High Performance Agile Crypto Modules

Chandana G. Gamage[1], Jussipekka Leiwo[2], and Yuliang Zheng[1]

[1] Monash University, PSCIT, McMahons Road, Frankston, Vic 3199, Australia, {chandag,yuliang}infotech.monash.edu.au
[2] Vrije Universiteit, Division of Sciences, De Boelelaan 1081A, 1081 HV Amsterdam, The Netherlands, leiwo@cs.vu.nl

Abstract. This paper examines the impact of the primary symmetric key cryptographic operation on network data streams, encryption of user data, have on the overall traffic throughput. The encryption function which provides the basic confidentiality security service is studied from two cryptographic design perspectives based on the ability of a network cipher unit to dynamically re-parameterize itself while processing a high speed data stream. The designs studied in this paper were chosen based on their suitability for high speed network operation and their flexibility in satisfying dynamic security requirements. We develop analytical techniques to model the performance of each scheme.
Keywords. Network security, High performance, ATM networks, Performance modelling

1 Introduction

In secure communication networks with an end-to-end security association, the basic security service of data confidentiality is provided by a symmetric key cipher. The two basic classes of symmetric key ciphers are the stream cipher and block cipher. Stream ciphers encrypt individual characters in a unit of plaintext one character at a time. For network communication, this generally means bit-by-bit encryption. In contrast, block ciphers such as DES [11] encrypt a fixed size group of characters at a time. Stream ciphers are useful in a communication network that does not buffer protocol transfer units but process a continuous bit stream. However, this is not the case with modern network designs which utilize a fixed length protocol transfer unit commonly referred to as a packet or a cell and variable length transfer units termed segments consisting of a number of fixed size packets or cells. Therefore, in network security, our discussion is limited to encryption function provided by symmetric key block ciphers. Apart from its suitability for use in providing data confidentiality packet oriented data networks, the cryptographic function of block encryption is a central element in providing several other network security services such as data integrity and message authentication. A single block encryption algorithm can be used in the construction of both the encryption function and the message authentication code (MAC) function [12] to provide data confidentiality and data authentication respectively as shown in figure 1. A MAC value simultaneously provides both data integrity and data origin authentication.

E. Dawson, A. Clark, and C. Boyd (Eds.): ACISP 2000, LNCS 1841, pp. 74–88, 2000.
© Springer-Verlag Berlin Heidelberg 2000

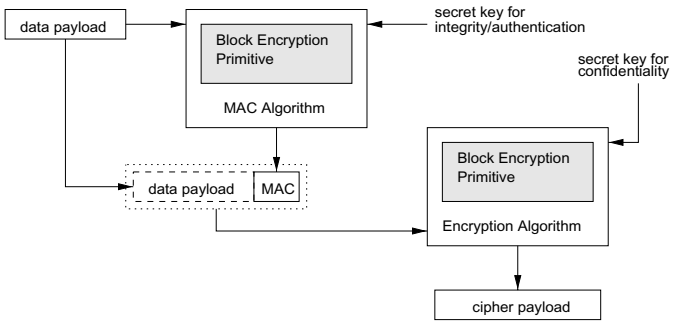

Fig. 1. A function block schematic for generating an encrypted payload with a MAC authenticator value

In the next section, we discuss specific network quality of service (QOS) parameters of interest for analyzing performance of secure communication channels. Thereafter, section 3 presents key-agile encryption technique and section 4 presents algorithm-agile encryption technique. The main reasons for selecting these two encryption techniques are given in section 3.1 and section 4.1, respectively. Apart from the efficiency and flexibility of operation provided by these two schemes in providing basic security services at the network layer of high speed communication networks, they have been used in several research projects [20, 22] to build secure ATM test networks. This is significant for secure high speed network design as ATM is the preferred network technology for multi-service broadband networks (B-ISDN) [13]. The analytical model and numerical example for key-agility (in sections 3.4 and 3.5) and algorithmic-agility (in sections 4.2 and 4.3) provide a basis and justification for use of these techniques in high speed network implementation. The paper concludes with remarks on the effect of several other QOS parameters on secure network performance.

