MLS-Net and SecureParser®: A New Method for Securing and Segregating Network Data

Robert A. JOHNSON

Director of Security Products Architecture, Unisys Corporation

2476 Swedesford Road, Malvern, PA 19355

ABSTRACT

A new method of network security and virtualization is presented which allows the consolidation of multiple network infrastructures dedicated to single security levels or communities of interest onto a single, virtualized network. An overview of the state of the art of network security protocols is presented, including the use of SSL, IPSec, and HAIPE IS, followed by a discussion of the SecureParser® technology and MLS-Net architecture, which in combination allow the virtualization of local network enclaves.

Keywords: Encryption, Crypto-splitting, SecureParser®, IPSec, SSL, HAIPE IS, vitualization.

INTRODUCTION

Securing network data while in motion is an increasingly important requirement for today's enterprise networks. A variety of pressures, whether regulatory, financial, or mission-related are forcing network managers and architects to consolidate and virtualize local networks while utilizing the public internet as their inter-enclave backbone.
Simultaneously, within this context of shared infrastructures, the need to guarantee the integrity and confidentiality of network data is rapidly growing.

In this paper, we will first review the state of the art of various network protocols that secure data while in motion. In the process, we will uncover shortcomings in those methods which will highlight a couple of key capabilities which are lacking. In particular, we will see that data is not secured while in motion within a local enclave, and as a result, separate local network infrastructures must be maintained to physically separate data associated with different security levels or communities of interest.

Next, we will examine a solution to both of these shortcomings, called MLS-Net. MLS-Net utilizes a data security technology called SecureParser® created by Security First Corp. The combination of MLS-Net and SecureParser® closes the gap in local enclave network security so that separate enclaves for different security levels need not be maintained and all data can be intermixed within the same local network infrastructure.

NETWORK SECURITY

Let us begin with a little background. In this section, we will examine the state of the art in network security, specifically how data is protected while in flight between clients and network resources, between server-based applications, and between different enterprises' networks.

There are a few different protocols that protect data while it is in motion, and many different products that implement those protocols. The discussion that follows focuses on the protocols, not the associated products.

As with most any networking problem, the problem of data security can be addressed at different layers within the protocol stack. In general, solutions are targeted at the session layer, network layer, and below, as shown in Figure 1.

Figure 1 – Security At Different Network Layers

OSI Layer	Security Methods
Presentation	
Application	User credentialing
Session	SSL
Transport	
Network	IPSec VPN, HAIPE-IS, SSL VPN
Link	MLS-Net
Physical	Link encryptors

Wide-area network (WAN) links are often protected by hardware-based link encryptors, which are point-to-point devices which encrypt all data flowing across the link. Although certainly appropriate for radio and satellite links, link encryptors are only useful when using point-to-point links owned by the enterprise. More and more, such private networks are being replaced by other technologies such as Secure Sockets Layer (SSL) or Virtual Private Networks (VPNs) running over the public internet.

SSL

Starting at the top of the protocol stack, the Secure Sockets Layer (SSL) protects data associated with a particular client/server or application/application session. SSL uses TCP/IP as its transport.

Theoretically, any TCP/IP session between any pair of applications could utilize SSL services. In reality, however, SSL is primarily used for browser-based communications over the HyperText Transport Protocol (HTTP) – the basis of the world-wide web. For example, when you buy a product through an online website, the communications dialog that carries your personal information such as address, phone number, and credit

card number, is protected by SSL.

SSL relies on a Public Key Infrastructure (PKI) for its cryptographic key management. PKI-based encryption technology utilizes pairs of encryption/decryption keys. Pairs of these keys are related mathematically in a very special way. When a piece of cleartext data is encrypted with one of the keys, say A1, it can only be decrypted by its pair, A2. Likewise, if A2 is used to encrypt some data, only A1 can decrypt it.

This relationship between A1 and A2 means that one of the keys, say A1, can be made public while the other, A2, is kept private. So, if Bob wants to send a private message to Alice, and guarantee that only Alice can read it, Bob will encrypt the message with Alice's public key (A1). Alice then uses her private key (A2) to decrypt the message. Since A2 is kept secret by Alice, only she can decrypt the message.

While very useful for browser-based applications that include a human in the loop, SSL becomes inefficient for high-volume transaction environments, especially when many distinct sessions must be set up and torn down repeatedly. This is because SSL peers must obtain a key pair for each session, and then exchange their public keys prior to the start of their dialog. The associated processing overhead can become too burdensome in high-volume situations.

