Chapter 16
Cryptography in Electronic Mail

Hosnieh Rafiee
Hasso Plattner Institute, Germany

Martin von Löwis
Hasso Plattner Institute, Germany

Christoph Meinel
Hasso Plattner Institute, Germany

ABSTRACT

Electronic Mail (email) is a very important method of communicating across the Internet, but the protocols used to handle emails during transmission, downloads, and organizational processes are not secure. Spammers and scammers misuse these protocols to propagate spam or scams across the Internet for advertising purposes or to gain access to critical data, such as credit card information. Cryptographic approaches are applied as a tool to help in securing email components, such as the header, data, etc. This chapter classifies the approaches used according to the protection mechanisms provided to the email components, and it also briefly describes these approaches. Because scammers are continually trying to crack current algorithms, the most recent improvements in email security using cryptography are covered in this discussion. An explanation is given as to the need for verifying both receivers and senders in this process. Finally, the authors examine how the use of these approaches will work in IPv6 as compared to IPv4.

INTRODUCTION

In computer terms, email (e-mail) is short for electronic mail. It is a current method of transmitting data, text files, digital photos, and audio and video files from one computer to another over the internet. This phenomenon did not become popular until 1990 and now it is a major business in personal communications. Compared to sending mail via the post office in the traditional way (snail mail), email is faster and cheaper. Messages can be sent at any time to anywhere and the recipient can read it at his or her convenience. The same message can be sent to multiple recipients at one time and the message can be forwarded without having to retype it.

Early email was not invented; it just evolved. Early email was just a small advance on what we know these days as a file directory—it just put a message in another user's directory in a spot where

DOI: 10.4018/978-1-4666-4030-6.ch016

Copyright © 2013, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

they could see it when they logged on. Just like leaving a note on someone's desk.

The first documented email system was MAILBOX, used at the Massachusetts Institute of Technology. Another early program used to send messages, on the same computer, was called SNDMSG (Tomlinson, 1971).

Some of the mainframe computers of this era might have had up to one hundred users - often they used what are called "dumb terminals" to access the mainframe from their work desks. Dumb terminals just connected to the mainframe—they had no storage or memory of their own and all work was done by the remote mainframe computer.

Today, a standard protocol called Simple Mail Transfer Protocol (SMTP) (Klensin, 2008) is used to send and receive mails and transport them across multiple networks (SMTP relay) by establishing a two-way transmission channel between a SMTP client and server over the internet or networks.

Here, two problems are encountered. The first is related to spamming. Spam mail is unsolicited email. It is also known as "junk" email that is typically not wanted by the user who is receiving it. The second is related to scamming. Scam mail is an email that is also unsolicited, but is attempting to acquire money or personal information from the recipient. Spammers and criminals profit from the use programs that misuse the SMTP protocol.

The U.S. Congress passed a law (15 USC Chapter 103, 2011) in 2003 that was designed to curb spam. This law makes it illegal to send messages that use deceptive subject lines and false return addresses, providing fines for as much as 6 million dollars and possible prison terms for violators. The law states that all messages, solicited or unsolicited, must have a valid postal address and an opt-out mechanism so that recipients can prevent future email solicitations. The email system is also vulnerable to hackers who can attach malicious programs to an email in hopes of infecting other computers whose resources they can then use in further attacking scenarios. This could damage the reputation of Internet Service Providers (ISPs) and/or expose critical personal information to criminals.

Email remains the most important application on the internet and is the most widely used facility that the internet has. Now more than 600 million people internationally use email. One can thus see how important it is to make it as secure as possible.

This chapter focuses on the use of cryptographic approaches to resolve the security issues inherent in SMTP. Reference will be made to many different possible cryptographic approaches based on what part of the total message they address; envelope or content. Each approach will be classified accordingly. Thus, there are cryptographic approaches for securing the SMTP envelope, such as verifying the users' authenticity to reduce spam and forged messages, and for securing the content of the message to prevent exposing critical data, such as credit card information, etc. to criminals. The necessity of verifying receivers, as well as senders, in order to avoid forged messages, will also be discussed. We start with a short introduction about electronic mail, SMTP, and problems of misusing SMTP. We discuss the advantages and disadvantages of these approaches and then introduce the most recent improvements and modifications made to enhance these approaches. Finally, we describe how to use these approaches in future internet networks, i.e., IPv6.

ELECTRONIC MAIL (EMAIL)

Email Object

An electronic mail message (or email for short) is a digital message that can be transferred over communication networks. An email consists of two components (Klensin, 2008):

- **Envelope:** The envelope is something that an email user will never see since it is part of the internal process by which an email is routed. It's added automatically by your

e-mail program when you press "Send", and it's removed automatically by the recipient's mail server just before the letter (without the envelope) is placed into their mailbox. Your email program connects to your outgoing mail server, and tells it your email address (the "Sender"), and the address(es) of the recipient(s). This is called the envelope. The data in the envelope is required by the Mail Transfer Agent (MTA) (Gellens & Klensin, 2011), that is, the SMTP clients and servers that provide a mail transport service.

- **Content:** The content is sent in the SMTP DATA protocol unit and has two parts:
  - **Header:** the body is always preceded by header lines that identify particular routing information of the message, including the sender, recipient, date, and subject. Some headers are mandatory, such as the FROM, TO and DATE headers. Some mail systems do not have the equivalent of a SMTP envelope, so when a message leaves the Internet environment, it might be necessary to insert the SMTP envelope information into the message header section.
  - **Body:** The body is the part that we always see as it is the actual content of the message contained in the email.

Email Transfer Protocols

Email transfer has been available as a computer application since the early 1960s, even before the evolution of the Internet. The first attempt to provide a protocol for electronic mail delivery on different platforms was supplied by the Address and Routing Parameter Area (ARPA), which proposed the Simple Mail Transfer Protocol (SMTP). It was developed for sending and receiving mail, and transporting it across multiple networks (SMTP relay). SMTP is an application layer protocol within the TCP/IP suite. The protocol is based on the "snail" mail (Van Staden & Venter, 2010) architecture; electronic mail is sent from one "post office" to the next, until the mail is delivered to the intended "mailbox."

In October 2008, the latest version of this protocol (Klensin, 2008) was released describing the musts of backwards compatibility for this protocol. This, in a nutshell, stated that SMTP Service Extensions defined in previous versions, which are not in regular use, might not be described in later RFC documents, but are expected to remain available. Since SMTP is limited in its ability to queue messages at the receiving end, it is usually used with one of two other protocols, POP3 or IMAP. SMTP is a simple text-based protocol. It thus benefits from other standard protocols, such as Multipurpose Internet Mail Extensions (MIME) (Freed & Borenstein, 1996), for sending non-text mail formats because SMTP supports only 7-bit ASCII characters, and thus, cannot transmit data in a binary format or other such formats.

SMTP Vulnerabilities

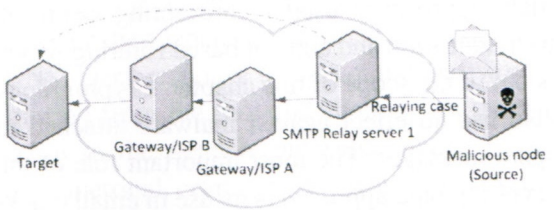
SMTP is not a secure protocol because the information that is transmitted in the SMTP is transmitted in plain text. It can thus be easily manipulated. The vulnerabilities inherent in SMTP allow for email abuse. Email abuse occurs when electronic mail is used to advertise unethically, harass, or annoy the email recipient. Abuse takes different forms—such as, spam, scam, email threatening (malware, viruses) and email cracking. Unsolicited electronic messages sent to people who do not choose to receive them are called spam. However, unsolicited messages sent to multiple accounts are not necessarily spam—for example, the occasional funny mass messages sent from friends to friends and back again. In these cases, the sender is known, whereas with spam, the sender is unknown. Scam is used to refer to unsolicited electronic messages that are sent to someone with the intention of getting him to give money for a service or product