2 Secure Communication and QOS Parameters

In the design of high performance network security systems, a detailed analysis of capacity requirements and deliverable throughput is essential. For end-to-end network performance modelling analysis, capacity is represented by the product of number of simultaneously active *secure channels* and the average channel bandwidth which determine the quantity of traffic that the network can sustain. Similarly, throughput is represented by bounds on the number of allowable active channels and allowable transmission losses which determines the amount of traffic successfully transmitted [10]. Inadequate or incomplete analysis of available and required system capabilities could easily lead to vastly under-performing systems. For example, over-estimation of required processing capacity of a security module could result in a conservative design approach that implements only modest and potentially inadequate security capabilities. Similarly, under-estimation

of required processing capacity could result in secure systems that fail to provide expected performance guarantees. This is a particularly important issue in multi-service high speed networks that have been designed to negotiate and then provide a guaranteed QOS for network users. In this respect, we use ATM networks as the basis for discussing the relationship between network QOS and cryptographic performance in secure high speed networks. The ideas presented and results derived in this paper are applicable to other types of networks with QOS support such as TCP/IP networks operating under resource reservation protocol (RSVP) [24]. As the main focus of the study presented in this paper is the impact of symmetric key cryptographic techniques on secure real-time data transmission, the QOS parameters we consider relate to data transfer and not to call control for the connection-oriented ATM networks. The QOS parameters of concern are bit error ratio (BER), cell loss ratio (CLR) and cell insertion ratio (CIR).

BER is defined as the ratio between number of bit errors that occur in a transmission system and the total number of bits transmitted and is mainly dependent on the transmission system being used including its physical characteristics (such as for copper, fiber optic, etc.) and the operational environment (such as electro-magnetic interference). The use of optical fiber technology in high speed networking has greatly reduced the expected value of BER. If the bit error occurrence in a link is a random process (as the case in modern high speed optical links), then the probability of an error free transmission of an ATM cell is $(1 - \text{BER})^{384}$. Here, only the cell payload of 48 bytes (384 bits) is considered for bit error detection as an unrecoverable bit error in the cell header portion will result in the cell not being allocated to any particular stream.

CLR is defined as the ratio between number of cells lost in transmission and the total cells transmitted over a period of time. The main reasons for specifying a CLR for ATM network connections are the cell discard due to buffer overflow and unrecoverable bit errors in cell headers. For our analysis, both BER and CLR can be considered as a single factor affecting performance as cells with bit errors in encrypted payloads also need to be discarded on integrity check failure at the same layer on which decryption is done.

The cell insertion occurs due to bit errors in the header causing mis-routing of cells onto wrong channels when the error in header address field matches with a correct switching label at a ATM node. CIR is defined as the ratio between number of cells misrouted to a destination and the total number of cells delivered to that destination address. CIR also causes loss of cryptographic synchronization and discarding of several cells. Thus, in the rest of the paper, our reference to CLR actually refers to compound effect of cell losses due to BER, CLR and CIR under encrypted cell transmission.

For secure cell transmission, we may consider a cell with even a single bit error as a lost cell as a single bit error in an encrypted payload can expand randomly on decryption of the cell causing bit error expansion within a cell. This intra-cell error expansion will make any use of error correction codes largely ineffective. Depending on the mode of encryption [12] used and the construction of the

encryption unit, bit errors may propagate to adjacent cells also. For example, if the cipher block chaining (CBC) or cipher feedback (CFB) mode is used, then a bit error within one encrypted cell payload will spread through rest of the cell stream on decryption. While the output feedback (OFB) mode does not cause bit errors in the cipher payload to spread, this mode requires periodic synchronizing of the encryptor and decryptor unit (say, by using a special marker cell) to recover from possible cell losses. Both the CBC and CFB modes are self synchronizing with cell loss propagation limited to only one additional block. The other common mode of operation, electronic codebook (ECB) has only intra-block bit error propagation and the cell stream is self synchronizing on cell losses with no loss effect propagation. However, ECB is not recommended for use in many applications as it is vulnerable to both substitution or reordering attacks and cryptanalytic attacks due to repeating plain text blocks producing identical cipher text blocks. For data networks, any unrecoverable errors in transmission of a protocol unit at a given layer (for example, a packet at network layer) detected at a receiver usually invokes an error recovery procedure at the same layer or at a layer above. As the standard error recovery mechanisms is to request retransmission of the entire protocol unit of transfer, the benefits of limited error propagation or self synchronization of an encryption algorithm is of limited value in network applications.

In summary, the major effect of a cell that was lost or was in error is the loss of cryptographic synchronization for adjacent cells in a stream of encrypted cells for widely used encryption modes resulting in more than one cell being lost and potentially a larger block of cells mapping to an upper level protocol data unit (PDU) being discarded.

