

Centrally-Secured Email and Non-Repudiation

Neeraj Chaudhry, Aaron Arthurs, Rajesh Akula, Sucharita Madala

Dept. of computer science and computer engineering

University of Arkansas

nchaudh@uark.edu, ajarthu@uark.edu, rakula@uark.edu, smadala@uark.edu

Abstract

In order to solve the problem of establishing a secure communications between the two end users that do not trust each other, two mail servers function as trusting parties. These mail servers consider all the security issues that are involved in a secure environment, uses well known encryption ciphers and digital signature to solve security problem and Non-repudiation. The goal is to provide a secure means of communication between the two end users, such that no user deny about the service they received or created also we have varied the file sizes used as messages that are being transferred between two the end users in order to compare the timing issues that are involved with each encryption algorithm.

We have explained the use of various cryptographic ciphers to acquire the necessary results and proved that each cipher has its pros and cons. Some are good for security and some are fast.

1. Introduction

Cryptography is a study of encryption or decryption of encoded data or ciphers so that it can be decoded by specific individuals. It can be characterized by: type of encryption operations used – substitution/transposition/product, number of keys used – single/double, way in which plaintext is processed – block /stream. Accordingly, two types of encryption are used: symmetric encryption and asymmetric encryption. [1]. Therefore, if data needs to be transferred safely and secretly from one end to other these encryption ciphers play an important role. Sending a message from one end to other end looks simple but it has to go across a number of systems and channels in order to reach at the other end safely, the systems can be mail servers, web servers and channels can be coax cable, fiber, electromagnetic radiations etc. This makes it easier for an intruder to eavesdrop from different locations and get access of the shared information without letting the users know about them and become a threat. Thus, this requires a need of certain encryption algorithms which are hard to crack but easy to use. Hence, the invention of DES algorithm set the path for future work and arrival of more secure and

fast encryption algorithms like 3DES, AES, BLOWFISH, RSA, etc.

Certain issues evolved for network security like maintaining control Access, Authenticity, Confidentiality, Data Integrity, Non-repudiation mainly for sending emails. [3]. and these are taken care of by using either of these encryption algorithm. These issues have started to increase so much as the world becomes tightly interconnected with the network devices as with the growth of low-cost broadband networking for homes and small enterprises. Organizations are feeling a great need of network security. [2]. The dependence on cyber infrastructure has increased so much that the researchers and experts have felt that there can be no financial transaction possible without it, even the larger infrastructure e.g., power grid, gas pipelines depend heavily on it. [4]. Various methodologies have been used by looking and thinking of the attack at the beginning stage of designing of security stages. [5].

The main goal of this project is by looking into various security issues to provide a secure means of communication between two non trusting parties which have their friends in between the communication to securely transfer the messages across the other end by taking the help of renowned encrypted ciphers used in today's world, compare the performance of these symmetric as well as signature algorithms depending upon the size of the message used for transmission, countering the denial-of-service attack with the help of digital signatures by providing certificates to the involved parties. We are using the Java JCE cryptographic package in our study.

Section 2 explains about the background of the various ciphers and their function, section 3 describes the explanation of the overall structure. Section 4 and 5 consists of results and conclusion. Section 6 consists of references and section 7 talks about the future work needs to be done.

2. Background

Cryptography is designed to solve various security problems like providing confidentiality, authenticity, non-repudiation, data integrity and for which various encryption ciphers are developed and used. For confidentiality symmetric cipher employed are DES, 3DES, AES, BLOWFISH, etc. Data encryption standard (DES) is developed by Edward Schaefer, supposed to be the fast and secure but with cryptanalysis and brute force attacks it is easily broken in very less amount of time, so it has given a way for other ciphers to arise such as 3DES, 3DES with 3 keys and AES, out of which AES is suppose to overtake because of its large key size, but is still in progress.[1]. Public key encryption cipher, RSA, developed by Rivest, Shamir, and Adleman [9], is used for secure communication by providing secret keys by which session keys are obtained to successfully generate a session between two end systems. For authenticity and non-repudiation, digital signature is a technique to sign the message or digital certificates are provided in order to resolve the dispute which can arise if a sender or receiver denies the service offered or received [7].

