engineering techniques (e.g. spoofing become SMS-spoofing) in order to manipulate people into performing actions or divulging confidential information (van der Merwe et al., 2005).

Grillo A., Lentini A., Me G. and Rulli G. (2008).
TRUSTED SMS - A Novel Framework for Non-repudiable SMS-based Processes.
In *Proceedings of the First International Conference on Health Informatics*, pages 43-50

This paper aims to describe a novel framework to exchange secure SMS: Trusted-SMS. Due to the well known, or yet undiscovered, security weaknesses of SMS, it's important, for some kind of applications, to enforce security layers, such as confidentiality, integrity and non repudiation. The goals of this paper are concerned with:

a Expressing the requirements of the Trusted-SMS framework;

b Analyzing and designing a framework meeting those needs;

c Characterizing that framework,with a sketch of a case study.

In the remaining of the paper, the Section 2 sketches the current level of development of modern methods and products that involve a secure SMS exchange. The Section 3 describes Trusted-SMS framework; this section deals with framework needs and goals in required features, architectural design and some technical details. The Section 4 presents a prototype system compliant to Trusted-SMS framework as case study. Section 5 presents some conclusions, and points to the future trends.

2 STATE OF THE ART

The huge success of some modern network protocols, not designed with security in mind, poses many concerns about fraudulent intrusions and undiscerning violation of the privacy. As for the TCP/IP case, the SMS suffer the lack of many features that are desirable or needed on an insecure network, as the GSM can be considered. For this reason, the companies offer some security-enhancing products, which allow to protect private information exchanged via SMS. The available solutions can be divided into two different models: "'Peer-to-Peer"'(P2P) and "'Client-Server"'(CS); the former focuses on the SMS exchange which involves only mobile devices, while the latter focuses on the SMS exchange which occurs between a mobile device, as the Client, and a remote computer, as the Server.

Many solutions belong to the P2P model, such as Message in a Bottle (Chirico, 2007), Spider-SMS (Barbi, 2007), Kryptext (Kryptex, 2007), MultiTasker (MultiTasker, 2007) and CryptoSMS (CryptoSMS, 2007). Products as Fortress-SMS (FortressSMS, 2007) follow the CS model, this can be view as a generalization of products based on P2P model. The level of security of the previous solutions is strictly related to the key-distribution schema and the security algorithm applied. Miabo and Spider SMS cannot be used for sending and receiving information as part of business transactions and personal communications. They don't fit security needs because the former relies on a PGP schema as public key infrastructure and the latter relies on a random key generator.

Fortress-SMS relies on a Rijndael engine with cipher block chaining (Advanced Encryption Standard compliant) as encryption algorithm; CryptoSMS uses three overlapping strong encryption schemes, employing both block and stream ciphers. Fortress-SMS and CryptoSMS are not fully documented about their execution times; to the best of our understanding encryption schemas adopted by these products require long processing times on mobile devices (Waadt et al., 2005).

Many products allow both the SMS digital signature and the SMS encryption; Trusted-SMS allows SMS digital signature or alternatively SMS signature on SMS encrypted.

Furthermore, the GSM specifications (3GPP, 2007) don't define any mechanism for ensuring integrity of SMS content and authentication of SMS sender. SMS digitally signed can be used to avoid SMS-tampering, ensuring integrity and authentication of the sender (Center, 2007). SMS encrypted are suitable to avoid unauthorized access to SMS content.

3 TRUSTED-SMS FRAMEWORK

Trusted-SMS is a complex framework for allowing secure SMS exchange between involved entities. This framework enables a user to send or receive SMS messages digitally signed with standard Elliptic Curve Digital Signature Algorithm (ECDSA) and optionally encrypted with standard Elliptic Curve Integrated Encryption Schema (ECIES). According to the digitally signed SMS, this framework can be used to avoid SMS from tampering ensuring integrity and non repudiation: the recipient can detect the tampering during the SMS verification process. With encrypted SMS, this framework can be used to avoid SMS from unauthorized access. If someone intercepts an encrypted SMS he cannot eavesdrop the SMS content.

