**SANDIA REPORT**
SAND98–2787
Unlimited Release
Printed December 1998



# Final Report for the Scaled Asynchronous Transfer Mode (ATM) Encryption Laboratory Directed Research and Development Project

Lyndon G. Pierson and Edward L. Witzke

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.


Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

## DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SAND98-2787
Unlimited Release
Printed December 1998

# Final Report for the Scaled Asynchronous Transfer Mode (ATM) Encryption Laboratory Directed Research and Development Project

Lyndon G. Pierson, Edward L. Witzke
Advanced Networking Integration Department

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0806

## Abstract

This effort studied the integration of innovative methods of key management, crypto synchronization, and key agility while scaling encryption speed. Viability of these methods for encryption of ATM cell payloads at the SONET OC-192 data rate (10 Gb/s), and for operation at OC-48 rates (2.5 Gb/s) was shown. An SNL-Developed pipelined DES design was adapted for the encryption of ATM cells. A proof-of-principle prototype circuit board containing 11 Electronically Programmable Logic Devices (each holding the equivalent of 100,000 gates) was designed, built, and used to prototype a high speed encryptor.

## Acknowledgments

The authors wish to express their gratitude to the following individuals for their help in this project: Larry Pucket and Terry Hardin for their work on the design and layout of the PLD11, Jeff Ingle, Mark Bean, Brian Weeks, Troy Young, and Gerry Trombley of NSA for their collaboration, and Mike Sjulin and Mike Vahle for their support of this work.

The authors would also like to thank project members Robert L. Hutchinson, Perry J. Robertson, D. Craig Wilcox, Thomas D. Tarman, Richard J. Granfield, Luis G. Martinez, L. Byron Dean of Sandia National Laboratories, Karl Gass of Utah State University, Peter Sholander, formerly of Sandia National Laboratories, and Hans Rodriques de Miranda, formerly of RE/SPEC, for their contributions to this work.

Informal collaborative relationships were initiated with research efforts into related areas of interest at Motorola and at Fore Systems, Inc., which enriched the exchange of ideas regarding of this work.

## Contents

## Figures

# 1. Introduction

Different applications have different security requirements for data privacy, data integrity, and authentication. Encryption is one technique that addresses these requirements. Encryption hardware, designed for use in high-speed communications networks, can satisfy a wide variety of security requirements if the hardware implementation is key-agile, key length-agile, mode-agile, and algorithm-agile. Hence, context-agile encryption provides enhanced solutions to the secrecy, interoperability, and quality of service issues in high-speed networks [30][20]. Moreover, having a single context-agile encryptor at an ATM aggregation point (such as a firewall) reduces hardware and administrative costs. This work focussed on scaling of a single algorithm key-agile encryptor to achieve high data throughput.

Three encryption prototypes have emerged, representing the results of previous research efforts to scale encryption to high speeds. The Microelectronics Center of North Carolina (MCNC) "Enigma2" prototype ATM end-to-end encryptor [14] is previously the fastest known, operating at 622 Mb/s. Sandia's "Scalable ATM Encryptor Prototype"[21], developed in 1995, is the second fastest, operating at 155 Mb/s. The NSA "Milkbush" prototype encryptor [28] is the third fastest, operating at about 100 Mb/s. NSA's efforts have evolved into the production ATM end-to-end encryptor called "FASTLANE" [13], which operates at up to 622 Mb/s. In order to meet DOE/ASCI objectives, ATM encryptors for higher data rates (2 to 10 Gb/s and even higher) will soon be required. This research effort explored techniques required to scale encryption speeds to 10 Gb/s and faster, while maintaining reliable communications synchronization.

# 1. Background

Encryption rates have lagged behind increasing communication rates. This is primarily due to the complexity of the encryption process, requiring hundreds or thousands more instruction cycles (and/or more transistors) to encrypt each bit than the number of instruction cycles (or transistors) required just to communicate the bit without encryption. To prevent encryption from lagging behind communication rates requires parallel encryption methods. Using these parallel methods, less expensive transistors can be used to accomplish extremely high encryption rates. Thus, as high speed (and more expensive) transistors make higher and higher communication rates achievable, encryption rates can "keep up" with increasing communication rates.

Most methods of encryption can be scaled using parallel methods for operation at high speeds, but typically can only be decrypted by using an identically scaled decryption process. This has worked well in the past, in which both ends of a communication system operate at the same constant bit rate. Now, "Variable Bit Rate" (VBR) communication services are evolving which utilize the flow control through ATM switching systems to perform the "Rate Adaptation" between the end communication interfaces. In order to efficiently use the "Variable Bit Rate" communication services of new ATM/SONET networking technology, a method of scaling encryption for high speed is needed, but which will also interoperate with slower less expensive interfaces (implemented with much less parallelism). Furthermore, this encryption must be key agile, applying different keys to the different "virtual circuit" streams of data being interleaved through the channel to be encrypted to different destinations. The reliable, efficient and secure synchronization of the decryption process with the encryption process is an additional challenge to the practical deployment of extremely high speed encrypted communication systems.

Recent Sandia research and experience in high speed encryption has evolved methods of scaling encryption while maintaining interoperability of highly scaled and lesser scaled (and less expensive) implementations [21]. These methods involve the elimination of "feedback" around the encryptor's non-linear encryption function to allow interoperability of scaled and unscaled implementations. These "modes of operation" are to be combined with innovative "pipelining" of the cryptovariable context with the data being encrypted, so as to allow fast context switching without "emptying the pipeline". This fast context switching is required to achieve "key agility" at high data rates. An extremely efficient synchronization method, prototyped in other Sandia projects, was shown to be viable for encryption at the ATM/SONET OC-48 (2.5 Gb/s) rate and for application to encryption of ATM/SONET OC-192 (10 Gb/s) links [22].

The scalability of most of the techniques intended for application to this project has been prototyped independently. The combination of these techniques is expected to achieve a scalable, secure, reliable communication system. The algorithm scaled in this project was the Data Encryption Standard (DES) [8], since this slow, computationally complex algorithm is representative of a wide class of algorithms of interest.

Encryption can protect proprietary information as it passes from one end of a complex computer network to the other, even through untrusted intermediate systems, such as on the Internet. Encryption technology has many other uses including encrypting disk files and producing digital signatures. Efficient high speed communication systems, being of a real-time nature, often require encryption systems that optimize throughput while minimizing network traffic delay. Additional requirements may include minimizing error magnification, deterring message playback attacks, interoperability between faster and slower encryptors/decryptors, and quick recovery from cryptographic synchronization loss.