that does not actually exist. Scam phishers try to send messages by manipulating the SMTP header or body to make them look legitimate. They search open mail relays, an SMTP server that allows anonymous users on the internet to send mail via this server, in order to deceive inexperienced users on the internet. They do this by using scammers' bluffs, which offer users lucrative business opportunities and other such things. They can also send messages to a list of mail addresses. These mail addresses are obtained by crawling through web pages on the internet using different software. For example, a scammer might ask you to cash foreign checks for which you will receive remuneration. Another example is where you are told you won prize money from a lottery (which you did not even enter). The difference between spam and scam is spam emails are emails that are sent with the intent to get you to buy a product. The product might exist, but usually does not work as described, and is not worth the money. So, the difference is that spam -while obnoxious and undesirable, is not illegal. Scams, on the other hand, are illegal. Spamming techniques are often employed by scammers. Threatening e-mail (malware attached to email) is usually sent in mass to many users with the intent to slow productivity of, or cause damage to, the recipient's computer system. These malicious programs, malwares or worms, called bots, are attached to messages sent to people in the hopes of infecting other computers on the internet. These infected computers are called zombies and they allow attackers to gain full access of the computer's resources. The attacker now controls the computers and can do Denial of Service (DoS) and phishing attacks (Suwa, Yamai, Okayama, & Nakamura, 2011) and can further propagate their attacks against other computers by using the computers that they now control. A cracker is someone who gains access to somebody's mailbox by usurping their passwords and bypassing all other security measures. They do this to gain information about the user that they can then use for their gain.

Email spoofing is another attack in this category. It is the act of editing or falsifying the SMTP header and the envelope information to hide the true origin or root of an email. Spoofing is also used to add fake validity to the content of an email by using a well-known and trusted domain, as the originating domain in order to perpetrate a phishing attack. Phishers are able to create emails with fake "Mail From:" headers in order to impersonate any organization they choose. In some cases, they may also set the "RCPT To:" field to an email address of their choice, whereby any customer replying to the phishing email will be sent to them. The growing press coverage over phishing attacks has meant that most customers are very wary of sending confidential information (such as passwords and PIN information) by email—however, in many cases, these types of attacks are still successful. The sniffing attack (Trabelsi, Rahmani, Kaouech, & Frikha, 2004) is not readily detectable. It can be accomplished by simply downloading free sniffer software from the Internet and installing it onto a Personal Computer (PC), which then becomes one of the infected computers on the network (botnet). A sniffer captures all packets and sends the important data to the attackers' computer. This data may consist of the passwords used to authenticate during an FTP session or the message of an email contained in SMTP packets that contains critical information.

Relay hijacking (see Figure 1) occurs when a malicious node finds an unsecure "SMTP Relay Server1" and misuses it to send messages to the target through other trusted relays, which are based on the trust of the "SMTP Relay Server1."

Figure 1. Relay hijacking



CRYPTOGRAPHIC PROTECTION MECHANISMS

People are unaware of sniffing attacks, and thus, they might include some critical personal information in their emails, which would be beneficial to thieves. For example, you might send an email to a friend asking them to watch your house while you are gone. The information contained in that email might define when you will be gone and where the location of the extra key is so that that person can gain entry to the house to check on it. A thief using a sniffing attack could garner this information and use it to steal everything from your house.

Scammers and attackers are actively looking to usurp critical information that is sent via emails. They are also interested in sending forged emails in order to pretend to be someone that they are not in order to obtain critical information from a user, such as password(s), bank account number(s) with associated PIN(s), etc. It is thus very important to insure the security of this critical information during the transfer process. Cryptography offers various methods for taking plain text data and transforming it into unreadable data for the purpose of securing the data during transmission. Using this approach, a key is used on the send side to encrypt the data, and a matching key is used on the receive side to decrypt the message.

It is important to remember that most spam messages are prevented by using approaches (Rafiee, Von Loewis, & Meinel, 2012) other than cryptographic. Cryptography thus plays a minor role in the prevention of spam. Cryptography plays no role in the prevention of scam except when the scammer uses a spamming approach—such as, forging headers or bogus domain. The two main approaches used in preventing scam are user education and content based filtering which is out of the scope of this chapter. Cryptography also has no effect against malware attached to spam messages. The most important role of the cryptographic approach is its use in email cracking which was explained in a prior section. This is because the cryptographic approach can be used to sign a domain, hash passwords, encrypt mail contents and prevent forged messages.

For the purposes of this chapter, cryptographic approaches will be classified into two main categories based on their usage in securing email components; securing the envelope and securing the content. Figure 2 shows the classification of cryptographic approaches used for each email component, i.e., the envelope and the content.

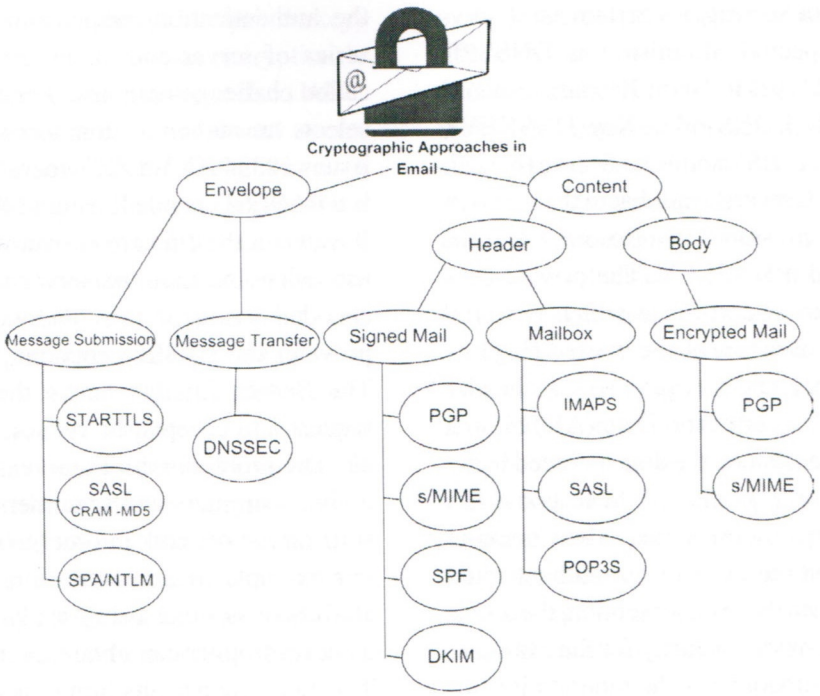
Securing the Envelope

SMTP wraps an email in an envelope for transmission. The envelope specifies what system is transmitting the mail, who the mail is from, and who it is to. This envelope might be likened to the envelope used in "snail mail." This is used by mail transport software to route and deliver the email. Because the envelope is processed before the data carrying the content, it is cheaper to reject spam based on envelope information than on content information. Likewise, the IP address of the sending system is available and can be compared with Blacklist and White list databases or other types of spam filtering. To protect this critical data, in order to reduce forged data such as forged "From", there are some cryptographic approaches available. These approaches are used to secure messages during the transfer process and during the message submission process from one MTA (Gellens & Klensin, 2011) to another.

Message Transfer

When transferring a message from your e-mail client, such as Microsoft Outlook or Thunderbird, the sending Mail Transfer Agent (MTA) (Gellens & Klensin, 2011) handles all of the mail delivery processing until the message has been either accepted or rejected by the receiving MTA. It can thus be sent either directly to the target domain, such as example.com if the "To" field is xx@example.

Figure 2. Cryptography classification for secure email



com, or to another e-mail server that is providing a relay service. Moreover, as the email clears the queue, it is routed along a host-to-host chain of servers. Each MTA in the internet network needs to ask for an IP address from the Domain Name System (DNS) in order to identify the next MTA in the delivery chain. The DNS is simply a database that defines the relationship between the name of a computer, such as http://www.example.com, to an IP address, such as 10.10.1.1. This process is called DNS resolving. DNS relies on a distributed database with a hierarchical structure. Email servers also require some specialized records in the DNS database like the MX record.

