Challenges and New Technologies for Addressing Security in High Performance Distributed Environments[†‡]

Rossen Dimitrov
rossen@cs.msstate.edu

Matthew Gleeson
gleeson@erc.msstate.edu

Department of Computer Science
Mississippi State University
Box 9637, MS 39762

Abstract

This paper discusses security implications of some of the latest technologies in distributed high-performance systems. Rapid developments in gigabit-per-second networks, network host interface architectures, and message-passing communications systems are among these, all posing challenges to traditional distributed security models. Recently, a number of novel approaches for increased communications efficiency have been introduced in various application environments. The security assurance characteristics of these approaches are often disregarded or evaluated without sufficient depth, rendering them suspect in secure applications. The goal of this paper is to point out important new networking technologies, reveal the challenges and obstacles they impose to security, and propose protection and assurance mechanisms that need be taken into consideration before these technologies are widely accepted in practice.

This paper reviews three components of an integrated high-performance distributed environment: System Area Networks (SAN), network host interfaces, and protocols for internetworking multiple SANs. The approaches and techniques leading to high communication efficiency are outlined and their affects on networked system security are investigated in turn. Solutions for increasing the level of trust in the reviewed distributed systems are proposed. These solutions can be implemented at the lowest software layers of the communications systems which achieves a higher degree of security mechanism effectiveness and at the same time introduces a minimal processing overhead. Furthermore, not all data traffic in a given system need be concerned with security issues; therefore, such traffic should accept minimal performance degradation resulting from the increased assurance of secure traffic in the same system.

Keywords: security, distributed systems, System Area Networks, Virtual Interface Architecture, PacketWay, Myrinet.

1 Introduction

Advances in information technology often arrive ahead of the definition and implementation of adequate security measures in computer systems. Similar is the case with the newly emerged networking technologies reviewed in this paper. Although at different stages of maturity they all have strong credentials for wide acceptance in practice. These technologies employ architectures that differ substantially from the traditional data communications models based on Local Area Networks (LAN); therefore, the study of these new architectures for potential security deficiencies and new security paradigms is necessary and timely. Consequently, the goals of this paper are first to point out the unique features

[†] Supported by DARPA, order D985 from USAF Laboratory F30602-96-1-0329 and DARPA order D350 USAF Laboratory F30602-95-1-0036. Additional support is acknowledged from the National Science Foundation CISE ASC Grant ASC-9422381 and Career, Grant ASC-9501917.

[‡] Jointly sponsored by Dr. Rayford Vaughn and Dr. Anthony Skjellum – faculty members at Mississippi State University, Department of Computer Science.

of a high-performance distributed system, reveal the vulnerabilities these features introduce into system's security, and finally propose measures for increasing the degree of protection and assurance. The rest of the paper reviews typical representatives of SANs, network host interfaces, and SAN internetworking protocols and suggests an integrated system architecture with effective security mechanisms.

Significant research has been conducted in defining security models and enforcement mechanisms for building secure distributed systems. Security recommendations and assurance criteria like the *Trusted Network Interpretation* [13], the *Trusted Computer Systems Evaluation Criteria* [18], and the *Guideline for the Analysis of Local Area Network Security* [14] have been proposed and accepted for specifying the goals and evaluating the efforts in securing data communication networks. So far, the main focus of the research, standardization and practice in distributed environments has been on systems based on LAN [1, 16] and secure protocols based on the *ISO seven-layer OSI model* [10]. The *Reference Monitor* [2] concept is widely used for implementing security policies in standalone systems and most distributed security models extend this concept to building a *Network Trusted Computing Base (NTCB)* by providing secure communications between networked computers guarded by discretionary and mandatory access controls [16].

The computational power of modern workstations has increased tremendously in recent years. New networking technologies such as ATM [17], Myrinet [6], and Fibre Channel [17] achieve high data rates and low hardware latency at extremely low bit error rates (BER). These technologies can be used in clusters of workstations capable of performing intensive parallel computations, teleconferencing, and time-critical tasks in a cost-effective manner. However, advances in CPU productivity and network performance cannot be efficiently exploited by the traditional LAN models and the OSI protocol stack, both now imposing intolerably high communication and system overheads. These traditional communication models generate an excessive number of context switches to the kernel and make intermediate data copies. Both of these factors significantly degrade the overall system performance. In contrast, the new high-performance SANs facilitate efficient host interface architectures and novel software solutions for low overhead and high bandwidth data transmission. Such solutions are proposed in a number of research efforts among which are U-Net [3], BDM [9], and Fast Messages [15] as well as the recent industry specification of the Virtual Interface (VI) Architecture [7]. PacketWay [8] is an IETF experimental standardization effort for an efficient SAN internetworking. A major goal of PacketWay is to avoid the unnecessary overhead of conventional protocol stacks as TCP/IP and exploit native high-performance characteristics of each SAN.

