# Compaq DSNlink and WorldWire Security
## White Paper

*Michael Spratte and Steve Roscio*
*Compaq Computer Corporation*

*December 18, 2000*

## INTRODUCTION

This paper discusses the security issues of the design, architecture, implementation, and deployment of the Compaq DSNlink and Compaq WorldWire service tools.

### DSNlink and WorldWire Service Tools

The DSNlink and WorldWire service tools provide a suite of components that customers and Compaq Computer Corporation use to establish and maintain secure electronic communications between their respective sites. Both service tools are part of Compaq's Distributed Services Network (DSN) architecture. DSNlink applications allow customers to:

- Create, review, and augment service requests with a Compaq Customer Support Center
- Search Compaq problem/symptom/solution databases
- Copy files to and from a Compaq Customer Support Center

An additional DSNlink application, DSNlink remote login, allows a Compaq support engineer, with an appropriate authorization on the remote system, to establish a telnet-like session on a customer's system and perform command-line operations.

The main WorldWire application, the TransPortal, is a virtual router that allows application clients on customer systems to connect to companion application servers on systems at a Compaq site and vice versa.

Several products and service tools produced by Compaq Computer Corporation, including Compaq Remote Support (CRS), System Management Console (SMC), Compaq Analyze, Compaq Crash Analysis Tool (CCAT), and Revision and Configuration Management (RCM), rely on WorldWire for electronic connectivity between Compaq Customer Support Centers and customer sites. The WorldWire TransPortal can also be configured for use with third party, off-the-shelf applications.

DSNlink and WorldWire use the same Network Exerciser Application, Netex, to test the connection between the customer site and Compaq. Netex can troubleshoot routing issues and security settings.

The current implementations of these service tools are *Compaq DSNlink Version 3.0 for OpenVMS™*, *Compaq DSNlink Version 3.0 for Compaq Tru64™ UNIX*, and *Compaq WorldWire Version 2.8* that runs on Microsoft Windows NT and Windows 2000.

### DSN Connections

WorldWire and DSNlink share the same communications software core that creates a **DSN connection**. A DSN connection provides secure peer-to-peer communications in a potentially hostile networking environment using different network protocols. Features include:

- Capability for a single logical connection over heterogeneous network protocols, including TCP/IP, WorldWire RAS (TCP/IP over Microsoft Windows RAS), DECnet, X.25, and the DSNlink modem protocol
- Strong cryptographic authentication
- Customer-controlled authorization
- Data confidentiality via selectable encryption
- Data compression
- Business-based routing

### Security is the Highest Priority

Security has been designed into DSNlink and WorldWire, not added on. Our commitment to security follows through all aspects of the product lifecycle: design, architecture, implementation, and deployment. More specifically:

- We start with an architecture where functionality has been cleanly partitioned into layers, each with clearly defined roles.
- After establishing identity through strong cryptographic authentication, a DSN connection is encrypted. Before accepting a connection, each end authorizes usage.
- We use well-known and well-respected algorithms for authentication, encryption, and random number generation.
- Installing and using DSNlink or WorldWire does not increase a system's exposure to attacks.
- We prescribe a protocol for simple business practices.
- The user has control of security parameters.
- All applications log every connection.

### DSN Development Practices

We feel that good software engineering practices promote not only a higher quality product, but a more secure one as well:

- DSNlink's multi-platform implementations are based on a single source code pool; the same code runs on all supported platforms. Enhancements and bugs fixes performed for one operating system are automatically incorporated on the other operating systems. Thus we can concentrate on the algorithms used, not the implementation differences.
- The small development team has years of experience working together providing remote customer support solutions.
- Since 1995 DSNlink has been deployed worldwide to over 20,000 customers in North America, South America, Europe, Asia and Australia.

### Assumptions

When discussing security and DSN, we need to make a few assumptions:

1. Only the two end systems of a DSN connection, the DSN client and DSN server, can be trusted.
2. The two end systems of a connection are physically secure. We must emphasize that anyone who has physical access to a system can control it and, hence, defeat any security mechanisms including strong cryptography.
3. Individuals who have either physical access or privileged access to an end system are trusted. If an attacker gains access to an authentication key, he can either impersonate the customer to Compaq or impersonate Compaq to the customer.
4. Any underlying network cannot be trusted and a connection over any of these networks is susceptible to many attacks including impersonation, eavesdropping, connection hijacking, and malicious data modification.
5. Any system involved in routing any segment of a DSN connection, including routers and firewalls, is not necessarily trusted.

And some points to keep in mind when evaluating security:

- The security of the whole system is only as good as the security of its weakest link. The weakest link is most likely a human.
- It is impossible to prove that a system is secure. However, one can prove that it is insecure. For example, in November 1999, Science Applications International Corporation (SAIC) certified that "the security features and assurances provided by Windows NT 4.0 with Service Pack 6a and the C2 Update with networking meet the C2 requirements of the Department of Defense Trusted Computer System Evaluation Criteria (TCSEC) dated December 1985." Yet, Windows NT regularly falls victim to attacks.
- Complexity decreases security.
- Security takes effort and dealing with it is not always convenient. Generally speaking, security tends to suffer at the expense of user-friendliness.

### Risk

DSN aims to protect information that a customer will transfer to a Customer Support Center. The most precious information is passwords. Passwords are used when a Compaq support specialist logs into a customer system. Passwords can also be embedded in a crash dump that a customer wishes to send to Compaq for analysis. With an appropriate username/password combination and access to the system, an attacker may gain enough access to cause severe damage, such as steal intellectual property or proprietary information such as a database of credit card numbers.

Generally, the risk of an attack increases with the value of the information needing protection. Contrarily, as the cost of compromising a system increases, the likelihood of an attack decreases.

## DSN ARCHITECTURAL OVERVIEW

To understand DSN security, you need to first know the architecture and the roles and responsibilities of the layers in the architecture.

## DSN Domains, Nodes and Applications

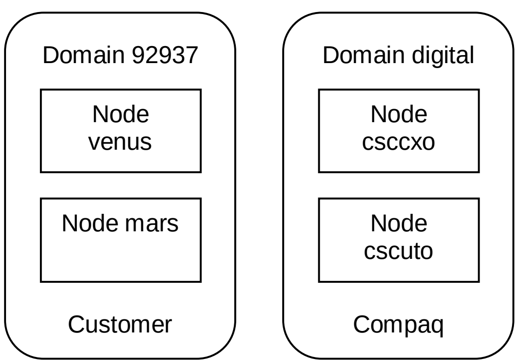
DSN uses domain and node names to identify individual systems. Furthermore, each connection is associated with an application.

A **DSN domain** is a character string that identifies a contractually related group of systems owned by a DSNlink or WorldWire customer. The DSN software uses domain values for identification, authentication, authorization, obligation fulfillment, and network routing. There are often several computer systems within the same domain. Compaq uses a customer's access number, obligation identifier, hardware serial number, service ID, or contract number as the DSN domain name. The Compaq Customer Support Centers uses two domain names, "digital" for DSN customers and "compaq" for WorldWire customers[1]. Authentication keys are identified based on this source domain and destination domain relationship.

A **DSN node name** identifies a system within a DSN domain. This relationship allows multiple nodes running DSNlink and WorldWire to use the same domain name. A DSN node may exist in more than one DSN domain, which allows one DSN node to choose among multiple access numbers. Usually the DSN node name is the IP host name or DECnet node name of the system.

Figure 1 shows two domains, "92937" and "digital" which represent a customer and Compaq. Domain 92937 contains two nodes, "venus" and "mars". Domain digital contains two nodes, "csccxo" and "cscuto."

**Figure 1**
**DSN Domains and Nodes**




A **DSN application** consists of two parts, a client, which requests a service, and a server, which provides the service. We specify a server as a 3-tuple:

*{domain, node, application}*

We can define a **DSN connection** as the communication path between a client and server. A DSN connection can consist of a single network hop using a single network transport (Figure 2) or multiple hops using different network transports for each hop. A system that routes a DSN connection is called a DSN gateway (Figure 3). All DSN systems may act as a gateway. No additional or special software is required.

