

UNISYS
imagine it. done.

Consulting • System Integration • Outsourcing
Infrastructure • Server Technology

The Unisys Stealth Solution and SecureParser: A New Method for Securing and Segregating Network Data

Robert A. Johnson

White paper



secure

This white paper presents a new method of network security and virtualization that allows the consolidation of multiple network infrastructures, each dedicated to a single security level or community of interest, onto a single, virtualized network. The overview of state-of-the-art network security protocols includes the use of SSL, IPSec, and HAIPE IS, followed by a discussion of the SecureParser® technology and Unisys Stealth architecture, which in combination allow secure sharing in consolidated networks.

Table of Contents

Introduction	5
Network Security	6
Secure Sockets layer (SSL)	6
Virtual Private networks (VPNs)	6
Internet Protocol Security (IPSec)	7
High Availability Internet Protocol Encryptor (HAIPE)	8
Secure Sockets Layer Virtual Private Network	9
Security Enclaves within GIG—A Closer Look	9
Securing the LAN	10
Multilevel Security	10
Communities of Interest	11
Unisys Stealth Solution Overview	12
Local Enclave Network Security	12
Crypto-Splitting: SecureParser Overview	12
Unisys stealth Solution Architecture	15
Conclusion	17
References	18
About the Author	19

This page is intentionally left blank.

Introduction

Securing network data while it is in motion is an increasingly important requirement for the enterprise networks of today. 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 review various state-of-the-art network protocols that secure data while it is in motion. These reviews uncover shortcomings in those methods that highlight a couple of key capabilities that are lacking. Specifically, one shortcoming is that data is not secured while in motion within a local enclave. As a result, separate local network infrastructures are often maintained to physically separate data associated with different security levels or communities of interest (COIs).

We also examine a solution to these shortcomings—namely, the Unisys Stealth Solution for Network. The Unisys Stealth Solution including SecureParser from Security First Corp. closes the gap in network security so that separate enclaves for different security levels or COIs need not be maintained, and all data can be securely intermixed within the same network infrastructure.

The Unisys Stealth Solution including SecureParser from Security First Corp. closes the gap in network security so that separate enclaves for different security levels or COIs need not be maintained.

Network Security

In examining state-of-the-art network security, we specifically consider how data is protected while in flight between clients and network resources, between server-based applications, and between different enterprise networks.

Several different protocols protect data while it is in motion and many different products implement those protocols. In this discussion, we focus on the protocols, not the associated products.

As with most networking problems, you can address the problem of data security at different layers within the protocol stack. In general, you target solutions at the session layer, network layer, and below, as shown in Figure 1.

OSI Layer	Security Methods
Presentation	
Application	User credentialing
Session	SSL
Transport	
Network	IPSec VPN, HAIP-ES, SSL VPN
Link	Unisys Stealth
Physical	Link encryptors

Figure 1. Security at Different Network Layers

Wide-area-network (WAN) links are often protected by hardware-based link encryptors, which are point-to-point devices that 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.

Secure Sockets Layer (SSL)

Starting at the top of the protocol stack, 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 worldwide 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—for example, 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—for example, 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 encrypts the message with Alice's public key (A1). Alice then uses her private key (A2) to decrypt the message. Because A2 is kept secret by Alice, only she can decrypt the message.

Although 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 disadvantage occurs 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.

Virtual Private Networks (VPNs)

Another approach puts the security processing in the network layer. You use network-layer protocols to form VPNs that 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, a hotel, or even an Internet café—and securely access network resources within the enterprise as if the

workstation was connected directly through 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. Which type of burden is more palatable to an enterprise is what often determines the type of VPN deployed.

Internet Protocol Security (IPSec)

The most popular form of VPN in use today is based on the Internet Protocol Security (IPSec) protocols. IPSec is a set of protocols that allows the transport of secure information between two enclaves that 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 IPSec use.