VPNs

Another approach puts the security processing in the network layer. Network-layer protocols are then used to form Virtual Private Networks (VPNs) which can run over public networks such as the internet.

VPNs are very popular in enterprises with mobile or home-based workforces. In general, a client workstation or laptop can plug into any public network – at home, at a hotel, or even an internet café – and securely access network resources within the enterprise as if the workstation was connected directly via an in-office wall jack.

The freedom and flexibility of VPNs are the driving force behind the trend toward work-from-anywhere network-centric computing environments. As with anything else, however, freedom and flexibility come with a price. The price is typically an additional administrative burden. What type of burden is more palatable to an enterprise is what often determines what type of VPN is deployed.

   IPSec: The most popular form of VPN is use today is based on the Internet Protocol Security (IPSec) protocols. IPSec is a set of protocols that allow the transport of secure information between two enclaves which are connected by an open, public network. The enclaves could be as simple as a single user's laptop, or as complex as an entire corporate intranet. Figure 2 shows an example of how IPSec is used.

Figure 2 – IPSec Example
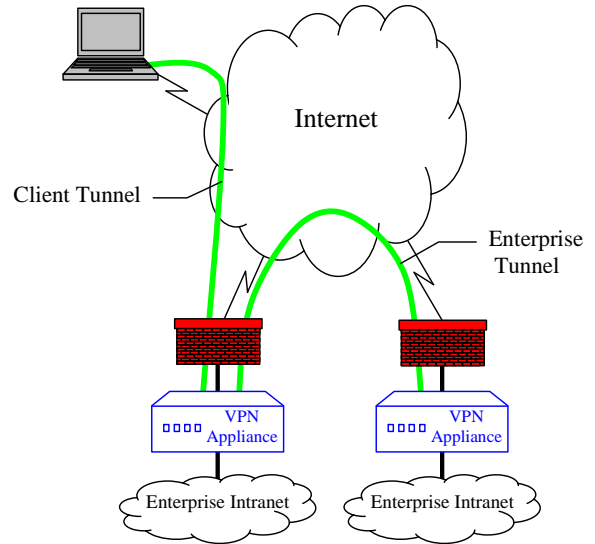
IPSec is an integral part of IP Version 6 (IPv6), and an optional enhancement to IPv4. Since it is an add-on to IPv4, most operating systems' native IPv4 stacks do not support IPSec directly. As a result, two types of deployments may be required. On the client workstation, software that adds the IPSec functionality must be installed that integrates with the native protocol stack. On the enterprise side, however, it is more efficient to deploy a VPN appliance, which implements the IPSec protocols on behalf of the entire corporate intranet.

IPSec can operate in two modes: Transport Mode, which encrypts only the data portion of the packet and leaves the IP header intact; and Tunnel Mode, where the entire packet, including the original IP header, is encrypted and a new IP header is added.

Transport Mode is primarily used within an intranet, where it is desirable for attributes of the original frame, such as Time To Live (TTL), source routing information, and/or quality of service (QoS), to be preserved.

Tunnel Mode, on the other hand, hides all of the information of the original frame, and hence is mainly used when most application options would not be supported, such as when tunneling through the public internet. Tunnel Mode is most often used to allow client access to the intranet from home or on the road.

IPSec in general, and Tunnel Mode in particular introduce issues with router-to-router communications. In general, IPv4 routing protocols such as Router Information Protocol (RIP), Open Shortest Path First (OSPF), Internet Group Management Protocol (IGMP), and others do not function well or at all when IPSec is in use. This forces the network to be thought of as enclaves of non-IPSec networks connected by IPSec virtual connections. This is appropriate when accessing an enterprise intranet from the road, or when an enterprise is geographically dispersed and it makes sense to connect the enclaves via the internet for financial or other reasons.

   **HAIPE IS:** The U.S. Department of Defense is moving

their operations toward a vision of net-centricity that includes the concept of the Global Information Grid (GIG). The GIG will encompass all DoD IT resources networked together, yet still retain the necessary distinctions of security classification. These security classifications can take many forms, but it is simplest to think of them as representing three levels: Unclassified, Secret, and Top Secret.

Data classified at a high level (e.g. Top Secret), may not be accessed by a person who is cleared for only a lower level of access (e.g. Secret or Unclassified). Today, separate networks are maintained for each security level, and rigorous policies and procedures are in place to try to ensure that no malicious or unintended declassification of information occurs.

One of the core attributes of the GIG, however, is that geographically dispersed resources will be interconnected via the "black" network, i.e. the public internet. In essence, the GIG will use the internet as its backbone. This makes sense, since the core IP protocols, upon which the internet is built, were originally designed by the Defense Advanced Research Projects Agency (DARPA) to be used for military purposes in time of national emergency.

