Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

SECURITY ISSUES IN HIGH LEVEL ARCHITECTURE BASED DISTRIBUTED SIMULATION

Asa Elkins

Integrated Data Systems
14160 Newbrook Drive, Suite 210
Chantilly, VA 20151, U.S.A.

Jeffery W. Wilson

Department of the Navy
Naval Sea Systems Command
2531 Jefferson Davis Highway
Arlington, VA 22242-5160, U.S.A.

Denis Gracanin

Department of Computer Science
Virginia Tech
Northern Virginia Center
Falls Church, VA 20043, U.S.A.

ABSTRACT

The United States Department of Defense (DoD) has, over the past several years, emphasized the need to employ simulation based acquisition (SBA) in engineering and development. Distributed simulation introduces an information assurance challenge and details of a simulation must be guarded from unauthorized access. The High Level Architecture (HLA) and its Run-Time Interface (RTI) do not define support of mandatory access controls (MACs) or discretionary access controls (DACs) required to provide necessary protection levels. We review of some current MLS approaches for HLA/RTI simulations to illustrate the deficient Multi-Level Secure (MLS) components in HLA and present options for a secure HLA interface built at the network layer. An initial implementation of a proposed solution is presented. We discuss experimental results, limitations of our implementation and future research directions.

1 INTRODUCTION

The successful engineering and development of highly complex systems depends heavily upon modeling and simulation. The need to conduct trade-off analyses and to manage risk throughout the system engineering process drives the creation of models and simulations at varying levels of abstraction. From individual components, through aggregation of those components in lower level systems, to the final ensemble, system engineers require tools of varying levels of fidelity and sophistication. This system-oriented approach applies equally to the public and private sectors – characterizing expected system performance before significant expenditure of wealth is important to both industry and government.

Humans have built models and simulations for thousands of years. As systems have become more complex, so have the tools used to build those systems. The advent of the digital computer, and the development of numerical methods to describe systems and their environments digitally, have brought increasing pressure to characterize systems and their expected performance in operational environments. The fidelity of individual system and environmental models has increased to keep pace with demand.

The United States Department of Defense (DoD) has, over the past several years, emphasized the need to employ simulation based acquisition (SBA) in engineering and development. SBA provides a disciplined way to improve the decision making process, reducing risk in acquiring increasingly complex systems.

The ubiquity of digital computers, and the communication infrastructure that connects them, has generated a demand to reduce or eliminate the manual exchange of information between and among models and simulations; “sneaker net” solutions are being replaced by technologies that link models and simulations within a common infrastructure. The High Level Architecture (IEEE 2000) is an example of such an infrastructure, and is the focus of the work described in this paper. The Defense Modeling and Simulation Office (DMSO) has developed the High Level Architecture (HLA) to substantially improve software interoperability of reuse among DoD simulations. HLA is essentially a set of rules that simulation developers use that will allow for interoperability and reuse of simulations. The rules require an object model that describes each simulation and federation (a group of simulations), describe basic levels of support for interactions between simulations, define federation-wide services, and specify the interface between the simulations and the run-time infrastructure (Kuhl, Weatherly, and Dahmann 2000). Interactions between simulations in a Federation Execution are controlled by the Run-Time Infrastructure (RTI). RTI provides simulation an Application Programmers Interface (API) for Federation Management, Declaration Management, Object Management, Ownership Management, and Time Management. These are generic services designed to service a broad range of simulation applications.

Several factors are causing an increase in reliance on distributed simulation over such networks. The diffusion of subject matter expertise in a given domain, short-duration industrial alliances to solve particular problems, and fixed sites for specific high-cost devices are among the factors that necessitate distributed simulation. Distributed simulation is also required because we have recognized that if two or more systems are to interact on the battlefield to accomplish a mission, it is only logical to demonstrate this interaction “on the bench”. The United States Department of Defense is putting shape to a nascent concept known as Network Centric Warfare (Alberts, Garstka, and Stein 2000). The central tenet of NCW is that information collection, processing, management, dissemination, and understanding is the key element of both deterrence and, when deterrence fails, success on the battlefield. NCW necessarily leverages networks – collections of humans, sensors, computers, and the communication links that tie them together into an ensemble. As the NCW concept matures, the DoD is faced with the need to describe, in operational terms, the military utility that can be expected when people and computers are interconnected. The evaluation of military utility is most affordably done in a distributed simulation environment – amassing the individual fighting units necessary to conduct an objective evaluation is difficult to do, when those assets are already either preparing for or fully employed in deterrence and peacekeeping missions.