**Figure 2**
**Direct DSN Connection Between a Customer and Compaq**

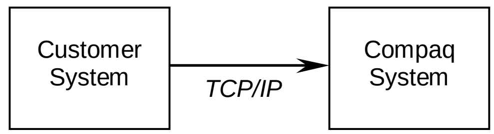
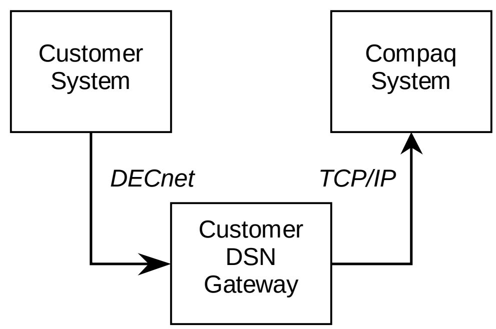



**Figure 3**
**A Multiple Hop DSN Connection Through a DSN Gateway System**




## DSN Architectural Layers

The architecture, a client/server model, defines five layers: DSN Application, DsnSession, DsnGateway, DsnTransport, and the networks[2]. The layers are arranged as shown in Figure 4.

[1] The DSN domain name "digital" refers to the origins of DSNlink at Digital Equipment Corporation.

[2] The names and spelling of DsnSession, DsnGateway, and DsnTransport derive from the class names used in the object-oriented model.

**Figure 4**
**DSN Communications Layers**

<table>
<tr><td colspan="5">DSN Application</td></tr>
<tr><td colspan="5">DsnSession</td></tr>
<tr><td colspan="5">DsnGateway</td></tr>
<tr><td colspan="5">DsnTransport</td></tr>
<tr><td>TCP/IP</td><td>X.25</td><td>DECnet</td><td>Modem</td><td>WWRAS</td></tr>
</table>

Although this architecture resembles the seven-layer ISO OSI Reference Model, there is not a strict adherence. In reality, the top four layers reside in the OSI application layer. Conceptually, though, we can visualize a different mapping. The networking layer most closely maps to the OSI physical, datalink, and network layers; the DsnTransport layer maps to the OSI transport layer; the DsnSession layer maps to the OSI session and presentation layers; the DSN application layer maps to the OSI application layer. The DsnGateway layer appears juxtaposed in that it performs routing like the OSI network layer, but it is positioned between a network-like layer and a session-like layer.

- At the base of the DSN architectural stack are the **networks** used for communications. The operating system provides this functionality in the cases of **TCP/IP**, **DECnet** and **X.25**. The **DSN modem** protocol is a Compaq developed implementation of two protocols, HDLC LAPB and ISO 8073 TP2, which provide error-free communications over PSTN and ISDN connections. The DSN modem protocol is available on OpenVMS and Tru64 UNIX systems. **WorldWire RAS** (WWRAS), TCP/IP using RAS, is available only on Windows NT and Windows 2000 systems.

- The **DsnTransport layer** selects the appropriate network and resolves differences between message-oriented protocols such as X.25 and stream-oriented protocols such as TCP/IP by providing a stream-oriented interface to the DsnGateway layer. This layer also resolves operating system differences so that the upper DsnGateway layer has a uniform interface to a network, regardless of transport or operating system.

- The **DsnGateway layer** allows an unlimited number of systems to communicate using heterogeneous protocols. This layer chains one or more DsnTransport layer connections and makes them appear as a single logical connection. For example, Figure 3 shows a customer using DECnet within his intranet and then TCP/IP to connect to Compaq. The DsnGateway layer makes these two different network connections appear as a single logical DSN connection.

  The DsnGateway layer also makes routing decisions to provide best-cost connection routing. This layer automatically maintains the **DSN route map**, the internal routing database used for next-hop decisions.

- DSN assumes that the DsnTransport and DsnGateway layers provide no guarantee of identity, confidentiality, or data integrity. The **DsnSession layer** addresses these data security issues by providing strong cryptographic authentication and end-to-end data encryption. Secondary functions of this layer include data compression, locale information (time zone and language information), a CRC-32 data integrity check, data flow control, and data checkpointing.

  Authentication is the process of establishing identity without fear of impersonation. The DsnSession layer uses a three-way challenge-response handshake. Each handshake message is signed using a hash-based message authentication code (HMAC) generated with a secret key. As with any secret key-based authentication method, both parties must guard the confidentiality of the keys.

  The DsnSession layer provides data confidentiality with the DES, triple-DES, RC4, and RC5 encryption algorithms.

- The **DSN Application layer** provides services for a specific application. A DSN application consists of a client process and a server process. Although most DSN application clients run on customer systems with the respective server running on a Compaq administrative system, called a DSN host, some application clients do run on the Compaq DSN host with the server running on a customer system.

  When a DSN application server has successfully accepted an incoming connection from the DsnSession layer, it then checks to see if the client is authorized to use the DSN application. Thus, verification of a client's identity does not guarantee access to the application.

## SECURITY IN EACH ARCHITECTURAL LAYER

The only certain method to secure a system is to disconnect it from any network, including telephone modem lines, and lock it in a room to protect it from human contact. However, even though this system may be secure, it is not very useful. Obviously, there is some middle ground between this and total accessibility on the Internet.

DSN is not meant to make a system more secure, but to allow it to communicate with Compaq without increasing the risk of attack. DSN is intended to work within an existing security infrastructure.

In this section we examine each layer in DSN and discuss possible weaknesses.

### DsnTransport and the Underlying Networks

This section describes how hostile the networks we use can be, especially the Internet, and why a DSN connection does not trust the networks. We present the reality of how insecure most networks are. We highly recommend Bruce Schneier's extremely readable book, *Secrets and Lies: Digital Security in a Networked World*, which discusses many network security issues.

DSN assumes that the underlying networks used by the DsnTransport layer offer no security. Most attacks against a system with a network connection include spoofing (impersonation) and denial-of-service (DoS). The network identifies its peer solely on the network address information. For example, if Alice calls Bob on the telephone, Bob can use the caller ID feature to identify that Alice's phone was used to call him. But how does Bob know that he is speaking with Alice? Furthermore, does Bob trust the phone company enough to believe that the caller ID system is working truthfully? Has someone found a way to attack[3] caller ID? We can apply the same suspicions to any networking technology.

Most protocols support some type of access restriction mechanism. This may prevent attacks from outside an organization, but offer no protection from attackers within an organization.

#### *Motivations for Attacking a System*

Attacking a computer system is a crime. It is not an exaggeration to call an attacker a criminal. G. Jack Bologna sums up the why, when, where, and how of computer-related crime in a concept called MOMMs, an acronym for Motivations, Opportunities, Means, and Methods. Knowing what motivates an attacker helps when evaluating risk. Economic, egocentric, ideological, and psychotic motives lead to the commission of a computer crime. Economic motives include the desire to secure financial gain. Egocentric motives include the need to show off the perpetrator's talent. Ideological motives include the feelings to seek revenge against someone or something that they believe is oppressive or exploitive. Psychotic motives include a distorted sense of reality and delusions of grandeur or persecution.[4] Note that although we are discussing these motives in the context of computer crime, these same motives can be seen among burglars, murderers, and terrorists.

Now that we have considered what motivates a network attack, what sort of attacks can we expect to see? The next sections will look at the opportunities, means, and methods. Although we are primarily interested in network attacks, we must keep in mind attacks that take advantage of buggy or poorly configured software as well as mail attachments and Trojan horses.

#### *Denial-of-Service (DoS) Attacks*

In March 2000 an individual orchestrated an attack on several high-profile Internet companies, including Yahoo!, e-Bay, and Amazon.com, by flooding their Web sites with high volumes of network connection requests. This crippling attack, called denial-of-service (DoS), blocked Internet users from accessing these companies' Web sites which directly resulted in revenue loss. This type of attack not only limits incoming access, but can also affect outgoing access. The goal of a DoS attack is not to steal information, but to disable network connectivity that in turn can financially affect the target.

DoS attacks are not limited to the Internet. Imagine an attacker continually dialing a telephone line. This can generate a nuisance big enough to make the owner disconnect it. Inadvertent DoS can also occur from within the organization. Imagine your teenage daughter talking to her boyfriend for hours on the telephone preventing you from receiving an important call from your doctor.[5]

#### *Traffic Analysis*

By just observing activity, an attacker can infer a lot of information. In the hours preceding the U.S. bombing of Iraq in 1991, pizza deliveries to the Pentagon increased one hundredfold. Perhaps news organizations, if they have not yet done so, should monitor pizza traffic to selected buildings in the Washington, D.C. area. When pizza traffic picks up, a press hound could alert his peers regarding a potential scoop. As far as we know, this is legal activity. This is called traffic analysis. An attacker studies communications patterns, not the communications themselves.