2 System Area Networks

SANs are usually built by close proximity point-to-point links interconnecting network switches and end nodes. SANs support gigabit-per-second or higher data rates while introducing hardware latency in the order of only a microsecond. The signal-to-noise ratio of these networks is extremely high and a bit error rate of 10^{-15} or lower is common. All these advanced features come at a relatively low price that makes SANs an attractive and cost-effective alternative for intensive concurrent high-performance computing.

2.1 System Area Network Applications

A major difficulty in implementing high assurance access control mechanisms in a SAN is the nature of the applications using the network. Some of the fundamental computer information security models such as the *Bell-LaPadula* confidentiality model [4] and *Biba* integrity model [5] rely on mediating the access of subjects to information objects labeled according to a specific classification level. These models have been extended and successfully applied to LAN-based distributed systems that are used for applications like Network File Systems (NFS) [16] and database management systems [11]. In such systems, information objects (e.g., data records, files, or directories) exist explicitly and the NTCB

can employ appropriate labeling mechanisms for implementing multi-level security policies.

As opposed to the LAN-based distributed systems, the typical SAN applications are parallel scientific simulations, exchange of multimedia streams, and transmission of real-time data flows. In these domains, data transferred between network nodes is often generated at run time and do not exist beyond the life span of the applications. The information objects are normally data buffers that reside in application's memory and are dynamically updated and exchanged between processes participating in a distributed parallel task. Consecutively, a systematic labeling of information objects in a SAN environment is virtually impossible. Therefore, the control, monitoring, and guarding functions of the NTCB cannot be implemented in the same manner and with the same level of assurance as in the traditional distributed systems.

2.2 Myrinet

Myrinet [6] is a typical representative of SAN. It has been initiated as a research project at CalTech and the University of Southern California. This network is now a commercially available product. Presently, there are numerous Myrinet installations in National laboratories and academic research groups. Building components of Myrinet are cut-through network switches, network host interface adapters, and point-to-point full-duplex links [6]. The host adapters are implemented as microprocessor systems capable of executing custom Myrinet control programs (MCP) [6, 12]. These programs can be created, loaded, and controlled by application processes running in user mode. This programmability offers a unique flexibility for implementing different high-performance software architectures.

The MCP executed on the adapters can perform a large part of the initial data processing on incoming and outgoing data packets. In addition to that, Myrinet adapters are equipped with Direct Memory Access (DMA) engines for accessing host system's memory without the participation of the host CPU [12]. The hardware architecture with a system CPU executing user programs and an intelligent network adapter performing communication tasks is referred to as a two-level multi-computer [6]. This architecture frees the host processor from immediate responsibilities for data transmission tasks and achieves overlapping of computation and communication. This leads to a better resource utilization and significant improvement of the overall performance.

2.3 Myrinet Frame Format

In order to exploit the available high-bandwidth efficiently, Myrinet use a simple frame format. Only the most critical system information is included in the frames. This reduces the overhead information transmitted over the network and improves the effective bandwidth available to user processes. Myrinet frames consist of a source route, a frame type field, a payload, and a trailer (Fig. 1). The cut-through routing function is based on source routes that are dynamically interpreted and consumed by intermediate network switches [6]. The network addresses of the source and destination nodes are not included in the frame headers.

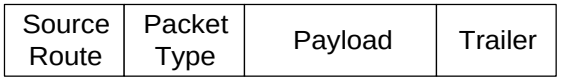

Fig. 1. Myrinet frame format

The absence of source identifier in the frame format leaves peer-node identification and authentication to higher software layers. For instance, in the seven-layer OSI network model employed by traditional LAN and WAN systems, data encryption as well as secure identification and authentication are specified at the Presentation layer, right below the highest Application layer [17]. However, in SAN-based high-performance systems, such higher intermediate layers may implement only limited functionality for synchronization and notification or may not even exist. The architectures used in these systems improve the communication efficiency through collapsing software layers and minimizing the processing at the intermediate software layers. In order to achieve maximum performance, a large number of the existing Myrinet clusters follow this strategy and use only a minimal software protocol stack.