Models and simulations can interact at varying levels of fidelity as systems are being developed; this interaction can be used to ensure functional allocations are correct and interfaces are properly described and implemented. Correction of deficiencies early in the system life-cycle results in increased performance while avoiding cost. Distributed simulation also offers a way to explore ways in which existing or emerging systems can better work together to accomplish a mission, and can be used to support development and conduct of operator training.

Distributed simulation introduces an information assurance challenge. The need to protect national security and proprietary interests by ensuring that the details of a simulation (and hence the detailed design of the simulated system) are guarded from unauthorized access. In the case of national security interests, not only must the simulation be guarded, but in many cases, the stimulation and response is classified. For proprietary interests, the government must develop ways to encourage innovation by a large number of small companies and institutions while protecting intellectual property in a marketplace dominated by a few large corporations (NSTISSC 2000).

This paper present an option for a secure HLA/RTI interface built at the network layer using IPSec protocol. Section 2 describes three examples of operational scenarios for secure distributed simulations. Section 3 provides and overview of the related work. Section 4 describes the proposed methodology while Section 5 provides some experimental results and analysis. Section 6 discusses limitations of the implementation and future directions for research.

2 OPERATIONAL SCENARIOS FOR SECURE DISTRIBUTED SIMULATIONS

There are important security-related performance and cryptography challenges that arise when the linkages among models and simulations are accomplished using wide area networks. The practical problem is making sure that performance requirements are met while satisfying national and commercial industrial security requirements. In particular, it is necessary to obtain adequate performance from the RTI while using standard Internet protocols. The issue is if those standard Internet protocols can be used to meet both national security and industrial security needs and requirements. Significant benefit is obtained if public and private needs can be met with a single method.

To put the practical problem in proper context, three operational scenarios are described. The first two scenarios highlight some of the national security issues associated with distributed modeling and simulation. These scenarios share a number of similarities; this is to be expected because system engineering problems are fundamental in nature; they arise whenever complex systems are engineered and developed, regardless of the entity responsible for doing the work. Differences that arise are typically driven by legislative or regulatory requirements.

2.1 Distributed Development

In this scenario, we examine the case where two or more industrial partners are engaged in cooperative development of components that will be integrated into a single system. For instance, one company may be developing a sensor that must be integrated with a command and control element built by a different company. Responsibility for integrating these components lies with a third company. The government has significant interest in ensuring these components work together to accomplish a mission, and that they can be installed in a vehicle. All partners have equal clearance and need-to-know for both the stimulation data set and the results of the modeling and simulation effort. Because this is a development effort, they also have a need to exchange large volumes of data, including voice and video. These partners have a business relationship for this project only; in other development projects, they are head-to-head competitors. For business reasons, the partners have elected not to collocate; they desire to interconnect their development environments for the purpose of completing this single development effort.

To execute their contractual obligations, and to comply with U. S. Department of Defense directives, instructions, and regulations, the industrial partners must comply with physical and information assurance requirements. Physical security requirements demand strict and verifiable access control, and involve, among other things, locks, alarms, and physical inspections. Information assurance is typically provided by National Security Agency (NSA)-developed and produced communications security (COMSEC) equipment, though commercially produced COMSEC equipment is available for use with NSA-developed keying material.

2.2 Distributed Test and Evaluation

In this scenario, the government is interconnecting systems in a distributed modeling and simulation environment for the purpose of characterizing the performance of the ensemble in a simulated operational environment. Each of the systems in the ensemble was acquired from different industrial partners, at different times, and to different interface standards. The modeling and simulation environment must be distributed because of the large fixed sites that support the effort. In some cases, there are hardware-in-the-loop simulations that cannot be moved. Additionally, significant cost can be avoided by precluding the need for collocation of people to support the work.

Because the systems involved in this distributed simulation are operating at different levels of classification, not only must the stimulation and response information be classified, provisions must be made to protect information at higher levels of classification from being inadvertently divulged at lower levels of access. It is also important, within a single classification level, to restrict the flow of information only to those who have a need to know. As with the distributed development scenario, separate equipment is used to provide communications security.