Obviously, when data of any security level is transported across the internet, it must be protected. Figure 3 shows how homogeneous security level enclaves can be connected to other enclaves of the same security level through the use of VPNs.

Figure 3 – Security Enclaves Within the GIG
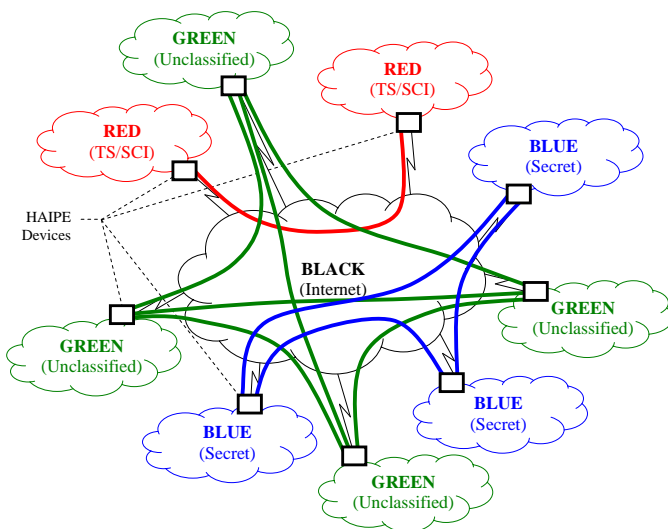
The VPN technology used within the GIG is called High Assurance Internet Protocol Encryptor Interoperability Specification (HAIPE IS, or just HAIPE). HAIPE is an enhanced version of IPSec. It, too, has two modes, a Tactical Mode which is functionally equivalent to IPSec's Transport Mode, and a Strategic Mode, which is likewise equivalent to Tunnel Mode.

HAIPE devices (the actual "Encryptor" from the name) are standalone appliances which replace normal IP routers and their associated routing protocols. Current implementations of HAIPE require the devices to be statically configured with enough of the network configuration to function as routers on behalf of their local enclaves. There is ongoing research investigating the use of dynamic discovery protocols, similar to those used by the Domain Name Service (DNS), to reduce the administrative overhead required to maintain the configuration information.

Another area of high maintenance costs is that of key management. The administration and maintenance of the key management and distribution system is complex and places a large burden on the responsible network managers.

This combination of high maintenance costs for network configuration and key management, especially when the extra dimension of multiple security levels is added, has limited the deployment of HAIPE to date.

SSL VPN: The final VPN technology we will discuss is one that is becoming very popular in the commercial sector. SSL-based VPNs are a hybrid of the SSL technology discussed above and the VPN concept. The perceived benefit of SSL VPNs is their lower administrative overhead.

This lower overhead stems from the fact that no client software needs to be installed by hand on the client machine, as is required for IPSec-based VPNs.

When an SSL VPN is in use, any client, whether it be a corporate laptop on the road, a PC at home, a handheld PDA, or a kiosk in an internet café, can securely access the enterprise's intranet. The client establishes SSL connections to all web-enabled resources within the enterprise through an appliance behind the enterprise's firewall.

For non-web-enabled resources, Java applets are automatically downloaded which enable access to those resources.

Unlike IPSec, which opens the entire intranet to a validated user, the SSL VPN appliance actively mediates access to all resources. The appliance is configured with extensive user authorization policies, which it consults when a client attempts to access a particular web page or other type of portal.

On the surface, it appears that SSL VPNs substitute the installation and maintenance of client VPN software with the configuration and maintenance of ever-changing user access policies, so the value of such a trade-off may not be obvious. The configuration of the appliance is centralized, however, whereas the installation and repair of IPSec client software is not. This alone, is a significant differentiator for understaffed IT shops. Add to that the enabling of PDAs, smart phones, and the like, and the attraction of SSL VPNs becomes apparent.

SSL VPNs, however, are only appropriate for remote client access. There is an implied client/server relationship between the endpoints of the VPN, which is not symmetric. Therefore, SSL VPNs are not appropriate for enclave-to-enclave communications as required by the GIG.

WHAT'S WRONG WITH THIS PICTURE?

So, what is wrong with the picture, as shown in Figure 3 above, of secure enclaves connected via HAIPE to each other across the

black internet? Well, two things at least. First, there is no security for data in motion within the local enclaves; and second, although they are physically disjoint from a network perspective, many of the enclaves are not geographically disjoint. In fact, they are often parallel networks within the same buildings and offices. Figure 4 illustrates this point.

