

# Secure Mail Transfer Protocol (SecMTP)

Hathai Tanta-ngai, Tony Abou-Assaleh, Sittichai Jiampojarn, and Nick Cercone, *Fellow, IEEE*

**Abstract**—Simple Mail Transfer Protocol (SMTP) is a common protocol for transferring email messages worldwide. SMTP is anonymous and provides only the basic functions required to transfer a message, namely specifying a sender, recipients, and message text. SMTP was designed without consideration for security. We propose a new protocol, Secure Mail Transfer Protocol (SecMTP), to provide user-to-user secure email services with a seamless interface with existing SMTP clients. SecMTP is an extension service for SMTP. The SecMTP architecture addresses the major security goals: confidentiality, integrity, authentication, non-repudiation, and certification. We describe the SecMTP specifications and architecture, and compare it with the current mail transfer technologies.

## I. INTRODUCTION

Electronic mail (email) has become one of the most common methods of communication in the 21st century. The main reasons behind the success of email are the wide availability, ease of use, and affordability. Simple Mail Transfer Protocol (SMTP) [1] is the primary protocol for transferring email messages worldwide. As the title indicates, SMTP was designed to be as simple as possible. Thus, it incorporated only the most basic functions required to transfer a message, namely specifying a sender, recipients, and message text.

SMTP is anonymous by nature. It does not require the sender to authenticate itself. The protocol relies on the sender to specify to the receiver identification information. As a result, it is trivial for the sender to hide its identity. Further, the message is transferred from source to destination in plain text without any protection. It is fairly simple to intercept a message and view its contents, as well as modify the contents and the sender identification data. In other words, security was not taken into consideration in SMTP.

There is an ever-growing modern communications need for a mechanism that combines the simplicity, availability, and affordability of email with security. There are five major security goals that are significant in email communication: confidentiality, integrity, authentication, non-repudiation, and certification. Confidentiality protects a message from eavesdropping.

Confidentiality is important at the level of user-to-user communication and at the level of node-to-node communication. The former one guarantees that the intermediate nodes that are used to deliver the message from source to destination are unable to access the contents of the message. The node-to-node confidentiality protects the message from eavesdropping while being transferred from one node to another on a network.

In some cases, confidentiality might not be as important as integrity. The message might not contain sensitive information, though it is crucial that it is delivered untempered with.

Authentication is important for several reasons. The sender may require assurance that only the intended receiver actually

receives the message. Similarly, the receiver may have policies regarding the acceptance and rejection of messages based on the sender's identity. Sender authentication is important to give the validity to the content of the message.

Non-repudiation prevents the sender from denying sending the message. Similarly, it prevents the receiver from denying receiving the message.

Certification means receiving credibility from a trusted third party regarding the identity of the sender and the receiver, as well as certifying that a particular server conforms to a set of security requirements for sending secure messages.

A number of extension services for SMTP (ESMTP) [1] exist that attempt to partially address some of the security issues, such as SMTP over Transport Layer Security (TLS) [2], and Authentication [3] service extensions. We provide an overview of these extensions in section 2. The use of these extensions is optional. Thus, they are based on best-effort and operate at the node-to-node level. They do not provide any guarantees for user-to-user level security as the message may pass through intermediate nodes that do not support these extensions.

We propose a new protocol—Secure Mail Transfer Protocol (SecMTP), which is an extension service for SMTP. The goal of SecMTP is to provide guaranteed services for user-to-user security by ensuring that messages travel only through compliant secure nodes. SecMTP addresses the five security goals mentioned earlier. It uses existing extension services whenever possible to avoid reinventing the wheel and introducing conflicting standards, and defines new functionality for the features that are not currently standardized as an SMTP extension service.

