1998 International Computer Symposium
Workshop on Cryptology and Information Security
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

A BLOCK-BASED APPROACH TO SECURE ATM NETWORKING

Chandana Gamage, Jussipekka Leiwo, and Yuliang Zheng

Peninsula School of Computing and Information Technology
Monash University
McMahons Road, Frankston, Vic 3199, AUSTRALIA
E-mail:{chandag,skylark,yuliang}@pscit.monash.edu.au

ABSTRACT

There is increasing deployment of asynchronous transmission mode or ATM as the preferred data transmission technology for network backbones as well as local area networks requiring high bandwidth and low latencies. At the same time network applications continue to be developed on packet oriented network protocols such as IP for the Internet and are mostly interactive in nature. This results in majority of the traffic on cell based ATM backbone networks to be packets and bursty in their traffic characterization. Also, protection of data transmission has become a major issue with growing awareness of both network users and operators to the potential for security failures and costs associated with such failures. Therefore, development of secure high speed network protocols that can efficiently support packet based bursty traffic on ATM networks is an important requirement.

The main contribution of the research work described in this paper is a comprehensive design of a secure ATM network protocol for transmission of blocks of cells that can map to upper level protocol data units such as IP packets. The described protocol satisfies a range of network data transmission security criteria as well as being efficient in its use of computational time and redundant bit space for cryptographic operations.

Keywords

Network Security, High Speed Networks, ATM, ABT, Cryptography, Signcryption

1 INTRODUCTION

Recent developments in networking technologies, particularly protocol support for applications and high bandwidth network access, have given rise to two important trends:

ATM for network backbone. A network built on ATM technology is capable of carrying several different traffic types (realtime (voice, video), non-realtime (data), etc.) in a single configuration. This is done by establishing virtual connections (VC) with specific quality of service (QoS) characteristics to suit a particular traffic type. ATM is also highly scalable in terms of geographic spread, number of users and bandwidth. These capabilities coupled with enhanced flexibility of ATM connection management to dynamically control different classes of services and QoS have made it the preferred technology for building wide area network (WAN) backbones.

IP for application delivery. Development of new applications, specially those that are network centric, is fueled by the growth in Internet based technologies and their expansion into new user categories. Therefore, it is likely that the primary networking protocol for user applications will continue to be the IP protocol. This is evidenced by the wide ranging adoption of IP by both hardware and software developers. Thus, the traffic on WAN links will be dominated by IP packets.

This evolving network scenario creates a requirement for efficient IP packet transmission over cell based high speed ATM networks. Furthermore, as network security is increasingly becoming a major issue in data communication, newer network protocol designs need to incorporate security functionality directly at the network protocol layer. An example of this approach is IPv6 [4]. When incorporating security features into ATM networks, the particular strengths and weaknesses in the micro cellular ATM switching networks must be carefully considered. This allows protocol designers to maximize the effectiveness of security measures and also to minimize potential performance degradations.

1.1 Implications of ATM Layer Design to Security

ATM cell switching is done in hardware to achieve high network throughput. Thus, this layer has to be very simple and for example, it is not possible to perform extensive error checking at the switching layer. This requirement has significant implications for implementation of security services at the ATM layer as it is not possible to directly integrate any security related processing at the switching layer of a *pure* ATM switch. Furthermore, congestion control in ATM is based on cell discard at switching nodes on buffer overflow. This has severe repercussions on security as the loss of even a single cell result in the loss of *crypto-context synchronization* of multiple cells in a particular secured cell stream.

Once the crypto synchronization of a cell stream is lost, the end-nodes must resynchronize and restart the secure cell transmission from the last recoverable cell position. It is possible to use the cell loss priority (CLP) bit in ATM cell header to give a higher priority for those cells that carry crypto-context synchronization information to reduce the probability of those control crypto cells being discarded on congestion. However, maintaining of a synchronized crypto-context on an end-to-end basis is highly inefficient and expensive for transmission links with any significant cell loss probability.

1.2 Related Work

Several approaches to building secure ATM networks have appeared in the literature. In the link layer technique described in [17], inter-switch transmission links are secured using link encryption between ATM ports. This method builds a low level secure network infrastructure mostly suitable for service providers as the link encryption keys are pre-distributed and securely stored at switches and have comparatively long lifetimes. Similarly, at the ATM layer, secure networking products [7, 13] are available for encrypting permanent virtual circuits (PVC) on an end-to-end basis to provide *secure virtual tunnels*. These static VCs also rely on secret keys exchanged at the time of PVC setup. A more flexible scheme for secure ATM communication at the cell level is described in [5, 10, 19]. These publications present the key agile secure ATM connection technique in which each switched virtual connection (SVC) is individually secured using a dynamic session key exchanged between the ATM connection endpoints during the three-step call establishment phase. The secure SVC capability available in end-point ATM switching nodes allow the establishment of *secure virtual private networks* [9, 10].

