

# TSRG based Certified Mail Service (TCMS)

**Gamal A. Hussein**

Arabic Micro Systems

[www.aramics.com](http://www.aramics.com)

[gamal\\_ams@hotmail.com](mailto:gamal_ams@hotmail.com)

**Fatma Helmy**

Private University in Cairo.

[fatmahelmy2000@yahoo.com](mailto:fatmahelmy2000@yahoo.com)

## ABSTRACT

This paper proposes an optimistic protocol for certified email. It satisfies most of the required certified email properties using Two Stage Random number Generator (TSRG ) cryptosystem. These properties include Fairness, Sending Receipt, Non-repudiation of origin, Non-repudiation of receipt, Authenticity, Integrity, Confidentiality, Timeliness, and Temporal Authentication. A built-in TSRG RNG is a distinguishable primitive in the TSRG cryptosystem design, where instantaneous real time One Time Pad (OTP) like data is generated. The TSRG RNG implements the simple idea of reseeding the RNG at unpredictable instants to an unpredictable state, creating a new RNG model before the attackers can acquire enough information to identify the current model. The protocol aims to combine TSRG security, easy implementation and feasible deployment. The paper contains an overview of the system (System Description and a detailed plan of work as well as the services that should be offered through a brief scenario). A detailed protocol, system implementation issues and other design decisions are discussed.

## 1. Introduction

Electronic mail allows users connected to the Internet to exchange messages containing text and/or attachment files. The simple interface of mail clients as well as the spreading diffusion of the Internet and its services has determined a large success for the email service. Nowadays, email has become the most widely used means in daily communication on the net because it is faster, less expensive and more convenient. Furthermore, it is considered to be a novel and an attractive communication channel. Business cards now include also email addresses in addition to normal contents. Due to its features, email is increasingly used in place of ordinary mail. For example, submissions of papers for publication in conferences or journals are usually done via email. However, the usage of email for official events creates some problems because, in its simplest form, the email service does not have many desirable features. Indeed, the email service is based on the Simple Mail Transfer Protocol (SMTP), which does not guarantee the delivery and the integrity of the messages. The content of the messages are stored and transmitted in plain text. A simple daemon program at the SMTP server can change the content of the email without being noticed by the sender or by the recipient. Furthermore, whenever an email message is received, there is no assurance on the identity of the originator of the message. RFC 2298 [Fajman, 1998] defines a MIME content-type for Message Disposition Notifications (MDNs).

Ordinary mail offers services such as sending and delivery receipts. The sending receipt and the delivery receipt are something that the sender can show to prove the message origin and destination. Moreover such receipts have also timestamps that prove the time of sending and reception. However, MDNs are not enough to satisfy all the properties usually guaranteed with ordinary mail because they can be easily forged.

Certified email tries to deal with these problems using certified email protocols. Distributed protocols for certified email can be divided into two groups: optimistic and in-line protocols [Shnier, 1998] [Abadi, 2002]. In both cases, there is a Trusted Third Party (TTP), which is an "entity" ensuring the fairness of the protocol. In inline protocols, the TTP is actively involved in each message exchange, while in optimistic protocols the TTP gets involved in case of dispute. Certified email protocols have to guarantee several standard properties like **Fairness, Sending Receipt, Non-repudiation of origin, Non-repudiation of receipt, Authenticity, Integrity, Confidentiality, Timeliness, and Temporal Authentication.**

Recently many researches have been working on finding certified email protocols that satisfy the above properties, and several email protocols have been proposed. In in-line protocol, the TTP is actively involved in each message exchange. This means that the TTP has to be always available and capable of great performance. In optimistic protocols [Blundo, 2003] [Asokan, 1997] [Asokan, 1998], the sender and the receiver first try to exchange the message by themselves, without the intervention of the TTP and rely on the TTP only in the cases when a dispute arises. The proposed protocol uses also an on line Time Stamping Server (TSS) [Ateniese, 2001]. The TSS is able to certify the starting time of a transaction and provide the sender with a temporal mark which can be verified by the other parties.