Unfortunately, the DNS is not secure enough. In particular there is currently no proof that the DNS server hasn't been corrupted. This has serious consequences for e-commerce and for the control of critical infrastructure. Ariyapperuma and Mitchell (2007) surveyed DNS attacks, such as man-in-the-middle (MITM) attacks, where the recipient of data from a DNS name server has no way of authenticating its origin or verifying its integrity. This is because the DNS does not specify a mechanism for servers to provide authentication details for the data they push down to clients. The resolver process has no way of verifying the authenticity and integrity of the data sent by name servers.

Domain Name System Security Extension (DNSSEC)

DNSSEC is an extension of the DNS (Arends, Austein, Larson, Massey, & Rose, 2005) used to validate DNS query operations. It verifies the authenticity and integrity of query results from a signed zone. In other words, if a DNSSEC is available from the requestor client to the resolver/caching nameserver to the authoritative nameservers, then the client has a level of assurance that the DNS query response is signed and trustworthy, starting from the root and chaining all the way down to the domain and subdomains. It uses asymmetrical cryptography which means that

separate keys are used to encrypt and decrypt data in order to provide security for certain name servers with their respective administrators. DNSSEC adds four record types to DNS; Resource Record Signature (RRSIG), DNS Public Key (DNSKEY), Delegation Signer (DS), and Next Secure (NSEC, NSEC3). Public keys are available to the world in DNS zones and are stored in a Resource Record (RR) type called a DNSKEY. The private keys are stored in a local certificate which is stored on the server. It uses two of the unused flag bits in the DNS query and answer message header (AD and CD). The Authentic Data (AD) bit in a response indicates that all the data included in the answer and authority portion of the response has been authenticated by the server. The Checking Disabled (CD) bit indicates that unauthenticated data is acceptable to the resolver sending the query. Moreover, if it provides security for the example.com domain and subdomains, the zone administrator will then electronically sign the zone and place this signature in the RRSIG. NSEC3 records are used to provide proof that a name does not exist by providing a range of names that do not exist. When the DNS server would normally reply with an empty answer, the NSEC3 record is signed with the corresponding RRSIG record in order to confirm that the domain name does not exist.

Message Submission

SMTP (Postel, 1982) is a simple protocol that is used for sending and receiving messages across the internet networks. Originally it did not support authentication. Thus spammers, scammers and attackers were able to misuse this schema in order to send their spam or malwares. SMTP Authentication is a feature which was introduced by Myers (1999) to protect mail servers from spam. The simplest authentication mechanism is "Plain". The client simply sends the unencrypted password to the server. All clients support the "Plain" mechanism. Two other authentication mechanisms are used for Authentication purposes; "Login" and SASL (CRAM -MD5). "Login" is the authentication mechanism consisting of a series of server and client message exchanges called challenge-response. For example, a client selects an authentication mechanism, a Server issues a "334 XXYYZZ" where "334 XXYYZZ" is a BASE64 encoded string of the "Username:". It waits for the client to answer with the BASE64 username and then the server sends a BASE64 encoded string of the "Password:" The client provides the BASE64 encoded string password. The Server finally checks the authentication request and accepts the request if everything is ok. The problem with Plain text authentications is that spammers and scammers can also easily sniff on the network in order to steal passwords. For example, if authentication via mail server and client is done using a plaintext password, an eavesdropper can obtain the user's password. This not only permits him to access anything in his mailbox, but, in a worse case, he can use the same password for all of the other critical applications or servers that need to be protected. It thus gives him access to anything else that the user has protected using the same password. So Siemborski and Melnikov (2007) proposed an extension to SMTP in order to enable it to support secure authentication.

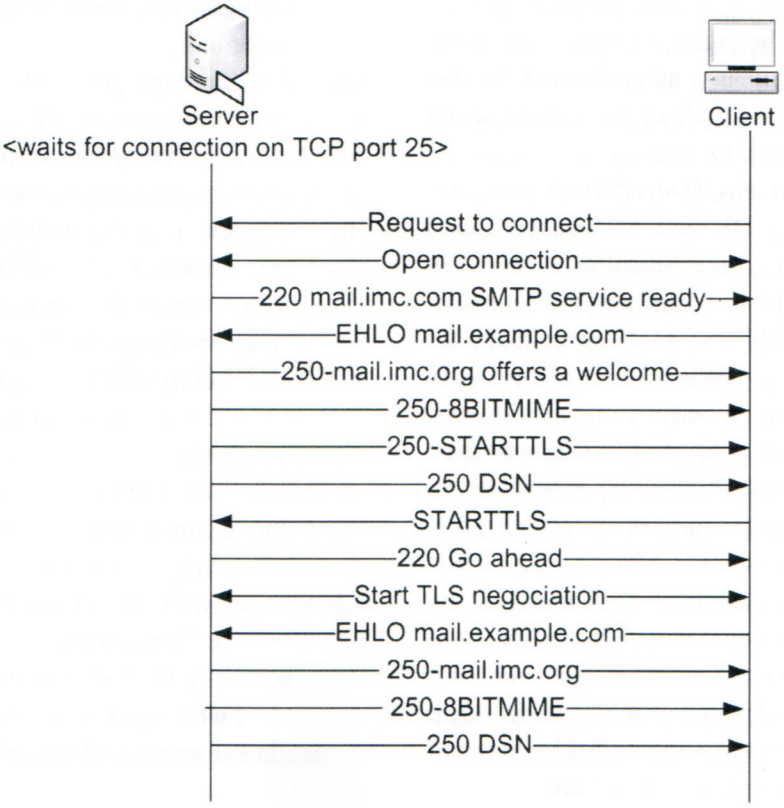
SMTP -STARTTLS (TLS Secure Password)

The Transport Layer Security (TLS) protocol was first proposed in 1991 (Dierks & Allen, 1999). Later it was updated and improved (Turner & Polk, 2011). Some of the improvements to this protocol are the replacement of the MD5/SHA-1 combination with cipher-suite-specified Pseudo Random Functions (PRFs). Additional ability was provided to the client/servers enabling them to specify their hash function and more support was offered for authenticated encryption with additional data modes. TLS is based on SSL3 to provide integrity and privacy on data. The au-

thentication mechanism in this protocol is based on asymmetric cryptography (public/private key) such as RSA, DSA, etc. The data encryption is based on symmetric cryptography such as AES, RC4, etc. The negotiation of a shared secret is secure and it is not possible for attackers to eavesdrop on it. In the authentication process, a TLS client sends a message to a TLS server, and the server responds with the information that the server needs to do the authentication itself. The client and server perform an additional exchange of session keys, and the authentication dialog ends. When authentication is completed, SSL-secured communication can begin between the server and the client using the symmetric encryption keys that are established during the authentication process. The keys for this symmetric encryption are generated uniquely for each connection. TLS is independent of application protocol. Therefore, the other protocols such as SMTP can layer on top of the TLS protocol transparently. Hoffman (2002) offered an extension to SMTP, called STARTTLS, which enabled it to use TLS. STARTTLS will thus help in authenticating clients or servers by ensuring the identities of the parties engaged in secure communication. It protects SMTP against password disclosure. By doing this it prevents some types of attacks against SMTP. An example of this type of attack would be man-in-the-middle.

When a client wants to start a STARTTLS session, as shown in Figure 3, it sends a request to the server, the connection to the server is established and the server answers the client's request with a 220 message. The client then sends a EHLO message which the server responds to with a welcome message. The message exchange continues as shown in Figure 3 until a secure session is established between the client and the server.

Figure 3. STARTTLS session between a client and a server



SASL (CRAM -MD5)

The old version of SASL used the Challenge-Response Authentication Mechanism (CRAM) (Klensin, Catoe & Krumviede, 1997), which is a cryptographic mechanism. This mechanism made use of the Message Digest 5 (MD5) algorithm to protect passwords against eavesdropping during transmission. Like the "Login" mechanism that is explained in the section "Message Transfer," it encoded the username in a BASE64 string. But it used the following MD5 algorithm in Hash-Based Message Authentication Codes (HMACs) (Krawczyk, Bellare & Canetti, 1997) to hash the password.

digest = MD5((‘secret’ XOR opad), MD5((‘secret’ XOR ipad), plain_data))

Where iPad is the byte 0x36 repeated B times, opad is the byte 0x5C repeated B times, and ‘secret’ is a string known only to the client and server. The client then sends this digest to the server. When the server receives the client message, it verifies the digest. If the digest is correct, the server assumes that the client is authenticated. So this mechanism provides both origin identification and replay protection for a session.