Trusted-SMS is composed by three main entities:

Service Supplier: this entity is responsible for providing services;

End User: this entity is the consumer of the supplied services;

Certification Authority: this entity is responsible to ensure trusting to the involved entities.

3.1 Required Features

The expected use cases and the resulting requirements suggest the following list of features, trying to depict the Trusted-SMS framework. For the sake of simplicity, we divide the framework required features in three subsets, which are related to the previously identified entities:

Service Supplier entity features:

1. each Service Supplier owns a Large Account number; clients of a Service Supplier use its Large Account number as its identifier;
2. the Service Supplier has to be able to generate, send and receive different kind of SMS;
3. each SMS, which Service Suppliers have to send to Customers, must be ECDSA-Signed using the CA private key (KPrivCA);
4. each SMS, which Service Suppliers receive from a Customer, must be ECDSA-Verified using the Customer public key (KPubMSISDN);
5. the Service Supplier has to manage a large number of information as end user personal data which has to be stored in a persistent memory storage;
6. the Service Supplier wants to preserve SMS content also from CA access.

End User entity features:

1. the subsystem placed on the End User platform, a smartphone, has to be realized as a MIDP 2.0 java mobile application;
2. End User platform has to be able to manage different kinds of SMS;
3. each End User is characterized by a key-pair: a public key(KPubMSISDN) and a private key (KPrivMSISDN);
4. each SMS, which Customers have to send to Service Suppliers, must be ECDSA-Signed by client application using its own private key (KPrivMSISDN);
5. each SMS, which Customers receive from Service Suppliers, must be ECDSA-Verified by client application using the CA public key (KPubCA);
6. the subsystem placed on the End User platform must be defined on a parameters set which defines the elliptic curve; that elliptic curve is used for generating/verifying ECDSA signatures (this curve is shared with server application);
7. the subsystem placed on the End User platform must be defined on a parameters set which defined the hash function; that hash function is used for creating a digest which must be signed with the identified signature algorithm;

Certification Authority entity features:

1. a single server realizes the Certification Authority (CA); each Service Supplier knows this server.
2. the CA must be able to sign SMS using ECDSA with its own private key (KPrivCA);
3. the CA must be able to verify SMS using ECDSA with an appropriate public key (KPubMSISDN);
4. the CA has to manage a large number of confidential information (e.g. end users' keys), which has to be stored in a persistent memory storage;
5. each keypair is characterized by a temporal validity;
6. the CA subsystem offers a graphical user interface in order to allow administration of some services;
7. inserting/deleting end users in/from CA repository has to be an administration task;
8. updating end user status (e.g. able/disable) has to be an administration task; a disabled user is temporally excluded from whole system, this feature is activated when end user's devices are lost or stolen, when contract's temporal validity is expired and is in line for renewed;
9. updating end user key status (e.g. able/disable) has to be an administration task, an end user with a disabled key cannot sign SMS while he can continue to verify information sent to its device.
10. inserting/deleting service suppliers in/from CA repository has to be an administration task;
11. updating service supplier status (e.g. able/disable) has to be an administration task.

3.2 Architectural Design

The architectural design evolves naturally from identified required features. Figure 1 proposes the overall architecture of Trusted-SMS framework. Despite of previous definition of the framework as result of three main entities, figure 5 shows six components:

Published Services: it represents the common space where Service Suppliers interact with End Users advertising services and products.

Service Supplier: it represents the subsystem that realizes the Service Supplier entity features previously identified.

Certification Authority: it represents the subsystem that realizes the Certification Authority entity features previously identified.

End User: it represents the subsystem that realizes the End User entity features previously identified.

SMS Gateway: it represents a generic Short Message Service Center (SMSC).

CA Administrator: it represents the front end to CA administration services.