A common technique for improving the protection of networks and host systems against

undesired traffic and availability attacks is packet filtering at intermediate or end-point nodes based on the source network addresses. This technique fails in Myrinet. First, the network switches pass all packets to the next switch or node only according to the leading word in the source route. This word has meaning only to the current network switch and after its interpretation is stripped out of the packet. Second, the host interface adapters cannot differentiate incoming packets based on the source address. Consequently, the filtering protection mechanism cannot be implemented at the data link-layer. As it was shown earlier, the higher software layers in Myrinet systems are optimized for maximum efficiency and in most cases these systems do not implement filtering at higher layers as well. In addition to the availability threats, the absence of network addresses also introduces a data integrity vulnerability of undetected packet source masquerading. So, as a result of the absence of source identifiers and collapsing software layers, distributed systems based on Myrinet are unable to provide a high assurance peer identification and authentication as well as mechanisms for protection against availability and data integrity attacks.

Following, several solutions for securing the traffic in Myrinet networks are proposed. First, a new frame type (Fig. 2) that specifies a security

Source Route	Secure Type	Security Attributes	Payload	Trailer

Fig. 2. A Myrinet frame with security attributes

attributes field in the frame can be used to overcome the absence of source and destination identifiers. This security attributes field will include identification and authentication information as well as classification labels for implementing distributed multi-level security systems. On a network concerned with security, the traffic not providing appropriate security attributes may be discarded which will improve protection against availability attacks and assurance of the identification and authentication procedures. The advantage of this proposed solution is that it can be implemented at the data-link network layer leading to several benefits:

- Early detection of undesired traffic for minimal waste of system resources
- Completeness of the mechanism, i.e., it cannot be bypassed
- Minimal degradation effect of system performance.

Another approach for securing Myrinet traffic is by using an extension to the trailer to prevent masquerading by third parties. This security trailer would take the form of a hardware-accelerated digital signature of the header and payload to thwart substitution attacks, and should include a source-destination-pair sequence number to prevent replay attacks. This trailer must be decoded with adequate performance by the recipient. This presumes a secure, out-of-band mechanism for providing hosts with digital signature capabilities of other hosts. Alternatives exist in which the routes between hosts are securely distributed, and where a get-based model is used for data transfer. This approach is beyond the scope of the this paper, but is the subject of ongoing research at Mississippi State University.

2.4 Memory Mapping and MCP

The MCP executed on the Myrinet host interface adapter performs the initial processing of incoming packets. This program can be used to implement certain techniques for secure identification and authentication of peers as well as encrypting outgoing and decrypting incoming packets. The architecture of Myrinet host interfaces is designed so that network adapter resources can be mapped directly into the virtual address space of user processes [12], thus, giving them high access privileges uncontrolled by the host operating system. This mechanism improves the flexibility and communication efficiency through bypassing the kernel in the critical data path, but at the same time provides little protection to the MCP and the data buffers in the adapter's local memory.

The memory mapping mechanism violates two of the fundamental requirements of the Reference Monitor, namely, completeness and isolation [2]. Firstly, bypassing the operating system makes the application of any mandatory or discretionary access control policy infeasible, and

secondly, the software that may be implementing certain security measures cannot be protected from an unauthorized access. Furthermore, by mapping network interface resources in its memory, any user mode process can gain access to vital information stored in the adapter's local memory. This leads to another major vulnerability of the memory mapping mechanism - the integrity of the MCP. By destroying, modifying, or substituting MCP functionality, an user with malicious intentions can disclose information, destroy sensitive data, and ultimately, prevent the system from performing its services. The latter is achievable through the use of Myrinet adapter's DMA engines to attack host operating system code and data memory segments or flooding the network with packets to other nodes. Some limited protection of the memory space is provided, but is not comprehensive. U-net [3] is one of the pioneer research projects implementing the memory mapping mechanism and at the same time ensuring protection to the control component residing on the network adapters as well as between multiple user processes on the same host.

It summary, the Myrinet architecture offers a number of high-performance solutions to its users, but at the same time introduces certain security vulnerabilities. Major sources of both improved performance and security deficiencies are the programmability of the network host interface and the memory mapping mechanism. In the next section of the paper, a new network host interface standard specification is presented. Its goal is to also employ the technique of bypassing the host operating system in the critical data path and simultaneously with that increase the level of protection.