The rest of the paper is organized as follows. Section 2 discusses papers, protocols, and standards related to secure mail transfer technologies. Section 3 introduces the Secure Mail Transfer Protocol; its specifications, and its architecture. Section 4 compares SecMTP with the existing secure mail transfer technologies and discusses the advantages and the shortcomings of SecMTP. Section 5 summarizes our results and section 6 states our future work directions.

## II. RELATED WORK

There are many current technologies that provide the mechanism to achieve some degree of security for transferring email over internet. We discuss those technologies in terms of confidentiality and integrity, authentication, notification, and non-repudiation. At the end, we discuss user and web applications that provide services for secure email.

### A. Confidentiality and Integrity

SMTP Service Extension for Secure SMTP over Transport Layer Security [2] provides a mechanism for communicating

over a secure socket by using the Transport Layer Security (TLS) Protocol Version 1.0 [4]. TLS guarantees confidentiality and integrity of the data stream between two nodes. Server authentication is done by using certificates that certify the domain name and ownership of the server to the client. The drawback of this extension is that the message is only secure if all the intermediate links also use TLS. If any of the intermediate nodes does not support the TLS extension service, the sender does not have the ability to request that the message is delivered using secure connections only. Moreover, even if the message is communicated from the sender to the receiver by the server using a TLS connection, the message is secured only between communication channels. At the nodes, the message is stored in a standard plain-text format. As a result, this mechanism does not protect the message if the server is compromised.

### B. Authentication

SMTP Service Extension for Authentication [3] allows SMTP client to authenticate itself to the SMTP server, and perform an authentication protocol exchange. This extension is a profile of the Simple Authentication and Security Layer (SASL) [5]. The authentication protocol exchange consists of a series of server challenges and client answers that are specific to the authentication mechanism. This SMTP Service Extension provides the authentication between SMTP client and SMTP server not between sender and receiver. Hence, there is no information for sender and receiver to ensure the identity of each others. As a result, the authentication is not satisfied in the user level. Moreover, this mechanism is indicated by the user, server can not guarantee the user authentication if they do not request this service.

### C. Non-repudiation

Sender non-repudiation can be easily guaranteed by using digital signature. If the sender signs a message, the receiver can use this signature to verify and prove that the sender in fact sent the message. However, to prove that the reader accessed a message required a trusted third party (TTP). There are three scenarios for non-repudiation using TTP: on-line TTP, off-line TTP, and hybrid systems. In the on-line TTP scenario [6], the sender sends a message and a digital signature to a TTP. The TTP requests a key and digital signature from receiver. When the receiver satisfies the request, the TTP sends the message to the receiver and a report of delivery to the sender. A number of drawbacks arise in the approach. First, the TTP must always be online waiting for messages. Second, TTP becomes a bottleneck since every single message from sender to receiver is sent through the TTP. And lastly, the TTP can learn the contents of the message without authorization. The light on-line TTP scenario [7] addresses these issues. The sender sends an encrypted message directly to the receiver. The receiver must send a digital signature to the TTP in exchange for the key to decrypt the message. TTP forwards the digital signature to the sender.

The off-line TTP scenario [8] is an optimistic approach. It is based on the assumption that the sender and the receiver can

honestly communicate most of the time without contacting a TTP. The sender encrypts the message with TTP's public key and sends it to the receiver. Then, the receiver sends its digital signature to the sender. Finally, the sender sends the original message (without encryption with TTP's key) to the receiver. If the sender does not send the message after receiving the digital signature, the receiver can contact TTP to decrypt the encrypted message, at which point a notification is sent to the sender, which eliminates the exploit where the receiver request a decryption without sending a digital signature to the sender.

TRICERT [9] is a hybrid method that combines the on-line and the off-line approaches. It distributes the task of the TTP optimistically to less trusted hosts and the TTP is contacted only if there is a dispute.

### D. User Applications

Security goal can be achieved by deploying specialized user applications. Pretty Good Privacy (PGP) [10] is the most popular freeware and commercial application that provides email security through plug-ins for most of the popular email clients. It uses public key cryptography and digital signatures to offer confidentiality, integrity, authentication and non-repudiation of sender. However, PGP does not have a mechanism to prove that the recipient read the message.