3 Key-Agile Encryption

3.1 Encryption at Physical Layer

One of the simplest method to secure communication between two network end-points is to agree on a cryptographic context including a symmetric key for traffic encryption using off-line mechanisms. This allows host-level security management at end-points. Thereafter, confidentiality and integrity services for the user data part of a traffic stream can be delivered through encryption and data integrity functions at the highest possible speed at the physical or link layer as no further on-line security related negotiations take place. The main disadvantage of this approach is that it is not possible to secure the network traffic at a finer granularity such as per-user or per-application. Also, the static pre-configuration of cryptographic keys and consequent long-term key usage allows an attacker to capture a large amount of ciphertext messages encrypted with the same key (especially in a high speed network) which could facilitate an off-line cryptanalytic attack. Furthermore, as inter-networking units such as switches and routers need to process header information contained in a protocol transfer unit, the full traffic stream cannot be encrypted. If the header information also need to be protected, for example, to prevent traffic analysis, then the traffic streams need

to be secured on hop-by-hop basis with decryption at input and re-encryption at output in each inter-networking unit using the symmetric key shared between each unit. In addition to the disadvantages mentioned earlier, this makes the switching and routing nodes in a network highly vulnerable to attacks. Also, the amount of cryptographic processing required at each inter-networking unit would be excessively high resulting in lower network throughput performance. The type of network security provided by such a scheme would be public network-to-network rather than private user-to-user as the management of cryptographic keys in the network infrastructure would be outside end-user control. Therefore, even with the potential for fast implementation through hardware and transparency of operation, this type of physical layer or link layer schemes for securing traffic is not suitable for modern high speed networks.

With respect to ATM networks, the physical layer or link layer type secure communication would be the use of a single cryptographic context and an associated symmetric key to encrypt the cell stream through a physical link connecting two adjacent ATM switches or through all the VCs between two ATM connection end-points. As this solution essentially creates a secure virtual tunnel that operates at the physical layer allowing fast encryption of cell streams between the end-points, designers of secure networks does not have to consider the behaviour of actual traffic sources (such as continuous or bursty traffic) or protocol semantics of the cell stream (such as native ATM traffic that have cell continuity or IP over ATM type traffic that contain cell blocks). Again, as the scheme requires static keying of physical links, this approach is not suitable for use by ATM end-points that connect through public ATM infrastructure for wide area connectivity. Also, even in the case of a private ATM LAN, this scheme does not have the capability to distinguish between traffic streams requiring security services and those that have no security requirements. Therefore, the potential for computational and transmission resource wastage is quite high. Due to these drawbacks, it is necessary to consider a more flexible and secure scheme to protect ATM cell stream at the next level of granularity, that is, for the individual VC.

3.2 Encryption at a Virtual Layer

Protection of data communication at a virtual layer as opposed to the physical layer, involves identification of end-to-end flow of protocol transfer units and applying cryptographic operations to these flows separately using individually specified security contexts. In network protocol models that provide an end-to-end transport connection layer, this mode of operation can be implemented with flow identification by the connection label embedded in data packets (for example, TCP source and destination port address gives a flow identifier at transport layer). If the network protocol model only provides a datagram style hop-by-hop virtual layer, the packet flow differentiation has to be done using a combination of header information (for example, the IPv6 source and destination addresses, identification, protocol and options fields together defines a unique

flow label. In IPv6, the flow label header field combined with the two IP addresses provide the flow identifier).

For ATM networks, the term key-agile cell encryption refers to the scheme by which the traffic through each individual VC is secured using a unique short-term symmetric key [20]. Among the many advantages of this approach are the ability to dynamically negotiate a shared key for each VC at its connection setup time, ability to determine symmetric key characteristics such as the key length based on end-point security requirements and frequency of dynamic key updates for long-lived VCs. However, to achieve key agility, several changes are required for the basic cell relay function at the ATM layer. The two main changes required are:

1. Table look-up of the associated symmetric key at each end-point on a per-cell basis.
2. Execution of cryptographic operations including integrity checksum calculation and encryption (resp. checksum verification and decryption) on the cell payload.

As the cryptographic operations are performed only on the cell payload and header is sent in clear text, this cryptographic table look-up is required only at end-point ATM nodes. The standard ATM layer functions such as cell relay, header recalculation and queue management at intermediate switching nodes are not affected by this type of secure cell transmission.

3.3 Dynamic Key Look-Up

Each active VC in an ATM network is uniquely identified by a 24-bit combined virtual path identifier (VPI) and virtual channel identifier (VCI) address at the user network interface (UNI). As the particular symmetric key to be used for securing a cell payload is determined by this VPI/VCI pair, cryptographic operations can be done only after a payload had its header prefixed. Also, the ATM layer performs cell-level address-based statistical multiplexing using the VCI in each cell. Therefore, in theory, key-agile cell encryption or decryption may require a table look-up for each cell received at an incoming or outgoing port. As the symmetric keys are much longer than ATM address fields (minimum of 64-bit and up to 256-bit long keys), the usual technique is to use a dual table solution in which a larger table indexed by VPI/VCI values contain pointers to a smaller cryptographic key table [20].