Leach and Newman (2000) offered another version of SASL called DIGEST-MD5 that supports data integrity after an authentication exchange in addition to other types of protection. Compared to CRAM-MD5, DIGEST-MD5 prevents chosen plaintext attacks and permits the use of third party authentication servers, permits mutual authentication, and permits optimized re-authentication if a client was recently authenticated by a server. Using this mechanism helps to avoid such popular attacks as replay attacks, Man-In-The-Middle (MITM) attacks, and online and offline dictionary attacks. But the problem with this mechanism is not only that it lacks support for clients but also this mechanism is deemed obsolete according to the list presented by Melnikov (2011). Some of the problems explained in that list are:

1. There are too many modes and too many options presented without thorough explanations or adequate implementations to back them up. Documentation is lacking in all areas. Some of the options are in conflict with each other.
2. The DIGEST-MD5 document allows an extra construct and allows for "implied folding whitespace" to be inserted in many places which is confusing and many implementations do not accept it.
3. The DIGEST-MD5 document's concept of a "realm" is used to define a collection of accounts. One or more realms can be supported by A DIGEST-MD5 server. There is no guidance provided by the DIGEST-MD5 document as to how realms should be named or how to enter them in User Interfaces (UIs).
4. Because the use of username in the inner hash is problematic it is rarely done in practice. Because it is not compatible with widely deployed UNIX password databases changing the username would invalidate the inner hash.
5. The descriptions of DES/3DES and RC4 security layers are not adequate enough to allow for the production of independently developed interoperable implementations.
6. The entire authentication exchange is not protected by The DIGEST-MD5 outer hash, so this makes the mechanism vulnerable to "man-in-the-middle" attacks.
7. The DIGEST-MD5 cryptographic primitives that are being use do not meet today's standards:
   a. The MD5 hash is not strong enough against brute force attacks initiated using the powerful hardware available today (Kim, Biryukov, Preneel, & Hong, 2006).
   b. The RC4 algorithm is prone to attack when used as the security layer without discarding the initial key stream output

c. The DES cipher for the security layer is considered insecure due to its small key space

SPA/NTLM (Integrated Windows Authentication)

NT LAN Manager (NTLM) (Microsoft Corporation, 2012) also known as Secure Password Authorization (SPA) is the Windows Challenge-Response authentication protocol. NTLM uses three types of messages between the client and server. A message is based on the client, the format, and the use of the message. As with some other authentication mechanisms, it uses a base64 encoded data stream that is the same as POP3 or SMTP. NTLM credentials are comprised of a domain name, a username, and a one-way hash of the user's password. NTLM uses an encrypted challenge/response protocol to authenticate a user without sending the user's password over the network. Microsoft Corporation (2012) listed the following steps as outlines for this protocol:

1. (Interactive authentication only) A user accesses a client computer and provides a domain name, user name, and password. The client computes a cryptographic hash of the password and discards the actual password.
2. The client sends the user name to the server (in plaintext).
3. The server generates a 16-byte random number, called a challenge or nonce, and sends it to the client.
4. The client encrypts this challenge with the hash of the user's password and returns the result to the server. This is called the response.
5. The server sends the following three items to the domain controller:
   a. User name
   b. Challenge sent to the client
   c. Response received from the client
6. The domain controller uses the user name to retrieve the hash of the user's password from the Security Account Manager database. It uses this password hash to encrypt the challenge.
7. The domain controller compares the encrypted challenge it computed (in step 6) to the response computed by the client (in step 4). If they are identical, authentication is successful.

NTLM is widely deployed on current systems. The main disadvantage of this mechanism is that it is not very secure and that it is vulnerable to many types of attack like the credentials forwarding attack. Ochoa and Azubel (2010) have published a list of these vulnerabilities.

Securing the Content

One of the oldest shortcomings of electronic mail in the internet, and most of its predecessors, is that the content is not cryptographically secured in the standard protocols. In particular, the content can neither be reliably authenticated, nor is it protected against eavesdropping. In the first version of electronic mail used in the UNIX environment, the SNDMSG (Tomlinson, 1971), this did not pose an issue as mail was only exchanged locally between users of the same multi-tasking operating system. Here, the operating system provided both authentication (by trustfully inserting correct sender information), and integrity and confidentiality (by protecting the recipients mailbox using regular file system access control).

Once email started being transmitted over the network, both authenticity and confidentiality were lost. Many users did not consider this as a problem: the network operators were trusted to not perform eavesdropping (and intercepting email is indeed a criminal offense today in many jurisdictions). In addition, email originally was not used for any "critical" activity, such as

commercial transactions. Authenticity was not a concern because, in most cases, the sender could be reliably authenticated by just verifying that the content is plausible, as recipients were familiar with the senders.

Today most users still do not consider confidentiality of email as an issue as they continue to trust network operators to not intercept their data. Some users, however, are concerned about the ability of government-supported lawful interception. In addition, a threat to confidentiality exists during message submission and postbox access. This threat is also addressed by transport-layer encryption which was discussed in the subsections on "Message Submission."

On the other hand, message authentication is a real concern for many users, as users often receive spam and scam emails from fake email accounts, and phishing emails that try to impersonate a genuine sender by talking the user into performing some action. As a consequence, a variety of approaches to deal with message and sender authenticity have been developed.

Securing the Header

Email messages delivered using SMTP usually contain headers which describe the travel path of a message between senders and recipients (Table 1). However, even though most email users are not

Table 1. Header Fields of a SMTP Message (Resnick, 2001)

Field name	Application	Required	Description
From	originator fields	Yes	The sender of the message.
Sender		No	If different than "From"
Reply-To		No	specify to whom the response shall be sent. If not set replies go to "From"
To	Destination fields	No (but usually present)	the primary recipient of the message
Cc		No	The secondary recipients of the message. A copy of message sent to them
Bcc		No	an original copy of message sent to these recipients
Date	Originator's date	Yes	Date and time stamp for the message
Message-ID	Identification fields	No (but usually present)	Unique code applied in a time of sending
In-Reply-To		No	A mechanism to Coordinate responses
References		No	If any other message ID available
Subject	International fields	No (but usually present)	Title of the message
Comments		No	Description of the message
Keywords		No	Used for searching purposes
Return-Path	Trace fields	No	Used to trace messages through email systems
Received			
Resent-Date	new fields	Depends on the status of message whether it is forwarded or not	Used to forward message
Resent-From			
Resent-Sender			
Resent-To			
Resent-Cc			
Resent-Bcc			

concerned with this, it is important to remember that the data contained in the header is critical data that needs to be protected in order to prevent the forging of messages and in order to reduce spam. There are some cryptographic approaches available for use in securing messages during the client message fetching process (mailbox security).

Mailbox

When a user wants to access or manage his emails, first he needs to use an email client which is called the Mail User Agent (MUA) (Gellens & Klensin, 2011). The most popular MUAs are Outlook, Thunderbird, Opera Mail, Eudora email software, Pegasus Mail, etc. These MUAs use some standard protocols for receiving email which are stored on the server and in managing them. One of these protocols is Post Office Protocol 3 (POP3) (Myers & Rose, 1997). POP3 is a client/server protocol. It has been built into the Netscape and Microsoft Internet Explorer browsers. POP3 is designed to delete email from the server as soon as the user has downloaded it. However, some email clients allow for users or an administrator to set an option that will allow for a longer retention period or an indefinite retention period. In POP3, users can work offline on their emails, but any changes made will not be saved on the server. Users need to periodically check the server for new emails, and if any are present, download them from the server to their PC. For these reasons, Internet Message Access Protocol (IMAP) (Crispin, 2003) was proposed to enhance the receipt of email from a server. This protocol enables users to better organize their emails on the server by creating unlimited folders. IMAP can act like a remote file server. Since it keeps original emails on the server, the user has access to all his emails from anywhere, or he can connect to the server from different computers at the same time. A user can also work live on his emails and any changes made to any email will be stored on the server. This occurs because, when a connection is established with the server, IMAP maintains that connection with the server. Messages viewed once on the server are then cached on the computer in order to save bandwidth. Users can also set up a synchronization process in order to store permanent copies of messages locally. Like other non-secure protocols that have been explained in past sections, IMAP and POP3 are vulnerable to various types of attack (Newman, 1999), such as man-in-the-middle attacks. Also passwords and usernames are sent in plaintext and thus can be sniffed easily. This is the main reason that Newman (1999) wanted to use these protocols over TLS/SSL.