---

[3] We prefer to avoid the words "hack" and "hacker". We feel that the term "attacker" is more descriptive of individuals who attack computer systems. Steven Bellovin mentioned the term "vandals" in 1992. In *Secrets and Lies*, Bruce Schneier uses the term "hacker" for anyone who is intellectually curious about how something works. He admiringly refers to Marie Curie as a pioneering hacker. The term has only become derogatory in the past few years.

[4] G. Jack Bologna, "Computer Crime and Computer Criminals," from *Computer Security Handbook*, Third Edition, pp. 6.10-6.11, John Wiley & Sons, New York, 1995.

[5] Some may not consider this an attack, but it sure seems like it at times.

Burglars also do traffic analysis. They "case" a house or business to establish normal activity patterns of the residents as well as the police. Using this information, they can plan their crime accordingly.

DSN connections, as any network connection, are susceptible to traffic analysis. The fact that a connection is being attempted might be useful to an attacker. One countermeasure a customer can take is to perform random innocuous operations such as Netex tests or service request listings. This extra traffic can dilute the effect of traffic analysis.

### *Impersonation*

Just as criminals can steal or forge documents to impersonate someone, opportunities also exist for impersonation in a network. A compromised router can facilitate impersonation by forwarding packets destined for one system to another. For example, suppose that Alice, whose system is in Network A, wants to connect to her bank's Web site, www.bobsbank.com, in Network B. However, the router, under the control of Mallory, has been configured to route all connections destined for www.bobsbank.com to Mallory's system in Network C.

Domain Name System (DNS) spoofing is a form of impersonation. It is equivalent to distributing a telephone directory with wrong numbers in it. And it is not hard to do. On February 14, 2000, someone mounted a DNS spoofing attack on RSA Security Inc.'s Web site www.rsa.com (205.181.76.59). The attacker made it appear that the Web site had been compromised. However, what really happened was DNS poisoning. At least one DNS server was attacked and modified to return the Internet address 200.24.19.252 for www.rsa.com. This was actually a system registered in Colombia whose real name was bachue.udea.edu.co. This bad information rapidly percolated throughout the world shutting down legitimate access to this RSA Security Web server. Nigel Metheringham of VData described the incident thusly: "This is analogous to you following a sign saying 'bank this way' and finding yourself in a dark alleyway in a potential mugging situation." The lesson learned from this incident is that DNS is not secure. What hurts even more is that researchers have been aware of this attack since 1990 and have developed a secure DNS protocol that could have prevented this.[6]

### *Eavesdropping*

Traffic analysis gathers intelligence by watching where information goes without caring about the information itself. For example, we doubt that the number of pepperoni pizzas delivered to the Pentagon had any correlation to the location of air raids during operation Desert Storm. Eavesdropping is the attempt to glean actual specific information. The attacker can later use this information for mischief or personal financial gain. Only a small percentage of information an eavesdropper gathers may be useful.

Without encryption technology, data is susceptible to eavesdropping.[7] Several U.S. Government programs are dedicated to this activity. The FBI's "Carnivore" system runs on an FBI-owned PC that can be installed at an ISP to collect IP traffic as part of a court-ordered wiretap. Not much detail is publicly known about exactly how Carnivore works or what kinds of traffic it collects.[8]

There is also an alleged joint program between the U.S. and the U.K. called "Echelon." Echelon eavesdrops on all communications using artificial intelligence to gather information. The French government claims that Echelon's primary goal is for industrial espionage rather than military intelligence gathering.[9]

Eavesdropping is obviously a privacy issue. Digital: Convergence Corporation's *:Cue Cat* bar code reader has raised enough controversy to elicit this privacy advisory from the Privacy Foundation:

> The :CueCat is promoted as an easy way for consumers to visit Web sites on their PCs by scanning bar codes that have been included in catalogs, magazine articles, and printed advertisements. By using this device consumers no longer have to enter URLs in their browser to go to a Web site to learn more about a product, a service, or a particular subject.
>
> The Privacy Foundation has serious privacy concerns about the product because the :CRQ software, which accompanies the :CueCat device, appears to transmit all of the information that Digital:Convergence would need in order to record every bar code that every user scans. This tracking feature of the :CRQ software could be used by the company to profile an individual user.[10]

---

[6] Steven Bellovin, "Using the Domain Name System for System Break-Ins," *Proceedings of the Fifth Usenix UNIX Security Symposium*, Salt Lake City, UT, June, 1995. http://www.research.att.com/~smb/papers/dnshack.pdf

[7] Academically speaking, encrypted data could be decrypted using a brute-force attack. Here we are assuming at least 128-bit encryption using a good algorithm and key selection. This assumption thus precludes a computationally feasible brute-force attack.

[8] Steven Bellovin, and Matt Blaze, *Open Internet Wiretapping*, http://www.crypto.com/papers/opentap.html

[9] "French report calls for talks on US spy system 'Echelon'," *Agence France Presse*, October 11, 2000.

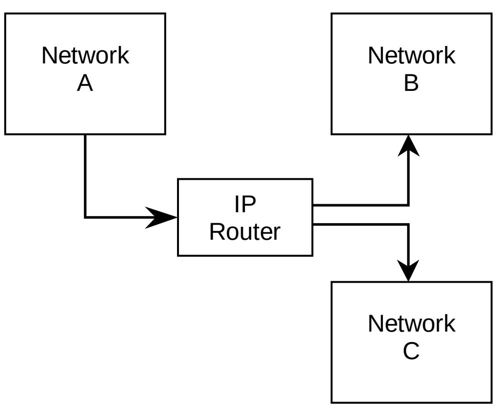
[10] http://privacyfoundation.org/advisories/advCueCat1.html

Devices such as these, although they may make the Internet more user-friendly, can be used for eavesdropping if used unscrupulously. Note that no one, including the Privacy Foundation, is claiming that Digital: Convergence Corporation is eavesdropping. We are using this as an example of how technology meant for convenience has potential for abuse.

*The Router Environment Facilitates Traffic Analysis, Impersonation, and Eavesdropping*

The Internet is really a group of IP networks interconnected with routers. Figure 5 shows three networks, A, B, and C, interconnected through an IP router. Because these routers are the crossroads of communications between networks, their immediate environment can become a target for launching a network attack. Not only must the actual router hardware be secured, but ancillary support systems and the physical facility must also be well secured. The physical communications lines are vulnerable to passive attacks, such as eavesdropping, and active attacks, such as impersonation. The router environment exists at every Internet Service Provider (ISP) and in all but the most trivial intranets. Again, router environments are prime areas for installing software and/or hardware that can view and record network traffic. This can be a side activity of a bored operator at an ISP.

**Figure 5**
**Intermediate IP Routers**




*Protecting Networks*

Any incoming network connection exposes a system to a network attack. Limiting incoming connections only to known entities reduces but does not eliminate a system's vulnerability to an attack. Most TCP/IP, DECnet, and X.25 networking packages allow a system administrator to restrict access to their systems. Firewalls can also prevent attacks by preventing or limiting incoming connection requests to a corporate intranet. Telephone modems with caller ID support and an appropriately configured script can restrict incoming calls.

*Modem and X.25 Protocols for High Security*

DSN offers the modem and X.25 protocols on OpenVMS and Tru64 UNIX platforms as a higher security alternative to the Internet.

It is harder to compromise the phone system. Attackers can hide in the anonymity of the Internet. The Internet also has more access points to exploit. Attacking the telephone system requires physical access. Access is obvious when someone is working on one of those green phone boxes. The telephone companies regularly perform line checks and monitoring. They are under government scrutiny in almost all countries. Connections are dedicated point-to-point virtual circuits. An Internet attacker can launch his invasion from halfway around the globe in a country with no laws governing this type of crime. A telephone line attack is usually done in the same city within a few miles of the target. Telephone hacking tools are not as ubiquitous as Internet hacking tools.