This symmetric key retrieval from a potentially very large table can therefore become a performance bottleneck in a high speed key-agile cell encryption scheme. There are several techniques to limit the negative impact on the performance:

1. *Caching of recently or frequently used symmetric keys.* This standard performance enhancement technique can be used by directing the dual table key access operation through a small cache of (VPI/VCI, symmetric key) tuples stored in high speed memory.

2. *Restricted range of VCIs for secure connections.* In the ATM standard, there are several blocks of VPI/VCI combinations that are reserved for specialized use. Among these are the address labels used for unassigned cells processed by the transport layer, cells introduced by the physical layer, idle cells for cell boundary resynchronization and OAM cells. Similarly, a range of VPI/VCI values could be defined for use in secure VC establishment by end-points. This approach will essentially limit the size of symmetric key table for fast access.
3. *Look-up table implementation in content addressable memory (CAM).* A look-up table constructed using CAM allows parallel search of its index entries, thus speeding up the content retrieval.

Although above techniques to reduce the performance penalty in cryptographic table look-up appear to be costly to implement (for cache and CAM) and arbitrarily restrictive (for limited VCI range), in actual high speed networks the table size for active VCs may be much smaller than the theoretical maximum possible due to other factors such as limited buffer space at multiplexors and allowed cell loss ratio (CLR). Therefore, before designing a costly or restrictive symmetric key table look-up scheme, it is important analyze the network operation to determine the actual performance requirements. Results of such an analysis could direct secure system designers to more affordable and realistic solutions.

3.4 An Analytical Model of the Key Agile Network Port

Consider N VCs multiplexed at an outgoing port with V_n, where $n = 1, \ldots, N$, as the transmission rate at time t for a VC labelled with number n. We assume the rates of the N circuits to be independent and identically distributed random variables. Now, we would like to determine the rate r such that the multiplexed link can carry the aggregate traffic of N VCs bounded by the allowable CLR. Alternatively, we could determine the number of traffic sources N for a given average transmission rate of r.

$$P\{V_1 + \ldots + V_N > rN\} \leq \text{ CLR} \tag{1}$$

To model fairly generic network traffic conditions, we further assume the input sources of the VCs to be homogeneous on-off sources with $P(on) = p$, peak rate a and the number of VC input sources in *on* state to have a binomial distribution. Then the probability that the aggregate transmission rate of active VCs exceeds the output rate of the outgoing link can be represented as

$$P\{V_1 + \ldots + V_N > rN\} = \sum_{n \geq \frac{Nr}{a}}^{N} \binom{N}{n} p^n (1-p)^{N-n} \tag{2}$$

Using the Bahadur-Rao theorem [3,23] we can determine an approximation for the above equation 2 as follows that would allow us to compute a value for N.

$$P\{V_1 + \ldots + V_N > rN\} \approx \frac{1}{\sqrt{2\pi}\sigma\theta_r\sqrt{N}} \times e^{-NI(r)} \tag{3}$$

where
$\theta_r = \frac{1}{a}\log\left(\frac{r(1-p)}{p(a-r)}\right)$,
$I(r) = \frac{r}{a}\log\left(\frac{r(1-p)}{p(a-r)}\right) - \log\left(\frac{a(1-p)}{(a-r)}\right)$ and
$\sigma^2 = r(a-r)$

3.5 A Numerical Example

To illustrate the operation of a high speed network under the above modelling assumptions, consider a port with VCs having a peak rate $a = 10$ Mbps (say, switched LAN interfaces) and rate $r = 5.33$ Mbps. Under these conditions, for $\lambda = 1/3$ and $\mu = 1/2$ gives $P(on) = p = 0.40$ at a mean rate $p \times a = 4$ Mbps. Now, the parametric values of equation 3 can be computed as $\theta_r = 0.05377$, $I(r) = 0.03597$ and $\sigma^2 = 24.89$. If the CLR is assumed to be 10^{-9} (for fiber optic links), the probability $P\{V_1 + \ldots + V_N > rN\}$ holds for $N \geq 500$.