Certified email protocols guarantee that a participant exchanges a message from a receipt, which the receiver should release at the end of the transaction. Indeed, the aim of such protocols is to provide a procedure for the secure exchange of messages, which is resistant to possible attempts of cheating by the participants. Beside security properties, such protocols should ensure that participants couldn't take any advantage of the procedure. Such protocols deal with the fair exchange of objects, i.e., at the end of the exchange, both participants get what they must have and nobody gets any other valuable information. Cryptographic techniques are used to obtain security properties on the messages exchange. Security services are protection mechanisms that enhance the E-mail security, and are able to detect, prevent, and/or recover from the previously mentioned threats. There is no single mechanism that provides all the security services.

Loss of life and fortune always follow the non-expert use of cryptography. For about three decades, three major cryptographic transforms have withstood the test of time. These transforms are One-Time-Pad, DES and RSA. Every few years, RSA key length must be duplicated to guarantee safe use which implies that the previously RSA encrypted messages are probably cracked by the current processing power and cryptanalysis algorithms. In the same time, RSA and similar asymmetric encryptions (e.g. Rabin, NTRU...) are based on hard to solve problems, which may be solved by unknown shortcut. DES has been cracked in few hours and the other symmetric transforms, like (IDEA, AES...), still have to be studied for a long time before a fair judgment on their strength. The one-time-pad (OTP) is the only up to date system that has been proven to be secure in the information theoretic sense. Another major variation is the use of pseudo-random number generation (PRNG) in place of the OTP. This is secure only as long as the PRNG is good enough to make the next bit unpredictable from all previous bits. This is similar to our objective of designing TSRG Cryptosystem. [Hussein, 2003, 1]

[Qiang, 2007] [Nugroho, 2006]. A built-in TSRG generator is a distinguishable primitive in the proposed cryptosystem design where instantaneous real time semi-true random data is created. Our previous published work proves that TSRG generated output is random and cannot be predicted using available technologies and mathematical theories under the condition that the state of the generator is not compromised [Hussein, 2002, 2]. Many well-behaved pseudo random generators can be adapted to be plugged in as TSRG generator primitive. TSRG Cryptosystem does not require sophisticated environment or present communication overheads as in the case of the everlasting cryptosystem [Rabin, 2002]. The proposed technique requires suitable message formats, which the receiver utilizes to interpret received data. The message format depends on the mode of operation (interactive, off-line) and the type of application to be used in [Hussein, 2003, 3]. The uncertainty in determination of the format and location of an encoded message, within basic random data, presents additional challenges for an attacker. This hybrid cryptosystem is based on an attack-oriented design appropriate to be **used in** a wide range of applications. E-Commerce, file transfer, E-mail clients, Access

control and VPN are typical Applications of the proposed cryptosystem. An illustration of the concept is shown in Figure 1.

The integration of the technology of ActiveX DLL, TSRG cryptosystem and De-facto standard MS Outlook and Exchange Server for realizing perfect Pluggable Certified Mail Service is the main objective of this project.