A modem daemon continuously monitors the modem for incoming calls. If an incoming connection attempt is not speaking the DSN modem protocol, the daemon hangs up. And after a successful connect occurs, only DSN applications are executed. The major disadvantage is the 56-kilobaud maximum speed over PSTN lines. (ISDN lines are also supported which can run faster.) On OpenVMS systems, DSN configures the serial port so that a login prompt will not be presented in the event that the modem daemon should terminate. The DSN configuration procedure for Tru64 UNIX systems provides similar security by only allowing non-login terminal ports to be selected for DSN usage. However, note that it is possible for a privileged user to manually change these settings on either operating system, thus exposing a system to greater risk. As with all security, these things should be continually monitored.

X.25 networks are not hotbeds of computer crime because they are not as ubiquitous and access is more expensive. X.25 software packages offer good access control. The default configurations provide fairly conservative access. Like telephone lines, X.25 establishes point-to-point circuits. Impersonation is difficult.

*Application Attacks*

Attacks on TCP/IP and UDP/IP applications are too numerous to mention all of them here. The UNIX sendmail program has been under constant attack. Sun Microsystems Network File System (NFS) uses a very naïve trust model. In a 1992 paper, Steven

Bellovin describes attacks on systems at AT&T.[11] Since then, not much has changed except that there are now more tools and more attackers. These vulnerabilities are why most people use firewalls.

*Web Browsers*

Web browsers are tremendous security leaks. It is no wonder why. They are incredibly powerful and complex general-purpose tools that open up unimaginable security holes. We like to call them the "black holes of common sense." It seems that commercial Web site operators are more interested in figuring out how to promote advertising than providing a secure environment. One must be very cautious and careful when using a Web browser for secure communications.

Secure Web sites use Secure Socket Layer (SSL) connections to provide authenticated and encrypted connections using a Certification Authority (CA). Because users rarely examine a secure Web site's certificate, a dishonest ISP can easily perform a man-in-the-middle attack on a secure Web connection by substituting the Web site's certificate. We recommend that you always examine the encryption level and vendor's certificate when connecting to secure Web sites. Unfortunately, on current Web browsers this is a manual operation that we feel is not obvious to the user. Although some might consider it an obnoxious feature, we wish that the browser vendors would automatically display a pop-up window containing the certificate and encryption information when initially connecting to a secure Web site. This is another example of ease-of-use at odds with security.

*Conclusion*

We have discussed the hostile environment of network transports to explain why DSN does not trust them. Most operating system networking software at this layer has facilities to ward off attacks. Prudent and responsible use of firewalls is a mandatory step in protecting a corporate network. However, none of these mechanisms can fix everything. That is why DSN offers the customer a selection of transports.

## DsnGateway

*Vulnerabilities*

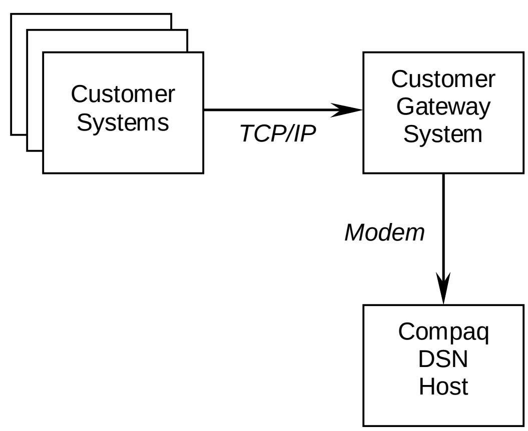
The DsnGateway layer learns routes. A denial-of-service attack is possible if a DsnGateway imposter (as opposed to a network imposter) sends a LEARN message after successfully establishing a DsnGateway connection. When a client attempts to connect back to the server, it would try the fake path learned in the route map. The DsnSession layer will catch this and not allow a connection to be made. However, enough information is logged to locate an imposter. Use of network access restriction mechanisms or a firewall greatly reduces this type of attack.

*DSN Gateway as a Concentrator*

The DsnGateway layer is devoted to routing and provides no additional security enhancements above what the underlying transports provide. Nonetheless, this layer facilitates a configuration that works well with firewalls and limits a customer's exposure to attacks.

DSN's routing ability allows multiple customer systems to funnel through a single DSN system, called a DSN gateway, providing a single point of connectivity with a Compaq Customer Support Center. If a customer is using the Internet, this single gateway system simplifies firewall configuration. If a customer is using the DSN modem protocol, only a single telephone line is necessary to support multiple systems. When using the X.25 or modem protocols, the customer's network is hidden from Compaq, thus retaining its autonomy. Network autonomy can also be maintained on the Internet when using an appropriately configured firewall and network address translation (NAT) using non-routable addresses such as the network 10 class A IP address block.

**Figure 6**
**Example DSN Gateway Configuration: Customer Connecting to Compaq Using the DSN Modem Protocol**

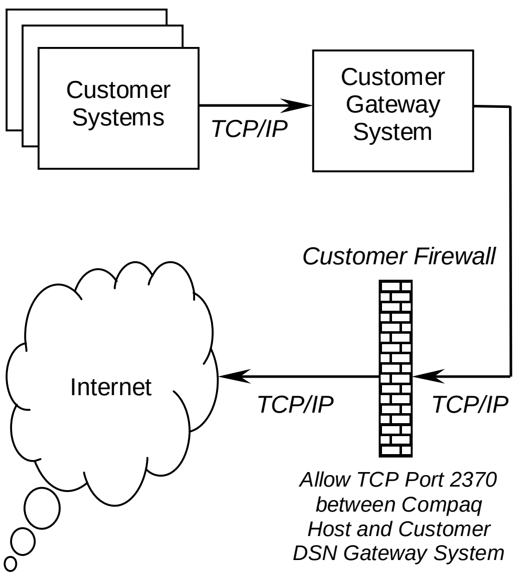



For example, consider a customer who is wary of using the Internet for sending sensitive support information. This customer has several systems that need to communicate with Compaq using a single modem line. Figure 6 shows this configuration. All systems would funnel through the customer's DSN gateway system that would then communicate with Compaq using the DSN modem protocol.

[11] Steven Bellovin, "There Be Dragons," *Proceedings of the Third Usenix UNIX Security Symposium*, pp. 1-16, 1992. http://www.research.att.com/~smb/papers/dragon.pdf.

Customers who wish to use the Internet might configure DSN as shown in Figure 7. Rather than making the firewall pass all connections between the customer DSN systems and Compaq, the firewall administrator need only allow connections between the customer gateway system and Compaq's gateway system using TCP/IP port 2370, thus simplifying firewall management.

**Figure 7**
**Example DSN Gateway Configuration: Customer Connecting to Compaq Using TCP/IP over the Internet with a Firewall**




## DsnSession

We have told you that you cannot trust any network and have described the risks of using a Web browser. In this section we will explain the features of DSN which enable customers to securely communicate with Compaq.

Two problems we discussed in the network layer, impersonation and eavesdropping, are solved in the DsnSession layer.

Authentication prevents impersonation; data encryption prevents eavesdropping. The DsnSession layer provides an end-to-end authenticated and encrypted communications link between a client and a server. Even though DsnSession data may pass through DSN gateway systems, because the data is encrypted end-to-end, an attacker on the DSN gateway systems cannot decipher the data. The security at this layer depends on both robust protocol design as well as strong cryptography. The DsnSession protocol is similar to the SKID3 protocol developed for RACE's RIPE project.[12]

### *Cryptographically Strong Authentication*

Most of us are familiar with using a user name and password to access a computer system. This is a form of authentication. An account number-PIN combination used by ATMs is another variant. However, this is only one-way authentication. How does the user know that he or she has connected to the correct system and not an imposter? Web browsers also have the same problem. How does a Web shopper determine that they are connected to Amazon.com's secure Web site and not an impersonator?

The DsnSession protocol's challenge-response handshake provides mutual authentication between Compaq and the customer. Using the default method, no passwords, either encrypted or unencrypted, are sent in the messages. Instead, a 160-bit signature is sent with each message. The signature is generated using a cryptographic hash of the authentication key and the message's contents. DSN uses a hash-based message authentication code (HMAC) algorithm as described in RFC 2104.[13] The default HMAC is constructed using two cryptographic hash functions, SHA-1 and RIPEMD-160.[14] We call this method SR160. An attacker would have to find weaknesses in both functions to successfully break the HMAC.