Just as different applications have different security needs, different users and communication sessions can have different needs. Symmetric end-to-end network encryption requires separate keys for each pair of communicating confidants. Each and any pair of communicating confidants can have multiple sessions (file transfer, virtual terminal, interprocess communication, etc.) proceeding simultaneously [35]. Each of these communication sessions therefore, can have different needs regarding session keys, cryptographic

robustness, and other encryption and communication characteristics. This requires fast context switching to encrypt different communication streams with different keys and characteristics for different destinations.

# 2. Architecture

This project has contributed extensively to the study of context agile encryptors as documented in [20] and [30].

The context-agile ATM encryption process resembles the ATM switching process. In particular, context-agile encryptors are similar to two-port ATM switches. For comparison, ATM switches modify cell headers and switch cells based on the "switching context" associated with each Virtual Path Identifier and Virtual Channel Identifier (VPI/VCI). The initial association of switching information with a virtual circuit may be a manual operation for Permanent Virtual Circuits (PVCs). The initial association might also occur automatically at connection setup time for Switched Virtual Circuits (SVCs). Then, for each incoming cell, the ATM switch performs an associative lookup, of switching information, based on the VPI/VCI found in each cell's header. This switching information maps the incoming VPI/VCI into the appropriate outgoing VPI/VCI. It also conditions the hardware to switch the cell out the proper port.

Context-agile ATM encryptors resemble ATM switches in that encryptors must retrieve information and make decisions based on the cryptographic context associated with each VPI/VCI. The initial association of the cryptographic variables, state, algorithm, etc. with each virtual circuit may be a manual operation or be performed at SVC connection setup time (or later) via the methods invoked for key management [31]. Once a cryptographic context is established for a virtual circuit, for each incoming cell, the encryptor performs an associative lookup of the cryptographic context, based on the VPI/VCI found in each cell's header. The encryptor then uses that cryptographic context to transform the incoming cell payload (plaintext or ciphertext) into the appropriate outgoing payload (ciphertext or plaintext). Finally, the encryptor typically routes the cell out the opposite port of a two-port device. Hence, in certain aspects regarding context lookup, signaling, and cell I/O, context-agile ATM encryptors resemble a two-port ATM-switch.

## *2.1 Key-Agile Encryption*

Key-agile encryption implementations (which are not also algorithm-agile) limit the context parameters to items such as key, initial variable, and present state. Key-agile software-implementations of cryptographic algorithms are usually straightforward. However, software-based encryption can raise both performance and security concerns. Hardware implementations provide higher performance; but, the efficient implementation of high-speed context switching in hardware is not as obvious.

Key-agile encryption hardware provides obvious benefits in both computer systems and high-speed communication networks. High-performance computers are often shared resources. Key-agile encryption allows each user on a workstation (or server) to use different key material. This cryptographically separates the users' traffic. High-speed network resources are also often shared resources. Indeed, each session, through a common network interface, may require separate keys. For example, an ATM encryptor may support end-to-end hardware encryption of multiple Asynchronous Transfer Mode Virtual Circuits (VCs) across a network operating at speeds from 155 Mbps (OC-3) to 10 Gbps (OC-192) and beyond.

Key-agile encryption also has some limitations. First, different systems may implement different security policies. Hence, a user that communicates with several other end-systems might need access to several different, shared key-agile encryptors. Robustness-agile encryptors (described in [20] and [32]) solve this problem. A second problem is that each ATM VPI/VCI (Virtual Path Identifier/Virtual Channel Identifier) combination (virtual circuit) has a different cryptographic context associated with it. Hence, any context-agile encryptor, whether key-agile or robustness-agile, must be able to switch encryption contexts very quickly as the ATM cells associated with various virtual circuits arrive for processing. Context-switching is further discussed later in a later section of this document.

Some commonly used encryption algorithms do not scale well for high speeds. This includes the class of encryption functions with feedback based on a combination of key and plaintext or ciphertext. These do not scale (in a parallel manner) *and* interoperate with units having different degrees of parallelism [21]. An example of such an algorithm in wide-spread use is DES in cipher block chaining (CBC) mode [10]. Filter generator [26] algorithms have been found to both scale well for high speed operation and to interoperate with unscaled or lesser scaled implementations [21]. An example of such an algorithm is DES in counter mode [27] [1].

## 2.2 Large Context Space

The number of potential encryption contexts (VPI/VCIs) and the amount of information per context may both be large. In general, there may be either $2^{24}$ possible User Network Interface (UNI) VPI/VCI combinations or $2^{28}$ possible Network-to-Network Interface (NNI) VPI/VCI combinations. Hence, implementing the cryptographic-context lookup with "straight indexed" (flat) memory may be too costly. Because the number of simultaneously active contexts is likely to be small, an efficient key-agile encryptor could use an associative memory lookup to determine the key and other cryptographic state information, associated with each cell stream. Clearly, encryption algorithms that must associate larger keys and greater state information will be more cumbersome (expensive) to implement than algorithms that require a minimum of key and state information. In either case, large content addressable memories (or the even larger sequential memories required) with access times on the order of ATM cell header processing times are expensive and/or unavailable. Hence, until large, inexpensive, and fast content addressable memories do become available, current designs compromise either the virtual circuit space over which circuits can be encrypted, or the cell processing latency, or both.

## 2.3 Cryptographic Synchronization

In order to protect against dictionary lookup and playback attacks, encryption modes that maintain a "cryptographic state" are employed. These modes require synchronization of the cryptographic state between the encryption and decryption processes. When an encryptor and decryptor pair have lost synchronization, the decrypted data stream is scrambled, which leads to excessive data loss. While some cryptographic algorithms or modes of operation are "self synchronizing", others require both initial synchronization and resynchronization after each cell loss. (Various specific methods of synchronization are addressed elsewhere [21] [30].) Since each virtual circuit has independent synchronization, the synchronization state information adds to the amount of information that must be associatively maintained for each encrypted virtual circuit.

## 2.4 Throughput

If the encryption or decryption process cannot keep up with the maximum possible cell arrival rate, then the cell traffic throughput on that virtual circuit must be throttled in some fashion to avoid cell loss. This can be done via Call Admission Control (CAC) at virtual circuit setup time (for constant, variable, and unspecified bit rate traffic) or by participation in the flow control after virtual circuit setup (for available bit rate traffic). In either case, the encryption/decryption devices must participate in the establishment and/or control of the VC, making it no longer "transparent" to the switching network.

## 2.5 Encryptor Architecture

The general architecture of a context-agile, Asynchronous Transfer Mode encryptor, addressing the issues described above, was jointly developed in collaboration between researchers at Sandia and at NSA [20]. The architecture is scalable in data rate, but is targeted at a proof-of-concept implementation operating at 10 Gbps (OC-192).