IMAPS

There are two mechanisms in use to protect IMAP; IMAP over SSL which is known as IMAPS and IMAP over TLS which is known as STARTTLS (Newman, 1999). The Secure Sockets Layer (SSL) Protocol Version 3.0 (Freier, Karlton & Kocher, 2011) is the last version of SSL. It has two security features; encryption and certification. It avoids fake identity scams by asking the contacted server to present its digital certificate to prove its identity.

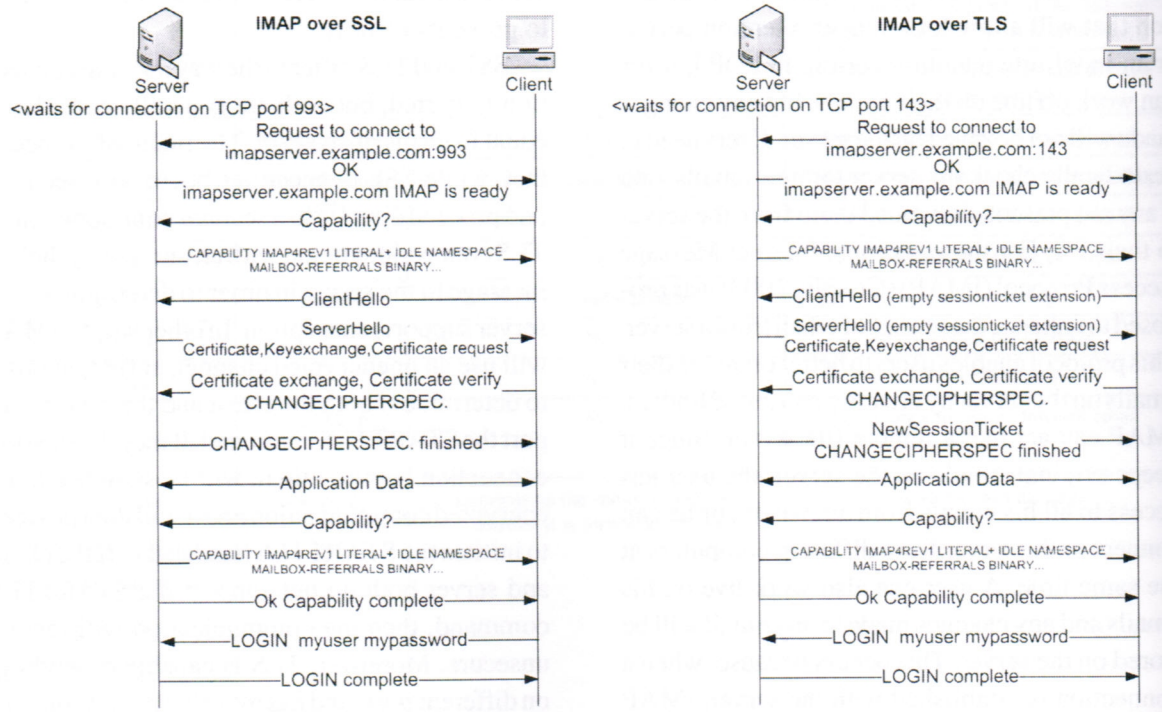
SSL and TLS differ in the way a secure connection is started, but both are generally considered equal in terms of security. The main difference is that, while SSL connections begin with security and proceed directly to a secured communication, TLS connections begin with an unsecured "hello" message to the server in order to determine if that server supports encryption. In other words, IMAP will use an unencrypted channel, at first, in order to determine if both the client and the server support the STARTTLS command. If they do, then the connection between them will be switched to an encrypted communication and it will then proceed to initiate the STARTTLS handshake. If the client and server both do not support the STARTTLS command, then the communication will remain unsecure. Moreover, TLS is capable of working on different ports and has more backward compat-

ibility than SSL. Figure 4 depicts the simple IMAP handshakes over SSL and TLS. IMAP (Crispin, 2003) supports a large number of commands. In the illustration, for instance, the CAPABILITY command requests a list of the supported server capabilities. The CHANGECIPHERSPEC (Freier, Karlton & Kocher, 2011), in Figure 4, is a message used to notify the receiving party that subsequent records will be protected under the just-negotiated CipherSpec and keys. Receipt of this message causes the receiver to copy the "read pending" state into the "read current" state. The client sends a CHANGECIPHERSPEC message, initiates the handshake key exchange and the certificate verification message process. The server then sends a message, similar to that of the client, after having successfully processed the key exchange message. The NEWSESSIONTICKET is basically a ticket consisting of the session state that includes, for example, the cipher suite and master secret in use.

SASL

As explained in the section "SASL (CRAM-MD5)", the Simple Authentication and Security Layer (SASL) (Melnikov & Zeilenga, 2006) is a framework which provides authentication and data security services in connection-oriented protocols via replaceable mechanisms. IMAP and POP are among the protocols that contain SASL support. If a protocol supports SASL, then it should include a command for identifying and authenticating a user to a server, and optionally, for negotiating protection of subsequent protocol interactions. If its use is negotiated, a security layer is inserted between the protocol and the connection. For authentication purposes, any of the mechanisms can be used that were explained in the prior sections, such as CRAM-MD5 (Klensin, Catoe & Krumviede, 1997), DIGEST-MD5, PLAIN, LOGIN and NTLM (Microsoft Corporation, 2012).

Figure 4. Simple IMAP handshakes over SSL and TLS



POP3S

Like IMAPS, POP3 can also be protected over SSL or TLS (Newman, 1999). The former is known as POP3S and the latter as POP3 over TLS. POP3S uses a separate SSL port, i.e. 995, where the SSL handshake procedure begins as soon as a client connects, and then, only after the session is encrypted does the regular protocol handling begin. Using two separate ports for plaintext and SSL connections was thought to be wasteful. This is why POP3 over TLS is the more popular of the two, as it uses the normal unencrypted port, i.e. 110. A client first connects to this unsecure port and immediately starts a STARTTLS command that changes the session to an encrypted one. However, the problem with POP3 over TLS occurs when a client does not support a secure connection or a client prefers using an unsecure connection instead of a secure one. This is when the POP3 over TLS tries to do a plaintext authentication, which is impossible in SSL, since the connection will be refused by the server.

Message Signing

Both Pretty Good Privacy (PGP) and S/MIME support message signatures. In email, it is essential that the message remains legible, even after signing, which causes both PGP and S/MIME to employ ASCII-armored data representations, where the original plain text becomes, literally, a part of the complete message. Additionally, S/MIME puts the MIME layer on top, allowing mail readers to recognize and display the original message; the signature then typically gets rendered as an attachment.

Again, each protocol specifies the set of supported algorithms. In addition to the asymmetric algorithms, digital signatures need to select hash algorithms. Also, some asymmetric algorithms are signature-only, so the list of algorithms for signing messages is longer than the one for encryption.

PGP requires a Digital Signature Algorithm (DSA) for the signature, and allows, but deprecates, RSA. SHA-1 is required as a hash algorithm with various alternatives also being supported (MD-5, RIPE-MD/160, SHA-256, etc.).

S/MIME requires RSA with SHA-256, and recommends various combinations of RSA and DSA with MD5, SHA-1, and SHA-256.

As with encrypting messages, verifying messages poses a challenge for the validation of the sender's public key. The challenge is slightly easier than that for encryption, though, because:

- The data format is backwards compatible, so the recipient is able to read the message even if he cannot verify the signature.
- The message typically includes an indication of how to establish trust. For S/MIME, the certificate is often included in the signed message, only requiring the recipient to receive and trust the CA certificate. Receiving the CA certificate is feasible, as the location of the CA certificate is included in the user certificate. For PGP, the key identifier is included in the message, which allows the user to retrieve the key from a key server (if the key was uploaded).