Figure 4 – Parallel Enclaves
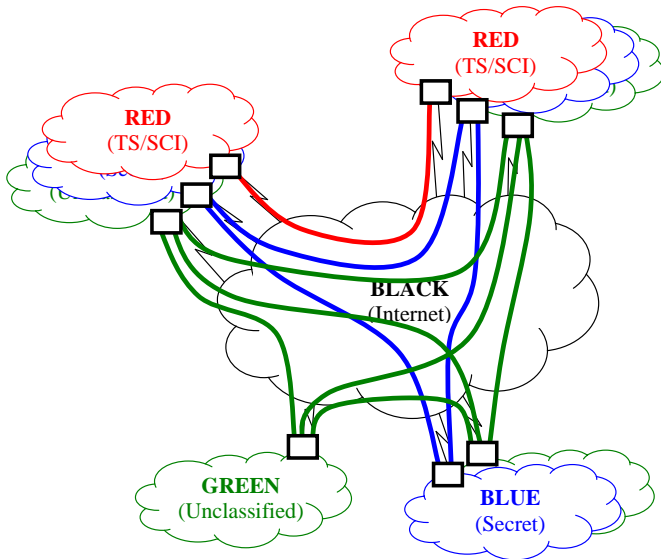
SECURING THE LAN

It is a truism that the vast majority of security threats come from insiders. Whether those threats are malicious or unintentional, sending all data within an enclave in clear text, even though that enclave only supports one security level, represents a risk that is not currently being addressed.

On the other hand, implementing IPSec between every pair of intercommunicating nodes in a local network is impractical in the extreme. So, in lieu of a workable LAN security technology, extreme measures are taken to physically secure the network infrastructure.

This is what leads to the parallel networks shown in Figure 4. Since the data is in the clear when traversing the local network, Top Secret data can not be present (and hence visible) on any network to which Secret or Unclassified clients are connected. The same is true for Secret data on Unclassified networks. So, a strict physical segregation of networks by classification level is implemented.

Collapsing parallel infrastructures: Administratively, implementing multiple parallel networks is fraught with problems. There is the obvious cost of obtaining, managing, and maintaining the necessary equipment, but there are also the intangible costs of lost productivity for those who must deal with two, three, or more workstations under their desks.

Multi-level security: What is needed instead is a method of intermixing data classified at different security levels on the same network in such a way that the data is protected from being received by any client that is not authorized to do so. This is the network version of the "holy grail" of multi-level security –

a single network infrastructure, including the switches, routers, email servers, and all of the other pieces of equipment needed to implement a network-centric enterprise.

In essence, the network becomes virtualized on demand to support the transport of different classifications of data. Figure 5 shows how the networks shown in Figure 4 could be consolidated into a set of shared local networks, each supporting multiple levels of security.

Figure 5 – Consolidated Security Enclaves
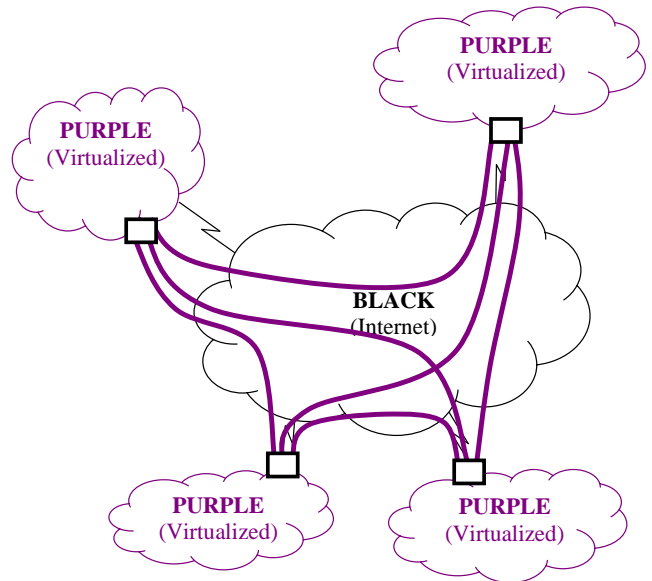
Communitites of interest: If we abstract the notion of a multi-level security network beyond that of the current tiered paradigm that supports Unclassified, Secret, and Top Secret designations, we arrive at a new paradigm that compartmentalizes network data by membership in flexible communities of interest (CoI), rather than rigid security level classifications.

Figure 6 compares the current rigid, hierarchical classification structure to a dynamic CoI model.