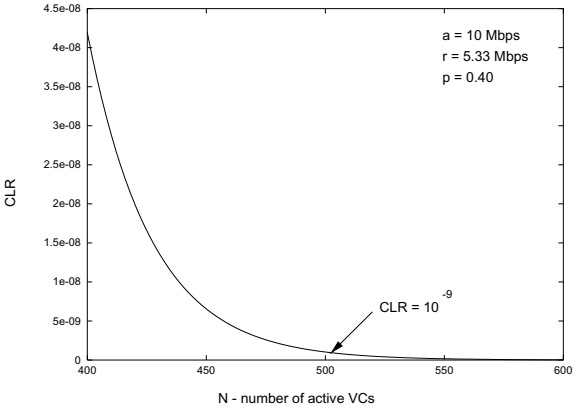

Fig. 2. A numerical example showing the limiting effect of CLR on the number of active VCs multiplexed onto a single ATM link

The results of the above numerical example show, that for an ATM outgoing port with an aggregate capacity of $N \times r = 2665$ Mbps (approximately an OC-48 link), the requirement is to maintain a cryptographic table with a maximum of 500 entries only as the number of active VCs are bounded by the CLR (see graph in figure 2). Also, more importantly, the port will have to look-up keys only 500 times every second or once every 2 millisecond given the common input source behaviour as on-off models. It is interesting to compare these performance values computed above with a simple derivation of key switching for an OC-48 link.

In this case, if we assume cell-by-cell key switching, a table look-up is required every (424 bits / 2488 Mbps) = 170 nanoseconds.

Above example clearly shows the unnecessarily high performance bounds a designer would obtain from a strictly theoretical derivation (a key look-up every 170 ns) against a more realistic value obtained by stochastic modelling (a key look-up every 2 ms). In real-time operation, the performance of the crypto unit is not limited by the average processing but by the peak instantaneous loads. As we have incorporated this behaviour into the analytical model through the peak rate a, we can conclude that key-agile cell encryption is a practical technique for implementing flexible security schemes for high speed networks.

4 Algorithm-Agile Encryption

4.1 Need for Algorithmic Context Negotiation

A high speed network that has only limited operational requirements and policies with regard to securing its data transmissions can standardize on a pre-agreed set of cryptographic parameters including algorithms for encryption and modes of operations, algorithms for digital signatures, key lengths, key update and access control policies and other cryptographic variables. Thereafter, the network can provide a secure per-connection end-to-end data transmission mechanism such as the key-agile cell encryption previously described in section 3 when only the session key is dynamically selected or updated. However, for a multi-service network spanning many operating and regulatory environments, a cell encryption mechanism based on a single algorithmic context (or few algorithmic contexts that are pre-configured for specific static connections) is clearly inadequate. When end-systems located in different operational and regulatory environments setup a secure connection, it is necessary to dynamically negotiate a cryptographic context including different algorithms acceptable to both end-users. For ATM networks, the industry standard security specification document [2] proposed by the ATM forum includes security signalling at connection setup time to carry out this task and provide adaptive secure data transmission. This need for dynamic cryptographic context selection is applicable to other network protocols that provide an end-to-end connection at a virtual layer such as the transport layer of TCP/IP protocol stack.

As a concrete example of the need for algorithmic agility, consider the export control regulation for cryptographic products enforced by many countries. While many countries allow strong cryptographic security (such as symmetric keys of 128 bit length or longer) within the national boundaries, strength of external secure communication is arbitrarily limited (such as short 40 bit keys). Use of different key lengths for the same algorithm itself constitutes algorithmic agility since the crypto units need to be reconfigured at real-time with new values associated with the key schedule. The situation becomes even more complex when multiple keys (such as in triple DES) are used to circumvent key length restrictions. Therefore, cell-level encryption devices in high speed trans-national

networks need to be designed with an array of crypto units that can operate in parallel to process multiple data streams simultaneously.

The term algorithm-agile cell encryption refers to the scheme by which traffic through different VCs may be processed according to a dynamically negotiated cryptographic context. The most commonly cited approach in literature is to implement a cryptographic module with multiple parallel processing units that are dynamically loaded with associated algorithmic context for the cell being processed currently in each unit [22,21].

However, in this paper, we consider using crypto units that are pre-configured for different algorithms and associated crypto variables. This method allows easier combination of key agility and algorithmic agility within a single security module. In the security module, one multiplexor unit can differentiate cells according to their algorithmic-context and feed to appropriate crypto unit while within that unit per-cell key and other related crypto variable look-up can be done.

The objective of the modelling and analysis done in the remainder of this section is to show the practicality of algorithm-agile cell processing using above described type of crypto units for normal high speed network operations under typical traffic loads and for reasonable number of algorithmic-context options. For this purpose, first we develop a system model based on the operational properties of an algorithm-agile crypto module. While the concept of algorithmic agility is applicable to any type of protocol scheme allowing end-to-end network connections in which users want dynamically specify a security context, the analytical model is developed for ATM, which is the most common type of high speed network implementation technology. Thereafter, a numerical example is used to evaluate the practical utility of the scheme under high speed network operational parameters.