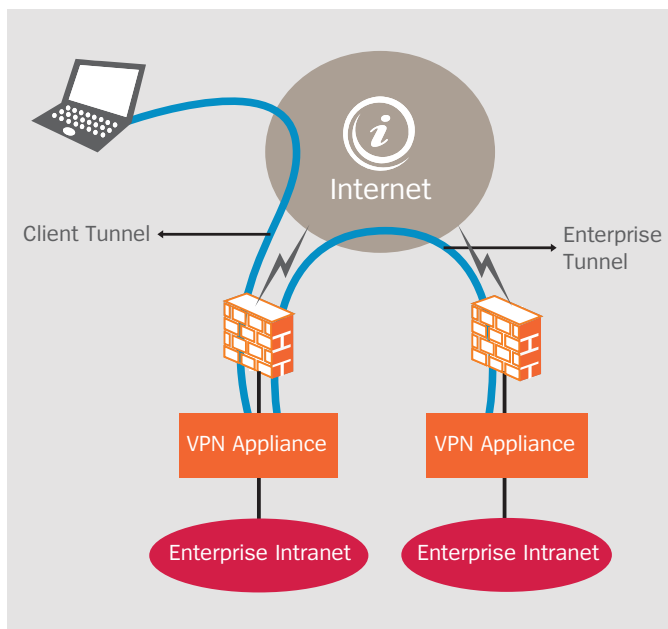


Figure 2. IPSec Example

IPSec is an integral part of IP version 6 (IPv6) and an optional enhancement to IPv4. Because it is an add-on to IPv4, the native IPv4 stacks of most operating systems do not support IPSec directly. As a result, two types of deployments might be required. On the client workstation, software that adds the IPSec functionality that integrates with the native protocol stack must be installed. On the enterprise side, however, it is more efficient to deploy a VPN appliance that 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
- Tunnel Mode, which encrypts the entire packet, including the original IP header, and adds a new IP header

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. Therefore, it is mainly used when most application options would not be supported, for example 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 restriction forces the network to be thought of as enclaves of non-IPSec networks connected by IPSec virtual connections. This approach 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 using the Internet for financial or other reasons.

High Availability Internet Protocol Encryptor (HAIZE)

The U.S. Department of Defense (DoD) 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 (for example, Top Secret) may not be accessed by a person who is cleared for only a lower level of access (for example, 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 are interconnected through the “black” network—that is, the public Internet. In essence, the GIG uses the Internet as its backbone. This usage makes sense, because 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 you can connect homogeneous security level enclaves to other enclaves of the same security level through the use of VPNs.

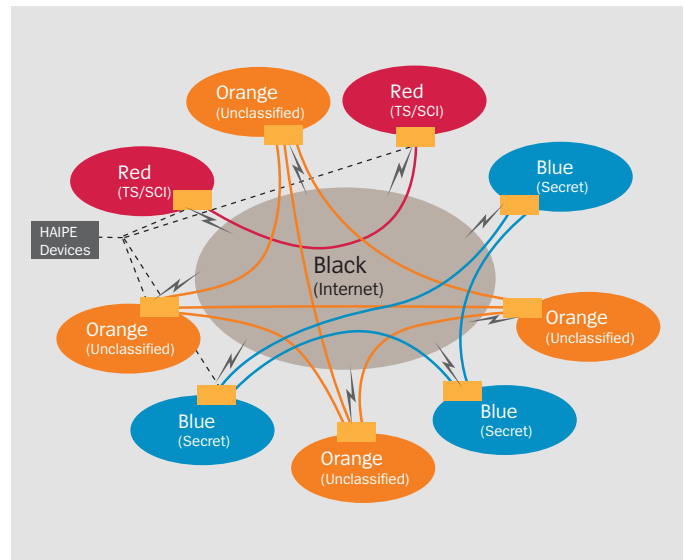


Figure 3. Security Enclaves within the GIG

The VPN technology used within the GIG is called High Assurance Internet Protocol Encryptor Interoperability Specification (HAIZE IS, or just HAIZE, pronounced “hay’ pea”). HAIZE is an enhanced version of IPSec. It has two modes, a Tactical Mode, which is functionally equivalent to the Transport Mode of IPSec, and a Strategic Mode, which is likewise equivalent to Tunnel Mode.

HAIZE devices (the actual “Encryptor” from the name) are standalone appliances that replace normal IP routers and their associated routing protocols. Current implementations of HAIZE require the devices to be statically configured with enough of the network configuration to function as routers on behalf of their local enclaves. Ongoing research is 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 key management. The administration and maintenance of the key management and distribution system are complex and place 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 HAIZE to date.