In further detail, digital signatures, in order to be eligible as non-repudiation evidence, should remain valid until its expiry date which is specified by some non-repudiation policy [6]. In the case of email, a full signature is composed of the signer's certificate, signed by a well-known certificate authority (e.g. Verisign), and the digital signature. The expiry date on the signer's certificate also applies to the expiry date on the digital signature. The digital signature is created by running the message through a secure, hash function (e.g. MD5 or SHA1) and encrypting the resulting hash value with the signer's private key via an asymmetric cipher (e.g. RSA). The signer's private key pairs with the public key stored in the signer's certificate.

3. Discussion

Fig.1. lays out the technique for Alice and Bob, two end users, to exchange messages. Alice and Bob do not trust each other so they have their trusting friends in the form of mail servers holding their mail boxes. It is through these trusting parties they are exchanging messages. In order to retrieve the messages, Alice and Bob connect to their mail servers through a secure channel (e.g. HTTPS) and from there they can get access to their mail boxes. It is these trusting friends which are providing all the necessary secrecy, confidentiality and authenticity by using each user's private keys and digital certificates. Mail servers talk to each other through a secure channel via secure shell client, SSH.

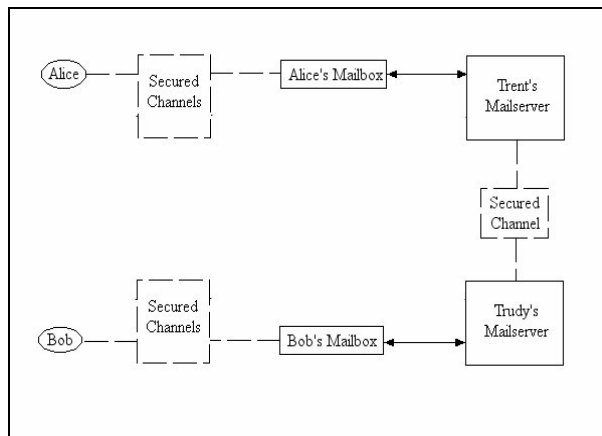


Figure 1. Overall structure of two trusting parties between end users.

Fig.2. shows how each mail server performs the security tasks; inside the mail server there are various components which perform their assigned tasks such as polling, maintaining a database of users, digital signing, and SSH:scp. The Poller (shown in Fig.3.), which is directly connected to each user's mailbox, continuously scans the users' mailboxes and retrieves the mails. When a user sends a message to another user in the same or different mail server, their message is first placed into their mailbox's send folder. The Poller will detect the mail, load the message(s) into memory, and clear the send folder. The sender takes the mail box's send folder messages and passes them on to the digital signer. Then the digital signer signs the message and encrypts them using the user's private key.

The user's data base connected to the poller consists of all the private keys and the certificates for each user. User Database here is maintained as a folder containing each user's certificate and RSA private key files, which are associated with the mail server. A mail server's administrator creates each user's certificate and private via, e.g. OpenSSL. In this project, certificates follow X.509 and are pem-encoded (private keys are also pem-encoded).