The architecture consists of the following major components. There will be both an input and an output physical I/O module, a cell Identification and Association Module, a Cell Router, one or more Cryptographic Modules, a Cell Combiner, a Key Management Module, and a Non-Real-Time Control Module. Figure 1 shows the connectivity relationship among the modules.

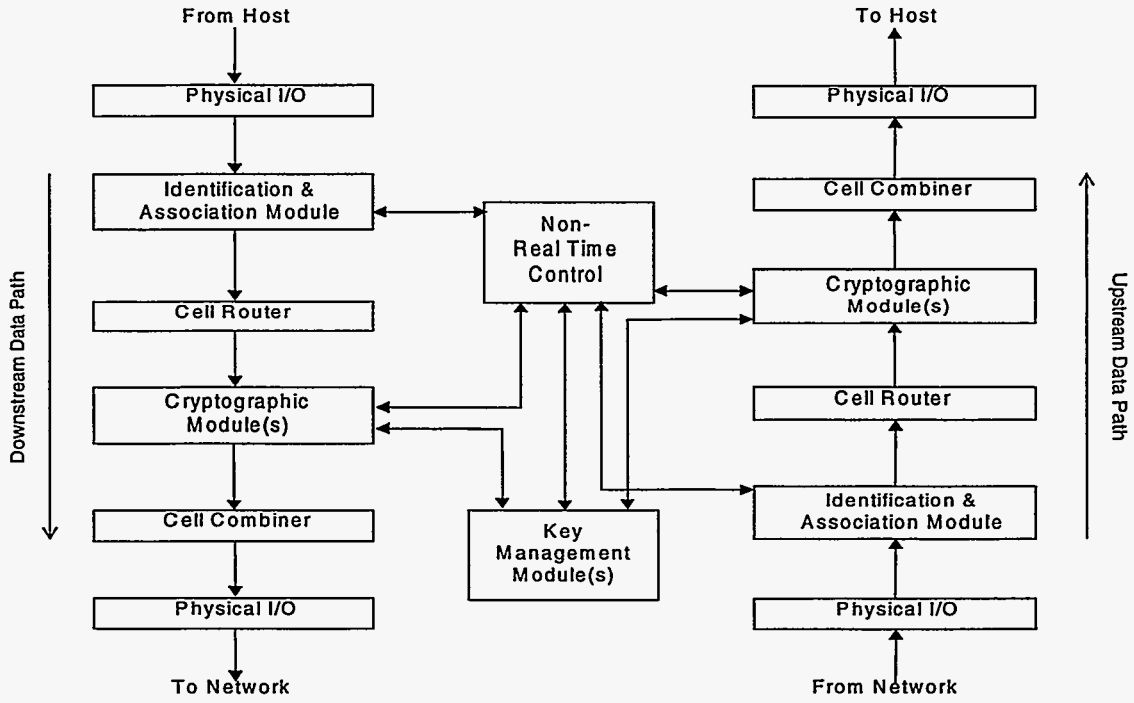


FIGURE 1. TOP LEVEL ENCRYPTOR ARCHITECTURE.

These modules or functions can be broken down into real-time and nonreal-time services. The separation of real-time and nonreal-time traffic is critical to the overall system design. Specifically, the system must be designed to accommodate all real-time traffic in a very efficient manner such that excessive delays are not experienced. All user traffic is handled in real-time through the physical I/O, identification and association, and cryptographic services. The nonreal-time services consist of network management functions such as signaling and certain types of OAM (Operation, Administration and Maintenance) processes. The real-time data path is referred to as the high-speed path, and the nonreal-time data path is referred to as the lower-speed path. A distinction is also made between general, ATM-specific functions and security services. The ATM specific functions can be implemented in a shell, which surrounds the security services. The purpose of the shell is to provide a generic, non-proprietary definition for a large portion of the encryptor. This may lead to commercially available devices into which a generic or custom (e.g. proprietary) security module may be inserted. Figure 2 shows the functional blocks, the shell boundaries, and the separation of real-time and nonreal-time functions.

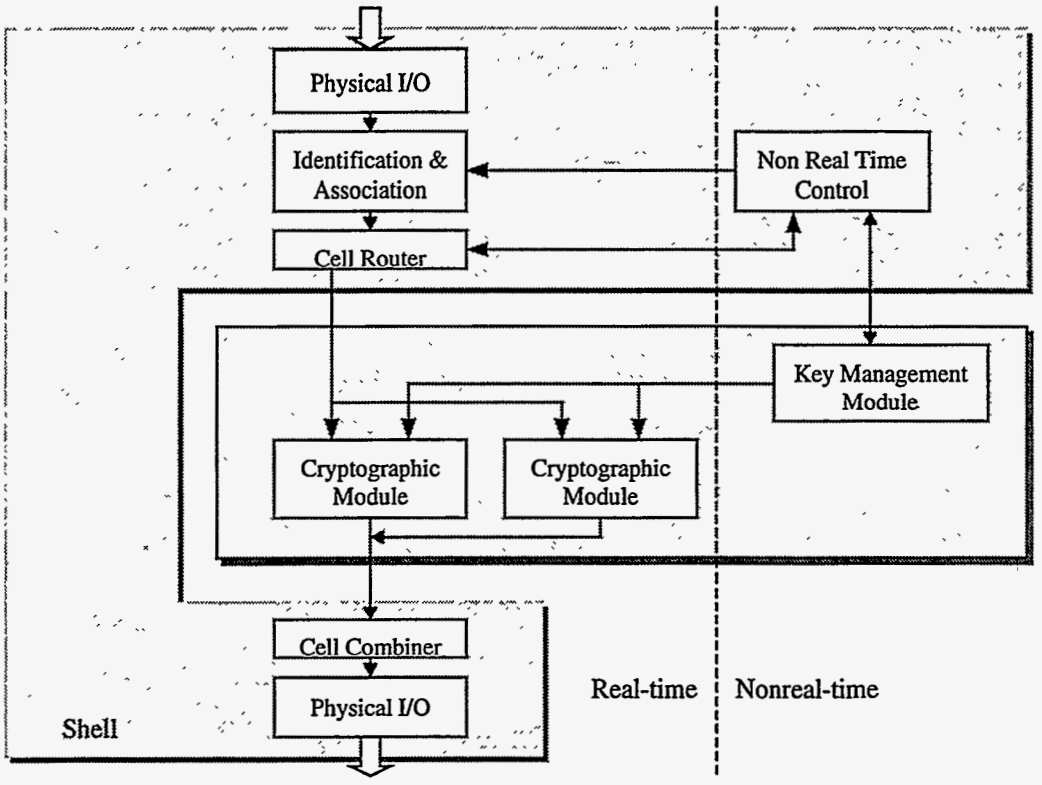


FIGURE 2. SHELL AND SECURITY MODULE RELATIONSHIPS.