3 System Area Network Host Interfaces

In the two-level multi-computer architecture, the role of network host interface adapters is not only to ensure correct access to the network medium and form the electrical signals, but also to perform a substantial part of the message processing and transmission. Thus, the host CPU is freed from the responsibility of handling interrupts generated by incoming or completed outgoing packets, thereby enabling overlapping of useful computation and communication (for systems that are not overly memory bandwidth limited). SAN interface adapters are specifically designed to target the functionality of the communication processors in the two-level multi-computer architecture. They are implemented as active intelligent devices based on specialized microprocessors or custom Application Specific Integrated Circuits (ASIC). SAN adapters often use DMA engines for transferring data between host system memory and on-board adapter's memory [12]. These DMA engines can act as masters on the I/O bus and transfer data directly to/from the user buffers without intermediate copies. The increased independence and processing capabilities of SAN network adapters facilitate a higher overall system performance by hiding latency, off-loading host CPU from immediate communication tasks, reducing the number of data copies, and allowing the use of simple software architectures.

3.1 Virtual Interface (VI) Architecture

Because of the high network data rates and high processing power of modern computer systems, the interface between user processes and network adapters becomes a major performance issue. In traditional models, all input/output operations are handled by the operating system on behalf of the user. However, in a SAN environment, the system overhead for context switches between kernel and user space becomes intolerably high. The VI Architecture is a new industry driven specification that defines the interface between high-performance SAN and computer systems [7]. The main purpose of the VI Architecture is to reduce the communication and system overhead by eliminating extra data copies and the involvement of the host OS in the critical message path. The two basic components of the VI Architecture are the user agent and the kernel agent (Fig. 3). The user agent resides in user space while the kernel agent functions as a part of the host operating system. Main VI Architecture abstractions are the Virtual Interfaces (VI) and the VI connections. A VI is a set of software mechanisms for data transfer, synchronization, and notification that are provided to user processes

through an Application Programming Interface (API). VI connections are logical links between two VIs existing on remote nodes [7]. Multiple VI connections can be established between two remote processes.

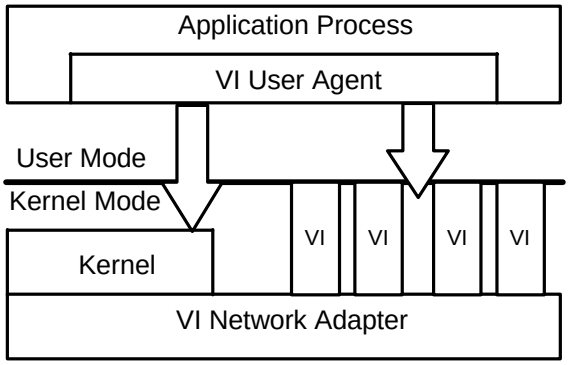

Fig. 3. Virtual Interface Architecture

In addition to its high-performance features, the VI Architecture offers several valuable solutions for increasing the level of assurance in a SAN based distributed system. For instance, one area of security concerns described in the review of the Myrinet host interface adapters was the lack of protection mechanisms that isolate user processes from each other and prevent attacks to the adapter control program. These issues are now addressed by the VI Architecture. Only the host operating system can directly access the memory and control registers of the VI Architecture network adapters; thus, the control program executed on these adapters is protected by preventing a direct user access to it. Instead, user processes interact with the network through the operations provided by the VIs. These operations are designed so that they deliver optimal efficiency and minimal system overhead while at the same time maintain the integrity of the control program and protect the information currently residing on the adapter from disclosure or modification.

3.2 VI Architecture Memory Model

The VI Architecture introduces an elaborate memory management model that substantially improves the protection in the host computer systems. Prior to performing any data transmission, the VI Architecture requires that user processes register all buffers used in the communications operations [7]. The registration procedure locks user memory pages in physical memory, creates a unique memory handle for each buffer, and associates a protection tag with this buffer. Protection tags are also associated with the VI attributes. Among other components, VI attributes include a maximum transfer unit (MTU) size, a reliability level, and a protection tag. A process can use a buffer to send or receive data through a VI only when two conditions are met: first, the memory handle of this buffer belongs to the requesting process, and second, VI's and buffer's protection tags match. As a result, user processes on a single node are protected from accessing each other's memory which prevents the threat for disclosing or destroying sensitive information.

Another important feature of the VI Architecture memory model is the use of virtual addresses in the interaction between user processes and the VI communication software and hardware. This feature is one of the major contributions of the VI Architecture and also has a positive impact on the security. In contrast, the Myrinet memory model requires that user processes obtain the physical addresses of the data buffers and supply them to the MCP. Consecutively, the MCP uses these physical addresses and initiates a DMA to the host memory. Without protection mechanisms for verifying the ownership of the physical addresses, the MCP can access any segment in the computer system's memory and even destroy information vital for the functionality of the operating system. This leaves Myrinet systems unprotected against malicious programs that can exploit the use of physical addresses to engage in confidentiality, integrity, or availability attacks.