As with the distributed development scenario, significant information related to the evaluation must be exchanged among participating sites, so bandwidth must be allocated to this additional traffic.

2.3 Commercial Industrial Security

As discussed earlier, the fundamental issues discussed here are the same as the national security case. The differences lie primarily in the cryptography used to implement information assurance requirements. Commercial users could be expected to use IPSec, perhaps as implemented in Microsoft® Windows® 2000 Server and Microsoft® Windows® 2000 Professional, to meet this requirement.

In this scenario, participants are interested in maintaining the confidentiality of their models and simulations. The detailed design of each company's product is a trade secret that must be protected from competitors. To demonstrate compatibility and interoperability, participants must engage in distributed simulation activities. To protect their trade secrets, they want to carefully restrict the amount of information divulged to other participants. The information assurance method used must support this selective release of information.

3 OVERVIEW OF APPROACHES FOR SECURE SIMULATION WITH HLA

HLA is a framework to combine simulations into a logical grouping, called a federation. Each partner in a federation is called a federate. Federations are typically implemented at a system high classification. Due to operational requirements, federations need to exchange data between unclassified and classified security boundaries, i.e. users need to be able to run federations that use information from many different sources. Classification is not the only concern, need-to-know and releasibility are additional factors that require consideration.

Two approaches are reviewed, one by Filsinger and Lubbes (1996) and the other one Bieber et. al (1998). Filsinger and Lubbes discuss the security requirements that distributed simulations, and HLA in particular, need to address. Their work outlines the security requirements, describes a system security concept for HLA, and describe current and future implementations. This work was summarized and augmented in Ozdemir paper (Ozdemir 1997). Bieber et al. examine the ONERA/CERT implementation of HLA/RTI that builds a secure sub-layer for RTI. Though the CERT approach does not address many of the MLS layers, it does deal with confidentiality of federation objects and properties and provides insight into potential implementation problems.

3.1 Filsinger and Lubbes Approach

According to the Defense Information System Security Program (DISSP) Goal Security Architecture, DoD information systems must be protected to: allow commercial carrier connectivity, allow distributed processing among multiple hosts, support multiple security policies governing unclassified and classified data, and support varied security protections.

These requirement require HLA to support: Federations operating at a range of security levels, simulations within federations operating at different security levels, transfer of object attribute responsibility among simulations operating at different classifications, confidentiality, integrity, and need-to-know policies, and reuse of simulations at different security levels.

The Ozdemir paper (Ozdemir 1997) summarizes the guidelines for adherence to the HLA requirements:

- HLA architecture must allow processing of MLS data among federates with users that do not have all the appropriate security clearances.
- Information must be prevented from leaking from high level of security to low level.
- HLA architecture will have to support processes that are capable of protecting information within a security classification and support processes that can be trusted to downgrade (sanitize) data.
- HLA must support security mechanism to allow object ownership and object attributes to be safely read or updated by any simulation with a federation.
- The implemented architecture must support enforcement of mandatory confidentiality, integrity, and need-to-know policies.
- Simulations must be reusable at different security levels at different times in different federations.

To support these principles, Filsinger and Lubbes proposed the following solutions: single security level, multiple single security levels with security guards and trusted agents, and a multiple MLS security domains.

Filsinger and Lubbes (1996) presented the single security level solutions to illustrate how HLA/RTI implementations currently work. There is a single security domain operating at a single classification in a system-high mode. This clearly does not support multiple security principles or domains or different classification levels. This example provides evidence that further work is needed to meet the aforementioned HLA guidelines.

The second example multiple security domains accommodates multiple security domains. Each security domain is described by a different security policy describing classification, releasability, and/or need-to-know restrictions. Guards and trusted agents are used to provide interdomain security communications. Guards have the capability to downgrade the classification of data without human review. To allow the guards to sanitize data, the data must be well structured and sanitization rules well-defined. Trusted agents create trusted channels between security domains. Trusted agents direct individual RTI requests to send data to the correct guard for that security domain. Trusted agents interface only with individual RTIs and have no mechanism to restrict data flow.

This second example requires that all communications between RTIs that are transmitted over an open network employ the proper cryptographic protections.

The third example that Filsinger and Lubbes present utilizes multiple security domains with federates hosted on MLS hosts. The primary difference between this implementation and the previous example is the requirement for all hosts to be MLS hosts and the use of distributed RTI implementation that supports MLS.