MTA Authentication

In addition to authenticating users sending email, authenticating the sender's Mail Transfer Agent (MTA) is also a useful technique for preventing faked emails. Users cannot generally assume that communication partners will be able to verify a signature due to the lack of trust; therefore, message signing is not widely used. In order to reduce spam, email administrators have been looking for alternative approaches to validate that an email message really comes from the user that claims to be the sender of the email.

Instead of having the user sign the email, the first mail transport agent receiving the original

message submission will sign the email, allowing receiving mail transport agents to verify that the mail was really submitted through that host. In addition, for an internet domain, the list of authorized MTAs is published. As a consequence, spammers sending email for a different domain can be recognized; in addition, spammers sending from your own domain, that do not go through the domain's official MTAs, can also be detected. We present two protocols that have been established for this application.

Sender Policy Framework

The Sender Policy Framework (SPF) (Wong & Schlitt, 2006) defines DNS records that list authorized source MTAs for messages originating from a domain. SPF defines a new resource record type (99); for compatibility, the TXT record type (16) can also be used. The value of this resource record defines, in a micro programming language, the list of valid senders for a domain, by specification of IPv4 or IPv6 address prefixes, or redirecting through MX, A, or PTR records. A receiving MTA uses the source sender identity as indicated in the MAIL FROM SMTP command, retrieves the corresponding SPF record, and checks whether the TCP peer address of the SMTP communication is authorized to send email. Optionally, the same check can also be applied to the host name in the SMTP HELO/EHLO command, although many legitimate senders currently fail this test as they put bogus data into the HELO command. No check of the (Resnick, 2008) "From: header" is performed.

With this framework, receiving MTAs may choose to reject email if the sender domain has an SPF record, or it may quarantine or flag the message.

In itself, no cryptographic mechanism is employed in this protocol. However, a potential threat to the protocol is poisoning of the recipient's DNS cache. This would be done to introduce fake SPF records to make the receiving MTA accept a message from an unauthorized host. This threat can be avoided by using DNSSEC in addition to SPF, in order to trustfully sign all DNS records involved (i.e. the SPF record itself, and any MX and A records it refers to).

DomainKeys Identified Mail (DKIM)

DKIM (Allman, Callas, Delany, Libbey, Fenton, & Thomas, 2007) involves a cryptographic mechanism that not only authorizes senders, but to also authenticates the originating MTA. As with SPF, DNS resource records are used to publish authorized information about the domain. However, instead of publishing policy, DKIM publishes public keys in the DNS. Compared to the other technologies, the authors of the specification point out that DKIM:

- Puts signatures into email headers, leaving the body untouched;
- Does not require users to have a priori trust in certain public keys; instead, the public keys are found through DNS lookups;
- Does not mandate any specific policy for the case that verification fails.

Each domain can publish any number of keys, typically one per sending MTA. The domain keys are put into the <keyname>._domainkeys.<domain> label, using a TXT record.

A MTA emitting a message creates a signature including certain selected header fields, and the body, signs this with the key, and creates a new email header. This header includes the key name, the list of fields included in the checksum, and the actual signature value. The supported signature algorithms are RSA-SHA1 and RSA-SHA256. Allman et al. (2007) point out those keys of 4096 bits and more will not fit into the standard DNS size limit for UDP messages of 512 bytes.

As the message is forwarded from MTA to MTA, possibly being replicated at a mailing list, it may get signed multiple times. Recipients can verify each individual signature separately.

The specification leaves any policy effect of DKIM verification to the local systems. They recommend that MTAs indicate verification results in additional email headers, allowing users to filter by these headers, rather than rejecting messages. For example, a spam filter might diagnose a phishing attempt when an unsigned message is received for a domain that is known to typically sign messages. In order to determine whether a sender would normally sign messages, the Author Domain Signing Practices (ADSP) protocol (Allman, Fenton, Delany & Levine, 2009; Leiba, Thomas, & Crocker, 2011) can be used.

DKIM, as currently specified, has a shortcoming that causes verification to fail even though the message was not substantially modified: transit MTAs may restructure MIME payloads, and in particular change the character encoding of text parts. This will break the verification, as DKIM's signature algorithm is not MIME-aware.

Securing the Body

Securing the body is critical when email exchanged between users contains vital data, such as credit card information or passwords. A particular common use case for email encryption is the exchange of passwords between system administrators. In this case, users just do not want to risk having their password intercepted. As discussed later, using email encryption is a challenge in practice, and the hurdles are too high for casual non-admin users.

Message Encryption

In today's email infrastructure, message encryption relies on asymmetric cryptography. Using only symmetric cryptography would require a sender and a recipient to share a secret, which is impractical by the nature of electronic mail. With asymmetric cryptography, the practical challenge is to obtain the public key of the recipient. Two strategies have been developed to provide this information to the sender:

- A directory service allows looking up public keys by certain criteria, such as the recipient's email address or the recipient's real name. Such directories often allow any user to post information, so the challenge here is to verify that the public key really belongs to the recipient.
- In the absence of a directory, the recipient can send his public key in a first interchange, which is then followed by the actual communication. With RSA, a typical approach is to have the recipient send a signed message first: the protocols discussed below will then allow for the inclusion of the public key along with the message. This also helps to verify that the recipient really holds the corresponding private key. Verifying that the public key really belongs to the recipient (rather than belonging to a man in the middle) still remains a challenge.

Pretty Good Privacy (PGP)

In 1991, Phil Zimmermann created the first version of the Pretty Good Privacy software (Zimmermann, 1995). The current version of the protocol is OpenPGP (Shaw, 2009) which is based on version 5.0 of the PGP software. PGP offers both digital signature and encryption. Encryption is performed in the following steps (section 2.1: Shaw, 2009):

1. The sender creates a message.
2. The sending OpenPGP generates a random number to be used as a session key for this message only.
3. The session key is encrypted using each recipient's public key.
4. These "encrypted session keys" start the message.
5. The sender optionally compresses the message.
6. The sending OpenPGP encrypts the message using the session key, which forms the remainder of the message.

7. The receiving OpenPGP decrypts the session key using the recipient's private key.
8. The receiving OpenPGP decrypts the message using the session key.
9. If the message was compressed, it will be decompressed.

Messages are transmitted as a series of packets, with packets being packed by packet type. Packet types include public keys, secret keys, encrypted session keys, signatures, compressed payload, encrypted payload, etc.

The preferred public key algorithm of PGP is Elgamal; RSA is also supported, but deprecated. Various symmetric algorithms for data encryption are supported, including IDEA, 3DES, CAST5, Blowfish, and AES. 3DES is required, AES-128, CAST5, and IDEA are recommended. As compression algorithms, ZIP, ZLIB and BZip2 are supported. None of the compression algorithms must be supported, but ZIP should be implemented.

In an email message, the encrypted data is typically put in ASCII armor, with a header tag indicating the kind of data, and the actual encryption result encoded in a radix-64 encoding.

PGP keys are mutually signed in a web of trust, where users can arbitrarily sign each other's public key. Global key servers can be used to upload and download public keys and signatures to public keys. Users then need to establish trust themselves in the claimed identities, but checking whether they can establish a chain of trust between people they personally trust, following signatures that those people made, then ultimately to the identity of the message recipient. In addition, people use other means of communicating public keys, such as posting them on their home pages, or maintaining key rings within communities.

S/MIME

The Secure/Multipurpose Internet Mail Extensions (Ramsdell & Turner, 2010) use the existing MIME framework (Freed & Borenstein, 1996). The current protocol version is 3.2. In addition, it is based on the Public Key Infrastructure (PKI) (Cooper et al., 2008) and the Cryptographic Message Syntax (CMS) (Housley, 2009). While we assume that the reader is familiar with PKI, we elaborate on CMS first.