Privacy-Enhanced Mail (PEM) [11], [12], [13], [14] is a set of standardized procedure for providing secure email. It is based on public keys and signatures, as well as certificates. It requires a key management system such as a centralized key server (Kerberos) to manage keys or a Certification Authority (CA) to manage certificates. PEM is also unable to provide non-repudiation of receiver.

### E. Web Applications

Several web sites provide secure email service [15], [16], [17]. They use a combination of the TTP approach for non-repudiation of receiver and secure client-server communications (HTTPS) to ensure confidentiality and integrity. The sender connects to the web site using a secure connection and submits a message. The web site notifies the receiver of the URL where the message can be retrieved. The receiver can be asked to provide a password in order to view the message. The pitfalls of this approach lie in trusting the web site with the contents of the message. Also, being limited to using an Internet browser to send and receive messages is inconvenient.

## III. SECURE MAIL TRANSFER PROTOCOL

### A. Overview

SecMTP is designed to incorporate security procedures into SMTP while maintaining the simplicity and compatibility that SMTP provides. The aim of SecMTP is to achieve the five security goals of confidentiality, integrity, authentication, non-repudiation, and certification.

The communication channels used in the SMTP protocol can be classified into user-server channels and server-server channels. In SecMTP, the TLS is used for all the channels

to provide node-to-node confidentiality and integrity. User-to-user confidentiality is achieved through user public key encryption and decryption at the sending and receiving end servers. Authentication is derived from the Authentication Extension Service and is incorporated into the message header as a digital signature, which also provides message and header integrity, and sender non-repudiation for the user-to-user communication level. The SecMTP compliant server is required to validate and sign the header that it adds. Both sender and receiver non-repudiation are achieved by using one of the TTP scenarios depending on the network structure. A procedure based on certification by a CA is used to identify secure servers. The incorporation of security into the message transfer is transparent to the user. The user may use an existing SMTP email client, such as Microsoft Outlook, to send secure message.

### B. Assumption and Limitation

We designed SecMTP's architecture, protocol specifications, and SecMTP Extension Service to SMTP. The SecMTP compliant server and client are also designed but we do not provide their implementation. All SecMTP compliant servers must be properly certified. One of TTP scenarios for non-repudiation has to be implemented to achieve non-repudiation on both sides. It is assumed that the SecMTP user trusts the integrity of the sending and receiving end servers but not the intermediate connections and nodes.

### C. The SecMTP Architecture

SecMTP architecture is based on SMTP architecture with the extension of security services. The SecMTP server is an SMTP server which implements SMTP Extension Service for Secure Mail Transfer Protocol (SecMTP). The SecMTP client is an SMTP client that supports SecMTP options. SecMTP server provides services for non-SecMTP clients via designated SecMTP port. A SecMTP client can connect to a SecMTP server via a SecMTP port a regular SMTP port, or an SMTP over TLS port. Figure 1 shows the basic structure of SecMTP connections between mail clients and SecMTP servers. The server-server connections are communicated as client/server in the same manner as SMTP. From node to node, The source machine is designed as a client and the destination machine is designed as a server. The certification is required in all SecMTP compliant servers. The security mechanisms are indicated in the user-to-server and server-to-server level. As a result, it implicitly achieved guarantee services in the user-to-user level. Next, The details of security mechanisms over SecMTP are discussed.

### D. The SecMTP Specification

Every communication channel in SecMTP uses TLS to guarantee confidentiality and integrity from user-to-server and server-to-server. SMTP Service Extension for Authentication is automatically required by all communication channels and the authentication header will be automatically added to the message by all intermediate servers with a digital signature.