3.2 Bieber et al. Approach

This example is based on the work published by the ONERA/CERT team (Bieber et al. 1998), based on the ONERA/CERT implementation of RTI. The example analyzes the security threats to ONERA/CERT's RTI implementation. Its focus is application of security to HLA RTI in a commercial sense and deals with confidentiality of technology.

The publication first describes the ONERA/CERT implementation. The implementation divides RTI into 2 parts, the RTI Ambassador (RTIA) and the RTIG. Each federate interacts with a local RTIA through the libRTI library. The RTIA processes exchange messages over a network with the RTIG process. The RTIG is outside of each federate. Only one RTIG process is used by all federations.

The example describes three kinds of interactions between the federate processes:

- Local libRTI calls between the federate and the RTIA to request service,
- Communications between RTIA processes on the same host, and
- Communications between RTIA and RTIG for inter-host processes.

The example continues to describe the security objectives of a security-aware HLA/RTI. The security-aware HLA/RTI implementation is responsible for distributing properly the values of sensitive federate object properties. The paper emphasizes that before the security objectives can be described, first the threats to the sensitive federate object properties must be understood.

The ONERA/CERT team identified 3 potential threats to its RTI implementation:

- Communications between local RTIA processes and the RTIG,
- Leak of sensitive object properties via supported RTI processes, i.e. broadcasting too widely the values of the sensitive object properties, and
- Direct request of sensitive object properties by unauthorized, but supported, federate processes.

The security objectives are designed to address the above suspect channels. To address the first channel, communications between RTIA and RTIG need to be protected from other components. For the second channel, the RTI processes need to protect sensitive from unauthorized federates. The third channel requires all federate information be isolated from other untrusted federates.

To apply the security objectives the ONERA/CERT leverage the security principles provided by the Security Assurance in Distributed Applications (SAIDA). SAIDA's main assumption is that no multi-level operating system is available for use, thus other means are required to enforce isolation.

To address the first concern (network isolation), a secure association between the federate, the RTIA, and the RTIG is required. These secure associations require a cryptographic solution.

The second objective (isolation with the RTI) requires a security filter between the RTI and RTIG. This requires a combined federation. A combined federation is made of several federations linked by an inter-federation gateway. In addition each federation object model must be extended with security attributes. The security attributes are used to associate each class, attribute, and federate with a security domain. RTI will need to be extended to filter messages based on the security domains. For example, based on HLA rules, the federates that subscribe to object properties are required to be notified of changes. The filter would prevent the subscription, if the security domains of the object and requestor were not complimentary.

The third objective could be accomplished by requiring that federates in the same security domain be hosted by the same machine. In addition the network that handles inter-host RTI communications needs to be dedicated to the simulation.

The ONERA/CERT implementation uses a trusted third party (TTP). The TTP operates a shared LAN where companies are free to connect machines hosting their federates. Each machine is allowed to host federates from only one company. All communications between company machines are mediated and authorized by the RTIG.

The example requires extensions to RTI services to add security domain filters for publication and subscription services (PublishObjectClass, PublishInteractionClass, SubscribeObjectClass, SubscribeInteractionClass). Theses messages are erased whenever the seucirty domain of the requing federate does not dominate and is not equal to the security value of the requested class. As the RTIG transmits the UpdateAttributeValue messages only to authorized subscribers, a federate from another company will never receive ReflectAttributeValue messages.

To ensure the confidentiality of communications between RTIA and RTIG, the example relies on the Secure Media Access Control (SMAC) protocol.

4 PROPOSED APPROACH

Enabling technologies are emerging rapidly. Increases in computational capacity continue unabated. Public and private institutions are investing to provide the communications bandwidth and protocols necessary to support distributed simulation. The protocols of interest in this problem area include the IP Security Protocol (IPSec) effort conducted on behalf of the Internet Engineering Task Force (IETF). The cryptographic community is advancing the state of the art in the strong encryption algorithms that will be necessary to provide the required level of information assurance. Recent developments include the efforts by the U. S. National Institute of Standards and Technology (NIST) to select the Advanced Encryption Standard (AES). Understanding the security issues that arise in a distributed simulation environment is important. We are interested to know if existing and emerging network layer security mechanisms will support distributed simulation. To investigate this problem, we examine a simple distributed simulation environment.