### 2.5.1 The Shell

The primary data path (that path that carries user data cells) through the shell is from the physical I/O through the identification and association function to the cell router. User traffic continues through the cryptographic module (outside the shell), and is returned to the shell at the cell combiner. Finally, the physical I/O output port provides data to the network. Additional support data paths include interaction between the nonreal-time control and the identification and association function for the purpose of maintaining connection and association information. Additionally, nonreal-time cells such as signaling cells are diverted from the real-time path at the cell router to the nonreal-time control for processing. Similarly, the nonreal-time control may insert cells via the cell router. Each functional block is discussed in the following sections.

For this project, the components shown in the shell have only minimum functionality. Scalability efforts have been directed primarily at the Security Module components. A thorough discussion of the shell components for the general case can be found in [20].

### 2.5.2 Content Addressable Memory

One approach for connection table lookup, is to use a Content Addressable Memory (CAM). This associative memory is the most efficient method to perform the table lookup function because the CAM needs to be only as deep as the number of connections to be identified, plus a select set of generic cells common to all connections such as cryptographic resynchronization cells. Content Addressable Memory is used by loading the search VPI and VCI fields into any unused location in CAM. At this time, the appropriate association information is loaded into a RAM corresponding to the now filled CAM location.

(The CAM finds the correct location when a cell's VPI and VCI is presented. The corresponding RAM then outputs the appropriate association information.)

CAM memory associations limit the number of connections due to limitations in CAM depth. Conventional SRAM does not limit the number of connections, but does restrict the size of the VPI and VCI field used for the address. In short, CAM is optimal for very wide associative lookups whereas RAM is preferred for narrower, deeper lookup tables On the surface, the advantage of the CAM approach may not be obvious. However, consider that each connection (fixed VPI/VCI) contains user data as well as OAM cells. The OAM cells require very different processing than the user data cells. The use of a ternary (1,0, don't care) CAM is more efficient at identifying specific types of cells on a given connection than a conventional RAM approach. Using CAM, there may be a single CAM entry identifying a cryptographic resynchronization cell (for all connections) and a single entry for each specific connection. When the CAM matches on both entries, it has identified the cell as being associated with a given connection as well as a specific cell type. Using a RAM approach, either a resynchronization entry is required for every connection (dramatically increasing memory size and address width requirements), or a secondary cell type lookup would be necessary.

Although a CAM implementation is more efficient, currently, there are no known CAM devices that will support operation at 10 Gbps. Therefore, although architecturally CAM would be the preferred technology, to construct a proof-of-concept implementation in the near term, Synchronous RAM and a reduced VPI/VCI table lookup approach may be used for the identification and association function.

### 2.5.3 Security Module

There are two primary cryptographic functions that the security module of an ATM encryptor must perform. First, key management must be provided for generating and maintaining traffic encryption keys. For the purposes of this discussion, it is assumed that the key management functions are performed independently of the encryptor itself. Commands to update the context memory are communicated from the key management entity to the encryptor via a protected channel. Second, the network device must perform confidentiality services, i.e. encryption/decryption. Further security service requirements such as traffic flow security and authentication are handled by the nonreal-time controller through ATM signaling or by other higher layer entities/applications.

#### 2.5.3.1 Confidentiality Services

Depending on the specific mode of operation, providing confidentiality services requires five basic components: key generator, key/plaintext mixer, state vector (SV) memory, cryptovariable (CV) memory and a cell processor. Ongoing exploration into Application Specific Integrated Circuit (ASIC) technology, Field-Programmable Gate Array (FPGA) technology, I/O capabilities, printed circuit boards and multi-chip modules will dictate the proper partitioning of these components. For the purposes of this discussion, it will be assumed that each component is a discrete ASIC, FPGA, or COTS device, as applicable. Figure 3 shows the relationship of these components.

The cell processor must provide processing on a per cell basis in real time. Fundamentally, the cell processor must perform one of two functions for each cell. Real-time cells are categorized as security related OAM cells (e.g. resync cells) or user data cells. The cell processor must process the cell differently based on the cell type. If it is a user cell, the cell processor must provide data appropriately to the key generator and mixer for proper encryption/decryption. If the cell is a resynchronization cell, OAM cell processing must be performed. The processing of the resync cells is described in the ATM Forum Security Specification [1], and is addressed subsequently.

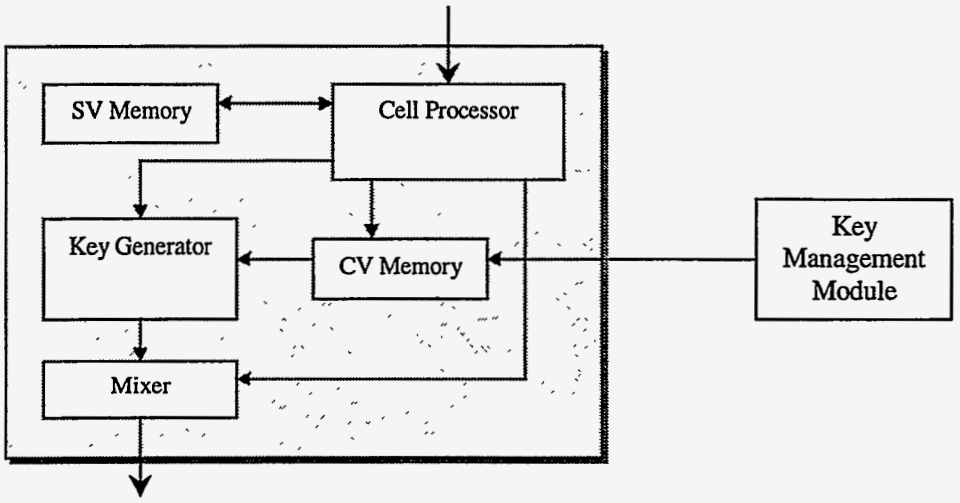


FIGURE 3. CRYPTOGRAPHIC MODULE

#### 2.5.3.1.1 Encryption/Decryption

In order to effectively encrypt or decrypt data at very high speeds, a non-feedback mode of operation is desired [36]. The two modes best suited for this application are Counter Mode and Electronic Codebook Mode. The following discussion assumes Counter Mode, as Electronic Codebook Mode does not protect against dictionary lookup or replay attacks.