DSN provides several cryptographically strong authentication methods: SR160, SHA1, RMD160, MD5_V3, and MD5. SR160, SHA1, RMD160, and MD5_V3 are RFC 2104-type HMAC algorithms. As previously stated, SR160 uses both the SHA-1 and RIPEMD-160 message digest functions in the HMAC algorithm. SHA1 and RMD160 use only one message digest function, SHA-1 and RIPEMD-160, respectively. MD5_V3 generates only a 128-bit signature using the MD5 message digest function with an HMAC algorithm. MD5 is present only for DSNlink V2 compatibility. It also generates a 128-bit signature, but does not use an HMAC function. Instead, a simpler, less secure message authentication code (MAC) generates the signature. DSNlink V3 and

---

[12] Research and Development in Advanced Communication Technologies in Europe, *RIPE Integrity Primitives: Final Report of RACE Integrity Primitives Evaluation (R1040)*, RACE, June 1992.

[13] RFC 2104, "HMAC: Keyed-Hashing for Message Authentication," Internet Request for Comments 2104, Hugo Krawczyk, Mihir Bellare, and Ran Canetti, February 1997.

[14] Cryptographic hash functions are also called message digest or secure hash functions. These functions are one-way. Given a specific hash value, it is computationally unfeasible to find a message that will hash to that value. Other well-known cryptographic hash functions are MD2, MD4, and MD5.

WorldWire always attempt to use SR160 authentication.

*Non-Cryptographic Authentication Methods*

DSN currently provides two non-cryptographic authentication methods, NOAUTH and PASSWORD. NOAUTH performs no authentication. PASSWORD is a plaintext exchange of passwords. These methods exist only for countries that prohibit strong cryptographic authentication. From a security aspect they are basically worthless.

*DSNlink V2 vs. DSNlink V3 Authentication*

The DSNlink V2 authentication handshake, designed in 1993, used a MAC consisting of a secret key prefixed to the message and then hashed with the MD5 message digest function.[15] However, this method is susceptible to message extension attacks whereby an attacker can append information to the end of the message and recalculate the signature. Further research led to the HMAC described in RFC 2104. Cryptanalysis of MD5 has since revealed enough concern that RSA Laboratories now recommends that applications replace it with the SHA-1 and RIPEMD-160 functions.[16] These issues led to the changes in the authentication protocol handshake used in DSNlink V3 and WorldWire. DSNlink V3 retains the V2 handshake for backward compatibility so that DSNlink V2, DSNlink V3, and WorldWire systems can interoperate.

*Authentication Key Files*

All methods except NOAUTH and PASSWORD use the same authentication key stored in a file named

*HMAC-DIGITAL-<domain>*

or

*HMAC-COMPAQ-<domain>*

NOAUTH and PASSWORD authentication use similar files whose names begin with NULL and PASSWORD, respectively. Regardless of the type of authentication selected, if the necessary key file does not exist, authentication fails. Thus as long as no NOAUTH or PASSWORD key files exist on the customer systems, customers need not worry that a Compaq impersonator will make a connection to their systems using these weak methods.

**It is extremely important that the authentication keys remain secret.** DSN stores these keys in files that have restricted access. The software installation procedures on OpenVMS and Compaq Tru64 UNIX systems create these files in directories that may only be read by privileged software. An attacker would have to gain privileged access to the system to read the contents. Normal user access is not allowed.

All DSN images on OpenVMS are installed with SYSPRV as well as some other privileges. The key files are not group- or world-readable and are owned by a DSN user that is a captive, non-interactive account. All privileges are disabled until the files need to be read.

DSN programs on Tru64 UNIX are "setgid." The key files are owned by root in a unique DSN group with group-read access and no world-read access (mode 640). The effective group ID is set to the DSN group only when accessing the key file.

Currently, DSN performs no file permission settings on Windows NT and Windows 2000 systems. Only NTFS supports ACLs, FAT does not. Unfortunately, our testing on Windows NT reveals that administrative shares, such as "C$", are automatically added at system boot time. These shares allow unrestricted access to domain administrators logged in to other systems. We have found that after every reboot, you will probably have the files exposed to anyone who can map your C drive. (Beware of C$ sprouting up after a reboot.)

*Data Encryption*

The DsnSession layer encrypts with well-known industry-standard algorithms. By default, DSN uses the triple-DES encryption algorithm with a 168-bit key.[17] The RC4 and RC5 algorithms using 128-bit keys are also available, as well as DES, using a 56-bit key.

Using a combination of Diffie-Hellman key agreement and a signed HMAC algorithm, a DSN connection negotiates two encryption keys, one for data sent from the client to the server, and the other for data sent from the server to the client. Each encryption key is used for a limited time. After an application client or server has sent one million bytes, it requests a key change from its peer and performs another Diffie-Hellman key agreement. This threshold is configurable.

*DES and Triple-DES*

The National Bureau of Standards, now the National Institute of Standard and Technology (NIST), standardized the **Data Encryption Standard (DES)** in 1977. It is a block cipher that encrypts data in 64-bit blocks. Its 56-bit key drew almost immediate

---

[15] Gene Tsudik, "Message Authentication with One-Way Hash Functions," *Computer Communications Review*, 22 (1992), pp. 29-38.

[16] M.J.B. Robshaw, "On Recent Results for MD2, MD4 and MD5," *RSA Laboratories Bulletin*, 4 (November 12, 1996).

[17] Three different 56-bit keys are used in three DES passes. The first pass encrypts the data with the first key, the second pass decrypts the data with the second key, and the third key encrypts the data with the third key. This Encrypt-Decrypt-Encrypt process is abbreviated EDE.

criticism because of its small size. In 1977 Whitfield Diffie and Martin Hellman argued that a special-purpose machine at a cost of $20 million could try all $2^{56}$ or 72,057,594,037,927,936 keys in one day.[18] By 1993, Michael Wiener had estimated that the cost had reduced to $1 million and the time to 3.5 hours.[19] However, both of these analyses were academic postulations. But in 1998 the Electronic Frontier Foundation (EFF) built a machine for $210,000 with an expected search time of 112 hours, thus validating the academic studies.[20]

Setting aside the key size issue, DES has successfully withstood over 20 years of intense cryptanalysis. The only major attack found so far is a chosen plaintext attack using differential cryptanalysis.[21,22] This attack effectively reduces the key to 48 bits. However, the design of the DsnSession protocol precludes chosen plaintext attacks, thus preventing differential cryptanalysis.

**Triple-DES** addresses the key size issue by encrypting the data three times with three different keys. With a 168-bit key, triple-DES has become a widely accepted high security algorithm. DSN uses both algorithms in cipher block chaining (CBC) mode.

The major disadvantage of triple-DES is that software implementations are not very fast because DES was designed for hardware implementation. With 800 MHz PCs now widely available, this lack of DES performance is not as important. However, we have many customers using older and slower systems, such as the VAXstation 3100. Our tests indicate that this older hardware can still perform triple-DES encryption at "not unreasonable rates."

*RC4 and RC5*

The **RC5** block cipher, a patented algorithm developed by Ronald Rivest (the 'R' of RSA) and licensed from RSA Security, Inc., addresses the DES key size and performance issues[23]. When comparing portable C versions on a MicroVAX 3100, RC5 software implementations are around 1½ times faster than DES and 4½ times faster than triple-DES. Although RC5 keys can be up to 2050 bits, DSN uses a standard 128-bit key. DSN offers RC5 as an alternative for situations where CPU performance may be an issue.

A small disadvantage of block ciphers is that when messages are sent, the message must be padded to be a multiple of the cipher's block size. For example, RC5 and DES both use 64-bit (8-byte) blocks. Messages that are not multiples of 64 bits in length must be padded with up to seven bytes. This can become an issue with slower communications links on certain protocols. The DSNlink remote login protocol can send many 2-byte packets. A 64-bit block cipher would require padding these out to 8 bytes, thus noticeably impacting performance. A stream cipher addresses this issue because it can encrypt one byte at a time; no padding is necessary. For the remote login and similar applications, DSN offers the option of the **RC4** stream cipher, also developed by Ronald Rivest and licensed from RSA Security, Inc.[24] RC4 is about one-third faster than RC5 or six times faster than triple-DES. DSN uses a 128-bit key with RC4.

RC4 has had more cryptanalysis than RC5, primarily because it is widely used in Web browsers for SSL connections (used by HTTPS). Both have only received a fraction of the analysis performed on DES. DES has been around almost twenty years longer, but RC4 and RC5 are both proprietary algorithms. RC5 has been patented and requires royalties. RC4's legal status is somewhat fuzzy. Many cryptographers are reluctant to spend large amounts of time reviewing proprietary algorithms because of the legal issues involved.

