Global Information Assurance Certification Paper

Defense Message System

William Wolfe
January 21, 2001

Background:

The Defense Message System (DMS) was initiated in 1988, when a study of the outdated Department of Defense's (DoD) dedicated point-to-point communication system, AUTODIN (AUTOmatic DIgital Network), revealed it was costing over $700 million each year to operate. In 1995, the Defense Information Systems Agency (DISA) awarded Lockheed-Martin the contract to provide DMS products and services. The idea was that DMS would allow the phase out of AUTODIN, reduce costs, and provide better services.

The Multi-command Required Operation Capabilities document (MROC 3-88), published by the Joint Chiefs of Staff in February 1989, outlined the initial requirements for the Defense Message System. DMS was to provide a secure e-mail capability to all of DoD, other U.S. government agencies, U.S. Allies, and Defense-related users. DMS was required to use Commercial-Off-The-Shelf (COTS) products as much as possible and utilize emerging technologies. From a security standpoint, it was required at a minimum to provide confidentiality, authentication, message integrity, and non-repudiation. Since 1989, MROC Change 2 (in 1997) has further refined the messaging requirements.

Since 1995, each military service (Navy, Air Force, Army, Marines, and Coast Guard) and other U.S. government agencies (i.e. DISA, DLA, NSA) have been working to integrate DMS into their existing infrastructure. Unlike the point-to-point communication used by AUTODIN, DMS utilizes a wide area network (WAN) called the Defense Information Systems Network (DISN). The DISN, maintained by DISA, is believed to be the best way to meet the MROC requirements regarding reliability and speed of service. Consequently, DMS message traffic almost exclusively utilizes the DISN.

What's so special?

In most ways, DMS is essentially the same as a commercial SMTP email system. Across the DISN, Message Transfer Agents (MTA) utilizing the x.400 protocol ensure the delivery of messages from one point to another. These messages are delivered to a message store on an email server. The majority of DMS email servers are either Microsoft Exchange or Lotus Notes. Utilizing the product's client (Outlook for Exchange or Lotus Client for Lotus Notes), the "end user" retrieves and processes their messages. Where it differs is in the implementation of security. DMS uses a Public Key Infrastructure (PKI). The public key is stored in certificates within the DMS x.500 directory. The private key is stored on a PCMCIA card called a FORTEZZA card (there are only three manufacturers of NSA certified FORTEZZA cards; Spyrus, Group Technologies, and Mykotronx). The user accesses the card (via a PIN) to perform signing and encrypting prior to message release. The card implements the Secure Hash Algorithm (SHA-1), the Digital Signature Standard (DSS) algorithm, the Key Exchange Algorithm (KEA), and the SKIPJACK encryption and decryption algorithm

(developed by NSA and utilized by the Capstone chip on the card). Unlike SHA-1, DSS, and KEA, the SKIPJACK algorithm was classified until July 1998.

For encrypting the message contents, symmetric (secret) key cryptography (SKIPJACK) is used, for the digital signature and providing the secret key to the recipient(s), the card relies on asymmetric key cryptography. The secret key used to encrypt the message is the Message Exchange Key (MEK), and the mechanism for providing the MEK is the KEA. The public key digital signature mechanism is the DSS algorithm, and the algorithm used to produce the hash for DSS is the SHA-1.

The first step in encrypting a message is to verify the recipient's certificate in the x.500 directory to ensure the certificate is still valid (has not been revoked, compromised, or expired). If the user's certificate is valid, the card then generates the MEK. The message is then encrypted using the MEK and the FORTEZZA SKIPJACK algorithm. For each recipient, a token is attached to the message header and the MEK is placed in the token. The token is then encrypted with the Token Encryption Key (TEK), using the intended recipient's public key (obtained from the recipient's certificate in the directory) and the originator's private key, via the Key Exchange Algorithm (KEA). Only an intended recipient can decrypt a token, and thus decrypt the message using the MEK and the SKIPJACK algorithm. Figure 1, from the DMS Recommended System Design Architecture (SDA) Document Release 2.2, presents a pictorial overview of the encryption and key exchange process.