3.3 VI Connection Management

Another security enhanced mechanism of the VI Architecture is its connection management model [7]. In the current specification, the VI connections are managed according to the client-server model. A process opens a passive connection and waits for VI connection requests. Then, a remote process makes a request for such a connection and supplies the attributes of its VI in the request. The process that has passively opened the connection can accept or reject the request.

This model of connection management suggests a mechanism for controlling the incoming requests for connections according to some criteria based on the remote node's address and VI attributes.

Although VI connections resemble the transport layer point-to-point connections as implemented by TCP in the TCP/IP protocol stack, they feature some significant differences. These differences are illustrated in the following example. An Ethernet LAN is used to connect a group of workstations. The IEEE 802.2 Media Access Control protocol specifies the mechanism for gaining control over the common transmission media. Once a node has gained control, it can generate an Ethernet frame that is broadcasted to all nodes (unless the network is switched). Each of the nodes receives the frame that may encapsulate a TCP segment requesting a TCP connection. The TCP segment carries the destination and source IP addresses and the destination and source TCP ports. The payload of the Ethernet frame is forwarded first to the IP software and then demultiplexed to the corresponding TCP port. Only then, the receiving node can decide whether to accept or reject the request for connection. By this time, the host system has performed certain processing using its resources even when the request is rejected. As opposed to this scenario, VI connections are established at the data-link layer and the requests can be evaluated at the immediate entrance of the host system - the network interface adapter. In case of a rejection, the higher software layers will not be invoked at all. This leads to a lower system overhead and at the same time to a tighter access control.

3.4 Security Augmented Memory and VI Connection Models

As it was shown earlier, a request for a VI connection can be evaluated based on the remote node's address and VI attributes. This, however, does not provide enough information for achieving secure connection establishment. One solution to this problem is the implementation of a user software security layer on top of the VI Architecture. This layer will perform secure identification, authentication, and possibly evaluation of classification levels. A different approach for secure connection management is the support of security extensions in the VI Architecture which requires modifications in the current specification. The former solution has a lower assurance level since it resides in user space, where user processes can bypass it. The valuable side of the VI connection establishment mechanism is that it is performed by the kernel agent of the VI Architecture, thus guaranteeing the involvement of the host OS. However, after the connection is established and user buffers are registered, the OS is eliminated from the data transfer. Therefore, an effective secure connection management can be achieved only if the VI attributes have a security related component which is examined by the VI Architecture kernel agent when the incoming requests are evaluated.

It is proposed in this paper that the VI Architecture specifically define a security component in the VI attributes (Fig. 4). This component may include parameters such as classification levels and information for secure identification and authentication. Using security attributes, the VI connection management will facilitate creation of low-level, end-to-end trusted connections and shift the implementation of security protocols to the data-link layer.

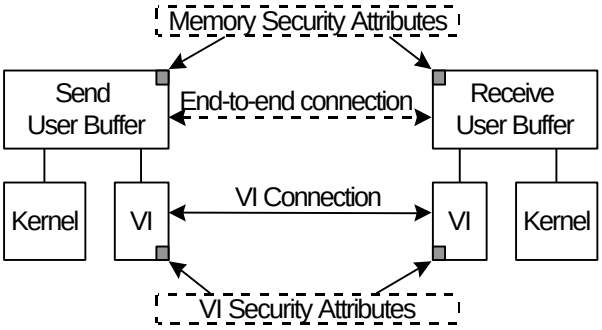

Fig. 4. Memory and VI security attributes and secure end-to-end communication

The proposition for VI security attributes extension can also be successfully applied to the memory attributes (Fig. 4). The VI Architecture specification may define an optional mode of operation in which a user buffer will be transferred over a VI connection only if the security attributes of both the memory region and the VI match and, similarly, an incoming message will be stored only if the originating VI and buffer have acceptable security attributes.

TCSEC require auditing of the "introduction of objects into user's address space" [18]. This requirement is considered too restrictive [1] for efficiency purposes. Memory and VI security attributes, when used adequately, will allow the VI Architecture to control and monitor the classification levels of information objects that will take the form of originating data buffers, VI connections, and destination buffers. Thus, each host system can implement audit functions by recording the requests for establishing VI connections. This mechanism will guarantee that all objects transferred over an audited connection comply with the conditions met at the time when this connection is established. Thus, the security augmented VI Architecture connection and memory management can be successfully used in implementing effective mandatory access controls (MAC) for higher degree of data confidentiality and integrity.