*AES and Rijndael*

On January 2, 1997, NIST announced the initiation of the **Advanced Encryption Standard (AES)** development effort and made a formal call for algorithms in September 1997. The call stipulated that the AES would specify an unclassified, publicly disclosed encryption algorithm, available royalty-free worldwide. In addition, the algorithm must implement symmetric key cryptography as a block cipher, support block sizes of 128-bits and key sizes of 128-, 192-, and 256-bits. The overall goal was to develop a Federal Information Processing Standard (FIPS) that specifies an encryption algorithm capable of protecting sensitive government information well into the next century. NIST expects the algorithm to be used by the U.S. Government and, on a voluntary basis, by the private sector.

The evaluation criteria for the AES included:

---

[18] Whitfield Diffie and Martin E. Hellman, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard," *Computer*, 10 (1977), 74-84.

[19] Michael J. Wiener, "Efficient DES Key Search," Technical Report TR-244, School of Computer Science, Carlton University, Ottawa, 1994.

[20] Paul Kocher, "Breaking DES," *CryptoBytes*, Volume 4, Number 2, (Winter 1999).

[21] Eli Biham and Adi Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, New York, 1993.

[22] Eli Biham and Avi Shamir, "Differential Cryptanalysis of the Full 16-Round DES," *Advances in Cryptology – CRYPTO '92 (LNCS 740)*, Springer-Verlag, 1993.

[23] Ronald L. Rivest, "The RC5 Encryption Algorithm," *CryptoBytes*, Volume 1, Number 1, (Spring 1995).

[24] No publicly available formal specification exists for RC4. However, alleged descriptions have been published which are output-compatible with certified implementations of RC4.

- Licensing requirements: the algorithm should be available on a worldwide, non-exclusive, royalty-free basis.
- Computational efficiency: the algorithm should be fast in both hardware and software.
- Memory requirements: not only should the implementation be compact in software (RAM and ROM usage), but in hardware also (gate counts).
- Flexibility requirements: the algorithm should be capable of being implemented securely and efficiently in a wide variety of platforms and applications (for example, 8-bit processors, smart cards, ATM networks, voice and satellite communications, HDTV, B-ISDN, etc.).
- Should be easily implementable in both hardware and software.

Thus, the technical goals of AES over DES are not just more security, but more flexibility in implementation.

In August 1998, NIST announced a group of fifteen candidate algorithms. This group was reduced to five finalists in April 1999. Then, on October 2, 2000, NIST announced that it had selected **Rijndael**, developed by Joan Daemen and Vincent Rijmen, as the AES. The NIST report states, "When considered together, Rijndael's combination of security, performance, efficiency, implementability, and flexibility make it an appropriate selection for the AES."[25] It is roughly 7½ times faster than DES or 2½ times faster than triple-DES on a 200 MHz Pentium Pro computer. Pending a successful review period, the AES process is expected to be complete by the summer 2001. At this time, AES will become an official standard.

Due to the timing of the AES announcements and because the AES finalist has not been completely standardized, DSN does not yet offer AES (Rijndael) as an option. However, we are very confident in the security of triple-DES as our primary encryption algorithm. Once AES is finalized, we expect to add it to DSN's suite of ciphers.

*Key Size*

A good cryptographic algorithm requires an adequately sized key. In 1999 Arjen Lenstra and Eric Verheul analyzed various encryption algorithms and their effectiveness when considering the increasing power of computers. Increasing the key length is an exponential function. Adding one bit to a key doubles the computation time. By default DSN uses the triple-DES algorithm with a 168-bit key. Lenstra and Verheul conservatively estimate that a 109-bit key should be sufficient through the year 2050. 128 bits should suffice until 2074. However, they state:

> Experience has taught us, however, that failures in cryptography almost invariably originate in some design error within the system as a whole, rather than in a wrong choice of cryptosystem or key size. In other words, it is better to concentrate on the quality of the overall design than to be fixated on the technology or key sizes used.[26]

*Cipher and Signature Suites*

Both the client and server have their own **cipher suite**, an ordered list of supported encryption algorithms. By default, this list is the following:

*{Triple-DES, RC5, RC4, DES}*[27]

An additional cipher, *NOENCRYPT*, is also available that specifies no encryption. These lists are used by the protocol to negotiate the encryption algorithm. The client sends its cipher suite to the server in the first message of the initial DsnSession handshake. The server searches its own cipher suite for the first matching cipher it sees in the client's cipher suite. For example, if the client's cipher suite is *{Triple-DES, RC4, RC5}* and the server's cipher suite is *{NOENCRYPT, RC5, RC4}*, the server will negotiate RC4 encryption. Essentially, the client determines the priority of cipher selection. If the server can not find a match, the connection is immediately terminated. This mechanism allows a customer to enforce a rule that requires all DSN connections to be encrypted, and the customer can choose which encryption algorithm is to be used.

A similar algorithm is used for negotiating authentication algorithms. Both a client and server have their respective **signature suites**. The default signature suite is as follows:

*{SR160, SHA1, RMD160, MD5_V3, MD5, NOAUTH, PASSWORD}*

As with the cipher suite, a customer can enforce rules to require a specific authentication algorithm. Note that by default, key files for NOAUTH and PASSWORD authentication do not exist, effectively disabling these methods. Customers should ensure

---

[25] James Nechvatal, Elaine Barker, Lawrence Bassham, William Burr, Morris Dworkin, James Foti, and Edward Roback, "Report on the Development of the Advanced Encryption Standard (AES)," NIST, October 2, 2000.

[26] Arjen Lenstra and Eric Verheul, "Selecting Cryptographic Key Sizes in Commercial Applications," *Pricewaterhouse-Coopers Cryptographic Centre of Excellence (CCE) Quarterly Journal*, Autumn 1999. http://www.cryptosavvy.com.

[27] Triple-DES uses a 168-bit key, RC5 and RC4 use a 128-bit key, and DES uses a 56-bit key. These algorithms are called out in the DSN configuration as TDES, RC5_128, RC4_128, and DES, respectively.

that files whose names begin with "NULL-" and "PASSWORD-" do not mysteriously appear in the authentication keys directories.

*Random Number Generation*

A good cryptosystem requires a good random number generator, which in turn requires a good initial value called the seed. In 1995 two University of California at Berkeley graduate students, Ian Goldberg and David Wagner, discovered a flaw in a Web browser sold by a famous company that is now owned by another famous company. This company's browser used the process ID, parent process ID, and the system time as the seed for their random number generator. The random number generator then was used to create the encryption keys for SSL (HTTPS) connections. Unfortunately, the seed hardly had 47 bits of entropy. That is, an attacker would need to try on average $2^{46}$ different numbers to guess the encryption key. Thus, the browser's 128-bit encryption was reduced to at most 47 bits because of a bad seed. Goldberg and Wagner also found that with a little cleverness the entropy could be reduced down to around 35 bits. Furthermore, if the attacker had access to the machine, he would only have to try on average 500,000 different numbers to calculate the encryption key, further simplifying the attack. The embarrassed famous company released a patch for this problem shortly after the attack was publicized in every major newspaper in the United States.[28]

DSN uses RSA Security, Inc.'s X9.62-compliant random number generator with the SHA-1 function. It generates the DsnSession handshake challenges, Diffie-Hellman key negotiation parameters, and encryption initialization vectors (IVs). The seed is derived from network, device, and process counters. Our analysis suggests that we reap at least 128 bits of entropy on OpenVMS and Tru64 UNIX systems. An attacker would have a one in $2^{128}$ chance in guessing the seed. Our Windows analysis is incomplete at this time.

*Public Key vs. Secret Key Authentication*

Public key cryptography addresses the key distribution problem. Using secret keys, if *m* entities wish to communicate with *n* entities, *mn* keys are required. For example, if 1,000,000 Web users each wanted to communicate with 1,000 on-line shopping stores, one would have to manage 1,000,000,000 keys and keep them secret. In public key cryptography, each entity is issued a public key that everyone can know, and a private key that the owner keeps secret. In our example, then, we would have $2(m+n)$ keys, or 2,002,000 keys, of which half would need to be kept secret. This is significantly fewer keys than the secret key scheme. Standards are being developed that define a system, called a public key infrastructure (PKI), that issues and manages these keys. Many vendors feel that PKIs are an essential and integral part of network security. Some reputable security experts, however, have reservations about PKIs.[29,30]

Public key cryptography also has the property of non-refutability. With secret keys, the two entities can impersonate one another to each other because they share the same secret. For example, a bank could impersonate one of their customers at an ATM to withdraw cash from that customer's account. In public key cryptography, two entities do not share any secrets and thus one can not impersonate the other. Referring to the previous example, the bank would be unable to impersonate any of their customers unless they stole a customer's private key.