The goal of our approach is the same as those presented by the provided examples:

- Support for common carrier communications systems.
- Compliance with the HLA rules.
- Minimal modifications to federation object models and RTI.
- Support for standard security principles, confidentiality, integrity, and need-to-know principles.

The alternative approach investigated in this paper is to implement RTI over Internet Protocol Security (IPSec). IPSec can interoperate with both IPv4 and IPv6. IPsec in conjunction with an innovative approach to public key infrastructure (PKI) could provide the basis for a secure implementation of HLA/RTI.

The PKI architecture could have 3 roots, 1 for Top Secret, 1 for Secret, and 1 for Unclassified. Need-to-requirements could be provided by implementing subdomains for the top-level domains.

4.1 IPSec Protocol

IPSec provides data privacy, integrity, and authenticity for network traffic. Additionally IPSec has the ability to protect against anti-replay attacks. IPSec provides the heightened security for client-server, server-server, and client-client communications (Kent and Atkinson 1998).

IPSec provides security services at the IP layer by allowing hosts a mechanism for negotiating security protocols, cryptographic algorithms, and associated cryptographic keys. Once the security services are negotiated by a pair of hosts, the services can provide authentication, replay protection, data integrity, and confidentiality to higher level protocols, including the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

To implement IPSec on a client or a server, an administrator must create and apply an IPSec policy to both the client and the server. As packets are processed for inbound or outbound connections, the IPSec driver compares IPSec policy against the packets. If the IPSec driver finds a match, it applies the appropriate actions. These actions include initiating key exchange, signing, or encrypting packets. The policy may also do nothing, depending on whether IPSec is required or only attempted (Microsoft 2000).

IPSec uses an on-demand security negotiation and key management service defined as Internet Key Exchange (IKE). IKE supports three standards-based authentication methods to establish the connection between computers: Kerberos, public key, and shared key. Once the computers have authenticated to each other, they generate bulk encryption keys to encrypt application data packets.

4.2 IPSec and RTI

IPSec can easily provide RTI with confidentiality. By hosting the RTI on an IPSec machine with a policy that requires IPSec, third party hosts cannot intercept and decode data protected by IPSec.

Thus, for Filsinger examples, the required cryptographic devices can be replaced by a proper IPSec implementation. The ONERA/CERT example used Secure Media Access Control (SMAC), which would also be redundant, if IPSec was implemented.

IPSec could be the basis for stronger security measures than simply encryption between participating RTI hosts. Consider combining IPSec with a Public Key Infrastructure. Start with 3 Certificate Authorities (CAs) for Unclassified, Secret, and Top Secret security domain. The Certificate authorities are arranged hierarchically with the Unclassified CA at the root level. The Secret CA would be subordinate to the Unclassified CA, and likewise the Top Secret CA would be subordinate to the Secret CA.

Because there are separate CAs, there can be separate certificate policies. The certificate policy for the Unclassified CA would allow hosts with certificates from the Unclassified, Secret, or Top Secret CAs to establish IPSec connections to Unclassified hosts. This would allow federates hosted in any domain to request information from Unclassified hosts. Similarly Top Secret hosts could request information from Secret hosts. Top Secret CA would have a policy that would deny requests from hosts that didn't have certificates from the Top Secret CA.

This hierarchical CA approach supplies a simple industry standard approach to providing security domains without specialized hardware. As discussed in Section 3, this approach could be extended to support need-to-know requirements by establishing additional subordinate CAs at the under the appropriate CA.

To fully support multi-level secure principles, security extensions to RTI and federate object models are required to allow federates with multiple security levels of data to provide the data to other objects.

First the federate object model (FOM) would need to provide a mechanism for requesting and storing private keys. Based on the data security requirements, the object would need to request a private key from the appropriate CA. Since the federate object models generally have a hierarchical structure, a private key would need to be supported at each level. Additionally the FOM would need to support Access Control Lists (ACLs) that enable the federate and RTI to allow or deny access based on the provided authentication information, provided via the private/public key implementation.

Consider as an example an object model used in an air traffic control simulation. The simulation would have a generic object for aircraft. This object would request an unclassified certificate for its location. This generic aircraft object would be used to create an object for a fighter aircraft. The fighter aircraft would have a private key for its location proprieties. In addition the fighter would have a weapons property. The weapons property would not be generally available and would require a separate private key, insure by a different CA. When an air traffic controller with a certificate from an unclassified CA attempts to access the fighter's location, the controller would be granted access. If the same controller attempted to query the fighter's weapon load, the request would be denied.