4.2 An Analytical Model of the Algorithmic Agile Network Port

Consider an algorithm-agile crypto module as shown in figure 3, that has m number of crypto processor units, each capable of processing cells that have been cryptographically secured according to a specific algorithmic context. Lets assume that each processor unit P_i is capable of processing p_i cells per time unit and that the combined crypto module receives n cells per time unit over its multiplexed input. We further assume the distribution of algorithmic context among arriving cells to be a Poisson distribution owing to its memoryless property. Therefore, if the algorithmic contexts are numbered as $0, \ldots, m-1$ then the probability function

$$p(x) = \begin{cases} \frac{\lambda^x e^{-\lambda}}{x!} & \text{if } x \in \{0, \ldots, m-1\} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

gives the distribution of algorithmic contexts among the cells that have arrived during a sample time unit. Here, λ is the mean value that determines the shape of the algorithmic context distribution.

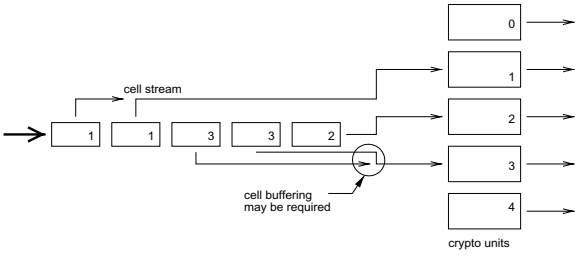

Fig. 3. Model of algorithmic agile cell processing

If we assume that each crypto unit P_i takes τ_i time units to process a cell (that is, $p_i \times \tau_i = 1$), then the amount of time T_i spent by each crypto unit in processing cell payloads is

$$T_i = n\ p(i)\ \tau_i \quad \text{for } i = 0, \ldots, m-1 \tag{5}$$

As the above model of cell processing consider a time window of single unit, for optimum performance by the crypto processing units without causing cell discard or excessive buffering, following inequality should hold

$$T_i = n\ \frac{\lambda^i e^{-\lambda}}{i!}\ \tau_i \leq 1 \quad \text{for } i = 0, \ldots, m-1 \tag{6}$$

4.3 A Numerical Example

To illustrate the operation of an algorithmic agile crypto module based on the above model, lets consider a 16 processor system with an OC-48 link capacity that input approximately 5.87 million cells per second (2488Mbps / 424bits) with an average cell processing time of 160ns. The graphs in figure 4 show the processor utilization profiles corresponding to different cryptographically secured cell traffic profiles as determined by various λ values. The numerical values chosen for the analysis conform to the optimality bounds given by equation 6.

The average value of 160ns to process the secured payload of a cell requires a sustained throughput of 2400Mbps (384bits / 160ns), which may seem difficult to achieve given current state of the art in VLSI based cryptographic processor cores. However, the throughput requirement can be easily reduced without an adverse effect on the overall system performance. Consider the crypto unit utilization graph for $\lambda = 2$ in figure 4 which represent a typical crypto traffic profile scenario with only 2 or 3 heavily used algorithmic contexts and the remaining modules used less frequently. For this particular case, the average value of τ can be increased from 160ns to 640ns while still remaining within the bounds set in equation 6. This will result in a required throughput of 600Mbps which is within the range of hardware implementations of widely used symmetric key cryptosystems such as DES [8].

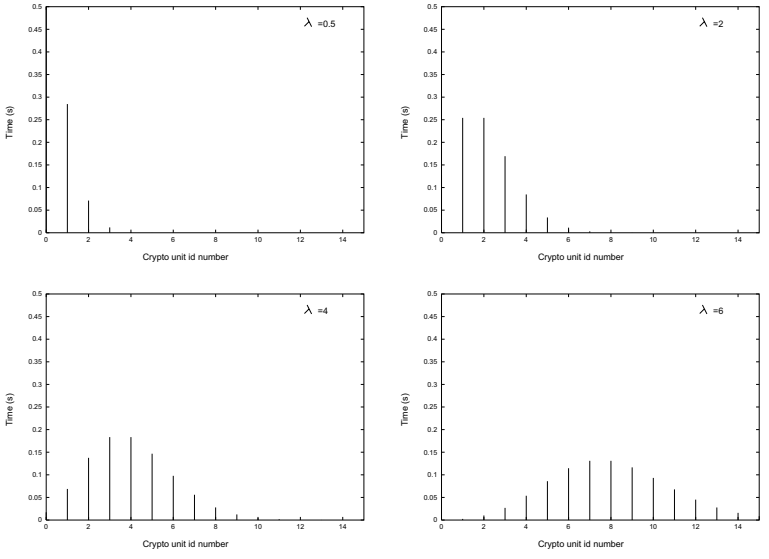

Fig. 4. The utilization of individual units in an algorithm-agile crypto module for different cell traffic profiles

The sample values used in the numerical example shown above are based on commercial VLSI crypto core units such as [1, at 640 Mbps] and [5, at 528 Mbps] that support the DES algorithm [11]. The table 1 gives a summary of the projected throughput requirement in terms of bit encryptions per unit time for different algorithm-context loadings. As newer encryption algorithms such as SPEED [25], Twofish [16] and Rijndael [9] that are are much faster than the DES algorithm comes into wider use, achieving the throughput requirements for high speed cell encryption with these new generation algorithms including the next generation Advanced Encryption Standard (AES [4,17]) is likely to be much more practical.