Secure Sockets Layer Virtual Private Network

The final VPN technology to consider is one that is becoming very popular in the commercial sector. SSL-based VPNs are a hybrid of the VPN concept and the SSL technology previously discussed. The perceived benefit of SSL VPNs is the lower administrative overhead.

This lower overhead results because 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 is 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 that 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 might not be obvious. The configuration of the appliance is centralized, whereas the installation and repair of IPSec client software is not. This difference alone is a significant differentiator for understaffed IT shops. Add to that the enabling of PDAs, smart phones, etc., and the attraction of SSL VPNs becomes apparent.

However, SSL VPNs 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.

Secure Enclaves within GIG—A Closer Look

So, what is wrong with Figure 3, which shows secure enclaves connected using HAIPE to each other across the black Internet? First, there is no security for data in motion within the local enclaves. Secondly, 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.

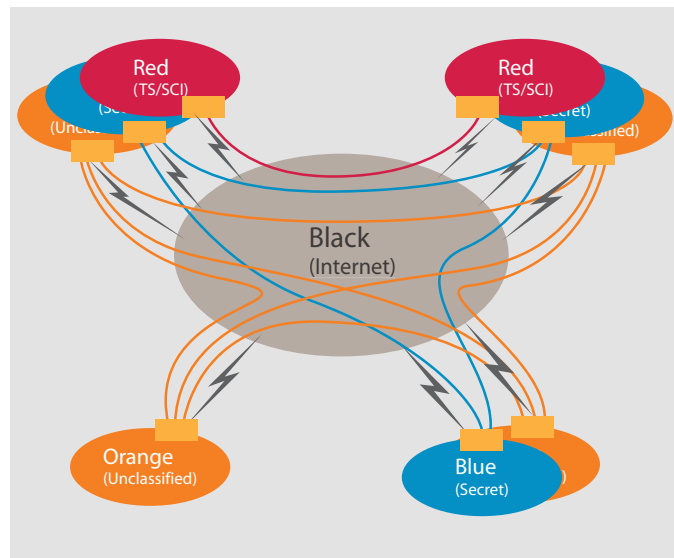


Figure 4. Parallel Enclaves

Securing the LAN

It is true 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 local area network (LAN) security technology, extreme measures are taken to physically secure the network infrastructure.

This physical security is what leads to the parallel networks shown in Figure 4. As the data is in the clear when traversing the local network, Top Secret data cannot 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.

Administratively, implementing multiple parallel networks is fraught with problems. There is the obvious cost of obtaining, managing, and maintaining the necessary equipment, plus extra cost for power, cooling, space, and weight. There are also the intangible costs of lost productivity for those who must deal with two, three, or more workstations under their desks.

Multilevel 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 network version is the “holy grail” of multilevel security—a single network infrastructure that includes 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

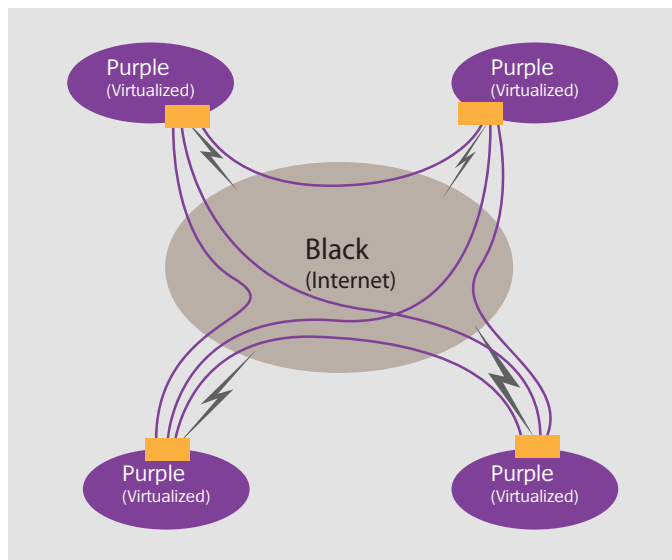


Figure 5. Consolidated Security Enclaves

Communities of Interest

If we abstract the notion of a multilevel 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 (COIs), rather than rigid security level classifications.

A COI is a group of people who must share information and not let anyone outside of their COI access that information. The same individual could be part of more than one COI, either one at a time or multiple at the same time depending on the organization's security requirements.