User data cells are placed in a FIFO buffer while the key stream is generated. First, the cell processor fetches the current SV from memory. This memory access is a second look-up as opposed to the initial Identification and Association look-up. The correct SV is selected based on the context information, which was passed to the cell processor along with the cell header (as described previously). Then, the SV is written to the key generator, updated and written back to SV memory. Assuming a 128 bit word-width interface to the key generator, the cell processor must update and write the local SV a minimum of three times to cover the length of an ATM cell payload. (Cell headers are never modified as they are required by the network.) Only the final value of SV is written back to SV memory.

Due to the cryptographic sensitivity of the CV, sometimes referred to as "key" [27], the cell processor only provides an index, or address, to the CV memory and never handles the CV data. Instead, the data is fed directly to the key generator. The cell processor is responsible for properly synchronizing the SV and CV data arrival times at the key generator. Finally, the key stream generated by the key generator is passed to the key/plaintext mixer along with the user data cell payload, which was buffered in a FIFO. The resultant mixer output is ciphertext, which is appended to the original, unmodified cell header to form the encrypted ATM cell.

#### 2.5.3.1.2 Cryptographic Resynchronization

In addition to user cell processing, the cryptomodule is required to process cryptographic information. Specifically, cryptographic resync cells must be extracted from and inserted into the cell stream, and processed accordingly. In order to eliminate bottlenecks and resulting backpressure on network traffic, all resync cell processing must be performed within a single cell period of 42.6 ns, for OC192 data rates [7]. Further, the cryptomodule must process two types of resync cells -- those destined for the cryptomodule (cell reception) and those originating from within the encryptor (cell insertion.)

The insertion of resync cells requires stepping the current SV to the new value and writing the new SV to memory and into the resync cell. Stepping the SV is described in the ATM Forum Security Specification [1]. It involves incrementing the jump number, setting the I/R (Initiator/Responder) bit accordingly, resetting the Sequence Number and Segment Number to all zeros and the LFSR (Linear Feedback Shift Register) to its preset value. The new SV is then written to the appropriate SV memory location and to the resync cell. In addition, a CRC-10 is computed and inserted at the end of the cell. The new SV stored in memory is used on the next cell arriving on the given connection.

The reception of resync cells requires verification of the CRC-10. Similarly, the jump number is checked to verify that it is greater than the current jump number. If the CRC-10 value or jump number is invalid, the cell is ignored and no resynchronization occurs. If both values are valid, the jump number is extracted and the other SV fields are reset. The new SV is stored in SV memory and is ready for use by the next cell on the given connection.

#### 2.5.3.2 Key Management

As in the nonreal-time control, the Key Management Module provides support services to the encryptor. The exact implementation is beyond the scope of this report. However, the following details are considered relevant to the architectural discussion.

The primary purpose of the Key Management Module to the encryptor is to generate a traffic encryption key (TEK) for each connection. The TEK is also known as the cryptovariable (CV). The security message exchange protocols are described in [1]. The security messages exchanged are necessary for the generation of a properly authenticated TEK. However, the exact algorithm or method of generating the TEK is not specified and may be proprietary.

As has been mentioned, the CV is critically sensitive information. Therefore, the visibility of this data is limited as much as possible. To this end, there is a protected path from the key management module to the cryptographic module. The scope of "protected path" is left to implementation. In some situations, such as if the key management services were provided remotely, the protected path may be cryptographically protected (i.e. encrypted). However, for local key management, a protected path may dictate a dedicated point-to-point connection between the key management module and the cryptographic module. For this discussion, we will consider the latter case.

The cryptomodule interface is directly connected to a dual port memory device. The key management module controls one read/write port, while the cell processor of the cryptomodule controls the other read-only port. Again, the motivation for a read-only port is to limit the exposure and possible modification of the CV data. While protecting the CV data is one issue, protecting the address of this data also presents unique security and assurance concerns. Ultimately, both the key management module and the cryptographic module receive the CV address from a single source, the nonreal-time control. The validity and trustworthiness of the address is directly related to the trustworthiness of the nonreal-time control subsystem.

# 3. Scaling Encryption Speed

In order to achieve high throughput encryption, the basic encryption algorithm or "nonlinear block cipher" must be accelerated, and the way the block cipher is used to encrypt data, called the "encryption mode of operation" must also be accelerated.

## *3.1 Accelerating the Nonlinear Block Cipher*

The Data Encryption Standard (DES) algorithm [8] was chosen for acceleration. This algorithm consists of 16 stages or "rounds" of interleaved nonlinear and linear operations. The DES has been studied extensively in the literature and is considered representative of "heavy-weight" algorithms. The fastest hardware implementation of the DES prior to this work was a Gallium Arsenide (GaAs) device made by Digital Equipment Corporation. The throughput of this device was reported to be 1 Gb/s. This implementation was not "pipelined", i.e., it iterated the data to be encrypted 16 times through a single "round" of hardware. Only a few were made in order to test the GaAs integrated circuit process intended for the manufacture of DEC's ALPHA computer processor. The few DEC DES chips that were made were used as part of the ENIGMA2 encryption prototype built by the Microelectronics Center of North Carolina (MCNC).

A pipelined design was chosen for acceleration of the DES. The algorithm was first implemented in the form of an EXCEL "spreadsheet" so that the proper functioning of the full algorithm could be studied. This also provided an opportunity to verify that the description of the nonlinear "S-boxes" were implemented properly by checking the operation of the spreadsheet on several known input/output test vectors. The description of the operation of these and other parts of the DES algorithm were then "cut" and "pasted" into a hardware description language for implementation in electrically programmable logic devices (EPLDs) and later, in a Complementary Metal-Oxide on Silicon (CMOS) Application Specific Integrated Circuit (ASIC).

The 16 stages of the fully pipelined DES design required four of the largest then available programmable logic devices. At the time this work was done, this required four Altera, Inc. 10K100 EPLDs, each containing reconfigurable logic equivalent to 100,000 gates (four rounds in each device). Simulations of this design showed that it would operate with a 50 ns clock cycle, yielding a throughput of 1.3 Gb/s. The pipelined design is shown in Figure 4.