Table 1. Average throughput requirement for crypto units under different traffic profiles ($m = 15$ and $n = 2488 Mbps$ are the parameters in equation 6)

λ	τ *(ns)*	*Active units*	*Throughput (Mbps)*
0.5	530	1	725
2	640	2/3	600
4	800	4/5	480
6	1060	6	365

In our model, we have not explicitly considered the possibility of more than one crypto unit supporting the same algorithmic context. In this instance, a

separate scheduler will have to be used to admit cells to individual units in the crypto processor group with the common algorithmic context. This will have no meaningful impact on our analysis which attempt to determine if an overall crypto module consisting of several separate units can process a multiplexed cell stream within performance bounds set by a high speed network environment.

5 Summary

The idea of key-agile encryption and details of a proof-of-concept implementation work done by Stevenson et al. [20] have appeared in literature preceding the work described in this paper. Also, design details of a cryptographic device called a *CryptoNode* incorporating similar ideas have been presented by Chuang in [6,7]. Independent of the work described in this paper, the concept of algorithm-agile encryption and related work has been presented by Sholander et al. [18], Tarman et al. [22,21] and Pierson et al. [14]. The use of a single key block in algorithmic agile systems, proposed by Smart [19], to assist in the rapid real-time algorithm selection can be combined with algorithm-agile crypto units to further improve performance.

To obtain operating parameters for the design of high performance security modules (specifically, high speed cell encryptors), designers can utilize any combination of analytical models, simulations and test implementations. Although both simulations and test-beds can benefit from results obtained through analytical models, there is a clear lack of work in this area. As real performance of high speed networks continue to increase and wide area networks grow in complexity, both simulation and test implementation will be more difficult. Therefore, analytical mechanisms such as ours that examines the performance of secure systems will be much valued tool for system designers.

Two other important data transfer QOS parameters are the end-to-end transfer delay (CTD) and the cell delay variation (CDV) or jitter. The analysis in this paper considered crypto modules to be delay units of fixed time duration that increase end-to-end transfer delay by a pre-calculatable value and thus cause no change in original negotiated CDV. However, as different crypto units within the security processing module are most likely to have different latencies due to differences in algorithms and key lengths, it will cause CDV in the multiplexed cell stream. Therefore, above modelling assumptions will be true only if delay equalization is done at the processing module at the cost of increased CTD. Otherwise, original QOS negotiation at connection setup time needs to consider the added CDV due to security related processing. To develop analytical models that can accurately represent the real-time behaviour of high performance secure networks, above security related delay analysis must be complemented with other factors such as connection admission control policies, job scheduling at security modules (processors) and management of shared resources such as buffers.

As the agile crypto units are designed for block-mode operation in uniquely identified end-to-end data flows, they can be positioned at several of the layers

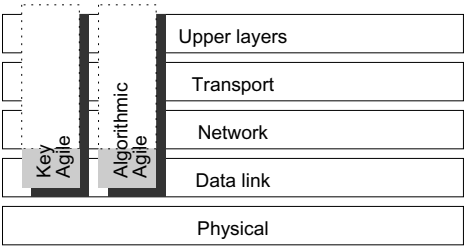

Fig. 5. The positioning of interfaces to the agile crypto units in a network protocol stack

in a network protocol architecture that have a defined protocol unit. However, as shown in figure 5, agile crypto units should be located nearer to the portion of a network protocol stack that is implemented in hardware to achieve the highest possible performance.