Figure 6 – Communities of Interest Security Model
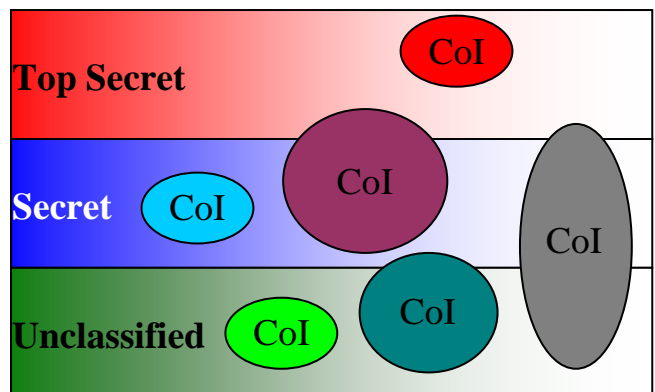
The CoI security model automates a true "need-to-know" capability for compartmented data. When necessary, a particular user can be authorized to participate in a CoI for as long as

needed. As a result, the CoI-capable network supports not just multi-level security, but also controlled data sharing/hiding to support Multi-National Information Sharing (MNIS) in coalition operations, or during cooperative joint operations with law enforcement or first responders.

MLS-NET OVERVIEW

We have demonstrated the desire for collapsing local parallel security enclaves into a single infrastructure that supports multiple security-based CoIs. Now, let us look at how that can be accomplished.

LOCAL ENCLAVE NETWORK SECURITY

MLS-Net is a network security architecture that allows the intermixing of data for different CoIs (or security levels) on the same network infrastructure. Using MLS-Net, there are no islands of cleartext within the grid of encrypted links or VPNs. Instead, all of the enterprise networks, including the local enclaves are protected. In addition, the local enclaves are protected in such a way that the physical separation of data for different CoIs is maintained.

Traditionally, data classified at different security levels is transported over physically distinct and non-interconnected networks. This physical separation is maintained all the way back to the users' desktops. MLS-Net consolidates the parallel networks into a single network, but still physically segregates the data by encrypting it and sending cryptographically-split pieces of each packet over different network paths. Data, then, is not segregated by CoI or security level, but rather by first a CoI secret (workgroup key), then by the random distribution of bits that make up the data.

The net result is that a snooper in the network will not see a coherent stream of packets between endpoints, but rather a disjoint set of encrypted partial packets. The data bits in the packets are cryptographically split in such a way that even if the snooper captured all of the partial packets that he could see at his location in the network, he could not restore the original data without also capturing all of the other partial packets spread throughout the network.

MLS-Net accomplishes this cryptographic splitting of data by utilizing the SecureParser® technology from Security First Corp.

Crypto-splitting (SecureParser®) overview: SecureParser® is not an encryption method, per se, but works in conjunction with standard encryption techniques like DES and AES to add a layer of physical security. SecureParser® takes an input buffer, shreds or "parses", the data at the bit level, then randomly assigns each bit to one or more output "shares". The distribution of the bits is controlled by a cryptographically secure pseudo random number. The resultant shares have the characteristic that a minimum subset of them is required to restore the original data.

SecureParser® operates on in-memory data segments of variable size. The SecureParser® parsing process for each segment is a nine step process, some steps of which are optional:

1) **External Key Pre-encryption** (optional): The original plain text is encrypted with an algorithm such as AES or DES. The key management for this optional step is external to the SecureParser® engine.

2) **Internal Key Generation**: In this step, two keys are generated for internal use by SecureParser®: an Internal Encryption Session Key, and a Split Session Key. These keys can be 128, 192, or 256 bits in length and are generated by a cryptographically secure pseudo-random number generator (CSPRNG).

3) **Internal Key Pre-encryption** (optional): The data segment is encrypted with the AES CTR or CBC algorithms using the Internal Encryption Session Key.

4) **All or Nothing Transform**: A form of "All or Nothing Transform" (AoNT) is used to transform the Internal Encryption Key into the Encryption Transform Session Key and the Split Session Key into the Split Transform Key. This step prevents key exposure when fewer than the minimum number of shares are present.

5) **Secure Keys**: The Encryption Transform and Split Transform Keys are divided into key shares using the Shamir key splitting technique. Each key share is distributed to one of the output shares. If requested, the Split Transform Key may also be encrypted with a Workgroup Key provided by the user.

6) **Parse**: The original plain or pre-encrypted data is shredded at the bit or byte level, and each of those pieces is randomly distributed to one or more of the output shares. The parsing algorithm uses the Split Transform Key to determine the distribution.