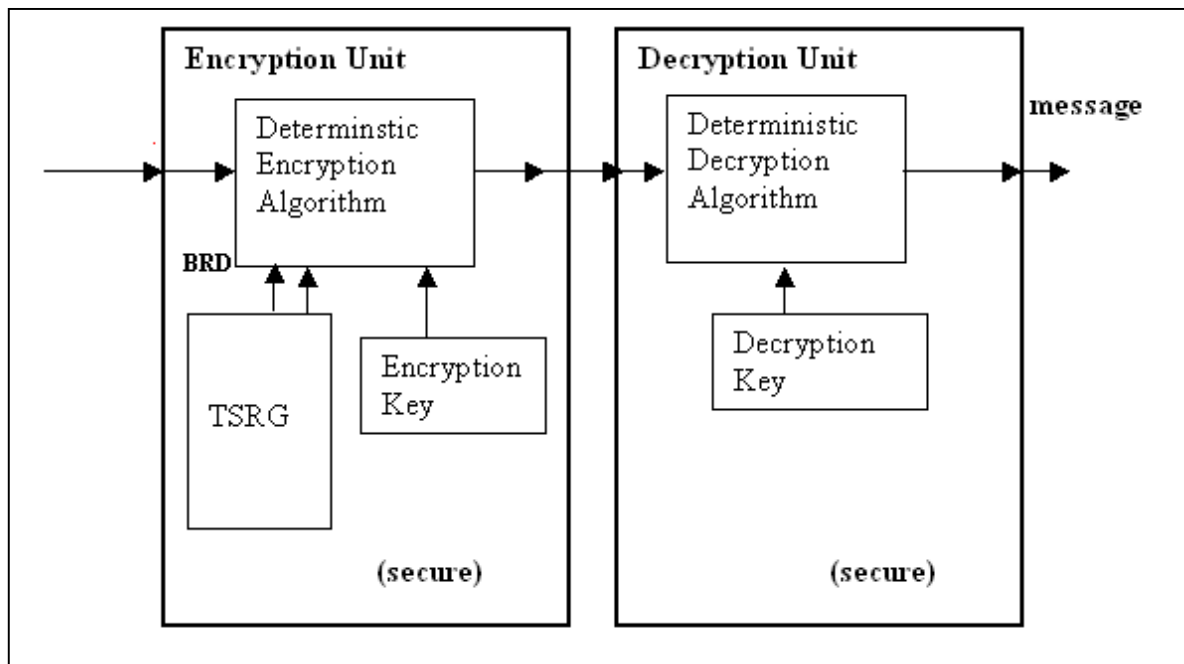


Figure 1: Block Diagram of TSRG Randomized Encryption.

## 2. Pluggable Certified Mail System

In this system, we propose a TCMS service that is designed to solve the fundamental problem of certifying message delivery and realizing the standard services of certified mail protocols. It uses the novel TSRG cryptosystem. In [Hussein, 2003, 2], one of the TSRG family member's mathematical model is derived then implemented. This model has a special cryptographic randomizer (IDEA in Special output feed back mode) cascaded with Lehmer generator. This slow generator is suitable for applications like Email where time is not the most significant issue [Hussein, 2002, 1]. Some drawbacks are discovered in this RNG like the leakage of the most significant bit of generator and the recovery of this bias requires sophisticated programming that degrades the RNG performance. Hence, the adapted generator have to be reselected and a new mathematical model must be built with all the derivations related to minimum length of attack, response time, repeatability and all other PRNG characteristics. Furthermore in [Hussein, 2002, 1], a Certified Pluggable Secure Email System is also designed and partially developed. In this system, the email client is developed from scratch, which prevents users from using their favorite email clients. This limitation suggests the development of our service as an extension of the de-facto standard exchange server and email client (i.e. MS SBS server and outlook). Consequently this must have an effect on the design of its

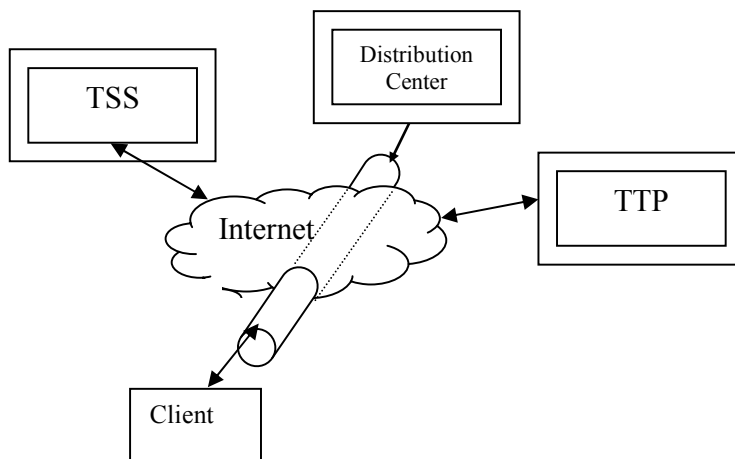
corresponding TSRG cryptosystem in both interactive and off-line modes. The design problem can be formulated as follow:

1. Enhance the TSRG by selecting one or more PRNG primitives and develop the corresponding mathematical model.
2. Design a TSRG cryptosystem to be used with the proposed protocol. It must realize the security services for the communication of the following:
  - a- The email client ActiveX and exchange mail server extension.
  - b- The email clients and TPP and TSS servers.
  - c- The email clients and the Distribution Center server.
3. Implement an immediate application for certified mail with the aid of TSRG cryptosystem.
4. Implement an application for securing downloading public keys and new fixes from the distribution center.