References

1. The SafeNet ADSP2141L from Analog Devices, Inc, March 1999. Available from `http://www.analog.com/`.
2. The ATM Forum, Mountain View, CA. *ATM Security Specification, Version 1.0*, af-sec-0100.000 edition, February 1999.
3. R. R. Bahadur and R. R. Rao. On deviations of the simple mean. *Annals of Mathematical Statistics*, 31(2):1015–1027, 1960.
4. O. Baudron, H. Gilbert, L. Granboulan, H. Handschuh, A. Joux, P. Nguyen, F. Noilhan, D. Pointcheval, T. Pornin, G. Poupard, J. Stern, and S. Vaudenay. Report on the AES candidates. In *Second Advanced Encryption Standard (AES) Candidate Conference*, Rome, Italy, March 1999. NIST.
5. CDI 2100 cryptographic processor from Cognitive Designs, Inc, May 1999. Available from `http://www.cognitive-designs.com/`.
6. S.-C. Chuang. Securing ATM networks. In *Proceedings of the Third ACM Conference on Computer and Communications Security (CCS'96)*, pages 19–30, New Delhi, India, March 1996. ACM Press.
7. S.-C. Chuang. Securing ATM networks. *Journal of Computer Security*, 4:289–329, 1996.
8. L. Claesen, J. Daemen, M. Genoe, and G. Peeters. Subterranean: A 600 Mbit/sec cryptographic VLSI chip. In E. Straub, editor, *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD'93)*, pages 610–613, Cambridge, MA, October 1993. IEEE Computer Society Press.
9. J. Daemen and V. Rijmen. The block cipher Rijndael. In *Proceedings of CARDIS'98: Third Smart Card Research and Advanced Applications Conference*, Lecture Notes in Computer Science, Louvain-la-Neuve, Belgium, September 1998. Springer-Verlag.
10. G. N. Higginbottom. *Performance Evaluation of Computer Networks.* Artech House, Inc, Norwood, MA, 1998.

11. National Bureau of Standards, U.S. Department of Commerce. *Data Encryption Standard. Federal Information Processing Standards Publications (FIPS PUB) 46*, January 1977.
12. National Institute of Standards and Technology, U.S. Department of Commerce. *DES Modes of Operation. Federal Information Processing Standards Publication (FIPS PUB) 186*, May 1994.
13. R. O. Onvural. *Asynchronous Transfer Mode Networks: Performance Issues*. Artech House, Inc, Norwood, MA, 1994.
14. L. G. Pierson, E. L. Witzke, M. O. Bean, and G. J. Trombley. Context-agile encryption for high speed communication networks. *SIGCOMM Computer Communication Review*, 29(1):9–9, January 1999.
15. R. A. Rueppel. Stream ciphers. In G. J. Simmons, editor, *Contemporary Cryptology: The Science of Information Integrity*, pages 65–134. IEEE Press, New York, 1992.
16. B. Schneier, J. Kelsey, D. Whiting, D. Wagner, and C. Hall. On the Twofish key schedule. In S. Tavares and H. Meijer, editors, *Fifth Annual International Workshop on Selected Areas in Cryptography (SAC'98)*, volume 1556 of *Lecture Notes in Computer Science*, pages 27–42, Kingston, Ontario, Canada, August 1998. Springer-Verlag.
17. B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson. Performance comparison of the AES submissions. In *Second Advanced Encryption Standard (AES) Candidate Conference*, Rome, Italy, March 1999. NIST.
18. P. E. Sholander and et al. The effect of algorithm-agile encryption on ATM quality of service. In *Proceedings of IEEE GLOBECOM'97*, pages 470–474, Phoenix, AZ, November 1997. IEEE Computer Society Press.
19. N. P. Smart. One key to rule them all. Technical Report HPL-1999-26, Extended Enterprise Laboratory, HP Laboratories Bristol, UK, March 1999.
20. D. Stevenson, N. Hillery, and G. Byrd. Secure communications in ATM networks. *Communications of the ACM*, 38(2):45–52, February 1995.
21. T. D. Tarman, R. L. Hutchinson, L. G. Pierson, P. E. Sholander, and E. L. Witzke. Algorithm-agile encryption in ATM networks. *IEEE Computer*, 31(9):57–64, September 1998.
22. T. D. Tarman, R. L. Hutchinson, P. E. Sholander, R. J. Granfield, L. G. Pierson, P. J. Robertson, and E. L. Witzke. Final report for the robustness-agile asynchronous transfer mode (ATM) encryption laboratory directed research and development project. Technical Report SAND97-2902, Sandia National Laboratories, Albuquerque, NM, December 1997.
23. J. Walrand and P. Varaiya. *High-Performance Communication Networks*. Morgan Kaufmann Publishers, Inc, San Francisco, CA, 1996.
24. L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. Rsvp: A new resource reservation protocol. *IEEE Network Magazine*, 7:8–18, September/October 1993.
25. Y. Zheng. The SPEED cipher. In R. Hirschfeld, editor, *Proceedings of Financial Cryptography'97*, volume 1318 of *Lecture Notes in Computer Science*, pages 71–89, Anquilla, BWI, February 1997. Springer-Verlag.