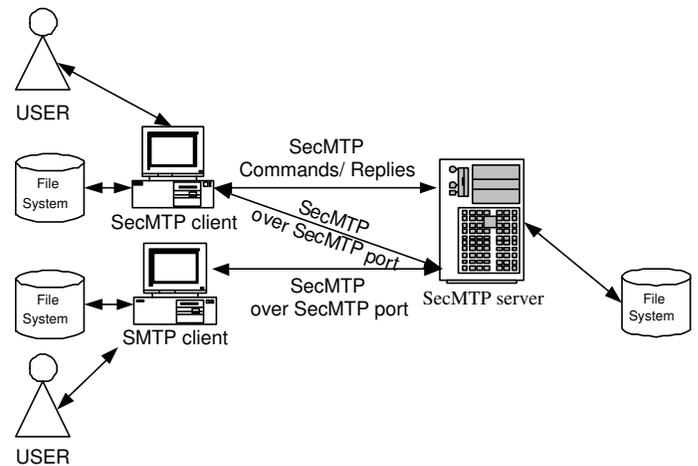


Fig. 1. SecMTP Basic Structure.

As a result of using TLS channels and authentication headers with trusted servers, SecMTP implicitly provides the user-to-user level of confidentiality, integrity, authentication and sender non-repudiation. The non-repudiation on both sides is achieved by using one of the TTP scenarios depending on the network structure.

SecMTP provides another level of security for users to explicitly request the SecMTP server to encrypt the message with receiver public key. SecMTP can also add a digital signature with sender private key before sending the message to TLS channel. To provide a seamless interface with users, including non-secMTP clients, the users private/public keys are stored at the server machine. SecMTP server will retrieve those keys for security operations with the user authentication mechanism such as login and password. This mechanism protects a message in the server and intermediate nodes. Digital signature with sender private key provides higher level authentication for the receiver.

Figure 2 depicts the timing diagram of a complete set of SecMTP connections from the sender to the sender's SecMTP server, to the receiver's SecMTP server, and finally to the receiver. The rectangular blocks indicate the operations at the server. The dashed rectangular blocks refer to optional operations.

To remain compatible with current systems, a SecMTP client connects to a SecMTP server through the standard SMTP port 25 by first establishing a TLS connection and then authenticating using SASL; or by starting a TLS connection on port 465 and then authenticating. To use a regular email client with SecMTP, the client has to connect to a predefined SecMTP port (e.g., 416). The SecMTP server must recognize this action and enable implicit use of SecMTP options. However, with this mechanism, the SecMTP will provide the default security level as defined by the server policy. Figure 3 illustrates the state diagram of starting a SecMTP connection. The transition edges show the commands that the client sends and the successful replies of the server. A SecMTP client can connect to SecMTP port for predefined setting and switch to user setting SecMTP by issuing the SECMTPT command.



at TCP port 25. The SecMTP server accepts the connection and waits for the client to start TLS negotiation followed by authentication. After the authentication succeeds, the client starts sending SecMTP commands. In this example, the client requests to add his digital signature to the message.

*Example 1:* :messages exchanged between SecMTP server and SecMTP client.

```
S: <waits for connection on TCP port 25>
C: <opens connection>
S: 220 foo.com SMTP service ready
C: EHLO bar.com
S: 250-mail.foo.org offers a warm hug
of welcome
S: 250 STARTTLS
S: 250 SECMTTP
S: 250 AUTH CRAM-MD5 DIGEST-MD5
S: 220 Go ahead
C: STARTTLS
C & S: <negotiate a TLS session>
C & S: <check result of negotiation>
C: EHLO
S: 250 ... AUTH CRAM-MD5 DIGEST-MD5 ...
C: AUTH CRAM-MD5
S: 334 ... C & S: <authentication session>
S: 235 authentication successful
C: EHLO
S: 250 ... SECMTTP ...
C: SECMTTP
S: 220 welcome SecMTP service ready
C: MAIL FROM:<Smith@bar.com>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: RCPT SIGN
S: 250 OK Digital Signature for Jones@foo.com
C: DATA
S: 354 Start mail input; end with
<CRLF>.<CRLF>
C: Data data data...
C: ...etc. etc. etc.
C: .
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission
channel
```