### 3. System Description and Detailed Plan of Work

The Objective of these system extensions is to securely exchange messages among parties with certification of delivery and the ability to dynamically upgrade any security primitive at the remote client. The main entities of this system are [Figure 2]:

- \* **The Trusted Transaction Party (TTP):** this server aids clients to communicate and to resolve in case of dispute.
- \* **The Time Stamping Server (TSS):** which certify the starting time of the transaction.
- \* **The Distribution Center Server (DC):** its responsibility includes Distribution of public keys and Upgrading client software
- \* **The Client workstation:** houses Email client application.



**Figure 2: TCMS Main Entities**

The use of the TSRG cryptosystem in TCMS has the objective of securing transactions between the clients and the (clients/servers). The protocol of the part that handles the relation between the client with TTP and TSS, is originally proposed by C. Blundo and others [Blundo, 2003]. In our

proposal we suggest major enhancements by using the TSRG cryptosystem instead of the customary hybrid cryptosystem used in their paper.

## TCMS Operational Protocol

TCMS has four types of message exchange (Client-Client, Client-DC, Client- TTP and Client TSS).

### 3.1 TTP-Client Protocol

The cryptographic primitives used in the protocol are:

- $\text{Sig}_A(\mathbf{m})$ : denotes the digital signature of the message  $\mathbf{m}$  using the private key of user A.
- $\mathbf{h}(\mathbf{m})$ : indicates the hash of message  $\mathbf{m}$  using some collision resistant hashing scheme.
- $\text{PK}_B(\mathbf{m})$ : denotes the encryption of message  $\mathbf{m}$  using the public key of user B.
- $\text{PK}_b^{-1}(\mathbf{m})$ : denotes the decryption of message  $\mathbf{m}$  with the private key of B.
- $\text{E}_k(\mathbf{m})$ : denotes the encryption of message  $\mathbf{m}$  using TSRG cryptosystem.
- $\text{ID}_S, \text{ID}_R$  are the source and destination identifiers respectively.

Suppose a sender S is trying to send a message to receiver R. The regular protocol between them is depicted as in Figure 3. The following section goes closely with [Blundo , 2003].

Since the protocol is optimistic, the interested parties (S and R), first try to communicate without the help of a trusted party TTP. If this phase completes without problems, the protocol terminates. If one of the parties is not satisfied by the communication exchanged in this phase, then it can start a recovery procedure involving the TTP. Hence we can identify in the protocol a normal behavior and a recovery behavior. In the normal behavior, the sender S needs first to get a timestamp on the message; then it exchanges messages with R in two rounds each consisting of two messages. If all the expected messages are received, the protocol terminates. Figure 3 provides a graphical description of the sequences of messages used in the normal behavior of the protocol. Let us denote by  $m_0$  the original email that S wishes to send to R. As the figure shows, there are six messages involved. The first two are exchanged by the sender S and the TSS. These messages are needed to get a timestamp on  $m_0$ . In more details:

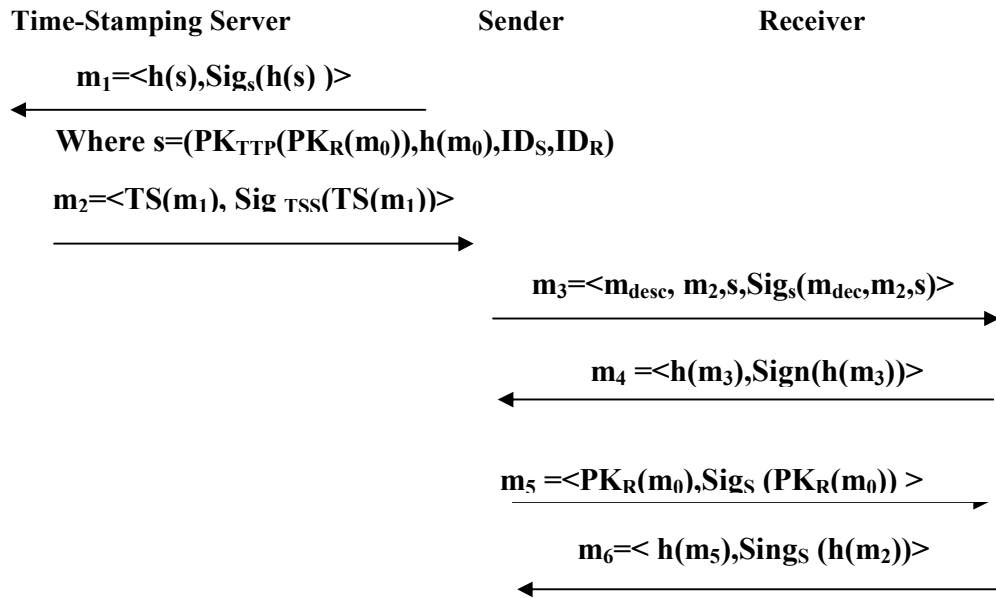
1. S sends to the TSS message  $m_1$  consisting of a hash and a signature of an encrypted piece of data, which in turn consists of the original email  $m_0$  , a hash of  $m_0$ , and the identifiers of the sender S and the receiver R.
2. The TSS replies with a message  $m_2$  consisting of a temporal mark and a signature of the received message  $m_1$ .