However, we feel that public key cryptography has one serious disadvantage in the DSN environment. If DSN were to use public key cryptography for authentication and an attacker were to steal Compaq's private key, each customer's system would become vulnerable. This issue is important to us because Compaq has roughly 20 CSCs throughout the world. We feel that allowing that many copies would expose Compaq's DSN private key to great risk. Because DSN uses secret keys and only one copy of the customer's key exists at Compaq, theft of a single secret key affects only one customer. Customers who feel that their keys have been stolen can easily request new ones.

*Data Integrity Features*

To help with data integrity each message contains a **CRC-32** checksum. We do not use this for message authentication. Although networks perform similar data checking, we have found that secondary checksums are valuable for detecting not only hardware errors, but software bugs as well. Many PCs do not have error-correcting or error-detecting memory and it is not uncommon for gamma radiation to induce memory errors. (Why do you think you need to reboot your home PC so often?) The CRC-32 also boosts confidence that the DSN software and the operating system are working correctly.

Additional data integrity features include **data flow control**, which can be used by an application to limit the amount of data "in the air." **Data checkpointing** allows an application to send data with the automatic

---

[28] John Markoff, "Security flaw is discovered in software used in shopping," *The New York Times*, September 19, 1995, pp. A1, D21.

[29] Don Davis, "Compliance Defects in Public-Key Cryptography," *Proceedings of the 6th USENIX Security Symposium*, Usenix Association, 1996, pp. 171-178. http://world.std.com/~dtd/compliance/compliance.pdf

[30] Carl Ellison and Bruce Schneier, "Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure," *Computer Security Journal*, Volume 16, Number 1, 2000, pp. 1-7. http://www.counterpane.com/pki-risks.html

acknowledgement that its DSN application peer received it. This feature ensures that the DSN application server and client are synchronized.

*Generation and Distribution of Authentication Keys*

Customer authentication keys are generated at Compaq using a high-entropy random number generator. Each key contains approximately 81 bits of randomness. Keys are not electronically transmitted (e-mailed) to the customer. Compaq's policy regarding passwords and secret keys requires voice transmission if the CSC contacted the customer or by traditional mail or fax.

*Known Weaknesses*

DSN currently does not perform message-level authentication. Only messages involved in the initial handshake are authenticated. This leaves encrypted connections using block ciphers such as triple-DES or RC5 prone to cut-and-paste attacks.[31,32] An attacker splices a block of previously encrypted data into an intercepted message. The idea is that an active attacker can meaningfully modify the data being passed between two communicating systems.

For example, consider an attacker who is familiar with the protocol of bank transactions. He knows that bytes 64 through 79 of the third message contain the destination account number in an electronic funds transfer (EFT) transaction. First, he needs to gain control of a routing point between a customer and a bank. He waits for the third message in an EFT transaction and alters the destination account field so that the funds are transferred into a different account. However, because he does not know the encryption key, he is unable to specify any account he wishes. Furthermore, the chances are quite slim that he can randomly choose cipher text that decrypts to a meaningful account number much less his own account number. However, suppose that bytes 16 through 31 of the first message contain the customer's encrypted phone number and zip code. The attacker places this information into the account number field of the third messages and voila! The third message decrypts normally with a perhaps erroneous account number. At best, the transaction is never performed because of an invalid account number. At worst, the modified account number does exist and belongs to the attacker.

The previous example may sound far-fetched, but it does illustrate one of the many attacks that protocol designers must consider. Given this scenario, why doesn't DSN perform message-level authentication?

- There is a potential performance hit in network bandwidth, especially on slower modem telephone lines.
- Attacks on data must be active. That is, the attacker needs to control a routing point.
- Our analysis shows that the difficulty of a successful cut-and-paste or data modification attack is on par with a brute-force key attack. The DsnSession message format is very sensitive to errors and contains a CRC-32 data check. The raw message is compressed and then encrypted. Because data compression takes 8-bit byte groups and encodes them as 9-bit to 16-bit tokens, a compressed DsnSession message will be bit-aligned rather than byte-aligned.[33] Furthermore, because DSN uses the block ciphers in CBC mode, at least two blocks (16 bytes or 128 bits) must be spliced. We feel that there is enough entropy in the compression-encryption process that an attacker will look for other weaknesses.
- Protocols that use short messages, like the DSNlink remote login protocol, are more susceptible to this type of attack in that the messages are usually smaller than a block cipher's block size. However, for remote login we use the RC4 stream cipher, which is not susceptible to cut-and-paste attacks.

We are still considering adding message-level to DSN sometime in the future.

## APPLICATIONS

DSN applications are designed to be function-specific and simple. This is in contrast to a Web browser whose power and flexibility make it insecure. The incorporation of Java, JavaScript, and ActiveX control technologies has enabled marvelous functionality but at a severe cost in security. Remember, as complexity increases, security decreases. As more features are added, more opportunities for attack are also added. The first version of Mosaic is probably the last secure Web browser that existed. Today's Web browsers are so complex that one assumes that they will have problems. It seems that within weeks after a new version of Microsoft's Internet Explorer or Netscape's Navigator is released, some security flaw has been discovered and announced.

---

[31] Steven M. Bellovin, "Problem Areas for the IP Security Protocols," *Proceedings of the Sixth USENIX Security Symposium*, Usenix Association, 1996, pp. 205-214. http://www.research.att.com/~smb/papers/badesp.pdf.

[32] D. Wagner and B. Schneier, "Analysis of the SSL 3.0 Protocol," *The Second USENIX Workshop on Electronic Commerce Proceedings*, USENIX Press, November 1996, pp. 29-40. http://www.counterpane.com/ssl.html.

[33] We have found the data compression algorithm extremely sensitive to data corruption. Decompression errors are discovered before the CRC-32 errors. During product development this property has been helpful in detecting software data corruption problems.

On the contrary, a DSN application is designed to be a simple, single-function component. User interfaces are captive and specific to the required functionality. The lack of complexity allows us to analyze the security better. We restrict the environment of DSN application servers so that if they are compromised, damage is limited.

## Access Control

DSN provides access control for both the client and server processes of an application. Customers not only have control of what servers Compaq is allowed to access on their systems, but they can also control what applications their users are allowed to access. The DSN domain, DSN node, user name, and time-of-day determine access. Access is explicitly allowed or denied.

*Local Authorization File*

System managers and administrators can control their users' access to DSN functions. Before establishing a connection to its DSN server, using the system's local DSN domain name, DSN node name, and user name, an application client checks the local authorization file (LAF) for a matching entry. The entry can explicitly allow or deny access.

The time-of-day check in the LAF allows an application finer access control than a file system access control list (ACL) does. The LAF is also operating system-independent in that not all file systems support ACLs. For example, the FAT on Microsoft Windows systems does not support ACLs. This platform-independent solution also provides a single management interface for customers who run multiple operating systems. Because the DSN node name is one of the keys in the LAF lookup, a single LAF can be shared in VMScluster and TruCluster environments.

*Remote Authorization File*

After a DSN server has successfully authenticated its client, it checks its remote authorization file (RAF) for access permission. If access is denied, the server immediately responds with a DsnSession "REJECT" message and terminates the connection. The application client will receive this message and display an appropriate diagnostic.

The RAF is similar to the LAF in that authorization is based upon the DSN application client's domain, node, user name, and time-of-day. The default RAF for customer systems allows only incoming connections from Compaq. Remote login is explicitly disabled as well.

## Server Contexts

All DSN application servers run under the root account on Tru64 UNIX, the administrator account on Windows, and a captive non-interactive account on OpenVMS.