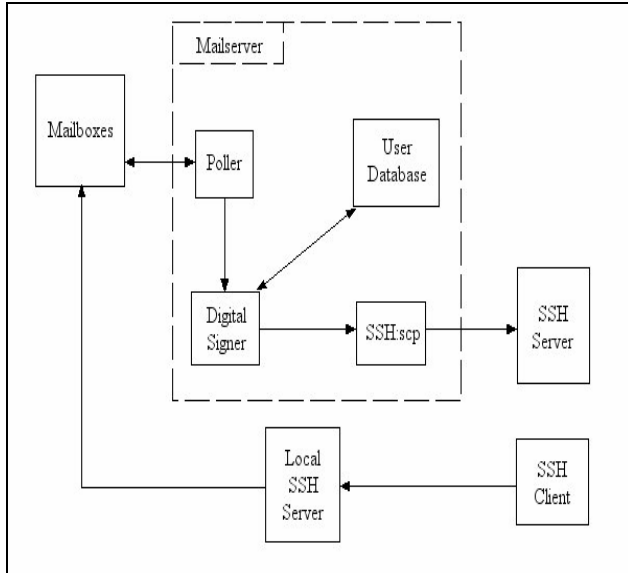


Figure 2. Mail server

In addition to maintaining each user's certificate and private RSA key, the mail server also maintains a list of trusted mail servers. More specifically, this 'trustees' list is represented by directory containing mail server certificates; mail server certificates are self-signed, which makes the mail servers root certificate authorities (CA). For instance, if Trent wants to trust himself and Trudy, Trent will store both his own and Trudy's certificate under a directory containing trusted certificates ('trent/trustees'). This trustee model is similar to how web browsers trust certain HTTPS sites by maintaining a list of trusted CAs.

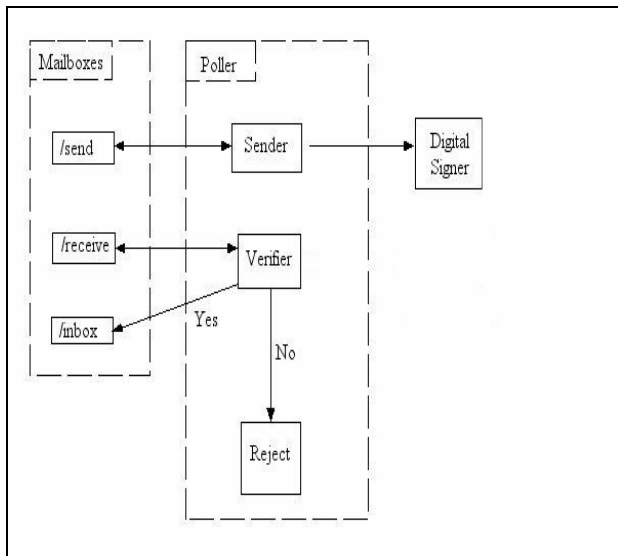


Figure 3. Poller

The steps taken when sending a message are as detailed. As an example, Alice, at Trent's mail server, wants to send a message to Bob, at Trudy's mail server. Alice creates a message file containing a subject, the intended receiver (bob@trudy, in this case), and the message itself. Alice saves the message in her 'send' folder ('trent/alice/send/message.txt'). Within X seconds, where X is the timed interval the poller runs, the poller will detect the message in Alice's 'send' folder, load it into memory, and remove the message file from Alice's 'send' folder. The poller then retrieves Alice's private key and certificate and adds to the message a timestamp, Alice's email address, a randomly-generated message ID, and Alice's certificate. Next, the poller signs the resulting message using the private key and a specified hash function, and saves the signed message to a temporary folder ('trent/queue'). With the signed message file in place, the poller creates an SSH session with Trudy's SSH server using a specified symmetric encryption cipher. For this project, Trent's public key (OpenSSH-encoded and stored in Trudy's authorized-key list), is used during SSH authentication. Once connected and authenticated, the poller opens an SCP client to transfer Alice's signed message into the 'receive' folder of Bob at Trudy.

On Trudy's side, Trudy's poller will detect Alice's signed message in Bob's 'receive' folder, loads it into memory, and deletes the signed message file. Next, the poller performs verification as detailed. First, Alice's certificate is extracted from the message. Then, the poller checks for Trent's certificate under the 'trustees' folder; if Trent's certificate is not listed in this folder, then the message will not verify. If Trent's certificate is found, then the poller will verify Alice's certificate against Trent's certificate, making sure Alice's certificate has been signed by Trent. If Alice's certificate verifies, then the poller will extract Alice's public key from her certificate; otherwise, the message is non-verifiable. Next, the poller decrypts Alice's signature in the message using Alice's public key. In the last stage of verification, the poller compares the computed hash value of the message against the decrypted hash value. If the two values match, the message verifies; otherwise the message does not verify and is placed into Bob's bulk folder ('trudy/bob/reject').