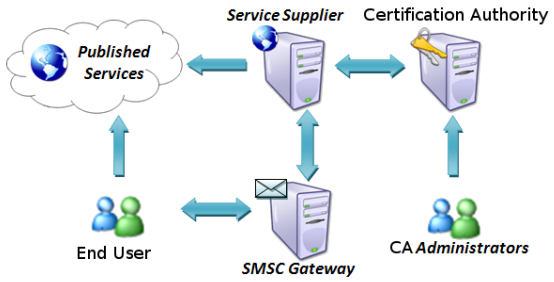

Figure 1: Overall architectre of Trusted-SMS framework.

This paper goes on to describe accurately identified subsystem.

Service Supplier subsystem: The Service Supplier subsystem can be divided in 4 main components (Fig. 2):

- a component that acts as general purpose end user interface, publishes provided services;
- a component that acts as interface with the sms gateway, sends and receives SMS to/from end users;
- a component that acts as interface with the Certification Authority in order to request a signature for a text or to verify an SMS signed by an user;
- a component that can be view as Service Supplier core, generates SMS text, interacts with others components in order to process and store end user requests and responses.

An end user request for a specific service could generally done by interacting with the service supplier directly or by a dedicated interface. The service supplier subsystem collects end user requests, automatically converts the request into a SMS text, requests to CA a signature for this text and finally forwards the SMS to the end user through the SMS gateway. In order to preserve confidentiality of the SMS content the signature is requested for a hash of that content. Reaction to an end user response involves the same service supplier subsystem components. The response is received through the SMS gateway interface and it is stored into the database. The service supplier interacts with the CA to verify the appended signature. The result of the verifying test is recorded in the database. If the requested service counts a kind of confirmation, the end user is informed by a new SMS communication or a message on the service supplier interface.

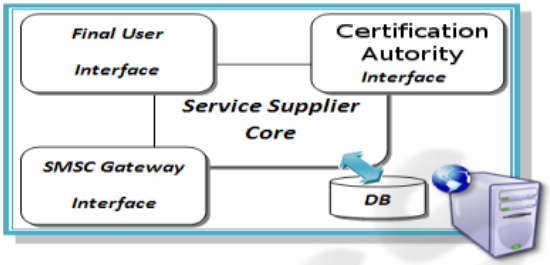

Figure 2: Structure of Service Supplier subsystem.

End User subsystem: The End User interacts with the remaining system by two different ways.
In the first case the End User chooses a specific service from the set of the published services; the Service Supplier of that service asks for confirmation. The End User has to confirm its choice to make use of requested service.
In the second case, the End User receives a communication about a service directly from the Service Supplier; only if necessary the End User answers to that communication. Service Supplier in this case can send service-communication without any explicit End User requests. In the first case described above, the End User realizes its interactions using two different channels; the first channel is used to select a service, the second one is used to accept the delivery of the selected service. In the second case these channels are coincident.
Separating the capability of requesting one service from the capability of accepting or denying the delivery of that service ensures an high level of security. If those channels rely on different technologies a potential attack becomes a very hard challenge. The main software component of the End User entity is represented by a midlet: a Java 2 Micro Edition(J2ME) application that can be executed on a personal mobile device characterized by low computational capability, such as a smartphone or a pda. The midlet has two main responsibilities; it has to verify the signatures that are carried by the incoming SMS and it has to generate the signatures that are carried by the outgoing SMS according to the CA and End User credential. Each midlet deployed on a mobile device has

to be provided with the CA public key and the End User private key; the former is used to verify CA signatures, while the latter is used to generate End User signatures.
End User credential (i.e. its cryptographic keys) are included in the midlet resources and protected by using a symmetric key injected into the midlet code. Received and sent SMS are stored in a secure separate repository, different from the usual mobile phone's SMS repository, secured and encrypted by user's PIN. Thus, if someone snaffles your mobile phone full of your private SMS he cannot read them because they are protected inside the repository.
Figure 3 shows the End User subsystem pointing out that three units made up this subsystem; SMS I/O Interface, which manages the SMS sending and receiving activities, and Graphical User Interface, which leads the interaction with the User, are coordinated by the End User Core. In order to store application data End User Core interacts with Record Management System(RMS). Cryptographic operations (i.e. signature generation or verification) are performed by the Core unit. Service Supplier Interface represents the second unit; listing all services offered by supplier and requesting for a service delivery represent the main issues that the second unit has to resolve.