SecMTP can adapt to control different levels of email security in the organization without any action by the users of both, SecMTP clients and non-SecMTP clients, by using SecMTP default settings. With the default settings, The SecMTP server will automatically set the security of email following the user or server profile, such as add RCPT SIGN if there is a default setting for digital signature to email messages. With this mechanism, SecMTP allows users and servers to set different policies for different levels of security services depending on the implementation. For example, the organization can set the policy for SecMTP server to add only digital signature to all emails inside the organization and include message encryption when emails are sent outside the organization.

#### IV. DISCUSSION

SecMTP provides the security services for transferring email over Internet. The main advantages of SecMTP over the

current secure email technologies are the ease of use and the guarantee of security services. SecMTP provides transparent security services to users. It enhances the predefined security service extensions for SMTP which are TLS, and Authentication, with the guarantee of services. While providing the same goals as PGP and PEM, SecMTP does not require the user neither to handle security operations by themselves nor require additional software. In comparison to secure email web applications, SecMTP users are not limited to use an Internet browser to send and receive messages. SecMTP allows the users and the server to set different policies for different levels of security services.

The advantages and the shortcomings of SecMTP are discussed below.

##### A. Advantages

SecMTP provides a seamless integration with existing systems. It is compatible with SMTP and current service extensions such as SMTP over TLS and Authentication extension services. Moreover, it does not require any specific action from the users. SecMTP provides user-to-user level of security, as well as user-to-server and server-to-server security. SecMTP can provide both best-effort security and guaranteed security.

##### B. Shortcomings

Security information of a message is stored in the header. Users must manually examine the header to access these information in non-SecMTP email clients. Non-repudiation is provided at the server level, not at the client level. Encryption and decryption are done at the server. Users must trust the end servers to provide security services. The SecMTP servers may become a bottleneck depending on the number of messages being processed and on the level of security they use. Moreover, SecMTP compliant email clients and servers are required in order to benefit from the full power of SecMTP.

#### V. CONCLUSION

We introduce a novel protocol—Secure Mail Transfer Protocol (SecMTP). SecMTP is defined as an extension service for SMTP. It provides a seamless operation for standard mail clients. SecMTP provides the security goals of confidentiality, integrity, authentication, non-repudiation, and certification for SMTP. It achieves the five security goals at various levels of user and server interactions:

- Authentication between user and server using the Authentication extension service.
- Authentication and certification between servers.
- Confidentiality and integrity of communication channels through the use of TLS channels.
- User-to-user confidentiality using public key encryption
- User-to-user authentication and integrity using digital signatures.
- Sender non-repudiation using digital signatures
- Both sender and receiver non-repudiation are achieved by using one of the TTP scenarios depending on the network structure.
- Guarantee security service using the STRICT delivery mode.

## VI. FUTURE WORK

Our goal is to implement a full-scale SecMTP server based on the SecMTP specifications, and architecture. The SecMTP server will support the SMTP over TLS, Authentication, and SecMTP extension services of SMTP. In addition, the server will have an underlying infrastructure that permits management of keys and certificates, and a mechanism for TTP non-repudiation.