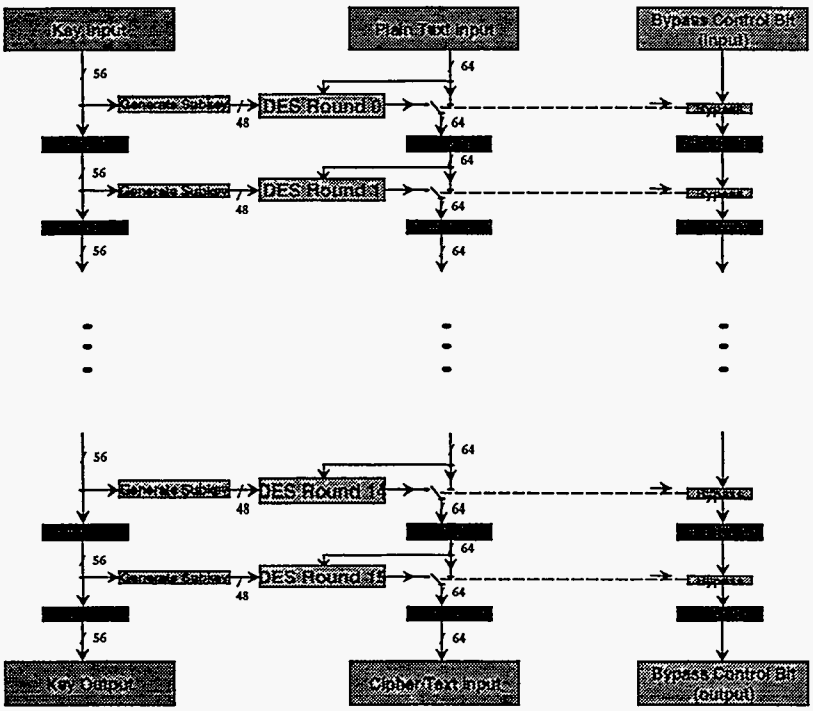


FIGURE 4. PIPELINED IMPLEMENTATION OF DES

## 3.2 Accelerating the Encryption Mode of Operation

Previous Sandia work [21] has shown that modes of operation which involve feedback around the nonlinear block cipher cannot be scaled and still interoperate with implementations employing different degrees of parallelism. The interoperation of implementations of differing degrees of parallelism is especially important for fast encrypting computer interfaces to communicate with lesser cost lower speed encrypting interfaces through the rate adaptation of an ATM network. Of the four most common encryption modes of operation defined in FIPS 81 [10], only Electronic Codebook (ECB) mode meets this criteria, but ECB mode fails to protect against "dictionary lookup" and "playback attacks" [21]. Counter Mode, (not defined in FIPS 81) involves no feedback around the nonlinear block cipher, yet protects against these attacks providing the "counter" involved has a long period. This research effort contributed heavily to the definition of a standard "Counter Mode" in ATM Forum's Security 1.0 specification. Even though variations of counter mode have been implemented and described in the literature, the Counter Mode specified in the ATMF Security 1.0 is the first documented "standard" implementation agreement to which multiple vendors can build interoperable Counter Mode encryption equipment.

Counter Mode has been found to scale well for high speed, yet also will interoperate with implementations employing different degrees of parallelism. This mode, described in general in [21], [27], and specifically in the ATMF Security 1.0 document [1], can be adapted to the DES encryption of ATM cells by encrypting 64, 128, 192, or 384 bits of the cell payload at a time in parallel. Since the ATM cell payload is 384 bits in length, encrypting more than 384 bits at a time involves the processing of multiple cells in parallel. Uncertainties in the arrival times of multiple cells and the potential for interleaving of cells from different virtual circuits to be encrypted with different keys make the processing of multiple cells in parallel difficult if not infeasible. Therefore, this project considered factors of parallelism only up to six 64 bit "parallel slices" (one entire 384 bit cell payload).

In order to achieve 10 Gb/s encryption throughput of ATM cells, and if a maximum parallelism of six 64 bit parallel encryption streams is to be used, then each of the 64 bit wide encryption streams must operate with a throughput of approximately 10/6 = 1.7 Gb/s.

## 3.3 Scaling for Low Cost at Low Speed

While Counter Mode has the advantages of being scalable and interoperable with dissimilarly scaled implementations while protecting against dictionary and playback attacks, it requires explicit synchronization of the decryption process with the state of the encryption process. Another mode of operation, Cipher Block Chaining, defined in FIPS 81 [10], lacks scalability and interoperability, but is self-synchronizing. Self-synchronization is an advantage that allows CBC encryptor implementations to be scaled to very low cost at very low speeds. The cost of explicit synchronization circuitry for Counter Mode, while small, cannot be eliminated from low speed low cost implementations. That a single mode of operation (such as Counter Mode) does not span the entire "cost-performance space" is one reason for the importance of "algorithm-agile" encryptors and is the subject of a separate SAND report [32].

CBC mode requires feedback around the nonlinear block cipher. For computation of the CBC mode of encryption operation [10], the result of encrypting the previous 64 bit word must be available (to be exclusive-or'ed with the next 64 bit input word) before the next input can be fed into the encryptor. This means that for CBC mode, the pipeline must be run "dry", with each input word fully clocked through all the pipelined stages before the next word can be input to the pipeline. In this situation, the encryption throughput of a pipelined nonlinear block cipher implementation would be 1/16 of the normal full pipeline throughput.

To accelerate the computation of CBC mode, it is noted that if all the pipeline latches of the synchronous pipeline implementation described above were removed from the design, an "asynchronous waterfall" of the 16 rounds of linear and nonlinear logic would be formed. Simulation of this alternate design showed that the 16 round encryption computation could be performed in 650 ns, shaving 150 ns from the time of 800 ns required by the synchronous pipeline design. For CBC mode in the EPLD implementation simulated, this would result in a modest throughput increase from 80 Mb/s to 98.5 Mb/s.

## 3.4 PLD11 Implementation

For the purpose of prototyping portions of a high speed encryptor, a general purpose reconfigurable logic circuit board was designed and built. This circuit board, called the PLD11 Multipurpose Programmable Logic VME board, is described in a separate SAND report [25].

The PLD11 board is a 9U VME board containing 11 Altera 10K100 Programmable Logic Devices, a controlled impedance clock tree, a VME interface, a programming interface, an OC3 (155 Mbps) interface and a serial port. The 11 Altera 10K100 Programmable Logic Devices are arranged to provide four 96 bit wide buses for a total of 384 parallel digital data lines in and out of the board that can operate up to 100 Mhz for an aggregate throughput of 38.4 Gb/s. The 14.44" X 15.75" board has over 1.1 million programmable gates that can be programmed through a serial interface. The board contains a clock reference and 50 ohm clock distribution tree that can drive each of the eleven 10K100 devices with two critically timed clock references. Five external clock references can be used to drive five additional PLD11 boards for a total of six boards operating all from the same synchronous clock reference. A system of six boards provides just under 7 million programmable gates.