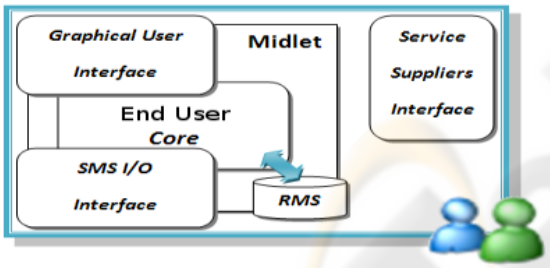

Figure 3: Structure of End User subsystem.

Certification Authority subsystem: The Certification Authority represents the trusted entity; End User and Service Supplier share the same confidence towards the CA entity. The CA subsystem has two main communication partners: the CA Administrator and the Service Supplier.
Interactions with CA Administrator are intended for managing users registered to Trusted-SMS framework. Service Supplier interacts with CA subsystem in order to obtain the generation/verification of signature carried by SMS.
The CA subsystem is supported in all its activities by a database, which allows the permanent storage of needed information. Database keeps information about End Users, Service Suppliers and CA itself. End Users information concern with personal data (e.g. name, surname) and personal credential(e.g. public key, public-key-validity flag, user-enabled flag). In order to insert new users, delete or enable/disable existent user, enable/disable users keys those information are accessed in a "'write-mode"' by CA Administrator with related services. In order to verify End User generated signatures, credential information are accessed in a "'read-mode"' by services related to the Service Supplier. CA information concern with CA credential and CA Administration privileges; the former are accessed by Service Supplier related services in order to generate a signature, the latter are used in authentication process while acquiring CA management control.
Finally, Service Supplier information are related with supplier data, which are accessed by Service Supplier related services in order to verify if that Service Supplier is licensed to use these services.
Figure 4 proposes the structure of the Certification Authority subsystem. Certification Authority Core provides cryptographic and database services to the other modules of this subsystem. These modules manage the communication with CA partners.

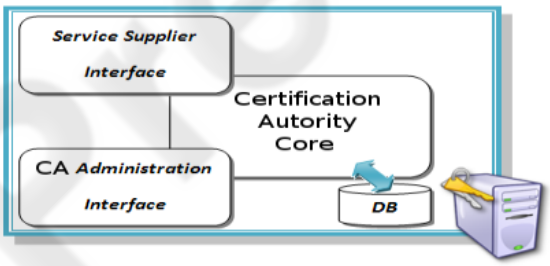

Figure 4: Structure of Certification Authority subsystem.

3.3 Technical Features

Elliptic curve cryptography (ECC) (Martinez et al., 2005) is a family of public-key algorithms that can provide shorter key lengths and, depending upon the environment and application in which it is used, improved performance over systems based on integer factorization and discrete logarithms (Higgins and Clement, 2001). ECC has its security based on a difficult mathematical problem. An elliptic curve can be thought of as a mathematical structure in which certain operations can be defined. These operations provide a one-way function that can be used to produce efficient cryptographic systems. The one-way function used in ECC is called the elliptic curve discrete logarithm problem (ECDLP). The ECDLP is similar to the one-way function on which DSA (of Standard et Technology, 2007) and Diffie-Hellman are based, and hence, elliptic curve analogs of each of

these algorithms have been defined. The most popular signature scheme which uses elliptic curves is called the Elliptic Curve Digital Signature Algorithm (ECDSA) (Jonson et al., 2001). The Secure Hash Algorithm (SHA) (F.I.P.S., 2002) hash functions refer to five FIPS-approved (F.I.P.S., 2007) algorithms for computing a condensed digital representation (known as a message digest) that is, to a high degree of probability, unique for a given input data sequence (the message).