The federation object model would need to be extended to support storing a private key with each object instance. This request for the key would be done at object instantiation via an extension to RTI.

Currently federation processes follow this order:

- A simulation executes the federation execution creation to create the federation.
- Interested simulations join the federation.
- Simulations publish their object classes.
- Simulations subscribe to published objects classes.
- RTI informs publishers of necessity to provide object updates.
- Simulations will instantiate an object.
- Simulations update attribute values.
- RTI forwards updated values to subscribed simulations.
- In a multi-level secure environment, this creates several problems:
    - Not all simulations should be able to join all federations.

  - All simulations should not be able to subscribe to all published object classes.

Transfer of ownership management needs to be restricted to only authorized users but it is not an issue because only subscribed objects can assume ownership. The following is needed too ensure data is afforded the correct handling instructions:

- Communications between simulations and the RTI need to be protected at a level commensurate with the security level of the data.
- Requests for data require a check of the requester against the data.

5 EXPERIMENTAL RESULTS

An initial investigation was conducted to verify that the current implementation of RTI would support the additional security provided by IPSec. This section does not attempt to demonstrate all the ideas described in the alternative approach, but simply attempts to demonstrate that RTI can natively support IPSec and attempts to measure the performance impact of IPSec on RTI.

To demonstrate that the current version of RTI supports an IPSec implementation without modification, a demonstration environment was constructed based on the IPSEC abilities built into the Windows2000 operating system. Two Windows2000 hosts, one server and one workstation, were configured with the available RTI implementation for 32-bit Windows. The server ran the RTIExec process and the workstation ran the sample RTI `Helloworld` federate, as shown in Figure 1.

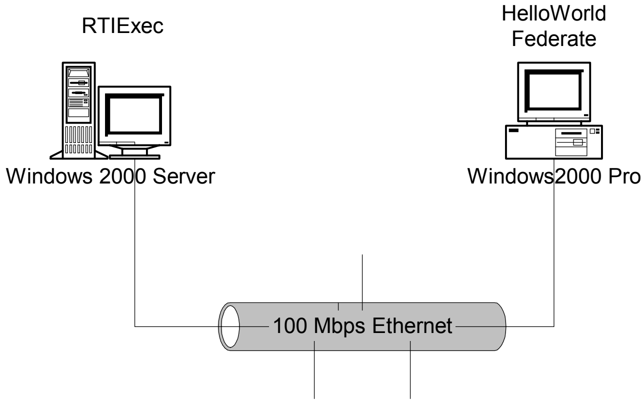

Figure 1: Network Configuration

A custom Windows2000 IPSec policy was created to demonstrate IPSec in Windows2000 and for the remaining tests. The steps outlined in the Step-by-Step Guide to Internet Protocol Security (IPSec) were followed to create a policy for the workstation and server. IPSec used the default Internet Key Exchange (IKE) method of Kerberos v5 authentication.

An IPSec policy called Partner policy was created on each machine. The policy used a custom filter, Partner Filter, and custom action, Partner Action, as part of the policy. A similar IPSec policy was created on the workstation to request security between the workstation and the server. This policy did not apply to the initial broadcast from the workstation to discover the RTIExec host, but applied to all subsequent communications, based on the rule.

Initially several tests were conducted to examine the network traffic caused by simply running the `HelloWorld` federate. The tests were conducted by executing the `HelloWorld` federate with an initial population of 100 and executing for 10 ticks.

Network traffic measurements were made with Microsoft's Network Monitor application. Number of packets captured and duration of the execution were collected. The numbers of packets represented the number of packets that were either broadcasts from the client workstation or packets sent between the server and the workstation. This was measured by starting the network capture, executing the federate, and stopping the capture. The filter was then applied to remove superfluous packets. The duration was measured as the time difference between the time of the initial broadcast from the client to locate the RTIExec and the final packet of the capture.

A sequence of tests was conducted with different number of federates and two machines shown in Figure 1. In addition, limited tests with more machines were conducted in preparation for more extensive experiments.

Table 1 provides simulation times for separate and simultaneous execution of federates, with or without IPSec on two machines. Each machine (Moltar and Arlel) had four federates running.

