GoScan

Security Overview

Summary

GoScan provides authentication and encryption to ensure security and document integrity. Security in GoScan Lite ("GSL") and GoScan ("GS") has two available security safeguards. The first is a secure transmission method and the second is encryption. If GSL or GS are used in a standalone mode residing outside of a firewall and wish to transmit images to a destination within a firewall, only secure transmission is available via HTTPS, FTPS or a custom web service.

If GSL or GS are transmitting images outside a firewall to GoScan Workgroup ("GSW") or GoScan Enterprise ("GSE") then secure transmission is augmented by encryption as described below.

Background

Public networks such as the Internet do not provide a means of secure communication between entities. Communication over such networks is susceptible to being read or even modified by unauthorized third parties. In addition to file encryption and encryption on a local disk, cryptography helps you create a secure means of communication over otherwise insecure channels, providing data integrity and authentication.

GoScan uses its own encryption algorithm class and generates the security keys and all the underlying network infrastructure programmatically – no user intervention is required. In short, you don't have to be a networking engineer to send data in a secured way.

Cryptography helps protect data from being viewed or tampered with and helps provide a secure means of communication over otherwise insecure channels. For example, GoScan data can be encrypted seamlessly, transmitted in an encrypted state, and later decrypted by the GoScan Server. If a third party intercepts the encrypted data, it would be nearly impossible to decipher.

In a typical GoScan situation where cryptography is employed, multiple workstations are used to acquire images simultaneously and then communicate to the GoScan Server over an insecure channel. GoScan ensures that communication remains incomprehensible by anyone who might be listening or attempting to fraudulently intercept a data stream through hacking techniques such as user impersonification or port spoofing. Since GoScan workstations may operate in remote locations, GoScan Server must be sure that the information received has not been modified by anyone during transmission and is in fact legitimate. To this end, GoScan uses a proprietary receipt functionality that assures the integrity of our images by validating the file name transmitted, transmission dates and timings, block-level integrity checks,

and lastly a binary comparison (e.g. checksum) of the source and destination file.
Technical Background Information

Cryptography is used to achieve the following goals:

- Confidentiality: To help protect a user's identity or data from being read.
- Data integrity: To help protect data from being altered.
- Authentication: To assure that data originates from a particular party.

To achieve these goals, GoScan combines algorithms and practices known as cryptographic primitives to create a cryptographic scheme. The following table lists the cryptographic primitives and their uses.

Cryptographic primitive	**Use**
Secret-key encryption (symmetric cryptography)	Performs a transformation on data, keeping the data from being read by third parties. This type of encryption uses a single shared, secret key to encrypt and decrypt data.
Public-key encryption (asymmetric cryptography)	Performs a transformation on data, keeping the data from being read by third parties. This type of encryption uses a public/private key pair to encrypt and decrypt data.
Cryptographic signing	Helps verify that data originates from a specific party by creating a digital signature that is unique to that party. This process also uses hash functions.
Cryptographic hashes	Maps data from any length to a fixed-length byte sequence. Hashes are statistically unique; a different two-byte sequence will not hash to the same value.

Secret-Key Encryption

Secret-key encryption algorithms use a single secret key to encrypt and decrypt data. GoScan secures the key from access by unauthorized agents because any party that has the key can use it to decrypt data. Secret-key encryption is also referred to as symmetric encryption because the same key is used for encryption and decryption. Secret-key encryption algorithms are extremely fast (compared to public-key algorithms) and are well suited for performing cryptographic transformations on large streams of data.

Typically, secret-key algorithms, called block ciphers, are used to encrypt one block of data at a time. Block ciphers[1] (like RC2, DES, TripleDES, and Rijndael) cryptographically transform an input block of *n* bytes into an output block of encrypted bytes. If you want to encrypt or decrypt a sequence of bytes, you have to do it block by block. Because *n* is small (*n* = 8 bytes for RC2, DES, and TripleDES; *n* = 16 [the default], *n* = 24, or *n* = 32 bytes for Rijndael), data values larger than *n* have to be encrypted one block at a time.

The block cipher classes provided in the base class library use a chaining mode called cipher block chaining (CBC), which uses a key and an initialization vector (IV) to perform cryptographic transformations on data. For a given secret key *k*, a simple block cipher that does not use an initialization vector will encrypt the same input block of plain text into the same output block of cipher text.

If you have duplicate blocks within your plain text stream, you will have duplicate blocks within your cipher text stream. If unauthorized users know anything about the structure of a block of your plain text, they can use that information to decipher the known cipher text block and possibly recover your key. To combat this problem, information from the previous block is mixed into the process of encrypting the next block. Thus, the output of two identical plain text blocks is different. Because this technique uses the previous block to encrypt the next block, an IV is used to encrypt the first block of data. Using this system, common message headers that might be known to an unauthorized user cannot be used to reverse engineer a key.

One way to compromise data encrypted with this type of cipher is to perform an exhaustive search of every possible key. Depending on the size of the key used to perform encryption, this type of search is extremely time consuming using even the fastest computers and is therefore unfeasible. Larger key sizes are more difficult to decipher. Although encryption does not make it theoretically impossible for an adversary to retrieve the encrypted data, it does raise the cost of doing so prohibitively. If it takes three months to perform an exhaustive search to retrieve data that is only meaningful for a few days, then the exhaustive search method is impractical.

The disadvantage of secret-key encryption is that it presumes two parties have agreed on a key and IV and communicated their values. Also, the key must be kept secret from unauthorized users. Because of these problems, secret-key encryption is often used in conjunction with public-key encryption to privately communicate the values of the key and IV.

If GoScan wants to communicate over an insecure channel, it might use secret-key encryption as follows. GoScan standardizes one particular algorithm with a particular key and IV. GoScan composes a data block and creates a network stream on which to send the message, encrypts the data block using the key and IV, and sends it across the

[1] GoScan adheres to MIL-SPEC standards through the usage of data encryption as defined in DOD 4120.24-M *Defense Standardization Program (DSP) Policies and Procedures*, March 2000, OUSD (Acquisition, Technology and Logistics).

Internet. GoScan does not send the key and IV to the GoScan Server; the server receives the encrypted text and decrypts it using the previously agreed upon key and IV. If the transmission is intercepted, the interceptor cannot recover the original message because he does not know the key or IV.

The GoScan Solution

GoScan is truly simply, smart, and <u>secure</u>. GoScan provides end-to-end security that is seamless to use and ensures the highest level of integrity possible.