Trusted-SMS relies on ECDSA to sign the exchanged data, thus avoiding unnoticed modification. The SHA1 hash function makes a digest of the data. Reducing a message in a digest lowers computational capabilities needed by the signing task, as we need, since the mobile devices involved in the system are very resource bounded. Trusted-SMS uses ECDSA with a 192 bit Elliptic Curves (EC) with a consequent 48 byte signature value. The Keys size of 192 bit agrees with the size of an SMS that is 140 bytes for binary and 160 bytes for textual SMS. In comparison to a 1024 bit RSA key, ECC (Elliptic Curve Cryptography) provides shorter keys, shorter encrypted messages and faster private key operations. Comparing the two cryptographic systems, a 1024 bit RSA key is considered to have the same secuity of a 160 bit ECC key. The cellular networks can be broken into two chief components - the radio, or ”air interface” and the wired backbone (out of scope of this paper). The air interface of a cellular network can be divided into control and traffic channels. Control channels (CCHs) are used for call setup and SMS delivery. Traffic channels (TCHs) are used for the duration of voice calls. It helps to think of control channels as a very small portion of radio frequency that allow cellular towers to send information pertaining to call setup, SMS delivery and network conditions (such as the availability of traffic channels) to mobile phones. The Short Message Service - Point to Point (SMS-PP) is defined in GSM recommendation 03.40 (3GPP, 2007). GSM 03.41 (3GPP, 2007) defines the Short Message Service - Cell Broadcast (SMS-CB) which allows messages (advertising, public information, etc.) to be broadcast to all mobile users in a specified geographical area. Messages are sent to a Short Message Service Centre (SMSC) which provides a store-and-forward mechanism. It attempts to send messages to their recipients. If a recipient is not reachable, the SMSC queues the message for later retry. Transmission of the short messages between SMSC and phone can be done through different protocols such as Signaling System 7 (SS7) within the standard GSM MAP framework or TCPIP within the same standard. Messages are sent with the additional MAP operation forward_short_message, whose payload length is limited by the constraints of the signalling protocol to precisely 140 bytes (140 bytes = 140 * 8 bits = 1120 bits). In practice, this translates to either 160 7-bit characters, 140 8-bit characters, or 70 16-bit characters. General Packet Radio Service (GPRS) is a packet-switched technology. GPRS that is an extension of GSM can be used as the bearer of SMS. Spoofing from a mobile is impossible unless you can forge GPRS radio traffic (Levy and Arce, 2004).

4 A CASE STUDY

Realizing a prototype developed with open source tool we illustrate the validity of our framework. We utilized Trusted-SMS framework in a specific healthcare environment. E-Health involving information and communication technologies that are at the service of a wide range of actors in the health sector, from doctors, hospital managers, nurses, data processing specialists, social security administrators to - of course - patients and citizens, is a good test-bed for our framework.
The Certification Authority performs its tasks by using Web Services. The W3C Web service definition refers to clients and servers that communicate using XML messages that follow the SOAP-standard (Lai et al., 2005). The CA publishes two different lists of web services, each one for its communication partners: the CA Administrator and the Service Supplier. The permanent storage of information is achieved using a mysql database and it is protected by symmetric key encryption (Zoratti, 2006).
Service Supplier publishes its services on an interface, from which end users can request the delivery of a specific service. Once the end user requests a service the related request is stored into a mysql database. For each request received the supplier use the signature generation service in order to produce the related SMS. Each received SMS is drawn from the database and it is processed in order to verify its signature validity. The Service Supplier subsystem utilizes the signature verification/generation services exported by CA as web services.
A SMS gateway is realized by an external hardware component that behaves as a gateway between GSM mobile phone network and web applications (AreaSX, 2007). This component provides the SMS sending/receiving capability using simple GET or POST HTTP transactions over Ethernet.
The End User subsystem is mainly represented by a Java 2 Micro Edition application compliant to MIDP 2.0 and CLDC 1.1 and installed on different models of