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

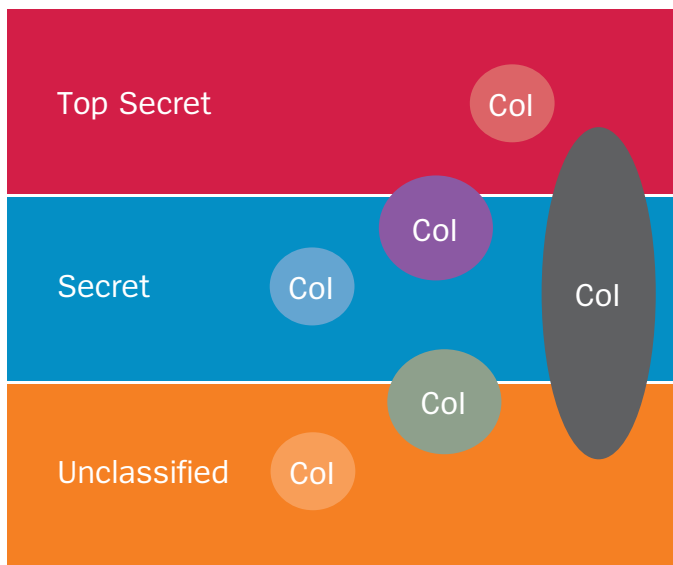


Figure 6. Communities of Interest Security Model

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 only multilevel 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.

Unisys Stealth Solution Overview

Considering the various protocols and security needs, collapsing local parallel security enclaves into a single infrastructure that supports multiple security-based COIs is a desired solution. Now, let's consider how that can be accomplished.

Local Enclave Network Security

The Unisys Stealth Solution is a network security architecture that allows the intermixing of data for different COIs (or security levels) on the same network infrastructure. Using Stealth, 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. The Unisys Stealth Solution 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 the network separately. Data is not physically segregated by COI or security level, but rather by a COI-specific secret (workgroup) key, then by the random distribution of bits that make up the data.

The Unisys Stealth Solution accomplishes this cryptographic splitting of data by utilizing the SecureParser from Security First Corp.

Crypto-Splitting SecureParser Overview

SecureParser is not an encryption method, 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 or "slices." The distribution of the bits is controlled by a cryptographically secure pseudo-random number. The resultant slices 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 sizes. The SecureParser parsing process for

each segment is a ten-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 slices 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 slices. 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 level, and each of those pieces is randomly distributed to one or more of the output slices. The parsing algorithm uses the Split Transform Key to determine the distribution. This patented process is unique to SecureParser.
7. Fault Tolerance: When fault tolerance is specified (see the M-of-N discussion that follows), each piece of shredded data is written to more than one output slice. This distribution allows the restoration of the original data with a minimum subset, as opposed to all of the slices.
8. Slice Authentication: Integrity information is written to each slice to allow the detection of corrupted

slices. In addition, a Message Authentication Code (MAC) may also be generated.

9. Post-encryption (optional): Each output slice might optionally be encrypted using a key provided by the user.
10. Distribute: Each slice is distributed to a separate storage location or transmitted in a separate network packet. 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.