While the above schemes differ in details of dynamic key exchange protocol used and in specific cryptographic techniques, they all exchange the session key during the call establishment phase using the large ATM call control messages with a maximum size of 64 KB. These large control frames at the service specific convergence sublayer (SSCS) of the ATM control plane provides the lengthy bit-spaces (in the order of 512 bits or more) required by common public key cryptographic primitives such as Diffie-Hellman [6], RSA [18] and ElGamal [8]. Also, they do not present any specific mechanisms for re-keying but suggest the use of initially exchanged session keys for the distribution of the updated session keys. This key chaining for re-keying has two main weaknesses

1. If existing session keys are used for distribution of key-updates, it sets up a key chain in which compromise of any single link will compromise all remaining links in the forward path of the chain.

2. In network applications such as multicasting, re-keying is generally used as an implicit mechanism to remove existing group members from a shared session key membership. In such instances it is not possible to re-key using the current session key.

1.3 Research Focus

Our focus in this research is to investigate a mechanism that will map an IP packet into a distinct block of ATM cells which in turn would allow this cell block to be cryptographically secured while being transmitted over an ATM WAN backbone. This per block-based approach distinguishes our work from other per cell-based security solutions. Our research aim is to develop a security protocol for end-to-end secure data transmission with minimum protocol overhead. We measure this overhead both in terms of redundant bits and need for dynamic cryptographic context management. Finally, we aim to design a secure network protocol that is capable of operating at high speeds over ATM networks.

1.4 Structure of the Paper

The paper is started by introducing the key transport problem in ATM networks and a possible solution using signcryption in section 2. In section 3 we discuss a particular mode of operation for ATM networks that use special cells for traffic management. This is followed by a discussion in section 4 which introduces modifications to the special cell structure that allow specific security functionality to be provided for the enclosing block of ATM cells. We use signcryption primitive to construct a cryptogram that is carried in these special cells to implement security services. In section 5 we discuss the assumptions made in designing the secure ATM protocol presented in this paper and also evaluate it against alternative schemes. We conclude the paper in section 6 by summarizing the main advantages of the proposed secure ATM protocol.

2 SIGNCRYPTION AND SINGLE CELL KEY TRANSPORT IN ATM

2.1 The Session key Transport Problem

When two participants wish to communicate securely, in general, a shared secret value called a session key is established. If a communication session is secured by a session key dynamically established as part of the connection setup, we term the transmission path a key agile network connection. Key agility is a desirable feature as every communication path is secured independent of each other. It also provides protection against replay and interleaving attacks on individual communication sessions.

Session key establishment can be done in two ways: (1) In *key material exchange*, both participants jointly generate a session key using key material from each participant. An example is the Diffie-Hellman protocol [6]. (2) In *key material transport*, one participant generates the session key and transmits it to the other participant. Alternatively, the key material necessary to derive a session key may be transmitted. An example is the Kerberos protocol [16]. In our research work, we concentrate on security for high speed ATM networks where security related protocol overheads are kept to a minimum. Therefore, our focus is on key material transport protocols which involve fewer interactions between participants of a communication session.

The session key transport problem in ATM networks considered in this paper is to provide a key transport protocol with following attributes:

1. The key material is transported in a single ATM cell payload of 48 bytes. This requirement is due to the need for a reduced protocol overhead in terms of redundant bits and also to eliminate the need for crypto-context synchronization between cells that carry the key material.

2. The key material transport protocol must have low computational overhead. This efficiency requirement needs to be satisfied for its use in high speed ATM networks.

3. The transported key material can be checked for integrity and authenticity. It should also provide non-repudiation by the sender.

4. The protocol must not involve a trusted third party such as a key distribution center (KDC). This attribute is required to support session establishment between participants that do not

share keys with a KDC and also to minimize the number of steps in a key transport protocol.

The above attributes 1 and 2 preclude use of conventional public key cryptosystems for the key transport protocol as they generate cryptograms of length in excess of 384 bits which will not fit within the 48 byte ATM payload. Also, the attributes 3 and 4 prevent the use of secret key cryptosystems in the protocol. Next we look at a new public key cryptographic primitive that is able to satisfy the above attributes when used in a key transport protocol.