## ACKNOWLEDGMENT

The financial support for this work was provided in part by the Natural Sciences and Engineering Research Council of Canada (NSERC)

## REFERENCES

- [1] J. Klensin, "Simple mail transfer protocol," RFC 2821, Apr. 2001.
- [2] P. Hoffman, "Smtplib service extension for secure smtp over transport layer security," RFC 3207, Feb. 2002.
- [3] J. Myers, "Smtplib service extension for authentication," RFC 2554, Mar. 1999.
- [4] T. Dierks and C. Allen, "The tls protocol version 1.0," RFC 2246, Jan. 1999.
- [5] J. Myers, "Simple authentication and security layer(sasl)," RFC 2222, Oct. 1997.
- [6] J. Zhou and D. Gollmann, "Certified electronic mail," in *ESORICS'96*. Springer-Verlag, 1996, pp. 160–171.
- [7] B. H. BM. Abadi, N. Glew and B. Pinkas, "Certified email with a light on-line trusted third party: Design and implementation," in *The 11th International World Wide Web Conference (WWW2002)*, May 2002.
- [8] B. Schneier and J. Riordan, "A certified e-mail protocol," in *13th Annual Computer Security Applications Conference*, Dec. 1998, pp. 347–352.
- [9] B. M. G. Ateniese and M. Goodrich, "Tricert: A distributed certified e-mail scheme," in *ISOC 2001 Network and Distributed System Security Symposium (NDSS'01)*, Feb. 2001.
- [10] (2003, Apr.) The pgpi project, the international pgp home page. [Online]. Available: WWW URL: <http://www.pgpi.org/>
- [11] J. Linn, "Privacy enhancement for internet electronic mail: Part i," RFC1421, Feb. 1993.
- [12] S. Kent, "Privacy enhancement for internet electronic mail: Part ii," RFC1422, Feb. 1993.
- [13] D. Balenson, "Privacy enhancement for internet electronic mail: Part iii," RFC1423, Feb. 1993.
- [14] B. Kaliski, "Privacy enhancement for internet electronic mail: Part iv," RFC1424, Feb. 1993.
- [15] I. Authentica. Authentica.
- [16] (2003, Apr.) Certifiedmail security solutions. [Online]. Available: WWW URL: <http://www.certifiedmail.com>
- [17] Z. Inc. Ziplip secure and control your email.

**Hathai Tanta-ngai** is a Ph.D. candidate in the Faculty of Computer Science at Dalhousie University, Halifax, Nova Scotia, Canada. Her research interests are on distributed systems, distributed and parallel computing, peer-to-peer systems, and Internet privacy. Tanta-ngai received the B.Eng. degree in Computer Engineering from King Mongkut's University of Technology Thonburi, Bangkok, Thailand, in 1997, and the M.Eng. degree in Computer Science from Asian Institute of Technology, Bangkok, Thailand in 1999.

**Tony Abou-Assaleh** is a Ph.D. candidate in the Faculty of Computer Science at Dalhousie University, Halifax, Nova Scotia, Canada. He received his B.Sc. in Computer Science degree with First-Class Honour from Brock University, St. Catharines, Ontario, Canada, in 2001, and his M.Math. in Computer Science degree from the University of Waterloo, Waterloo, Ontario, Canada, in 2003.

Abou-Assaleh research interests include natural language processing, HPSG, reasoning under uncertainty, machine learning, evolutionary computation, secure communications, and algorithmic problems.

**Sittichai Jiampojamarn** is an M.C.S. student in the Faculty of Computer Science at Dalhousie University, Halifax, Nova Scotia, Canada. He received his B.Eng. degree in Computer Engineering with First-Class Honour from King Mongkut's University of Technology Thonburi, Thailand, in 2002. His research interests focus on network security issues.

**Nick Cercone** is Professor and Dean of the Faculty of Computer Science at Dalhousie University, Halifax, Nova Scotia, Canada. Cercone's research interests include natural language processing, knowledge-discovery in databases, data mining, and design of human interfaces. He is the author of over 200 refereed publications and has graduated 60 graduate students in his career.

Cercone received the B.Eng. degree in Engineering Science from the University of Steubenville in 1968, the M.C.S. degree in Computer and Information Science from Ohio State University in 1970, and a PhD degree in Computing Science from the University of Alberta in 1975. Cercone worked for IBM Corporation in 1969 and 1971 on design automation.