7) **Fault Tolerance**: When fault tolerance is specified (see the M-of-N discussion below), each piece of shredded data is written to more than one output share. This allows the restoration of the original data with a minimum subset, as opposed to all, of the shares.

8) **Share Authentication**: Integrity information is written to each share to allow the detection of corrupted shares. In addition, a Message Authentication Code (MAC) may also be generated.

9) **Post-encryption** (optional): Each output share may optionally be encrypted using a key provided by the user.

10) **Distribute**: Each share is distributed to a separate storage location, or transmitted over a separate networking path. This step is external to the SecureParser® engine.

Figure 7 shows a schematic of the SecureParser® processing. The steps shown in dashed-line boxes are optional.

Figure 7 – SecureParser® Processing Steps
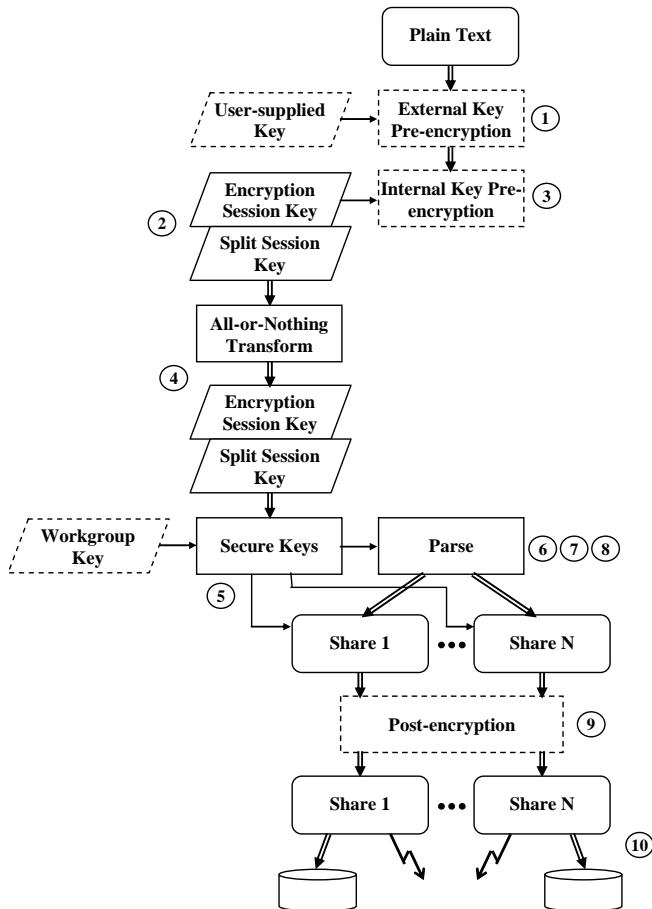
In the simplest mode of operation, i.e. without any of the optional functions, or even with the Internal Pre-encryption (step 3), the keys used (the Encryption Session Key and Split Session Key) are generated internally, they are then split, and the key shares are stored together with the data shares. As a result, no external key management is needed. Rather, access to the separate shares is, in essence, the "key".

This is a big advantage over using just straight encryption to protect the data, especially when the data must persist for a long period of time, such as backup/archive data. If an external key is used, that key must persist and be available as long as the data it is protecting exists. So, for situations where physical separation of the data shares is sufficient protection, key management is not a concern.

It was mentioned previously that each piece of split data is parsed into one or more shares. The reason the data may be put into more than one share is to allow for resiliency in the case where one or more of the data shares are lost or corrupted.

SecureParser® can be configured to support "M-of-N" redundancy – N shares are generated, but only M of them are required to restore the original data. So, in a 2-of-3 scenario, the original data is parsed into three shares such that any two of them can reconstruct the original.

There are many situations where this type of redundancy is a big

advantage. For disaster recovery purposes, mission critical data must be duplicated, often to a remote site. Without splitting, all of the data would need to be recovered before processing could continue. Using split redundancy, however, processing can continue on the remaining shares (which can still restore the original data), while a new set of redundant shares are created.

Note that even though the individual shares are smaller than the original data, there is no savings in the total amount of data. Indeed, if the configuration allows one share to be lost, the data storage or bandwidth needed multiplies. For example, in a 3-of-4 scenario, each bit must be in two shares, so the total data doubles. Or, if M-of-N is 2-of-4, each bit must be in three shares, since two may be lost. This triples the data

Figure 8 shows a notional example of a single character 'J' being parsed and restored in a 2-of-3 manner. It does not represent the actual internal algorithm, which is proprietary to Security First Corp.