*File Copy*

The DSN File Copy application allows Compaq and a customer to copy files to and from one another. The server is only allowed to read files from an outgoing directory and write files into an incoming directory in the customer's system. Thus the customer can control Compaq's access to all files. Customers move files to and from the outgoing and incoming directories. If the customer's authentication key were stolen, an attacker would only be able to copy files into the incoming directory or copy files from the outgoing directory. No other part of the system is exposed. By adding time specifications in the RAF, the customer can further reduce exposure.

*Mail*

The DSN Mail application sends mail between Compaq and customers using an authenticated connection. The mail application is used mainly by the CSCs to inform customers of service request updates.

The mail server takes the incoming message and relays it to the customer's mail system. On OpenVMS, DSN uses the native MAIL facility. On Tru64 UNIX and Windows, DSN relays the mail to an SMTP mail server. Tru64 UNIX uses the system's own (i.e., localhost) SMTP server. Because most Windows system do not run SMTP servers, the Windows installation process prompts for the IP address of an SMTP server within the customer's organization.

Electronic mail is probably the most widely abused and counterfeited application on the Internet. Although DSN can preserve authenticity between mail application peers there is no guarantee that something bad will not happen from the time it leaves the DSN mail server and arrives in the appropriate mail box.

If an attacker has stolen the authentication key, he can impersonate mail from the CSC. He could create a MIME-formatted message with a hazardous attachment. This attachment could contain a virus, worm, Trojan, or other attack.

*Remote Login*

The DSN Remote Login application allows a specialist at a Compaq CSC to log into an OpenVMS or Tru64 UNIX system similar to the telnet or rlogin TCP/IP applications. A CSC specialist will need a valid user name and password from the customer. The installation procedures create a RAF with an entry explicitly denying any remote login access from any source. Compaq recommends that customers should grant specialists access for a limited time using the RAF time fields.

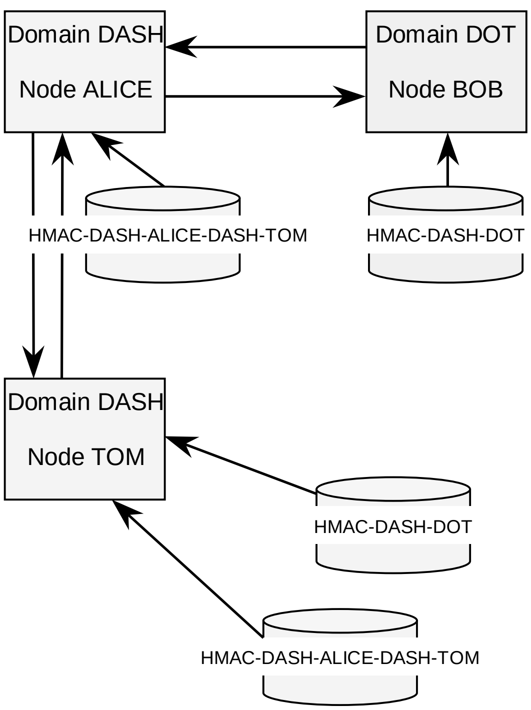
Two log files of all remote login sessions are created, one on the customer's system and one at Compaq. The log files record only what has been displayed, not the keystrokes entered. Passwords and other data entered when terminal echoing has been disabled are not recorded. The customer can view the file during the session using the UNIX "tail -f" or OpenVMS "TYPE/CONTINUOUS" commands.

Note that for an attacker to successfully gain access via the remote login server, he would need the authentication key, a user name and password, and the customer would have had to modify the time fields in the RAF to allow him in.

### K2 Cryptographic Services

The DSN K2 application[34], which is used only by WorldWire and internally within Compaq for DSNlink, facilitates a "low-key" secret key infrastructure. A K2 server grants use of a DSN authentication key to a DSN client or server without revealing the key's contents.

**Figure 8**
**K2 Performing Sign-Check Operations**




[34] K2, which stands for "second-order key resolution" or "low-key secret key infrastructure," derives its name from its recursive use of the DsnSession layer and the Himalayan mountain. Because we live in Colorado, we also like mountains.

In Figure 8 node ALICE in the domain DASH wishes to connect to node BOB in the domain DOT but does not have the authentication key HMAC-DASH-DOT. However, node TOM, also in domain DASH, does have this authentication key. ALICE creates an authenticated and encrypted DSN connection to TOM using the **node-level authentication key** HMAC-DASH-ALICE-DASH-TOM. Alice then passes all handshake messages destined for BOB to TOM to be signed. TOM signs them and then returns them to ALICE who sends them to BOB. ALICE can also check messages sent from BOB by sending them to TOM for verification. TOM returns an appropriate status. TOM does all of this work without ever revealing the DASH-DOT authentication key to ALICE.

Notice that K2 is a service that uses the DsnSession layer for establishing the secure links with a K2 server. Additionally, the DsnSession layer uses K2 for key operations. Thus, if TOM does not have the key, he could ask TED to perform the operation. This recursion can lead to infinite invocations of K2 without resolution. To prevent this, K2 enforces a hop limit of two, which allows only two levels of key trust. Thus, TED would not check any further in this example.

Support specialists at the CSCs use K2 to access customer systems from their workstations without needing personal copies of customer authentication keys. WorldWire uses K2 at customer sites to simplify key management. The primary Compaq authentication key need only be stored in one place, the Qualified Service Access Point (QSAP) system. Because the K2 server is a DSN application, it also uses the RAF. Thus, normal application access controls can be enforced.

## CONCLUSION

Today's networks can not be trusted without utilizing appropriate software. Compaq's DSN software provides security that makes the Internet and other networks useable for secure exchange of data between a customer and Compaq.

## BIBLIOGRAPHY

Here is a list of good resources for understanding security and cryptography.

Ross J. Anderson, "Why Cryptosystems Fail," *Communications of the ACM*, Volume 37, Number 11, November 1994, pp. 32-40. This paper examines several examples of security failures. It is available directly from the ACM at http://www.acm.org/pubs/citations/journals/cacm/1994-37-11/p32-anderson/

Dorothy Denning, *Cryptography and Data Security*, Addison-Wesley, Reading, 1983. Like a PDP-10, this is a classic in its field. Although a lot of information is dated, one can find comfort that the field of cryptography has very firm and well-founded roots.

Arthur E. Hutt, Seymour Bosworth, and Douglas B. Hoyt, *Computer Security Handbook, Third Edition*, John Wiley & Sons, 1995. This book examines all aspects of computer security.

Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996. This is an excellent resource for cryptography. Consider it the cryptographer's version of the *Physician's Desk Reference*. It gives more detail to and treats the subject more rigorously than *Applied Cryptography*. This book will explain the differences between HMACs and MACs. Although the entire text is available online at http://www.cacr.math.uwaterloo.ca/hac/ we highly recommend purchasing a copy.

Joel Scambray, Stuart McClure, George Kurtz, *Hacking Exposed: Network Security Secrets & Solutions, Second Edition*, Osborne/McGraw-Hill, 2000. This tells you how the Internet criminal works by discussing the methodology and tools. Some people may claim it is a "How To" book for hacking, but on the other hand, if you do not know the enemy, you cannot protect yourself from it.

B. Schneier, *Applied Cryptography, Second Edition*, John Wiley & Sons, 1996. This is a good introduction to cryptography and is almost universally recommended for newcomers to the subject.

B. Schneier, *Secrets and Lies: Security in a Digital World*, John Wiley & Sons, 2000. This book confirms all of the fear and paranoia you have heard about using the Internet. And if you weren't paranoid before you read this, you will be afterwards.

## ACKNOWLEDGEMENTS

The authors would like to extend special thanks to Donna Duncan of Writing Out West, Inc. and Sharon Rogenmoser of Compaq Computer Corporation for their editorial help and research.

## LEGAL NOTICES

DSNlink Version 3.0 for OpenVMS, DSNlink Version 3.0 for Tru64 UNIX, and WorldWire V2.8 contain encryption features subject to the U.S. Export Administration Regulations (EAR). The U.S. Department of Commerce classifies DSNlink as "retail encryption" exportable under License Exception ENC in accordance with Section 740.17 of the EAR.

DSNlink Version 3.0 is also available without encryption capabilities. This version is called DSNlink NE Version 3.0 for OpenVMS, DSNlink NE Version 3.0 for Tru64 UNIX, and WorldWire V2.8. The "NE" in the name stands for No Encryption; the NE kit contains no encryption code and therefore does not perform data encryption. However, all other security features described in this document apply.

---