To complete Alice's message transaction (assuming her message verifies), Trudy's poller will add to the Alice's signed message a receive-confirmation flag and Bob's certificate. The poller then signs the resulting message using Bob's private key. Then, Trudy's poller sends the doubly-signed message back to Alice's 'receive' folder using SSH. Finally, Trent's poller will detect the doubly-signed message in Alice's receive folder and perform the same verification process as with Trudy on Bob's

certificate and signature. If the message verifies, the message is placed into Alice's sent folder ('trent/alice/sent'); otherwise, the message is placed into Alice's bulk folder ('trent/alice/reject').

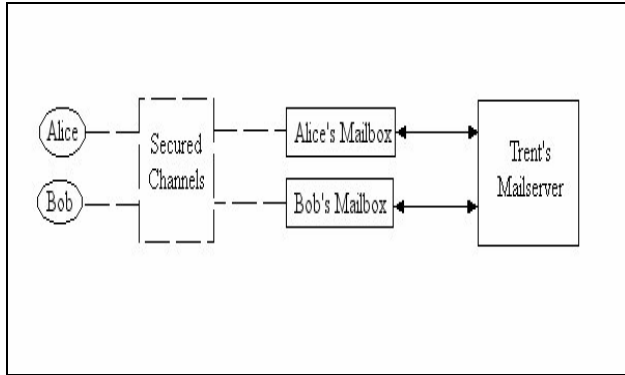


Figure 4. Only one trusting party between the users.

4. Results

RSA with SHA1 and RSA with MD5 algorithms were written, compiled and executed in Java using j2re1.4.2_08 (java runtime environment), j2ssh-0.2.7(SSH client), OpenSSL 0.9.7d(certificates and private key generation), OpenSSH_3.8p1(SSH server) packages for different symmetric encryption cipher's, like AES256, AES196, AES128, 3DES, BLOWFISH, in order to provide security, authentication and confidentiality, and are compared. All these issues have been considered and the goals have been achieved. We have achieved the secure communication and no involved party can deny about the service they received.

Below in various charts, we have shown the results that we acquired while changing the file size of the intervening message between two users. We used huge, large, medium, small and small initial files with sizes of 10,734KB, 1352KB, 76KB, and 1KB. 'Small initial file' is the timing result which we got when sever still has not included the encrypting ciphers. We found a difference using the same 1KB files which are named as small file and small initial file. We have received very interesting results while varying the file size, e.g. as shown in Chart1. Huge file, BLOWFISH symmetric AES128 with MD5, 3DES with MD5 and BLOWFISH with MD5 proved more secure and fast.

In chart 2, with Large file AES256 with SHA1 proves more secure and fast and as in chart 3, with medium AES256 with SHA1, AES196 with SHA1 and 3DES with SHA1 and AES256 with MD5 shows almost same results

but AES256 with SHA1 proved somewhat better in terms of speed. In chart 4, with small file AES 128 proved better from others.

In figure 5. We have explained which cipher is performing best in terms of time it is taking for the message to pass from user to one mail server and one from mail server to other mail server and from second mail server taking all the time for encryption , decryption and for signing a signature. AES cipher proves more secure and fast whether it's a small file or huge file. But in other cases other cipher also proved to be compatible with AES.

Symmetric Encryption Cipher	File Size (KB)	Time (msec)
AES 128 - MD5 , 3DES - MD5	10734	47922
AES 256 - SHA1	1352	7658
AES 256 - SHA1	76	1954
AES 128 - SHA1	1	1641

Figure 5. Comparing Best Cipher.

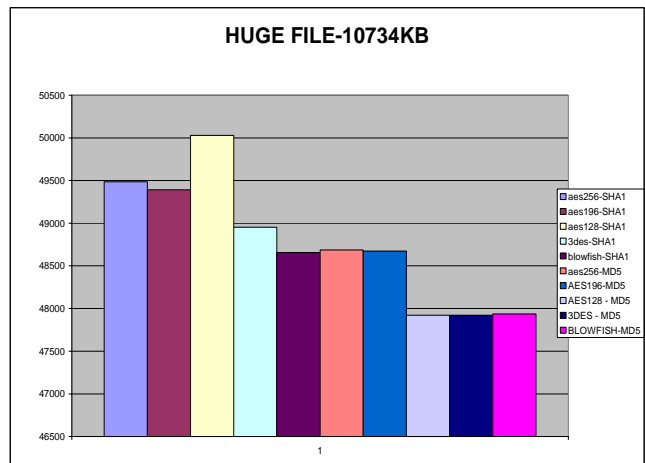


Chart 1. Huge file – (From Left to Right) AES256-SHA1, AES196-SHA1, AES128-SHA1, 3DES-SHA1, BLOWFISH-SHA1, AES256-MD5, AES196-MD5, AES128-MD5, 3DES-MD5, BLOWFISH-MD5