Having obtained a timestamp on its original email  $m_0$ , the sender is now ready to start interacting with the receiver. The four messages involved in the communication are explained below.

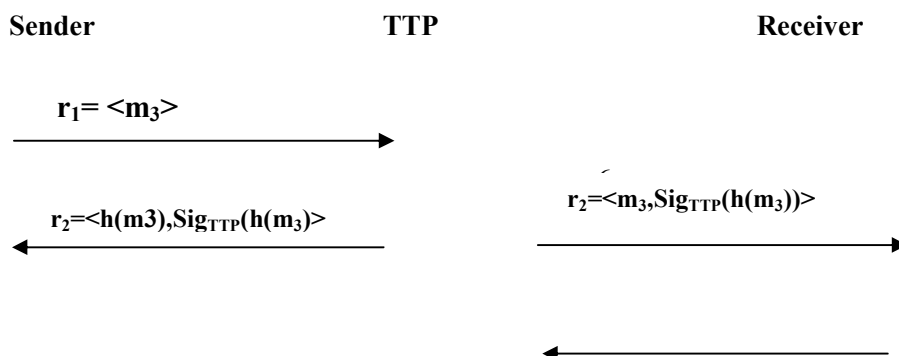
1. The sender S sends the message  $m_3$ , consisting of a description of the original message  $m_0$  (e.g., the subject of the email), the message  $m_2$  received by the TSS, and an encrypted piece of data that contains the original email  $m_0$ , the hash of  $m_0$  and the names of the sender S and the receiver R. At this point, the sender expects to receive a reply from R by a specific time. If a reply is not received by this time, then the recovery protocol, denoted by  $R_{1s}$  , is started as shown in figure 4.

2. Upon reception of the message  $m_3$ , R replies with message  $m_4$  consisting of a hash and a signature of  $m_3$ . The receiver now expects to receive another message from S by time within a time limit. If no reply is received, the recovery procedure denoted by  $R_{1R}$  is invoked by the receiver.
3. Upon reception of  $m_4$ , the sender S sends message  $m_5$ , consisting of an encryption of the original email  $m_0$  with the public key of R, and of a signature of S of such an encryption. The sender now expects to receive an acknowledgment from R by an expected time limit. If no reply is received, the recovery protocol  $R_{2S}$  is invoked.
4. Upon reception of the message  $m_5$ , the receiver is able to decrypt the message with its private key and thus obtain the original email  $m_0$ . The receiver replies with an acknowledgment message  $m_6$  consisting of a hash and a signature of  $m_5$ .

If all the expected messages are received within some given time bounds, then the protocol terminates without the need of further processing. However, if either the sender or the receiver does not get an expected message within the given time bound, then a recovery procedure is invoked. A message can be missed due to network congestion that delays the delivery of a message or to an attempt to cheat. In this case, the sender and the receiver are not satisfied therefore the intervention of the TTP is necessary to resolve the dispute over the email being sent. We remark that the channels between the TTP and the clients' nodes of the system are reliable, in the sense that the delivery of a message is guaranteed.



**Figure 3: The Normal Main Protocol**



## **Figure 4: The Recovery Protocol $R_{s1}$**

### **3.2 Client-Distribution Center Message Exchange**

The clients communicate with the DC to ask for downloading new DLL security modules or the public key of other client. The exchanged messages between clients and DC can be implemented using Windows Socket programming [Quinn, 1996] or more safely using VPN tunnel. An On-Line version of the TSRG cryptosystem have to be designed to fulfill the required security services in both cases.