Nokia mobile phones: series 40 equipped with Symbian OS 8.0 (i.e. 6630, 6680), series 60 2nd edition equipped with Symbian OS 8.1 (i.e. N70), and series 60 3rd edition equipped with Symbian OS 9.1 (i.e. N73, E70). Bouncycastle Crypto APIs offer support for realizing cryptographic operations. This APIs are an open source solution, which works with everything between J2ME and JDK 1.6 (Bouncycastle, 2007).

Next, we present the description of two use cases, both derived from different ways of the end user interaction with the framework described above.

In first use case, the End User (a patient) has a preliminary contact with a medical professional for receiving a complete check-up. As soon as the physicians have the results for physical examinations, he sends a SMS signed to the patient. The SMS carries personal information about patient health status protecting them with some cryptographic techniques. Signing the SMS content ensures patient on sender authenticity; moreover cryptographic protection guarantees patient privacy. In this way physicians can inform immediately the patient about the examination, if necessary they can ask for a secondary care.

The second use case, which refers to a more complex situation, will be described using a schematic representation (Fig. 5). In particular this case-study depicts the whole scenario including all interaction from the preliminary registration phase. In order to reach clearness in exposition, description is organized as a finite sequence of steps, which refers to Figure 5.

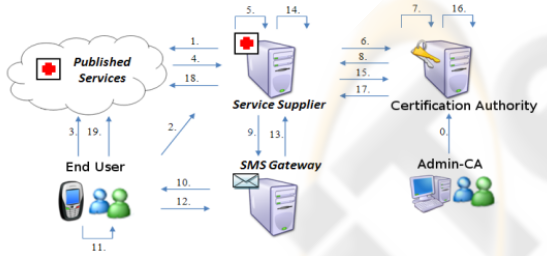

Figure 5: Structure of Certification Authority subsystem.

1. The CA-Administrator add End Users and Service Suppliers to the framework, this represents the zero step of the happy scenario of the case study. This step is executed for every End User/Service Supplier, who decides to subscribe to Trusted-SMS services.
2. The Service Supplier that delivers a set of services publishes them on its interface (such as a web site).
3. The End User (a patient) has an initial contact with a medical professional for receiving health examination.
4. The End User choices a specific service (such as request for its own case history, specific examination results, information about personal cure, reservation for a secondary care, buying a medical examination, etc.) from that interface and
5. its request reaches the Service Supplier.
6. The Service Supplier produces an SMS related to the request received.
7. Utilizing the signature generation service offered by Certification Authority, the Service Supplier signs the SMS content
8. The CA checks both the Service Supplier credentials and the End User credentials; only if the credentials are valid, CA begins the signature generation.
9. The CA sends the signed response to the requesting Service Supplier.
10. The Service Supplier forwards the digitally signed SMS to SMS Gateway, (10) which delivers it to the End User.
11. The End User verifies the signature carried by the SMS, if necessary accepts or refuses the service delivery and generate the signature for response SMS.
12. The response SMS is sent to the SMS Gateway,
13. which forwards it to the Service Supplier.
14. The Service Supplier requests to utilize the digital signature verification process.
15. The CA processes the Service Supplier request by checking the credentials of entity involved and
16. answer to Service Supplier with the status of verification process.
17. The Service Supplier realizes the End User choice (cfr. step 10 accept or refuse) if the End User digital signature was successfully checked.
18. The Service Supplier publishes on the web site the transaction status
19. The End User can control the effective status of the service delivery. Steps 18 and 19 are conditional on the specific requested service.

5 CONCLUSIONS

The growth of mobile technology has opened various opportunities, both in marketing and in M-Commerce applications; technologies such as cell phone networks are becoming integrated with other systems such as the Internet. The SMS have been very popular among mobile phone users because it is silent, fast, and cheap. Unfortunately, this may lead to new security risks, for example when conducting phishing attacks using mobile technology and especially the SMS.