3.5 Remote Direct Memory Access

The VI Architecture defines read and write remote DMA (RDMA) operations for increasing communications performance. While highly efficient, the RDMA operations impose certain security threats. According to the current specification, once a VI connection is established a remote process can initiate read or write RDMA without explicitly negotiating the transaction with the process that owns the remote VI and data buffers. The network interface adapter will perform the specified RDMA operation on behalf of the user process that is the target of the RDMA operation without any involvement of the host CPU, hence the host operating system. In fact, the processes owning the target buffers do not have any control on the data transfer and are notified only when the operation is completed. Consequently, no access control on the data transactions can be imposed. Confidentiality and integrity mandatory access controls require a clear distinction between access rules for reading and modifying information objects [4, 5]. The VI Architecture does not provide a means for such distinction. So, even when a control mechanism establishing VI connections with only authorized peers is implemented, confidentiality and integrity threats still exist.

A possible solution consists of adding explicit memory attributes that specify permitted memory accesses, namely, whether RDMA is allowed and what types of RDMA is allowed - read, write, or both. When a user process registers a memory region, it will specify the desired access controls. This information will be made available to the network interface adapters, which in turn will control the requests for RDMA to the particular memory segment. RDMA requests may be rejected if they are not enabled in the memory attributes of the target segment at registration time. This solution requires expansion of the memory protection attributes with components for access controls and certain modifications in the VI Architecture adapters behavior. The proposed solution will increase the level of protection without adding a significant extra processing overhead, so it will not negatively affect the communication performance.

3.6 Sensitive Information in the VI Architecture Network Packets

The last security consideration investigated in this section is related to the importance of the information contained in the VI Architecture packets exchanged across the network and the possible impact of the malicious use of this information. The packets consist of a header and a payload. The header carries control information items such as source and destination addresses, VI connection identifier, target buffer address, remote memory handle, and VI attributes. Capturing the VI connection identifier, the target buffer address, and its memory handle is sufficient for an subject participating in an active wiretapping to request read RDMA to the target buffer. Thus, without being a target of the data transfer, such masqueraders will be able to gain access to the information in the specified buffer and violate the information confidentiality. With the same success, the intruder will be able to generate RDMA write requests and attack the integrity of the data contained in the target buffer.

A potential solution that may increase the level of protection against the scenario described above is the use of optional or mandatory packet formats in which the sensitive information is encrypted using private or public keys. Currently,

the VI Architecture does not specify the packet formats and their implementations is left to the network vendors. A VI Architecture security extension may explicitly specify the secure packet formats and protocols with security attributes.

In summary, the VI Architecture specifies an efficient interface between a SAN and host computer systems that offers a number of new approaches for building high-performance communications systems. The VI Architecture is a step forward not only in providing high bandwidth and low overhead data transmission but also in introducing mechanisms for improved protection. The security assurance of a distributed system based on the VI Architecture could be further improved by employing the solutions proposed earlier in this paper. These solutions are chosen so that they increase the level of thrust without necessarily adding extra processing overhead. In many cases, these solutions require relatively simple security extensions to the current VI Architecture specification.

4 Secure Communications between System Area Networks

Though the secure VI Architecture solutions suggested in the previous section can provide higher degree of assurance within a single network, they do not address the need to connect such networks in a secure manner. Often it is necessary to connect multiple networks, each carrying information with different levels of sensitivity, in a way that does not leave the data vulnerable to intruders. In this section we introduce PacketWay and Secure PacketWay, an effort to provide secure communications for larger systems spanning multiple networks.

4.1 PacketWay

In the same way that the Internet Protocol (IP) is used to internetwork Wide Area Networks, the goal of PacketWay is to internetwork SANs. It attempts to provide improved performance by directly supporting and exploiting advanced features of modern networks, including source routing and cut-through routing. PacketWay is currently an IETF Experimental Standard [19].

PacketWay packets usually consist of a series of routing headers that specify the native routes through the heterogeneous SANs between two endpoints, followed by a main message header and data block. Each routing header encapsulates the native route through one of the SANs. Placing the routing headers at the front of the packet allows PacketWay routers to read only the first few bytes of a packet and immediately route the rest of the message without any further processing overhead. Hardware implementations can perform this "cut-through" routing quickly and keep latency much lower than a "store-and-forward" routing strategy.

The PacketWay specification does not mandate a standard error checking method that all nodes must support. Instead, it is assumed that each type of SAN will have native error-checking and error-correcting capabilities of its own that are well optimized. The PacketWay frame format provides a mechanism for indicating that a transmission error has occurred and which hop introduced the error via an Error Indication (EI) trailer following each packet. This allows the endpoints to devise their own mechanisms for circumventing faults and possibly detecting intrusions in the network.