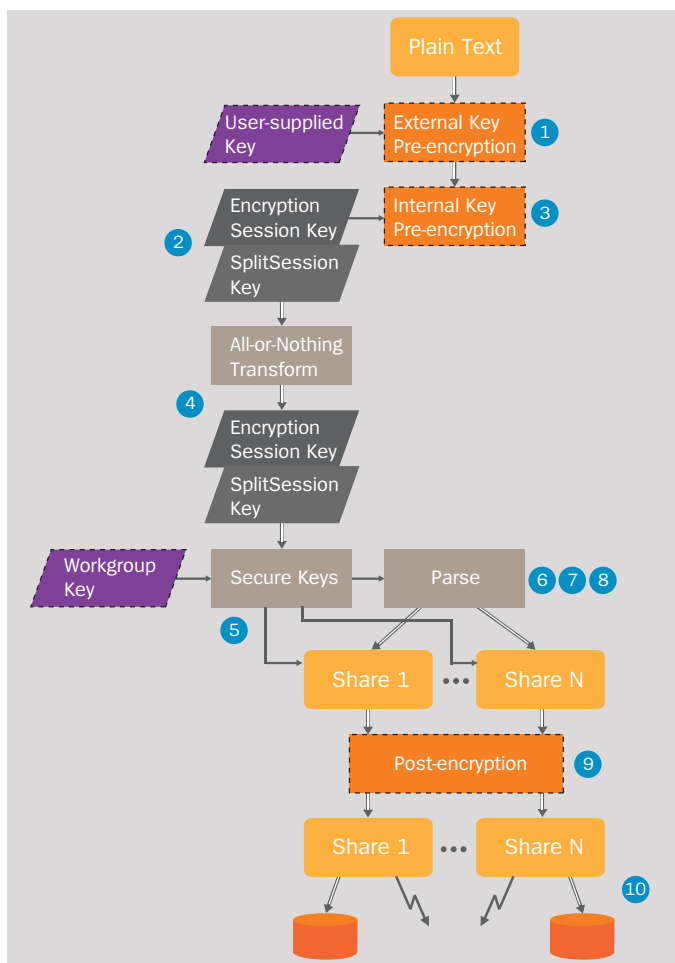


Figure 7. SecureParser Processing Steps

In the simplest mode of operation (that is, 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.

Optionally, SecureParser can include these keys with the data. In this case, these session keys are themselves split, and the key shares are stored together with the data slices. As a result, no external key management is needed. Rather, access to enough of the separate slices is, in essence, the “key.” This option 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 slices is sufficient protection, key management is not a concern.

As mentioned previously, each piece of split data is parsed into one or more slices. The reason the data might be put into more than one slice is to allow for resiliency in the case where one or more of the data slices are lost or corrupted.

You can configure SecureParser to support M-of-N redundancy—N slices 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 slices such that any two of them can reconstruct the original.

In many situations, 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, processing can continue on the remaining slices (which can still restore the original data), while a new set of redundant slices are created.

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

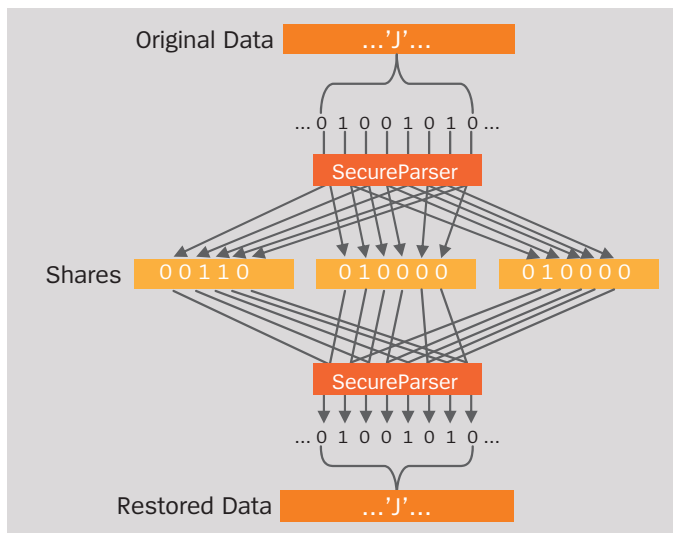


Figure 8. 2-of-3 Example

One additional wrinkle in the redundancy capabilities of SecureParser is the ability to specify a certain number (L) of mandatory slices. These mandatory slices 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 slices (L+N) are created, where the fourth is the mandatory slice and is always required. The sets of slices that can restore the original data are: {1, 2, 3, 4}, {1, 2, 4}, {1, 3, 4}, and {2, 3, 4}. If L was 2, slices #4 and #5 would be the mandatory slices and both would be required to recover the original data.

As noted, the Internal Encryption Session and Split Session keys can be each split and stored in the output slices. This scheme allows the data to be restored without using any external keys once a minimum subset of slices is located. For situations where additional security beyond the physical separation of slices 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.

Unisys Stealth Solution Architecture

Figure 9 shows a schematic representation of the components of the Unisys Stealth architecture.