The pipelined DES EPLD design described above was subsequently implemented by another project in 0.6 micron CMOS technology in Sandia's Microelectronics Development Laboratory foundry. Simulations of the performance of this ASIC indicate that it can achieve approximately 8 Gb/s if properly cooled. Actual tests have shown this ASIC to operate at 6.7 Gb/s, limited by the maximum clock rate of the test equipment. This ASIC was then bonded out to a printed circuit board providing an Altera 10K100 pin-compatible form

factor. These pipelined DES ASICs were then used in place of 10K100 parts on the PLD11 board to demonstrate higher throughput encryption than could be provided by the EPLD implementation.

# 4. Conclusions

Several key-agile encryptor prototypes, such as the Milkbush [28], Enigma2 [14], and Scalable ATM Encryptor [21] have been built. Single algorithm, key-agile encryption products are becoming available (CellCase [15], FASTLANE [13]). A general problem is the unavailability of fast content-addressable memories

This effort has developed a DES ASIC capable of operation at over 6.7 Gb/s. This showed how computationally complex, "heavy-weight" algorithms can be sped up, while retaining the ability to be scaled further by the use of several ASICs in parallel.

This work has also produced a prototyping board for reconfigurable hardware, that has the possibility of being used by other projects. This board can house over 1.1 million gates of programmable logic, has 96 bit wide data paths between programmable logic devices, contains a common clock distribution, and can be "ganged" together with other like boards to implement multi-million gate hardware designs.

This effort influenced and accelerated the deployment of scalable, variable bit rate Asynchronous Transfer Mode (ATM) encryption equipment needed to satisfy the Department of Energy (DOE) Accelerated Strategic Computing Initiative (ASCI) requirements for communication security. In particular, GTE, Inc. has started development of a new member of the TACLANE/FASTLANE family of encryptors. The GTE UltraFASTLANE will have an initial operational capability of 2.5 Gb/s (OC-48c), with an upgrade path to 10 Gb/s (OC-192c) as the electro-optic interfaces become available. As a result of the collaboration between this LDRD and NSA/R2, LDRD project personnel have been invited to participate in design reviews of the GTE UltraFASTLANE encryptor.

Further development of this technology is expected to: 1) Enable national defense applications requiring the secure exchange of massive amounts of data between widely separated sites, 2) Enable the interoperation of low cost, "low" speed encryptors for High Performance Workstations and super-high-speed encryptors for Massively Parallel Processing (MPP) computers, and 3) Significantly reduce the cost of encryptors for industry standard ATM/SONET data transmission interfaces, presently costing $30k per encryptor.

# 5. Bibliography


1. The ATM Forum Technical Committee, ATM Security Specification Version 1.0, Straw Ballot, STR-SECURITY-01.00, The ATM Forum, Mountain View, CA, December 1998.

2. The ATM Forum Technical Committee, Scaleable Parallel Interface for UTOPIA, ATM97-0537, ATM Forum, Mountain View, CA, December 1997.

3. The ATM Forum Technical Committee, *Traffic Management Specification, Version 4.0*, af-tm-0056.000, The ATM Forum, 2570 West El Camino Real, Suite 304, Mountain View, CA, April, 1996.

4. The ATM Forum Technical Committee, *UNI 4.0 Security Addendum*, ATM Forum BTD-SIG-SEC-01.00, The ATM Forum, 2570 West El Camino Real, Suite 304, Mountain View, CA, February, 1997.

5. The ATM Forum Technical Committee, *UTOPIA Specification, Level 1, Version 2.0*, ATM Forum af-phy-0017.000, The ATM Forum, 2570 West El Camino Real, Suite 304, Mountain View, CA, March, 1994.

6. The ATM Forum Technical Committee, *UTOPIA Specification, Level 2, Version 1.0*, ATM Forum af-phy-0039.000, The ATM Forum, 2570 West El Camino Real, Suite 304, Mountain View, CA, June, 1995.

7. Bean, Mark O., et al., Functional Architecture for a 10 Gigabit per Second Context-Agile ATM Encryptor, R2 Technical Report R22-002-98, National Security Agency, Ft. Meade, MD, February 1998.

8. Data Encryption Standard (FIPS PUB 46), Federal Information Processing Standards Publication 46, National Bureau of Standards, Washington, D. C., January 15, 1977.

9. Denning, Dorothy Elizabeth Robling, Cryptography and Data Security, Addison-Wesley, Reading, MA, 1982.

10. DES Modes of Operation (FIPS PUB 81), Federal Information Processing Standards Publication 81, National Bureau of Standards, Washington, D. C., December 2, 1980.

11. Diffie, Whitfield, and Martin E. Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, Vol. IT-22, No. 6, p. 644-654, November 1976.

12. Ellison, Frantz, and Thomas, *Simple Public Key Certificate*, Internet Draft, Internet Engineering Task Force, March, 1997.

13. http://www.nsa.gov:8080/programs/missi/kg75.html, June 1998.

14. http://www.mcnc.org/HTML/ITD/ANR/Enigma2.html, November 1996.

15. http://www.secantnet.com/product1.html, January 20, 1998.

16. The International Telecommunications Union, *B-ISDN DSS2 User-Network Interface Layer 3 Specification for Basic Call/Connection Control*, Recommendation Q.2931, February, 1995.

17. Kahn, David, The Codebreakers, Macmillan, New York, 1967.

18. Menezes, Alfred J., et al., Handbook of Applied Cryptography, CRC Press, Boca Raton, FL, 1997.

19. Peravian, Mohammad, and Tom Tarman, "Asynchronous Transfer Mode Security," IEEE Network Magazine, Vol. 11, No. 3, pp. 34-40, May/June 1997.

20. Pierson, Lyndon G., et al., "Context-Agile Encryption for High Speed Communication Networks," SAND98-1978J, to be published.

21. Pierson, Lyndon G., et al., Scalable End-to-End Encryption Technology for Supra-Gigabit/second Networking, SAND94-1622, Sandia National Laboratories, Albuquerque, NM, April 1997.

22. Pierson, Lyndon G., and Joseph H. Maestas, *Efficient Synchronization of ATM End-to-End Encryptors,* Sandia National Laboratories Technical Advance SD-5993, April, 1997

23. L. G. Pierson and E. L. Witzke, *Multiply-Agile Encryption in High Speed Communication Networks,* SAND97-1069C, Sandia National Laboratories, Albuquerque, NM, April, 1997.

24. Rivest, R. L., et al., "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Communications of the ACM, Vol. 21, No. 2, p. 120-126, February 1978.

25. Robertson, Perry J., et al., Final report and Documentation for the PLD11 Multipurpose Programmable Logic VME Board Design, to be published, Sandia National Laboratories.

26. Rueppel, Rainer A., "Stream Ciphers," in Gustavus J. Simmons (ed.), Contemporary Cryptology: The Science of Information Integrity, IEEE, New York, 1992.