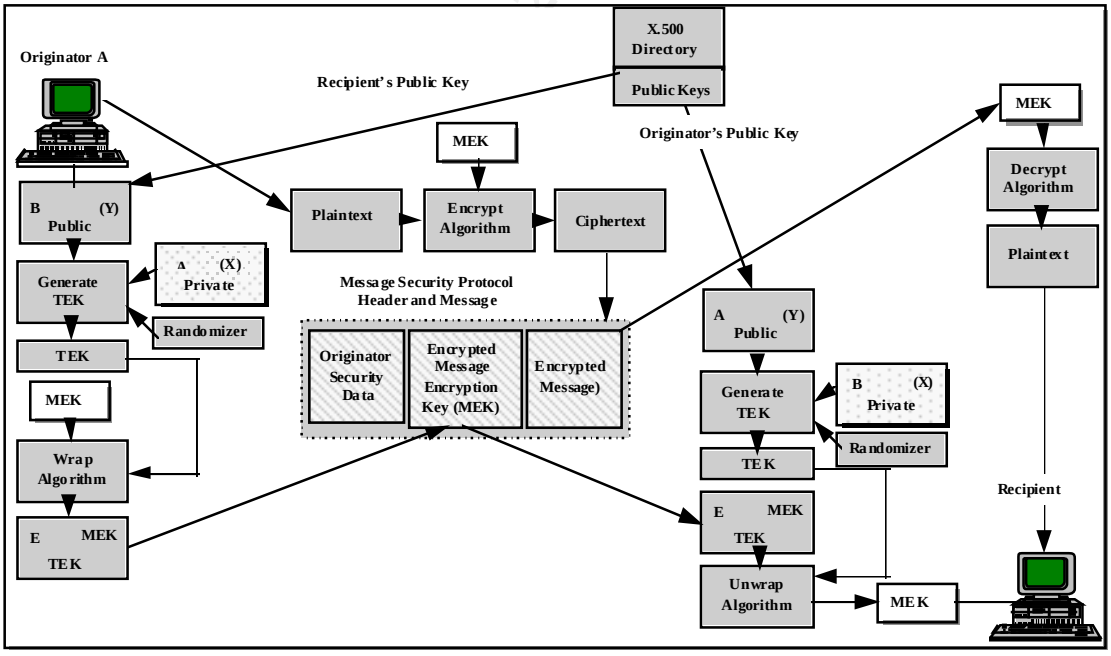

Figure 1. Passing Key Information

In order to decrypt, the recipient generates a unique pair-wise key by using the originator's public key, the recipient's private key, and the originator's random number. The token and then the MEK are decrypted using the pair-wise key. The message is

decrypted using the MEK and finally, the decrypted data is hashed and the result compared to the hash value in the token.

Message signatures are produced by hashing the message contents using a one-way hash algorithm implemented on the card to produce a 160-bit (20 byte) hash value. The private key, the hash value, and a random number generated by the card are provided to the DSS algorithm to create the digital signature. To verify a digital signature, a hash is calculated over the message content, and the hash value, along with the message originator's public key (after the certificate in the x.500 directory is validated), is used to calculate a signature value. If this signature value matches the signature value in the originator's message, then the signature is valid.

The services provided by the FORTEZZA technology are confidentiality, authentication, message integrity, and non-repudiation. All items that were required through the Multi-command Required Operation Capabilities document, published in 1989.

Confidentiality

Confidentiality protects information from being disclosed to unauthorized people. As in any public key cryptographic system, the encryption process provides confidentiality because the encrypted message cannot be read by anyone except the intended recipient.

Authentication

Authentication is the assurance that someone does not use a false identity when applying public/private keys. To receive a FORTEZZA card, the user must validate their identify and a certificate containing, among other items, their public key is created in the x.500 directory for that user.

Message Integrity

Message integrity is the assurance that the originator's message was unaltered in route to its final destination. The hash on the message content performs this message integrity function. This hash is then sent along with the message. The recipient's software, which has the same hash function, runs the content through it own hash function and compares the result to the hash sent with the message. They should be identical. The hash is unique to the message content, meaning that if the message content is changed in any way, even if a single character is changed, the hash will also change indicating that the message has been altered.

Non-Repudiation

Non-repudiation is the assurance that the person whose name appears on the message as the message originator truly did write and approve the message. In the FORTEZZA

technology, the digital signature is used for non-repudiation. The digital signature is basically the message hash encrypted with the signer's private digital signature key.

The recipient's FORTEZZA/MSP software decrypts the digital signature using the originator's public digital signature key and retrieves the message hash from the digital signature. The software re-computes the message hash using the message and the hash function, and compares it to the retrieved message hash. If the two match, then the originator must have been the person who sent the message because he or she is the only one who has access to that private digital signature key.