### **3.3 Client-TTP-TSS Message Exchange**

In section 3.1 , TTP-Client protocol is proposed which guarantees fair certified mail exchange.

## **4. TCMS Implementation**

A prototype implementation of the proposed protocol on Windows platform has to be developed in order to test the usability of the approach. In particular, the implementation relies on the development of a plug-in for Microsoft Outlook and of an extension to Exchange Server. Com-Add-Ins let developers extend Microsoft Outlook beyond its native capabilities. After the installation of the plug-in, users can continue to use the client application (MsOutlook) as usual for normal email message or decide to compose or read a certified mail messages. In this case all the messages exchanged during the phases of the protocol are automatically handled by the plug-in and users are requested only to confirm the actions. For the server side, an extension to MS Exchange Server has also to be developed through the generation of a DLL ActiveX which reacts to the reception of certified mail messages and generates the requested messages needed to execute the recovery protocols. To implement the server side of the certified mail service, we relied on the event model which is provided with the Web Storage System used by Microsoft Exchange Server. The plug-in (based on the design of a COM Add-In, which is an ActiveX DLL) is able to interact with applications coming with the Office package.

The plug-in is activated whenever the user decides to compose a new message. At this time a form is displayed on the screen asking the user if he/she wants to use the normal or the certified mail service. In the second case, the generation of a certified mail message is started by computing the necessary information and attaching them to the message or adding them to the message header. Some user-defined fields can be added to the message header besides the standard ones in order to

distinguish certified email messages and to hold the additional values which are contained into the message.

Whenever the Receiver gets a certified mail request from the Sender, a form containing the Sender and the Subject of the email is displayed, and he is asked to refuse or accept the mail. In the first case, the message is deleted from the Inbox. In the second case, the attachments are saved on the disk, a verification of the integrity and the authenticity of the message is performed, and both timestamp and time limit are checked. Finally a new message holding the key request is composed and sent to the Sender. In the normal case, the execution of the protocol continues automatically and messages are exchanged between the Sender and the Receiver till the end of the protocol. A timer, which is initialized whenever Outlook is started, checks that the expected messages are received within the time limit fixed for each phase of the protocol. If this is not the case, the corresponding recovery procedure is invoked automatically and the needed messages are composed and exchanged with the TTP to end the execution of the protocol.

Another Task of the Plug-In is to securely communicate with Distribution Center sever to get the public key of other users or to download new fixes as described before.

To implement this project , the following setup is required:

- 1- MS small business server
- 2- TTP server
- 3- Distribution Center Server
- 4- Minimum two clients computers

## **5. Project Stages Proposal**

The stages of the project are proposed to be in two stages as follows:

### **The first stage activities include:**

- 1 - Analysis of the problem using mathematical foundation (the problem components and their relationships must be defined... ). This will cover studying certified emails protocols
- 2 - Selection of hardware platforms and software environment (operating system, languages, and other tools).
- 3 - Enhance the TSRG generator to match certified email requirements.
- 4 - Design of TSRG cryptosystems to be used in client-client protocol in normal operation.
- 5 - Implantation of ActiveX Com-Add-in DLL application with user defined forms and without recovery protocols.
- 6 - Testing and validation of the implemented system components.

### **The Second stage activities include:**

- 1 - Design of the TSRG cryptosystem to be used in the protocol between the client and the TTP and TSS servers in the recovery modes.
- 2 - Implementation of the TCMS including the recovery protocols.

3- Testing and validation of the implementation in the previous step.

**The Third stage activities include:**

1 - Design of the On-Line TSRG cryptosystem to be used in the protocol between the client and the Distribution Center Servers.

3 - Implementation of the full TCMS system.

4- Testing and validation of the implementation in the previous step.

**Conclusions and Future work**

We have presented a proposal of an optimistic protocol for certified email. It satisfies most of the required certified email properties using TSRG cryptosystem. These properties include Fairness, Sending Receipt, Non-repudiation of origin, Non-repudiation of receipt, Authenticity, Integrity, Confidentiality, Timeliness, and Temporal Authentication. The protocol relies on an on-line trusted third party server as well as a time stamping server. A Distribution Center server is also assumed to support distribution of public keys and downloading security Plug-Ins. To achieve the previous tasks, a new TSRG must be developed and its mathematical model must be derived. The corresponding TSRG cryptosystems have to be designed to secure communications between the clients and the different servers. While the implementation is assumed to be deployed on the de-facto standard mail client and server, it can be easily implemented in other environments.