Table 1: Simulation Execution Times (in Seconds)

Federate			1	2	3	4
Moltar	Separ.	Normal	47.592	53.249	54.120	54.621
		IPSec	47.642	54.190	55.943	56.413
	Simult.	Normal	148.453	170.551	173.841	175.843
		IPSec	162.901	185.686	188.865	190.397
Arler	Separ.	Normal	106.729	117.382	120.286	120.306
		IPSec	108.661	120.446	121.748	122.139
	Simult.	Normal	175.207	175.933	176.764	176.815
		IPSec	188.835	189.496	189.666	189.771

Figure 2 provides a comparison of simulations times from Table 1 for eight federates (four on each machine) running separately. The increase of simulation duration due to use of IPSec is below ten percent.

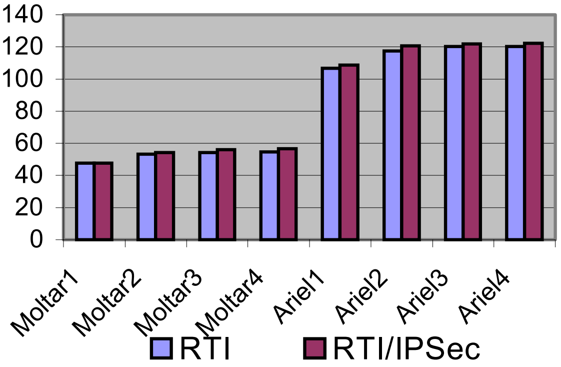

Figure 2: Simulation Times for Separate Federates

Figure 3 provides a comparison of simulations times for eight federates (four on each machine) running simultaneously. The increase of simulation duration is also below ten percent.

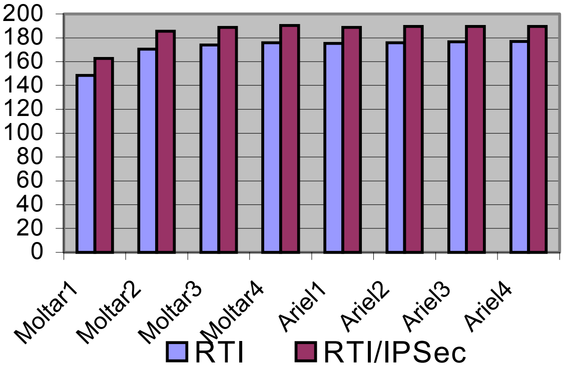

Figure 3: Simulation Times for Simultaneous Federates

5.1 Limitations of IPSEC for RTI

IPSec does not ensure the security of the host implementing IPSec. If proper physical security is not implemented, anyone with direct access could gain control of the machine. Once control of the machine is achieved, IPSec could be disabled or the authentication and encryption key could be stolen to allow 3rd parties to decrypt and access the data.

IPSec relies on the external authentication and encryption protocols. IPSec itself is a mechanism to negotiate security protocols. Any flaws in the negotiated protocol could be exploited to reduce the security provided by IPSec.

IPSec is going to have direct performance effects on simulations. Federate hosts are already burdened by the base operating system, networking stack, and federate execution itself, now is also going to be responsible for encryption prior to transmitting the data. Initial results with the HelloWorld federate showed a less than 10% increase in execution time. Results did show an increased impact as the size of the HelloWorld federate was increased.

IPSec is implemented using host level authentication and encryption. If an machine hosted multiple federates at different classification levels, the current implementation would not allow IPSec to be used to block communications between an higher level federate on one machine and a lower level federate on another, if that machine also had a federate that was of equal level to the first.

6 CONCLUSION

IPSec provides significant flexibility. In particular, it does not stipulate the authentication algorithms used in the IP Authentication Header (AH) protocol or the encryption and authentication algorithms used in the Encapsulating Security Payload (ESP) protocol. Microsoft® Windows® 2000 Server and Microsoft® Windows® 2000 Professional implements IPSec using DES for international use and 3DES for use in North America. Performance data should be obtained using a different encryption algorithm, such as the Advanced Encryption Standard, so that data can be compared with that obtained using 3DES.

The United States Government has stated, as policy, a preference to move away from traditional NSA-developed and produced COMSEC equipment and toward Commercial off-the-shelf (COTS) information assurance products (NSTISSC 2000). Effective on 1 January 2001, preference was given to COTS products that had been evaluated and validated to be compliant with appropriate national security requirements. Based on this direction, it is reasonable to assume that the marketplace will respond by offering COTS products that meet these requirements, and it is reasonable to assume that products that meet these stringent requirements would be preferred by the commercial sector. This action will blur the distinctions drawn in the operational scenarios above, by eliminating the NSA equipment and keying material.