Figure 8 – 2-of-3 Example
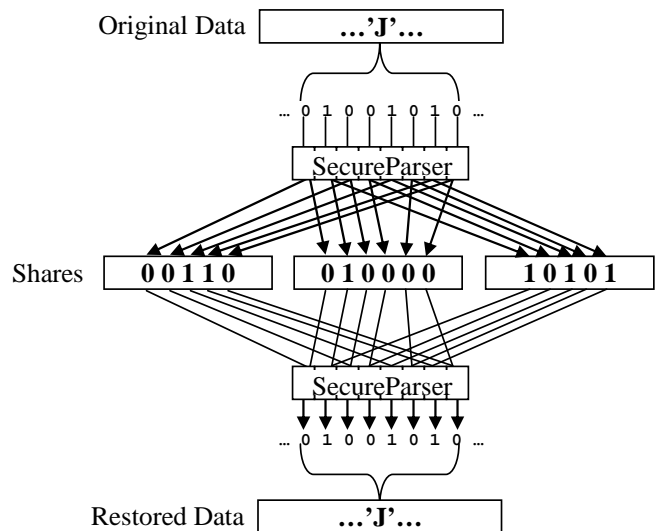
One additional wrinkle in the redundancy capabilities of SecureParser® is the ability to specify a certain number (L) of mandatory shares. These mandatory shares are required for restoration regardless of the additional M-of-N specification. So, in an L-and-M-of-N case, where L is 1, M is 2, and N is 3, four shares (L+N) will be created. The sets of shares which can restore the original data are: {1, 2, 3, 4}, {1, 2, 4}, {1, 3, 4}, and {2, 3, 4}. Note that share #4 is always required. If L was 2, shares #4 and #5 would be required.

As noted above, the internal Encryption Session and Split Session keys are themselves split and stored in the output shares. This allows the data to be restored without using any external keys once a minimum subset of shares is located. For situations where additional security beyond the physical separation of shares, is required, the Split Session Key can be encrypted with an external Workgroup Key. The Workgroup Key is a symmetric key that is also required during restoration.
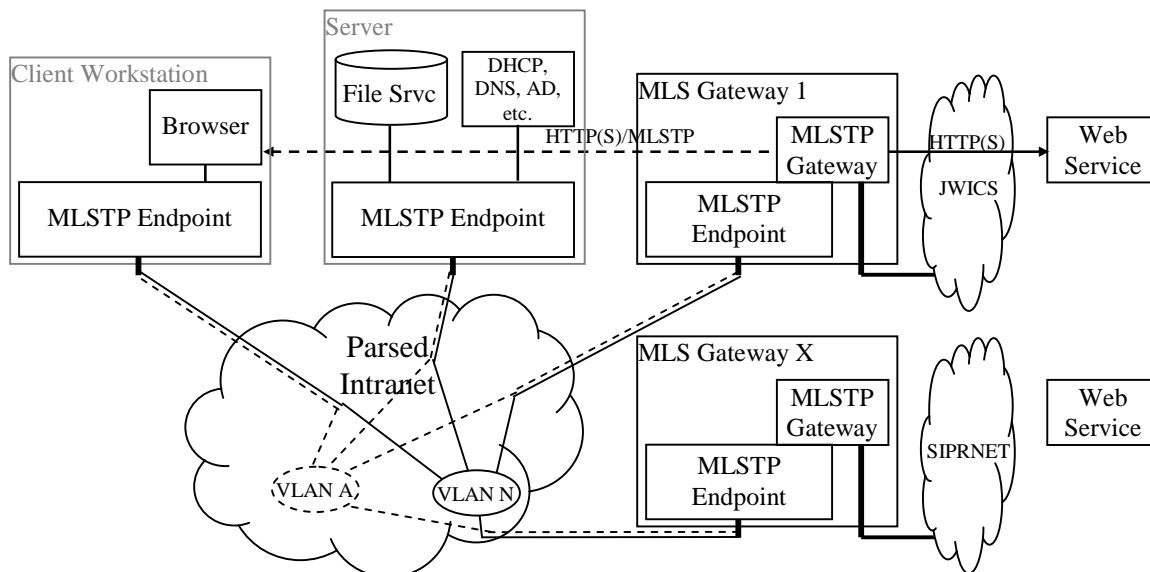
It has been demonstrated by Bellare and Rogaway [1] that this

crypto-splitting technique is a stronger form of encryption than straight AES.