SKIPJACK

SKIPJACK was de-classified by the U.S. DoD on June 23, 1998. Like DES, it is a 64-bit block cipher, but it uses an 80-bit key, rather than the 56-bit key of DES. It uses 32 rounds, twice as many as DES. SKIPJACK could not have been a submission for the Advance Encryption Standard (AES), since AES required a 128-bit block cipher and the ability to support key lengths of 128-, 192-, and 256-bits.
For those interested, the specifications of the SKIPJACK algorithm and modes of operation (OFB, CFB, CBC, and Codebook) can be found at http://csrc.nist.gov/encryption/skipjack-kea.htm. According to Bruce Schnier, "SKIPJACK's performance is good. Slower than Blowfish and some of the AES submissions, it's still about twice as fast as DES on 32-bit microprocessors." If for nothing else, SKIPJACK is unique in that it's the first NSA-developed encryption algorithm ever seen by the public. Simple in design, SKIPJACK is apparently susceptible to any changes to the original design. That is, if you change the original design, the result will be an algorithm that can be broken. Cryptanalysis today is providing evidence of this and an example can be found in a document entitled Initial Observations on the SKIPJACK Encryption Algorithm. According to the authors, reducing the SKIPJACK algorithm from 32 rounds to 16 rounds resulted in an algorithm that could be broken by an attack faster than a brute force (exhaustive search) attack.

Vulnerabilities

So, what are the weaknesses to DMS? A brute force attack on the encryption algorithm (SKIPJACK) would be to attempt the entire 80-bit key space. That translates to 2^80 keys (or 1 trillion trillion). While most agree today that 56-bit keys are no longer sufficient, this is still 2^24 more keys than the 56-bit key space. Another challenge to this attack would be to collect the encrypted data and in theory, that data should only be traversing the DISN, not the Internet. In this instance, traffic analysis may be the only thing not being well defended against.

One apparent weakness is the fact that DMS tries to utilize commercial software as much as possible. As such, it is susceptible to all the well-known attacks against Microsoft NT, Microsoft Exchange, HP-UX (platform for the x.500 directory), Lotus Notes, etc. By gaining root privileges on an email server, a hacker could gain access to the message stores. However, since the encryption/decryption occurs at the end user's machine, through use of the FORTEZZA card, the hacker would not be able to read the messages. Even if the hacker were to place sniffer software (tcpdump, ethereal, snort,

etc) on a machine on the internal network, messages are encrypted once they leave the client machine. However, those machines where root privilege was obtained could be used for other purposes.

It appears that if obtaining messages in the clear is the goal, then the point of attack would be the end user's machine. An application installed on the client machine that either sent a user's keystrokes, screenshots, or altered the client application (such as forwarding copies of all incoming/outgoing messages in the clear) could provide a way to obtain classified information. This would also provide a way to obtain the access PIN for the FORTEZZA card. This however, would take a great deal of time, as it would affect only one DMS user at a time. An unknown virus introduced by one DMS user could possibly affect several at time since the data is encrypted until it reaches the final destination. User awareness and anti-virus software with updated signatures would help reduce that risk.

A denial of service attack is certainly one method that could be used effectively. The MTAs on the DISN provide enough redundancy that should one MTA be flooded with data, other routes are available to allow message traffic to continue to be delivered. To be successful, a denial of service attack would have to be focused at the message store, the final delivery point for messages. This would affect not only incoming messages, but could also prevent users from sending outgoing messages. Even then, this would only affect a subset of all DMS users.

Summary:

After many years, implementation of DMS is gaining acceptance. According to the latest implementation statistics, over 1,000,000+ messages are utilizing the DMS infrastructure. As of this writing, version 2.2, which continues to utilize Windows NT, Lotus Notes 4.6, and Exchange 5.5, is due out soon. DMS version 3.0, scheduled for late 2001, will migrate the DMS environment to the newer versions of those products.

A recent study evaluated if and when DMS should change to an SMTP and software based PKI implementation. If that is to happen, the decision is yet to be made. Although SKIPJACK is an integral part of DMS, NSA is dedicated to the Advanced Encryption Standard (Rijndael) and implementing it as the successor to DES. It's obvious that DoD is determined to make DMS work. In order to succeed, DMS will need to continue to meet the requirement to adapt to emerging technologies.

Works Cited

Biham, Eli; Biryukov, Alex, Dunkelman, Orr; Richardson, Eran; Shamir, Adi. "Initial Observations on the SkipJack Encryption Algorithm." URL: http://www.cs.technion.ac.il/~biham/Reports/SkipJack/note1.html (15 June 1998).

Bricknell, Ernest; Denning, Dorothy; Kent, Stephen; Maher, David; Tuchman, Walter, "SKIPJACK Review – Interim Report." URL: http://www.epic.org/crypto/clipper/skipjack_interim_review.html (28 July 1993)

Defense Information Systems Agency, "DMS Recommended System Design Architecture Document Release 2.2" Document Number ZDD5026 (21 December 1999)

Defense Link, "Encryption Formulas Declassified." URL: http://www.defenselink.mil/news/Jun1998/b06231998_bt316-98.html (23 June 1998)

Schneier, Bruce. "Crypto-Gram." URL: http://www.counterpane.com/crypto-gram-9807.html (15 July 1998).

Last Updated: May 27th, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event