CMS is derived from PKCS#7 (Kaliski, 1998). It is an ASN.1 (Legg, 2007) based syntax for encrypted and signed data. In addition, it supports various auxiliary data records such as encrypted symmetric keys. Following the ASN.1 notion of object identifiers, arbitrary cryptographic algorithms can be used, allowing for the introduction of new algorithms without the need to change the data structures. Encrypted data is a record identified by the object identifier

{ iso(1) member-body(2) us(840)
rsadsi(113549) pkcs(1) pkcs7(7) 6 }

While CMS uses Distinguished Encoding Rules (DER) to transmit data, S/MIME now takes such messages and puts them into the MIME framework. In addition, it specifies certain cryptographic algorithms required for implementations to allow for interoperability. Finally, it also supports transmission of auxiliary data such as certificates and certificate revocation lists.

For key encryption, S/MIME specifies RSA as mandatory, and recommends RSAES-OAEP. For symmetric algorithms, AES-128 CBC is required, and AES-192 CBC and 3DES CBC are recommended. CMS messages are transmitted using the application/pkcs7-mime media type.

Trust in public keys is established through certificate authorities in the PKI. A common infrastructure for publishing certificates is LDAP, but there currently is no global directory for PKI certificates (unlike PGP). Instead, there may be organization-wide directories (such as installations of Microsoft's Active Directory), allowing for the retrieval of a certificate for a message recipient. In the absence of a directory, it is common to have the ultimate recipient (Bob) of an encrypted

message send a signed message to Alice first, as this will include Bob's certificate, which then gets cached in Alice's email client, allowing Alice to then send an encrypted message. Trust in Certificate Authorities is typically achieved by relying on a list of CAs which is provided by the operating system vendor, and extended according to local policies.

FUTURE RESEARCH DIRECTIONS TO SECURE EMAIL

The fight against spam and scam will be an ever ongoing process because as the cryptographic algorithms continue to improve and as new cryptographic algorithms are proposed for use along with the current protocols which secure email components during the transfer process, the scammers continue to come up with new techniques of their own to circumvent the new security. They are making use of more powerful hardware and more intrusive software packages in order to crack cryptographic algorithms that already exist. We will thus have to focus all of our resources on trying to stay one step ahead of the spammers and scammer of email systems.

Internet Protocol version 6 (IPv6) (Deering & Hinden, 1998) is the next-generation of internet protocol and is designed to solve security issues and the lack of addresses that are present in the older version (IPv4) of this protocol. On February 3, 2011, IANA allocated the final blocks of IPv4 addresses to the regional registries thereby exhausting the central address pool. Despite the fact that the protocols related to email, such as SMTP, IMAP, POP3, etc., are application layer protocols within IPv6 networks, spammers and scammers now have another exploitable area at their disposal which resulted from the expansion of the address space and the nodes' temporary addresses used for privacy or security reasons in IPv6. If spammers and scammers gain full access to the resources of one IPv6 node in a network, then they will be able to send spam and scam through a different authorized IP address from the same node. Moreover, in a subnet, there are 2^{64} IPv6 usable addresses making it easy for spammers and scammers to change their IPv6 addresses every second or to send out each spam or scam mail with a different address. This helps them hide their identity while they flood the network with spam and scam thus complicating the life of the system administrator. Moreover, due to the fact that IPv6 addresses are 128 bits in comparison to IPv4 addresses that are 32 bits, the regular DNS Address Resource Record (RR) was created to allow a domain name to be associated with a 128-bit IPv6 address. An example of these RRs is the four "A"s, i.e. "AAAA", to indicate that the IPv6 address is four times the size of the IPv4 address. The AAAA record is structured in the same way that the A record is in both binary and master file formats, but it is just much larger. As stated in an earlier section, in order to secure DNS, as it has a vital role in many protocols, like email protocols, especially in the transfer stage, DNSSEC (Arends, Austein, Larson, Massey & Rose, 2005) was proposed. Hoffman (2010) adds the latest extension to DNSSEC which specifies how the DNSSEC cryptographic algorithm identifiers in the IANA registries are allocated. When user A sends an email to user B, it is important for user A to ensure that user B, in the other domain, really receives that message without a scammer having eavesdropped and spoofed it on its way before it reaches user B. It is the same for user B as he wants to be sure that user A really sent that email. For average users this may not be as important as it is for say governors or people who have high positions in society as there is the potential for their reputation to be sullied by bogus information contained in fraudulent emails. More to the point, as also touched briefly in section "MTA Authentication", consider this hypothetical example. The head of a large company wants to expel one of his executive managers called X, but before doing so he wants to get the

opinions of his other executive managers. Since he is traveling on business he prefers to use email as the means of contacting all of the executive managers. The head of the company's email resides in a different domain than those of his executive managers. X finds out what is going on and being the devious person he is, he initiates a 'man-in-the-middle' attack. This attack enables him to capture the emails being sent to the executive managers and to then create a bogus "in his favor" response from each manager back to the head of the company. X wants to buy himself enough time so that his dismissal will not be imminent. He hopes to use this time to collect incriminating information against the head of the company so that X can blackmail him in order to keep his job. So in this example you can see how important it is to check both the sender and the receiver. If the email clients of those executive managers checked the receiver's domain, then X would not have been able to execute his attack. The same is true for the receivers. If they had checked the sender, this would not have happened and X would not have had an opportunity to send forged emails to the head of the company. This issue also applied to the content of emails. Several new studies focus on making better use of a cryptographic approach to protect data. Mantoro, Norhanipah and Bidin (2011) proposed an IPv6 framework for DNSSEC in order to provide for origin authentication of DNS data. DNSSEC is not widely implemented and this kind of framework can be the first steps for DNSSEC global implementation.

CONCLUSION

Electronic Mail (email) is a novel way of communicating using the internet. Unfortunately, protocols related to transfer, receive, or send emails do not have the protection mechanisms needed to protect the email components and to prevent spam, crack or the theft of critical data. For example, many scammers and spammers misuse SMTP, a protocol for sending and receiving emails, in order to disseminate their unwanted messages or to attach malicious programs. However, cryptographic approaches play a minor role in spam reduction. But they do have a high effect on the prevention of email cracking and forged messages.

To protect the mail data, verify authenticity, and, as a result, decrease spam and forged mail, many approaches have been proposed which were classified in this chapter as cryptographic approaches according to the level of protection afforded to email components. Their advantages and disadvantages, if any, were described. We explained the differences in the use of these approaches in the new generation of internet protocol, i.e. IPv6, in comparison to the old internet protocol, i.e. IPv4, and we covered the most recent studies in this area. We also explained the importance of verifying "sender domain" as well as "receiver domain."

REFERENCES

Allman, E., Callas, J., Delany, M., Libbey, M., Fenton, J., & Thomas, M. (2007). DomainKeys identified mail (DKIM) signatures. RFC. Retrieved from http://tools.ietf.org/html/rfc4871

Allman, E., Fenton, J., Delany, M., & Levine, J. (2009). DomainKeys identified mail (DKIM) author domain signing practices (ADSP). RFC. Retrieved from http://tools.ietf.org/html/rfc5617

Arends, R., Austein, R., Larson, M., Massey, D., & Rose, S. (2005). DNS security introduction and requirements. RFC. Retrieved March 2005, from http://www.ietf.org/rfc/rfc4033.txt

Ariyapperuma, S., & Mitchell, C. J. (2007). Security vulnerabilities in DNS and DNSSEC. In Proceedings of the Second International Conference on Availability, Reliability and Security. ACM. Retrieved from http://dl.acm.org/citation.cfm?id=1250514

15. Chapter, U. S. C. 103. (2011). Commerce and trade. Retrieved from http://uscode.house.gov/download/pls/15C103.txt

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., & Polk, W. (2008). Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. RFC. Retrieved from http://www.ietf.org/rfc/rfc5280.txt

Crispin, M. (2003). Internet message access protocol - Version 4rev1. Retrieved from http://tools.ietf.org/html/rfc3501

Deering, S., & Hinden, R. (1998). Internet protocol, version 6 (IPv6) specification. RFC. Retrieved from http://www.ietf.org/rfc/rfc2460.txt

Dierks, T., & Allen, C. (1999). The TLS protocol version 1.0. RFC. Retrieved from http://www.ietf.org/rfc/rfc2246.txt

Eastlake, D. (1999). Domain name system security extensions. RFC. Retrieved from http://tools.ietf.org/html/rfc2535