In this paper we proposed a framework to exchange secure SMS with respect to: (a) *security*, communication between principals are based on well cryptographic techniques, (b) *traceability*, transaction are not kept anonymous, (c) *usability and convenience*, cost of deployment and management for all subsystems involved (consumer, service suppliers, trusted service provider) are acceptable.

Furthermore, in order to set Trusted-SMS into a specific e-healthcare scenario, we made a prototype developed with open source tool. Since current trends in mobile phone technology move towards a direction of miniaturization and higher computational and graphical performance, allowing a complete transaction procedure in less than a minute, we believe that Trusted-SMS prototype can shows the validity of our framework in the field.

Finally, the case study covers only one of the possible uses of Trusted-SMS framework. The systems managing private information, or systems that schedule a booking/reservation procedure (e.g. system that can be used by theatre, stadium cinemas, airline company, university), or involving money transfers, can benefit of security features provided by Trusted-SMS framework.

REFERENCES

3GPP (2007). 3rd generation partnership project.

AreaSX (2007). Sms machine/htpp.

Barbi, L. (2007). Spidersms - sending and reception of encrypted sms.

Bouncycastle (2007). The legion of the bouncycastle.

Center, C. C. R. (2007). Sms spoofing - q-a with ccrc staff.

Chirico, U. (2007). Miabo - messages in a bottle.

CryptoSMS (2007). Cryptosms - protecting your confidential sms messages.

Dickinger, A., Haghirian, P., Murphy, J., and Scharl, A. (2004). An investigation and conceptual model of sms marketing. In *System Sciences, Proceedings of the 37th Annual Hawaii International Conference on*.

F.I.P.S. (2002). Secure hash standard, fips publication.

F.I.P.S. (2007). Federal information processing standards.

FortressSMS (2007). Fortresssms - phone based application to send and read encrypted sms text messages.

Higgins, J. Z. L. and Clement, M. (2001). Performance of finite field arithmetic in an elliptic curve cryptosystem. *Security Technology, CCST '05. 39th Annual 2005 International Carnahan Conference on*, pages 249 – 256.

Jonson, D., Menezes, A., and Vanstone, S. (2001). The elliptic curve digital signature algorithm.

Kivimaki, A. and Fomin, V. (2001). What makes a killer application for the cellular telephony services? *Standardization and Innovation in Information Technology, 2nd IEEE Conference*, pages 25 – 37.

Kryptex (2007). Kryptex - send and receive encrypted text sms.

Lai, K. Y., Phan, T. K. A., and Tari, Z. (2005). Efficient soap binding for mobile web services. *Local Computer Networks 30th Anniversary, The IEEE Conference on*, pages 218 – 225.

Levy, E. and Arce, I. (2004). Interface illusions. In *IEEE Security and Privacy*, pages 66 – 99.

Martinez, V. G., Avila, C. S., Garcia, J. E., and Encinas, L. H. (2005). Elliptic curve cryptography: Java implementation issues. *Local Computer Networks 30th Anniversary, The IEEE Conference on*, pages 238 – 241.

MultiTasker (2007). Multitasker - messaging made easy.

of Standard et Technology, N. I. (2007). Digital signature standard.

van der Merwe, A., Seker, R., and Gerber, A. (2005). Phishing in the system of systems settings: mobile technology. In *Systems, Man and Cybernetics, IEEE International Conference on*, pages 228 – 232.

Waadt, A., Bruck, G., Jung, P., Kowalzik, M., Trapp, T., and Begall, C. (2005). Qos monitoring for professional short-message-services in mobile networks. *Wireless Communication Systems, 2nd International Symposium on*, pages 228 – 232.

Zoratti, I. (2006). Mysql security best practices. *Crime and Security, The Institution of Engineering and Technology Conference on*, pages 183 – 198.