27. Schneier, Bruce, Applied Cryptography, 2nd edition, John Wiley & Sons, New York, 1996.

28. Semancik, William, et al., "Cell Level Encryption for ATM Networks and Some Results from Initial Testing," Conference Proceedings, DoD Fiber Optics '94, March 1994.

29. Sholander, Peter, et al., "The Effect of Algorithm-Agile Encryption on ATM Quality of Service," GLOBECOM 97, IEEE, Piscataway, NJ, November 1997.

30. Tarman, Thomas D., et al., "Algorithm-Agile Encryption in ATM Networks," IEEE Computer, Vol. 31, No. 9, pp. 57-64, September, 1998.

31. Tarman, Thomas D., et al., Final Report for the Protocol Extensions for ATM Security Laboratory Directed Research and Development Project, SAND96-0657, Sandia National Laboratories, Albuquerque, NM, March 1996.

32. Tarman, Thomas D., et al., Final Report for the Robustness-Agile Asynchronous Transfer Mode (ATM) Encryption Laboratory Directed Research and Development Project, SAND97-2902, Sandia National Laboratories, Albuquerque, NM, November 1997.

33. Trombley, G. J. and M. O. Bean, Technology Trends Influencing High-Speed INFOSEC Requirements, R2 Technical Report R22-003-98, National Security Agency, Ft. Meade, MD, February 1998.

34. Wilcox, D. Craig, DES ASIC Design, to be published, Sandia National Laboratories.

35. Witzke, Edward L., and Lyndon G. Pierson, "Key Management for Large Scale End-to-End Encryption," Proceedings, 28th Annual International Carnahan Conference on Security Technology, IEEE, New York, October 1994.

36. Witzke, Edward L., and Lyndon G. Pierson, "The Role of Decimated Sequences in Scaling Encryption Speeds Through Parallelism," Conference Proceedings of the 1996 International Phoenix Conference on Computers and Communications, IEEE, New York, 1996.

**Appendix I: LDRD Data**

This effort was funded by Sandia National Laboratories' LABORATORY Directed Research and Development program under Case 3512220000.

Awards: 1998 VeriBest Superior Systems Award for Most Unusual Design (PLD11 Board)

Publications:

Pierson, Lyndon G., and Edward L. Witzke, Mark O. Bean, Gerry J. Trombley, "Context-Agile Encryption for High Speed Communication Networks," SAND98-1978J, to be published.

Robertson, Perry J., Robert L. Hutchinson, Lyndon G. Pierson, Thomas D. Tarman, Edward L. Witzke, Final report and Documentation for the PLD11 Multipurpose Programmable Logic VME Board Design, to be published, Sandia National Laboratories.

Tarman, Thomas D., and Robert L. Hutchinson, Lyndon G. Pierson, Peter E. Sholander, Edward L: Witzke, "Algorithm-Agile Encryption in ATM Networks," IEEE Computer, Vol. 31, No. 9, pp. 57-64, September, 1998.

In addition, this project contributed heavily to the ATM Forum's Security Specification 1.0 [1], a document addressing many security issues in ATM networks, including the first definition of a standard, interoperable "Counter Mode" for DES encryption. Project member Thomas D. Tarman served as the editor of this specification.

Patents (applied for or issued): In preparation, based on Technical Advances SD-6088, SD-6238, and SD-6120 below.

Technical Advances:

Robertson, Perry J., and Edward L. Witzke, "General Purpose Programmable Accelerator Board (PLD Implementation)," SD-6088, August, 1997.

Pierson, Lyndon G., and Perry J. Robertson, D. Craig Wilcox, Edward L. Witzke, "10 Gb/s Triple DES Module," SD-6238, August, 1998.

Robertson, Perry J., and Lyndon G. Pierson, Robert L. Hutchinson, "Flexible Programmable Logic Board (PLD11), " SD-6120, November, 1997.

Copyrights (for Software): N/A

Employee Recruitment: N/A

Student Involvement: This project supported student effort at New Mexico Institute of Mining and Technology under the direction of Professor Victor Yodaiken.

Follow on Work (new activities, projects , sponsors): Collaborations from this project motivated GTE, Inc. to propose development of "UltraFASTLANE", a 10 Gb/s Type 1 encryptor patterned after their "FASTLANE" 0.622 Gb/s encryption product. The "UltraFASTLANE" encryptor is expected to satisfy initial DOE ASCI encryption needs in the 10 Gb/s arena. Personnel from this project have been invited to participate in the design reviews to assure that the "UltraFASTLANE" encryptor will meet DOE needs.

DISTRIBUTION:

| | | |
|---|---|---|
| 5 | | Jeff Ingle<br>National Security Agency<br>Attn: R222, R&E<br>9800 Savage Rd.<br>Ft. Meade, MD 20755-6000 |
| 1 | MS 0188 | C. E. Meyers, 4523 |
| 1 | MS 0188 | LDRD Office, 4523 |
| 1 | MS 0431 | S. G. Varnado, 6200 |
| 1 | MS 0449 | R. J. Granfield, 6237 |
| 1 | MS 0449 | R. L. Hutchinson, 6237 |
| 1 | MS 0449 | R. S. Tamashiro, 6237 |
| 1 | MS 0622 | J. F. Jones, 4600 |
| 1 | MS 0806 | L. B. Dean, 4616 |
| 1 | MS 0806 | L. G. Martinez, 4616 |
| 10 | MS 0806 | L. G. Pierson, 4616 |
| 1 | MS 0449 | T. D. Tarman, 4616 |
| 1 | MS 0806 | M. O. Vahle, 4616 |
| 10 | MS 0806 | E. L. Witzke, 4616 |
| 1 | MS 0812 | M. R. Sjulin, 4914 |
| 1 | MS 0874 | K. L. Gass, 1716 |
| 1 | MS 0874 | T. L. Hardin, 1716 |
| 1 | MS 0874 | P. J. Robertson, 1716 |
| 1 | MS 1074 | D. C. Wilcox, 1735 |
| 1 | MS 0624 | L. L. Pucket, 2984 |
| 1 | MS 9003 | D. L. Crawford, 8900 |
| 1 | MS 9011 | H. Y. Chen, 8910 |
| 1 | MS 9011 | P. W. Dean, 8910 |
| 1 | MS 9018 | Central Technical Files, 8940-2 |
| 2 | MS 0899 | Technical Library, 4916 |
| 1 | MS 0161 | Patent and Licensing Office, 11500 |
| 1 | MS 0619 | Review and Approval Desk, 15102<br>For DOE/OSTI |