Freed, N., & Borenstein, N. (1996). Multipurpose internet mail extensions (MIME) part one: Format of internet message bodies. RFC. Retrieved from http://tools.ietf.org/html/rfc2045

Freed, N., & Borenstein, N. (1996). Multipurpose internet mail extensions (MIME) part five: Conformance criteria and examples. RFC. Retrieved from http://tools.ietf.org/html/rfc2049

Freier, A., Karlton, P., & Kocher, P. (2011). The secure sockets layer (SSL) protocol version 3.0. RFC. Retrieved from http://tools.ietf.org/html/rfc6101

Gellens, R., & Klensin, J. (2011). Message submission for mail. RFC. Retrieved from http://tools.ietf.org/html/rfc6409

Hoffman, P. (2002). SMTP service extension for secure SMTP over transport layer security. RFC. Retrieved from http://www.ietf.org/rfc/rfc3207.txt

Hoffman, P. (2010). Cryptographic algorithm identifier allocation for DNSSEC. Retrieved from http://tools.ietf.org/html/rfc6014

Housley, R. (2009). Cryptographic message syntax (CMS). RFC. Retrieved from http://tools.ietf.org/html/rfc5652

Kaliski, B. (1998). PKCS #10: Certification request syntax version 1.5. RFC. Retrieved from http://tools.ietf.org/html/rfc2314

Kim, J., Biryukov, A., Preneel, B., & Hong, S. (2006). On the security of HMAC and NMAC based on HAVAL, MD4, MD5, SHA-0 and SHA-1. Retrieved from http://eprint.iacr.org/2006/187.pdf

Klensin, J. (2008). Simple mail transfer protocol. RFC. Retrieved from http://tools.ietf.org/html/rfc5321

Klensin, J., Catoe, R., & Krumviede, P. (1997). IMAP/POP authorize extension for simple challenge/response. RFC. Retrieved from http://tools.ietf.org/html/rfc2195

Krawczyk, H., Bellare, M., & Canetti, R. (1997). HMAC: Keyed-hashing for message authentication. RFC. Retrieved from http://tools.ietf.org/html/rfc2104

Leach, P., & Newman, C. (2000). Using digest authentication as a SASL mechanism. RFC. Retrieved from http://www.ietf.org/rfc/rfc2831.txt

Legg, S. (2007). Abstract syntax notation X (ASN.X) representation of encoding instructions for the generic string encoding rules (GSER). RFC. Retrieved from http://tools.ietf.org/html/rfc4913

Leiba, B., Thomas, M., & Crocker, D. (2011). Author domain signing practices (ADSP): Point and counterpoint. Internet Computing, 15(1), 76-80. Retrieved from http://doi.ieeecomputersociety.org/10.1109/MIC.2011.1

Mantoro, T., Norhanipah, S. A., & Bidin, A. F. (2011). An implementation on domain name system security extensions framework for the support of IPv6 environment. doi:10.1109/ICMCS.2011.5945627

Melnikov, A. (2011). Moving DIGEST-MD5 to historic. RFC. Retrieved from http://tools.ietf.org/html/rfc6331

Melnikov, A., & Zeilenga, K. (2006). Simple authentication and security layer (SASL). Retrieved from http://tools.ietf.org/html/rfc4422

Microsoft Corporation. (2012). Microsoft NTLM. Retrieved from http://msdn.microsoft.com/en-us/library/aa378749.aspx

Myers, J. (1999). SMTP service extension for authentication. RFC. Retrieved from http://tools.ietf.org/html/rfc2554

Myers, J., & Rose, M. (1997). Post office protocol - Version 3. RFC. Retrieved from http://www.ietf.org/rfc/rfc1939.txt

Newman, C. (1999). Using TLS with IMAP, POP3 and ACAP. RFC. Retrieved from http://tools.ietf.org/html/rfc2595

Ochoa, H., & Azubel, A. (2010). Windows SMB NTLM authentication weak nonce vulnerability. Retrieved from http://www.ampliasecurity.com/research/OCHOA-2010-0209.txt

Postel, J. B. (1982). Simple mail transfer protocol. RFC. Retrieved from http://www.ietf.org/rfc/rfc821.txt

Rafiee, H., Von Loewis, M., & Meinel, C. (2012). IPv6 deployment and spam challenges. *IEEE Internet Computing*, *16*(6). doi:10.1109/MIC.2012.97.

Ramsdell, B., & Turner, S. (2010). Secure/multipurpose internet mail extensions (S/MIME) version 3.2 certificate handling. RFC. Retrieved from http://tools.ietf.org/html/rfc5750

Ramsdell, B., & Turner, S. (2010). Secure/multipurpose internet mail extensions (S/MIME) version 3.2 message specification. RFC. Retrieved from http://tools.ietf.org/html/rfc5751

Resnick, P. (2001). Internet message format. RFC. Retrieved from http://tools.ietf.org/html/rfc2822

Resnick, P. (2008). Internet message format. RFC. Retrieved from http://tools.ietf.org/html/rfc5322

Shaw, D. (2009). The camellia cipher in OpenPGP. RFC. Retrieved from http://tools.ietf.org/html/rfc5581

Siemborski, R., & Melnikov, A. (2007). SMTP service extension for authentication. RFC. Retrieved from http://tools.ietf.org/html/rfc4954

Suwa, S., Yamai, N., Okayama, K., & Nakamura, M. (2011). DNS resource record analysis of URLs in e-mail messages for improving spam filtering. In Proceedings of the 2011 IEEE/IPSJ International Symposium on Applications and the Internet. IEEE. Retrieved from http://dl.acm.org/citation.cfm?id=2061659

Tomlinson, R. (1971). The first network email. Retrieved from http://openmap.bbn.com/~tomlinso/ray/firstemailframe.html

Trabelsi, Z., Rahmani, H., Kaouech, K., & Frikha, M. (2004). Malicious sniffing systems detection platform. *IEEE Computer Society*, *201*. doi:doi:10.1109/SAINT.2004.1266117.

Turner, S., & Polk, T. (2011). Prohibiting secure sockets layer (SSL) version 2.0. RFC. Retrieved from http://tools.ietf.org/html/rfc617

Van Staden, F., & Venter, H. (2010). Adding digital forensic readiness to the email trace header. [ISSA]. *IEEE Information Security for South Africa*, *1*. doi: doi:10.1109/ISSA.2010.5588258.

Wong, M., & Schlitt, W. (2006). Sender policy framework (SPF) for authorizing use of domains in e-mail, version 1. RFC. Retrieved from http://tools.ietf.org/html/rfc4408

Zimmermann, P. R. (1995). The official PGP user's guide. Retrieved from http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=5518

ADDITIONAL READING

Oppliger, R. (2009). SSL protocol, TLS protocol. In *SSL and TLS: Theory and Practice* (pp. 75–178). Norwood, MA: Artech House.

KEY TERMS AND DEFINITIONS

Cracker: Crackers transform computers into zombies by using small¬ programs that exploit weaknesses in a computer's Operating System (OS).

Cryptography (in Email Usage): The practice and study of techniques for secure communication or content of email in the presence of third parties (called adversaries). These techniques are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation.

DKIM: DomainKeys Identified Mail involves a cryptographic mechanism to not only authorize senders, but to also authenticate the originating MTA.

MIME: Multipurpose Internet Mail Extensions is an Internet standard that extends the format of email to support; text in character sets other than ASCII, non-text attachments, message bodies with multiple parts, and header information in non-ASCII character sets.

Scammer: The use of Internet services or software with Internet access to defraud victims or to otherwise take advantage of them, for example by stealing personal information.

SMTP: Simple Mail Transfer Protocol is a protocol used to send electronic mail across the Internet.

Spam: Spam is the use of electronic messaging systems to send unsolicited bulk messages, especially advertising, indiscriminately. Spam is often used for the purposes of scamming.

Spammer: A person who creates electronic spam.

SPF: The Sender Policy Framework defines DNS records that list authorized source MTAs for messages originating from a domain. With this framework, receiving MTAs may choose to reject email if the sender domain has an SPF record, or it may quarantine or flag the message.