LARGE FILE-1352 KB

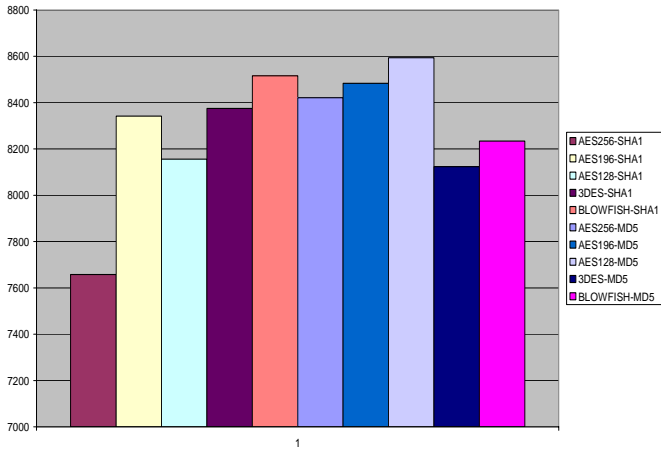


Chart 2. Large file - (From Left to Right) AES256-SHA1, AES196-SHA1, AES128-SHA1, 3DES-SHA1, BLOWFISH-SHA1, AES256-MD5, AES196-MD5, AES128-MD5, 3DES-MD5, BLOWFISH-MD5

SMALL FILE- 1KB

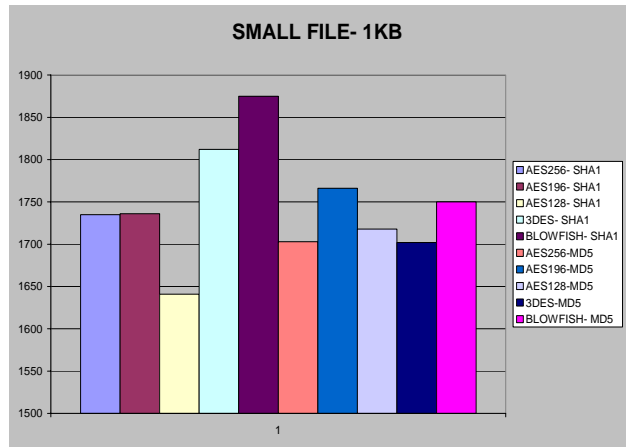


Chart 4. Small file(From Left to Right) AES256-SHA1, AES196-SHA1, AES128-SHA1, 3DES-SHA1, BLOWFISH-SHA1, AES256-MD5, AES196-MD5, AES128-MD5, 3DES-MD5, BLOWFISH-MD5

MEDIUM FILE- 76KB

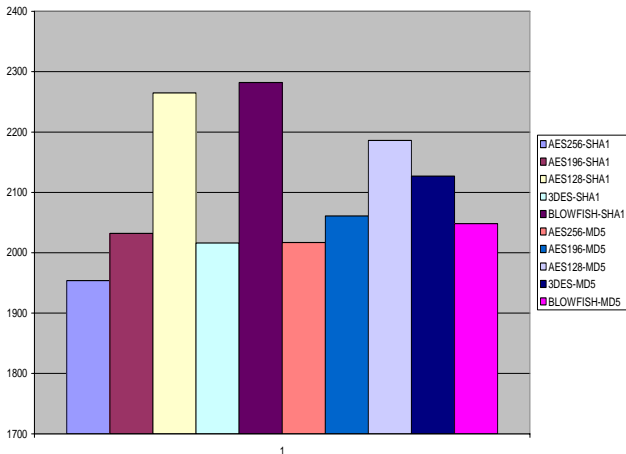


Chart 3. Medium file(From Left to Right) AES256-SHA1, AES196-SHA1, AES128-SHA1, 3DES-SHA1, BLOWFISH-SHA1, AES256-MD5, AES196-MD5, AES128-MD5, 3DES-MD5, BLOWFISH-MD5