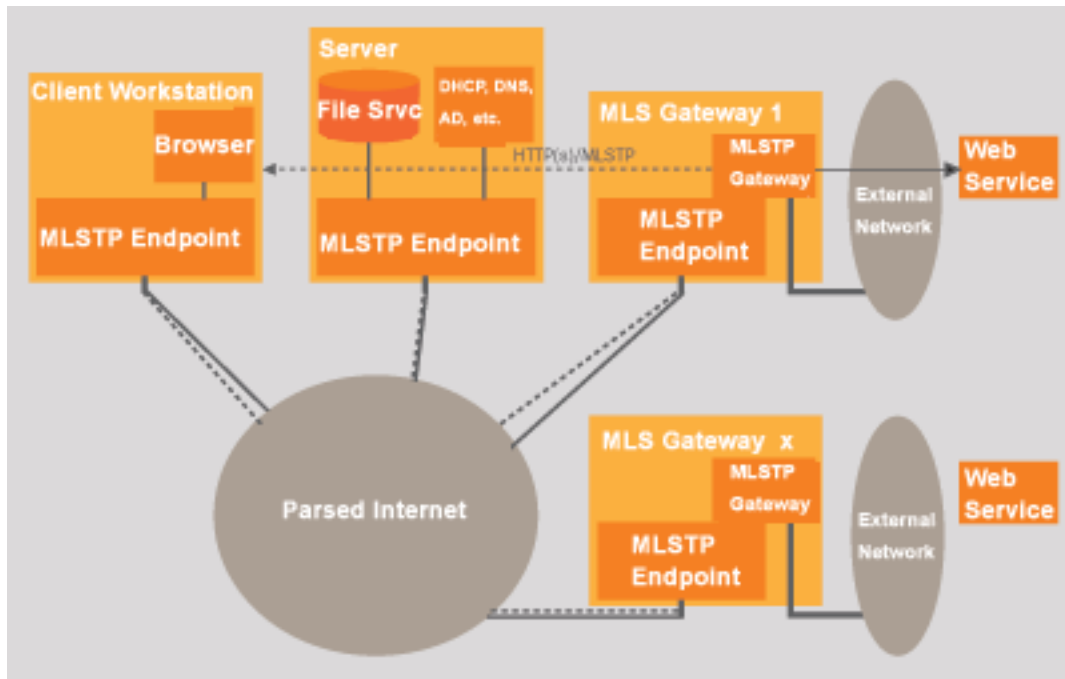


Figure 9. Stealth Solution Architecture

The important components of the Unisys Stealth Solution are the Multi-Level Security Tunneling Protocol (MLSTP), and the MLSTP Endpoint and MLSTP Gateway components that implement it. MLSTP uses SecureParser to encrypt and parse a sent packet, and to unparse and decrypt received packets. MLSTP uses the following SecureParser processes:

- **Step 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).
- **Step 3 Internal Key Pre-Encryption:** The data segment is encrypted with the AES CTR or CBC algorithms using the Internal Encryption Session Key.
- **Step 6 Parse:** The original plain or pre-encrypted data is shredded at the bit level, and each of those pieces is randomly distributed to one or more of the output slices. The parsing algorithm uses the Split Transform Key to determine the distribution.

- **Step 7 Fault Tolerance:** When fault tolerance is specified (see the M-of-N discussion previously presented), each piece of shredded data is written to more than one output slice. This distribution allows the restoration of the original data with a minimum subset, as opposed to all of the slices.
- **Step 8 Slice Authentication:** Integrity information is written to each slice to allow the detection of corrupted slices. In addition, a Message Authentication Code (MAC) may also be generated.

MLSTP enforces COI separation. Each COI is assigned a workgroup key. Workgroup keys are associated with a user, not a physical workstation, and are distributed to the workstation when a particular user logs on. 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). Servers receive their workgroup key(s) when they power on.

MLSTP is delivered as a network driver that sits just above the link layer (layer 2) in the network stack. Therefore, it is below the operating system and application activities, and above the existing network infrastructure. This positioning permits Stealth to be

incrementally implemented as opposed to a rip-and-replace scenario.

Let's assume Alice wants to Instant Messenger (IM) Bob, and they are both using Stealth. Alice types "Hi, Bob" in her IM window, and IM creates the appropriate message to go to Bob's workstation. Stealth, in Alice's workstation's network stack, now opens a connection to Bob's workstation. Stealth, using SecureParser, generates a new encryption session key and a new split session key. Stealth creates an "open session" request that contains both of these new keys. Stealth then encrypts, using SecureParser, this message with Alice's workgroup key, parses it into multiple slices also using Alice's workgroup key, and sends the resulting slices to Bob's workstation.