2.2 Overview of Signcryption

The public key cryptography method, *signcryption*, provides the signature-then-encryption function as a single primitive. The significance of signcryption is that the cost of signcryption operation is smaller than the cost of conventional signature and encryption operation combinations for comparable level of security. This cost reduction is achieved for both bit-lengths (space) and computational steps (time). Detailed description of signcryption, analyses of its cryptographic strength, example implementations and comparison of performance improvements are given in [20, 21, 23]. This section briefly summarizes the operation of signcryption.

	Parameters public to all:
p	a large prime
q	a large prime factor of $p-1$
g	an integer with order q modulo p chosen randomly from $[1,\ldots,p-1]$
$hash$	a one-way hash function whose output has, say, at least 128 bits
KH	a keyed one-way hash function
(E,D)	the encryption and decryption algorithms of a private key cipher
	Alice's keys:
x_a	Alice's private key, chosen uniformly at random from $[1,\ldots,q-1]$
y_a	Alice's public key ($y_a = g^{x_a} \bmod p$)
	Bob's keys:
x_b	Bob's private key, chosen uniformly at random from $[1,\ldots,q-1]$
y_b	Bob's public key ($y_b = g^{x_b} \bmod p$)

Table 1: Parameters for signcryption

The parameters used in generating a signcrypted message (i.e. a cryptogram) are shown in table 1. The individual cryptographic operations using these parameters are listed in table 2. In table 3, an example implementation of the signcryption primitive is shown.

The step (2) in table 3 should generate a hash value output that is sufficiently long (say, 128 bits), so that both k_1 and k_2 have sufficiently long key lengths (say, at least 64 bits). The compactness of the ciphertext, generated by signcryption, is such that it can digitally sign and encrypt a random bit sequence of length 64 to 70 bits into a cryptogram of bit length less than 384 bits. This random bit sequence is recovered by a recipient for use as a private key in subsequent communication with the original sender. While the compact cryptogram is small enough to fit inside the 48 byte

$c = E_k(m)$	Encryption of message m with key k, typically in cipher block chaining (CBC) or output feedback (OFB) mode. E is a private key cipher such as DES [14]
$m = D_k(c)$	Decryption of ciphertext c with key k.
$r = KH_k(m)$	Hashing a message m with keyed hash function KH with key k. For practical applications, $KH_k(m) = hash(k,m)$ where $hash$ is a one-way hash function such as SHS [15]
$\in_R [\ldots]$	This indicates an operation to choose an element uniformly at random from the set of elements $[\ldots]$

Table 2: Cryptographic functions for signcryption

ATM cell payload, the combined sign-and-encrypt operation of signcryption is computationally efficient for use in high speed network applications.

Signcryption of m by sender (Alice):
(1) $x \in_R [1,\ldots,q-1]$
(2) $(k_1,k_2) = hash(y_b^x \bmod p)$
(3) $c = E_{k_1}(m)$
(4) $r = KH_{k_2}(m)$
(5) $s = x/(r+x_a) \bmod q$
Unsigncryption of (c,r,s) by receiver (Bob):
($\bar{1}$) $(k_1,k_2) = hash((y_a \cdot g^r)^{s \cdot x_b} \bmod p)$
($\bar{2}$) $m = D_{k_1}(c)$
($\bar{3}$) if $KH_{k_2}(m) = r$ then accept m

Table 3: Example implementation of signcryption

In section 3 we discuss a particular mode of operation for ATM networks that use special cells for traffic management. We use signcryption primitive to construct a cryptogram that is carried in these special cells to implement security services.

3 THE ASYNCHRONOUS BLOCK TRANSFER MODE

The asynchronous block transfer (ABT) capability is a packet data oriented service mode recommended by ITU-T [11]. The defining feature of ABT is that it allocates network resources in a multi-service ATM network on per-block transmission requirements rather than on per-connection basis as in other modes such as constant bit rate (CBR) and variable bit rate (VBR) [2]. ABT is designed to benefit the large class of data applications with bursty traffic characteristics. These include applications in the Internet such as WWW, Telnet and RPC which can use ATM adaptation layer-5 (AAL5) as a link level transport mechanism. ABT operates at ATM layer level (as shown in figure 1) by reserving bandwidth for a connection either on an end-to-end basis (for delayed transmission mode of ABT) or a hop-to-hop basis (for immediate transmission mode of ABT) through bandwidth renegotiation. The VC

for which ABT protocol renegotiates bandwidth is initially established without any bandwidth allocation. This dynamic bandwidth acquisition is initiated by the source separately for each transmitted block.