4.2 Secure PacketWay Extensions

Secure PacketWay is an extension of the PacketWay standard that addresses secure communication in the PacketWay environment [8]. "Secure communication" is defined as the delivery of data between trusted parties in such a way that unauthorized parties cannot:

- Interpret sensitive messages;
- Forge messages such that they appear to be from another party;
- Alter sensitive messages without detection;
- Replay previously valid messages and have them accepted as genuine.

Secure PacketWay interconnects secure or "trusted" SANs by passing data through insecure or "untrusted" networks. Each secure SAN contains trusted nodes that communicate within the SAN using local security policies (e.g., secure data-link protocol or VI Architecture extensions). At least one node on the SAN is a Secure PacketWay router connected to other trusted SANs via an untrusted network.

The assumed threat to the inter-SAN network is an active adversary who has gained physical control of one or more network links in the untrusted network. It is assumed that any and all data crossing the untrusted network can be copied, deleted, or altered by the adversary.

All communication between trusted SANs takes place through Secure PacketWay routers, which are actually made up of two interconnected processes called "half-routers" (Fig. 5). Each half-router (HR) is a full-fledged node on a SAN, one on the trusted ("red") SAN and the other on the untrusted ("black") SAN. Conceptually, the red and black HRs within a single router are separated by a cryptographic device through which all inter-SAN traffic passes.

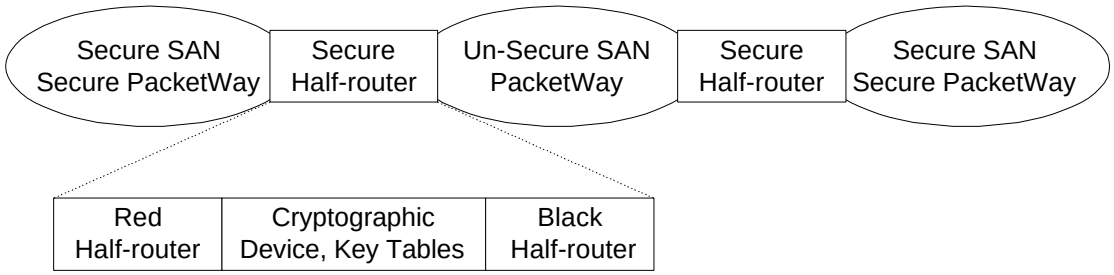

Fig. 5. Secure PacketWay model

When a packet is to be routed from the red HR across the black SAN, the cryptographic unit encrypts and encapsulates the sensitive packet inside a plaintext packet with appropriate native routes. This plaintext packet is then given to the black HR, which sends the packet through the black network to the other secure router. At the other end the process is reversed and the black HR hands the plaintext packet to the cryptographic unit, which unwraps and decrypts the sensitive data, which is finally sent to the destination by the red HR. Because red HRs only contact the black HRs through the cryptographic unit it is not necessary for either side to know anything about the other. The red HRs are logically connected by the cryptographic units in such a way that they cannot and do not know the nature of the network between them. Since all plaintext packets are sent and received by the black HRs, the adversary in the middle cannot determine the identities of the communicating parties on the red SANs.

The Secure PacketWay model of a "network layer firewall" is important because it allows communication within the trusted SAN to take place at full peak performance, and only traffic that might be compromised by going through the untrusted network must endure the overhead of encryption and encapsulation. Delegating security duties to the routers is even more important for embedded systems that have limited resources and strict performance requirements. It relies on the creation of encryption and decryption devices that can pipeline their operations at sufficient performance to keep up with the native performance, which is an area for significant additional study and inquiry.

5 Conclusions

This paper reviewed three components of a modern integrated high-performance distributed system: a System Area Network, a network host interface, and a SAN internetwork protocol. Representative technologies of each of these components were studied, namely, Myrinet, the Virtual Interface Architecture, and PacketWay. The decisions facilitating efficient data communications were pointed out and their impact on the security of the networked systems was investigated. The paper raised security concerns regarding some consequences of new approaches for efficient data transmission, and proposed solutions for increasing the level of protection and assurance. Main objective of these solutions is to introduce security procedures at the lowest layers of the communication system. Thus, the security mechanisms have greater effectiveness and, at the same time, cause minimal processing overhead.