Stealth running in Bob's network stack tries to use SecureParser with his workgroup key to unparse and decrypt this message. If successful, then Alice and Bob share a workgroup key, and Stealth in Bob's workstation knows the keys to interpret the messages from Alice. Stealth in Bob's workstation reverses the process and creates an "open session" message back to Alice's workstation with a different pair of keys. If Bob has multiple workgroup keys, Stealth attempts to unparse and decrypt with each one until the operation is either successful or fails.

If none of Bob's workgroup keys match Alice's workgroup key, then Bob's workstation does not respond to Alice—the message is ignored, and not sent up the network stack into Bob's workstation. If Stealth in Alice's workstation doesn't get a response from Bob's workstation and Alice has more than one workgroup key, Stealth in Alice's workstation tries each of her workgroup keys to determine if Alice and Bob share a workgroup key. If they don't share a workgroup key, they won't communicate. In that case, Alice can't even ping Bob—Bob's workstation doesn't exist for her.

These encryption and split session keys are unique for each direction of each conversation in a Stealth network. When the session terminates for any reason, the keys are destroyed. In the previous example, later if Alice and Bob need to communicate, Stealth creates a new pair of keys for each direction.

The Endpoints manage the tunnels, acting as an intermediary between the IP network stack and the low-level LAN drivers.

The feature of MLSTP that allows different COI data to intermix on the same network is the use of workgroup keys within the Endpoints. When a user is working in a particular COI or security level—for example, 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—for example, Secret—the Endpoint closes all tunnels that were established with the Top Secret key and reestablishes them as necessary with the Secret key.

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 or the black Internet through a HAIPE device.

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

Conclusion

Regarding the state of the art with respect to network protocols that secure data while it is in motion, we have determined 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 COIs.

The Unisys Stealth Solution for Network is a solution to both of these shortcomings. This solution 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 slices must be present to restore the original data. The Unisys Stealth Solution transmits split slices of IP packets over a LAN, WAN or wireless network enclave

References

A substantially similar version of this paper was published in the Proceedings of the Risk Management in Cyber-Informatics 2007 conference of the International Institute of Informatics and Systemics.

Several of the references below are internal whitepapers, which have been distributed to prospects at various conferences. You can obtain the white papers from the author at Robert.Johnson@unisys.com.

- [SCHN05] S. Schnitzer, R. Johnson, and H. Hoyt, "Secured Storage Using SecureParser®," Proceedings of the 2005 ACM Workshop on Storage Security and Survivability (2005).
- [SFC05-1] Security First Corp., "SecureParser® Beyond Encryption v3.5," white paper (2005).
- [SFC05-2] Security First Corp., "SecureParser® Design Specification v4.1," white paper (2005).
- [SFC05-3] Security First Corp., "SecureParser® Storage Overview v4.1," white paper (2005).
- [SFC06-1] Security First Corp., "SecureParser® Cryptographic Core Design v4.1," white paper (2006).
- [SFC06-2] Security First Corp., "SecureParser® Workgroup Key Usage Notes v 4.1," white paper (2006).
- [SFC06-3] M. Bellare, P. Rogaway, "Robust Computational Secret Sharing and a Unified Account of Classical Secret-Sharing Goals," unpublished manuscript (2006)

About the Author

Robert A. Johnson

**Director of Security Products Architecture,
Systems and Technology, Unisys Corp.**

Mr. Johnson is a 1981 graduate of Bucknell University, where he majored in Mathematics and Computer Science. Since 1981, he has held many positions—first with Burroughs, then with Unisys. His responsibilities have primarily included designing and implementing I/O and networking subsystems for the full range of computers from PCs to mainframes. Rob has also been responsible for managing innovation activities within the Unisys Systems and Technology engineering community and has acted as an engineering liaison to the Unisys Federal business unit. Currently, he is the lead architect for the Unisys Stealth Solutions program. Rob has published several technical papers and trade journal articles and holds many U.S. and foreign patents

For more information, please visit www.UnisysStealthSolution.com

©2008 Unisys Corporation.

All rights reserved.

Unisys and the Unisys logo are registered trademarks of Unisys Corporation. All other brands and products referenced herein are acknowledged to be trademarks or registered trademarks of their respective holders.

Printed in the United States of America 12/08

3839 3377-000