The provided implementation only addresses the feasibility of using IPSec to provide security for RTI implementations. The questions raised in this paper provide ample areas of research. The immediate question that affects this project is support of multiple federates, specifically the HelloWorld federate, hosted by multiple Windows2000 machines and other machines. Second question is what is the effect of having multiple hosts implementing IPSec communicating with an RTIExec that supports IPSec. Medium range studies would include how to integrate a PKI solution into an RTI/IPSec implementation. This study should identify necessary RTI or FOM extensions to leverage the PKI solution. Longer-term studies would investigate having user certificate be used with IPSec and the impact of multiple different user federates hosted relying on IPSec on performance of a simulation. Another longer-term study could investigate a more intuitive approach to addressing need-to-know requirements.

REFERENCES

Alberts, D. S., J. J. Garstka, and F. P. Stein. 2000. *Network Centric Warfare: Developing and Leveraging Information Superiority*. 2nd ed. Vienna, VA: DoD C4ISR Cooperative Research Program (CCRP) Publication Series. Available online via <http://www.dodccrp.org/Publications/pdf/ncw_2nd.pdf> [accessed April 1, 2001].

Bieber, P., J. Cazin, P. Siron, and G. Zanon. 1998. Security Extensions to ONERA HLA RTI Prototype. In *Proceedings of the 1998 Fall Simulation Interoperability Workshop*. Paper number 98F-SIW-086. Available online via <http://www.sisostds.org/doclib/doclib.cfm?SISO_FID_1709> [accessed April 1, 2001].

Filsinger, J., and H. O. Lubbes. 1996. System Security Approach for the High Level Architecture (HLA). In *Proceedings of the 14th Workshop on Standards for Interoperability of Distributed Simulation (Winter)*. Available online via <http://ftp.sc.ist.ucf.edu/SISO/dis/workshop/14th/papers/067.msw> [accessed February 6, 2000].

IEEE. 2000. *IEEE 1516-2000, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules*. Piscataway, NJ: The Institute of Electrical and Electronics Engineers, Inc.

Kent, S., and K. Atkinson. 1998. RFC 2401: Security Architecture for the Internet Protocol. Available online via <http://www.ietf.org/rfc/rfc2401.txt?number=2401> [accessed April 1, 2001].

Kuhl, F., R. Weatherly, and J. Dahmann. 2000. *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*. Upper Saddle River, NJ: Prentice Hall PTR.

Microsoft. 2000. Step-by-Step Guide to Internet Protocol Security (IPSec). <http://www.microsoft.com/windows2000/library/planning/security/ipsecsteps.asp> [accessed: April 1, 2001].

NSTISSC. 2000. *NSTISSP No. 11* - Subject: *National Policy Governing the Acquisition of Information Assurance (IA) and IA-Enabled Information Technology (IT) Products*. Ft Meade MD 20755-6716: National Security Telecommunications and Information Systems Security Committee (I42), National Security Agency.

Ozdemir, H. T. 1997. Security Issues in High Level Architecture (HLA). <http://www.npac.syr.edu/projects/cps714fall97hw1/cj97hto> [accessed February 6, 2000].

AUTHOR BIOGRAPHIES

ASA ELKINS is a Senior Engineer with Integrated Data Systems. He is a Microsoft Certified Systems Engineer and SANS Level 2 Certified. He received his M.Sc. degree in Computer Science from Virginia Polytechnic Institute and State University. His email address is <elkinsa@home.com>.

JEFFREY WILSON is a Captain in the United States Navy, and is a Distance Learning Instructor and a Ph.D. student in Computer Science at Virginia Polytechnic Institute and State University in Falls Church, VA. His research interests are in real-time computing and networking. He is a member of IEEE, ACM, and Sigma Xi. His email address is <Wilson.Jeff@vt.edu>.

DENIS GRACANIN is an Assistant Professor of Computer Science at Virginia Polytechnic Institute and State University in Falls Church, VA. His research interests include virtual reality and distributed simulation. He is a member of AAAI, ACM, IEEE, SCS, and SIAM. His email address is <gracanin@vt.edu>.