Security mechanisms are in general more effective when they are incorporated at the design

stage of networks, protocols, or host interfaces rather than being added consequently to fully operational and wide accepted products. The technologies reviewed here are at different stages of realization but they all have substantial support from industry and research organizations for fast acceptance in practice. There are already a significant number of Myrinet based clusters in defense and research laboratories that perform parallel computing. The standardization effort of PacketWay has also shown promising results. Independent working prototypes implemented at MSU and Sanders have been demonstrated to interoperate successfully. Also, the VI Architecture quickly gains supporters among the industry leaders in the area of high-performance computing. A beta testing versions of VI Architecture compatible networks are already available. The authors of the paper believe that the emerging standards in the area of high- performance distributed systems offer new paradigms for concurrent processing requiring new security solutions and that now it is an appropriate time for augmenting these standards with adequate mechanisms for security and protection.

6 References

[1] Abrams, M.D. and H.J. Podell, *Information Security: An Integrated Collection of Essays, Essay 16 - Local Area Networks,* IEEE Computer Society Press, Los Alamos, CA, 1995.

[2] Anderson, J.P., *Computer Security Technology Planning Study*, ESD-TR-73-51 , Vol. 1, Hanscom AFB, Mass., 1972.

[3] Basu, A., V. Buch, W. Vogels, T. von Eicken, *U-Net: A User-Level Network Interface for Parallel and Distributed Computing,* Proceedings of the 15th ACM Symposium on Operating Systems Principles (SOSP), Copper Mountain, Colorado, December 3-6, 1995

[4] Bell, D.E. and L.J. LaPadula, *Secure Computer Systems: Mathematical Foundations and Model*, M74-244, MITRE Corp., Bedford, Mass., 1973

[5] Biba, K.J., *Integrity Considerations for Secure Computer Systems*, ESD-TR-76-372, USAF Electronic Systems Division, Bedford, Mass., Apr. 1977.

[6] Boden, N.J., D. Cohen, R.E. Felderman, A.E. Kulawik, C.L. Seitz, J.N. Seizovic, and W.K. Su, *Myrinet: A Gigabit-per-Second Local Area Network*, IEEE Micro, Vol. 15, No.1, Feb. 1995.

[7] Compaq Computer Corp., Intel Corp., Microsoft Corp., *Virtual Interface Architecture Specification: Version 1.0, Dec. 16, 1997*, http://www.viarch.org/html/Spec/document/san_10.pdf (Downloaded: 1 Feb. 1998).

[8] George, R., J. Smith, F. Shirley, T. Skjellum, T. McMahon, G. Byrd, and D. Cohen, *Proposed Specification for Security Extensions to the PacketWay protocol*,http://www.erc.msstate.edu/labs/hpcl/packetway/secure.txt (Downloaded: 24 Jan. 1998).

[9] Henley, G., N. Doss, T. McMahon, and A. Skjellum, *BDM: A Myrinet Control Program and Host API.* Mississippi State University, Technical Report: MSSU-EIRS-ERC-97-3, 1997.

[10] International Standards Organization, *Information Processing Systems - Open Systems Interconnection Basic Reference Model - Part 2: Security Architecture*, ISO IS 7498-2, 1988.

[11] Jajodia, S. and R.S. Sandhu, *Information Security: An Integrated Collection of Essays, Essay 23 – Toward a Multilevel Secure Relational Data Model*, IEEE Computer Society Press, Los Alamos, CA, 1995.

[12] Myricom, Inc.a. *LANai 4.x.Documentation* http://www.myri.com/scs/documentation/mug / development/LANai4.X.doc (Downloaded: 22 Dec. 1996).

[13] National Computer Security Center, *Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria*, NCSC-TG-005, July 1987.

[14] National Institute of Standards and Technology, *Guideline for the Analysis of Local Area Network Security*, Federal Information Processing Standards, Pub. 191, Nov. 1994.

[15] Pakin, Scott, Vijay Karamcheti, and Andrew Chien. *Fast Messages: Efficient, Portable Communication for Workstation Clusters and MPPs*. IEEE Concurrency, Vol. 5, No. 2, April - June 1997

[16] Rushby, J. and B. Randell, *A Distributed Secure System*, Computer, July 1983, Vol.16., No.7.

[17] Tanenbaum, A.S, *Computer Networks*, 3rd ed., Prentice Hall, Upper Saddle River, NJ, 1996

[18] United States Department of Defense, *Trusted Computer System Evaluation Criteria*, DoD 5200.28-STD, Dec. 1985.

[19] Cohen, D., C. Lund, T. Skjellum, T. McMahon, and R. George, *The End-to-End (EEP) PacketWay Protocol for High-Performance Interconnection of Computer Clusters*, ftp://ftp.ietf.org/internet-drafts/draft-ietf-pktway-protocol-eep-spec-02.txt (Downloaded: 24 Jan. 1998).