SMALL INITIAL FILE- 1 KB

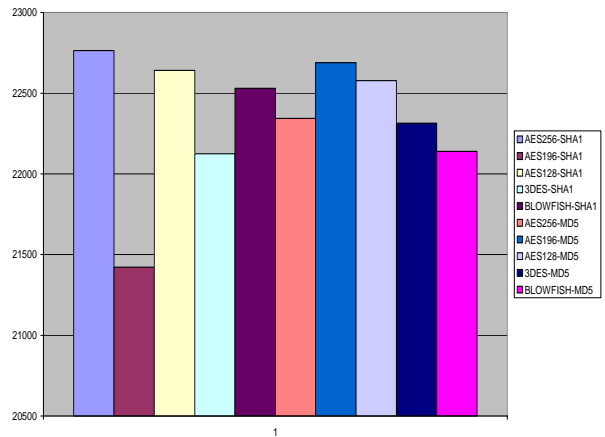


Chart 5. Small Initial file(From Left to Right) AES256-SHA1, AES196-SHA1, AES128-SHA1, 3DES-SHA1, BLOWFISH-SHA1, AES256-MD5, AES196-MD5, AES128-MD5, 3DES-MD5, BLOWFISH-MD5

5. Conclusions and Future work

Our goal of securing a communication between two untrusted parties is achieved using several encryption algorithm ciphers with digital signatures. Every message was signed by the help of trusting parties and the whole communication is saved from denial attacks. Two mail servers acting as trusting parties are controlling all the encryption and assigning digital signature to each untrusting party. These servers are connected via SSH client to securely communicate with each other. So the communication of messages between the end users and the servers is secure by some means. Already a lot of work has been done to compare the encryption algorithm by various researchers [10] and number of papers have been published on using these cryptographic schemes, but nothing has been done to focus on timing issues while simultaneously working to send the encrypted messages. In future, some protocol like SMTP is required between mail servers to communicate securely instead of using SSH which would be a better way for the servers to exchange messages securely.

6. References

- [1] W. Stallings, *Cryptography and Network Security: Principles and Practices* (2003), ISBN 0-13-091429-0.
- [2] P.W. Dowd and J.T. McHenry, "Network security: it's time to take it seriously", *Computer Volume* 31, Issue 9, and Sept. 1998 Page:24 - 28.
- [3] C.E. Landwehr and D.M. Goldschlag, "Security issues in networks with Internet access", *Proceedings of the IEEE Volume* 85, Issue 12, Dec. 1997 Page(s):2034 - 2051.
- [4] L.D. Paulson, "Wanted: more network-security graduates and research", *Computer Volume* 35, Issue 2, Feb. 2002 Page(s):22 -24.
- [5] D. Graft, M. Pabrai, U. Pabrai, "Methodology for network security design", *Communications Magazine*, IEEE Volume 28, Issue 11, Nov. 1990 Page(s):52 - 58.
- [6] Z. Zhou, R. Deng, "On the validity of digital signatures Magazine ACM SIGCOMM Computer Communication Review Volume 30, Issue 2 (April 2000) Page(s): 29-34.
- [7] R. Rivest, "The MD5 Message-Digest Algorithm", *RFC* 1321, April 1992.
- [8] D. Eastlake 3rd, P. Jones, "US Secure Hash Algorithm 1(SHA1)", *RFC* 3174, September 2001.
- [9] B. Kaliski, J. Staddon, "PKCS #1: RSA Cryptography Specifications", Version 2.0, *RFC* 2437 October 1998.
- [10] A. Dunn, "Environment-independent performance analyses of Cryptographic algorithms", *Proceedings of the twenty-sixth Australian computer science conference in research and*

practice in information technology - Volume 16, (2003)
Page(s): 265-274 ISBN - ISSN: 1445 - 1336, 0-909-92594- 1.

[11] T.Ylonen, "The SSH(Secure shell) Remote Login Protocol", *Internet Draft*, November 1995.