MLS-NET ARCHITECTURE

Figure 9 shows a schematic representation of the components of the MLS-Net architecture.

Figure 9 – MLS-Net Architecture

The important components of MLS-Net are the Multi-Level Security Tunneling Protocol (MLSTP), the MLSTP Endpoint and MLSTP Gateway components that implement it, and the VLANs within the parsed intranet.

MLSTP utilizes an N-split tunnels between each pair of Endpoints. Each packet that is transmitted between those endpoints is cryptographically split using SecureParser® into N shares, which are wrapped inside MLSTP packets, and which are then sent through N distinct network paths to the partner Endpoint.

To ensure that the share packets take separate paths through the network, and therefore maintain physical separation, the tunnels run over separate 802.1q virtual LANs (VLANs). The VLANs are rooted in different switches, resulting in unique spanning trees (i.e. paths) for each VLAN. Also, each Endpoint is represented by N IP addresses (one for each VLAN), which are on separate subnets. Router ports are assigned to those VLANs/subnets in such a way that the share packets comprising an original message are routed differently.

The Endpoints manage the tunnels, acting as an intermediary between the IP network stack and the low-level LAN drivers. The data is protected by the cryptographic splitting and by the separation of the different shares as they traverse the network. A potential hacker would have to snoop the network traffic at a point on at least M of the VLANs/routes to capture enough shares to restore the original data. Various obfuscation techniques, such as IDLE packet insertion and random transmission delays, are also employed to confuse a snooper, even if all of the traffic is captured. Of course, if all traffic is captured and the obfuscation techniques are defeated, the snooper still must break the cryptographic split and the AES encryption.

The feature of MLSTP that allows different CoI data to intermix on the same network, however, is the use of workgroup keys within the Endpoints. When a user is working in a particular CoI or security level, say Top Secret, all traffic is encrypted using a Top Secret workgroup key. If the user wishes to switch to a different CoI or security level, say Secret, the Endpoint closes all tunnels which were established with the Top Secret key, and reestablishes them as necessary with the Secret key.

Workgroup keys are associated with a user, not a physical workstation, and are distributed to the workstation when a particular user logs on. In the most common identity management system, Windows Domain Controller/Active Directory, the user's set of workgroup keys are wrapped in PKI certificates and stored in the user profile. The set of workgroup keys that is assigned depends on the user's authorizations within the enterprise's user management system (domain controller and Active Directory, for example).

The other component shown in Figure 9 is the MLSTP Gateway. The MLSTP Gateway acts as a proxy allowing communications between the local parsed intranet and external, single-security level networks such as JWICS or SIPRNET, or the black internet via a HAIPE device.

For webcasting, audio and video teleconferencing, or other collaborative applications, multicast traffic is also supported.

CONCLUSION

We have reviewed the state of the art with respect to network protocols that secure data while in motion. In the process, we have seen that the current network security protocols and practices are lacking a couple of key capabilities. Notably, data is

not secured while in motion within a local enclave, and as a result, separate local network infrastructures must be maintained to physically separate data associated with different security levels or communities of interest.

A solution to both of these shortcomings, called MLS-Net, was presented. MLS-Net utilizes a cryptographic-splitting technology called SecureParser® created by Security First Corp. SecureParser® splits data using advanced secret-sharing algorithms in such a way that a minimum number of the split off pieces, or shares must be present to restore the original data. MLS-Net transmits split shares of IP packets over different network paths within the local network enclave.

REFERENCES

Several of the references below are internal whitepapers which have been distributed to prospects at various conferences. They may be obtained from the author upon request. The author may be contacted at Robert.Johnson@unisys.com.

SecureParser® is a registered trademark of Security First Corp., all other product names are trademarks of their respective owners.

[1] S. Schnitzer, R. Johnson, and H. Hoyt, "Secured Storage Using SecureParser®," Proceedings of the 2005 ACM Workshop on Storage Security and Survivability (2005).

[2] Security First Corp., "SecureParser® Beyond Encryption v3.5," whitepaper (2005).

[3] Security First Corp., "SecureParser® Design Specification v4.1," whitepaper (2005).

[4] Security First Corp., "SecureParser® Storage Overview v4.1," whitepaper (2005).

[5] Security First Corp., "SecureParser® Cryptographic Core Design v4.1," whitepaper (2006).

[6] Security First Corp., "SecureParser® Workgroup Key Usage Notes v 4.1," whitepaper (2006).

[7] M. Bellare, P. Rogaway, "Robust Computational Secret Sharing and a Unified Account of Classical Secret-Sharing Goals", unpublished manuscript (2006)