We need to perform additional research to test the ability to use the TSRG Cryptosystem on certified emails. Data will be collected in a laboratory environment, followed by a small pilot if test results are encouraging. However, since the implementation schedules for such projects are fluid and subject to the impact of budgets and other pilot projects, this leads to the need for more work on interoperability, cooperation and coordination, all of which are suitable topics for continuing research and implementation

## References

- [Asokan, 1997] N. Asokan, M. Schunter, and M. Waidner. "Optimistic protocols for fair exchange", ACM Conference on Computer and Communications Security, 1997.
- [Asokan, 1998] N. Asokan, V. Shoup, M. Waidner, "Asynchronous protocols for optimistic fair exchange", In Proceedings of the IEEE Symposium on Research in Security and Privacy, 1998.
- [Abadi,2002] M. Abadi, N. Glew, B. Horne, B. Pinkas, "Certified email with a light on-line trusted third party: Design and implementation", In Proceedings of Eleventh International World Wide Web Conference, ACM Press, New York, US, 2002.
- [Ateniese, 2001] G. Ateniese, B. de Medeiros, M. T. Goodrich, "TRICERT: A distributed certified E-mail scheme", In Proceedings of the Symposium on Network and Distributed Systems Security (NDSS 2001), San Diego, CA, February 2001.
- [Blundo,2003] C. Blundo, S. Cimato, and R. De Prisco, "Certified E-Mail: Design and Implementation of a New Optimistic", IEEE Symposium on Computers and Communications - ISCC'2003, Turkey, June 2003
- [Fajman , 1998] R. Fajman, "RFC 2298. An extensible message format for message disposition notifications", March 1998.  
<http://www.ietf.org/rfc/rfc2298.txt> accessed 11/3/1998.
- [Hussein, 2002, 1] Hussein G., Dakroury Y., Hassan B., Badr A., "TSRG: Analysis and Design of a proposed RNG", DMS 2002, Sep. 2002, San Francisco, USA.
- [Hussein, 2002, 2] Hussein G., Dakroury Y., Hassan B., Badr A., "TSRG: Attack Oriented Design and Implementation ", SECI02, Sep. 2002, Tunis.  
<http://www.lsv.ens-cachan.fr/~goubault/SECI-02/Final/actes-seci02/pdf/003-ghussein.pdf>
- [Hussein, 2003,1] G. Hussein, Y. Dakroury, A. Badr, "TSRG Cryptosystem: Design and Implementation", In Proceedings of SETIT 2003, 17-21 March 2003, Susa, Tunis.
- [Hussein, 2003,2] G. Hussein, "Two Stage Random Generator Cryptosystem for Networks Applications Security", PhD thesis, Oct. 2003, Ain-Shams University, Faculty of Engineering, Cairo, Egypt.
- [Hussein, 2003,3] G. Hussein, M. W. David, "TSRG Randomized Cryptosystem", IEEE International Carnahan Conference on Security Technology (ICCST) ,October 2003, Taipei, Taiwan.
- [Shinier, 1998] B. Shinier, J. Riordan, "A certified E-mail protocol", In Proceedings of the 13<sup>th</sup> Annual Computer Security Applications Conference, 1998.
- [Quinn, 1996] Bob Quinn, Dave Shute, "*Windows Sockets Network Programming*", Addison Wesley Publishing Company, 1996.

**[Nugroho ,2006]** Nugroho Herucahyono ,”Study Design and Implementation *Stage Two Random Number Generator* “, *Engineering Program Information* , *Institut Teknologi Bandung*  
<http://www.informatika.org/~rinaldi/Kriptografi/2006-2007/Makalah2/Makalah-062.pdf>

**[Rabin,2002]** Yan Zong Ding, Michael O. Rabin: Hyper-Encryption and Everlasting Security. STACS 2002: 1-26

**[Qiang, 2007]** Qiang, Ping Lichun, WANG Chun-dong, “TSRG based on smart cards and security of authentication and access control”, "Journal of Tianjin University of Technology", 2007, 23 No. 06