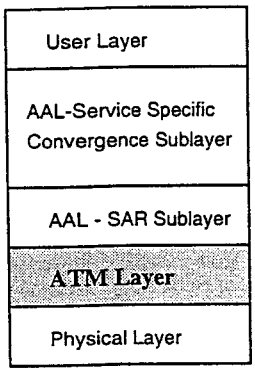

Figure 1: ATM protocol stack layers (for user plane)

A block in ABT can be any higher layer protocol data unit, a fraction or multiple of it that is delimited by two resource management (RM) cells. An RM cell is a special ATM cell used for in-band signaling and control of a VC. It has the standard ATM cell format for the header but carries a specialized payload depending on its management task [11]. The leading bandwidth-allocating RM cell defines the required peak cell rate (PCR) traffic parameter and cell delay variation (CDV) QoS parameter. The following RM cell is used for deallocation of resources used for the virtual path. In ABT protocol, nodes discard the entire flow of related cells (i.e. a block) if any cell from that block is dropped on congestion at a switch. This mode of operation, rather than partially forwarding a flow to the destination that will be eventually discarded and retransmitted by source, allows switching nodes to control link congestion while maintaining a high bandwidth utilization.

3.1 ABT with Delayed Transmission (DT)

In this mode, the bandwidth-allocating forward RM cell for a block of cells is sent to the network indicating the required QoS parameters. The actual transmission of the block is *delayed* until confirmation of acceptance by the network is indicated through a backward RM cell. The leading RM cell is processed by each switch along the VC to determine if the requested QoS can be guaranteed at that switching node. The block is sent to the network only if a positive confirmation is received. The transmission of the block of data is followed by the bandwidth-releasing RM cell.

3.2 ABT with Immediate Transmission (IT)

In this mode, the bandwidth-allocating forward RM cell for a block of cells is sent to the network indicating the required QoS parameters. Thereafter, the block is transmitted *immediately* following that RM cell. This block of data is followed by the bandwidth-releasing RM cell. The leading RM cell is processed by switches along the VC on a hop-by-hop basis to determine if the requested QoS can be provided. A switching node will discard the entire block as marked by the leading and following RM cells if the requested bandwidth cannot be allocated for the VC to support that block.

In section 4 we discuss modifications to the RM cell structure that allow specific security functionality to be provided for the enclosing block of ATM cells.

4 A SECURE ABT PROTOCOL USING RM CELLS

The ATM early packet discard (EPD) in block transfer mode for flow control of ATM cells is used for the implementation of application-segment oriented security protocol described in this paper. Our implementation of secure data delivery in ABT is based on using the RM cells to provide following fundamental security services.

Security Association. (SA) The SA gives a cryptographic specification in terms of algorithms for bulk encryption and message authentication code (MAC) generation, lengths of encryption key and hash value, type of encryption (stream or block) and mode of operation for signcryption (sign and encrypt, sign only or encrypt only). The SA is sent from the source to the destination and defines the actual cryptographic primitives used for the associated block of cells and is carried by the leading RM cell.

Block Encryption Key. (BEK) The BEK is a symmetric key used for encryption of the cell payloads belonging to the block and for generating a block-wise MAC. We use signcryption based public key cryptography for secure transportation of the BEK. The signed and encrypted key is sent from the source to the destination and is carried by the leading RM cell.

Data Confidentiality. Encryption of the cell payloads belonging to the transmitted block using the BEK provides confidentiality for user data so that only the holders of the BEK can decrypt the cryptogram. Data confidentiality is enabled by specifying encrypting mode of operation in the SA.

Data Integrity. A MAC is generated for the entire transmitted block, including the leading and following RM cells, using the BEK. The MAC is sent from the source to the destination and is carried by the following RM cell. Integrity of data transmission can be established by the successful verification of the MAC at the receiver node allowing receiver to detect tampering or accidental alteration of data while in transit.

Non-repudiation by the Sender. The combination of senders digital signature and a cryptographically secure hash of the message provide a single mechanism for non-repudiation of both content and transmission of a message by a sender. If the BEK is successfully recovered from the signcrypted RM cell payload using standard public key cryptographic operations and if that key can be used for the subsequent decryption of the received data block to obtain meaningful data (in an application-oriented sense), then the receiver can present both the BEK and the cell block as convincing evidence to an arbitrator. It should be noted that arbitration is essentially non-mechanical process and the task of security services is to provide cryptographically secure evidence to aid that process. This cryptographic

evidence will be used only in the event a sender maliciously denies sending a block of data to a receiver.

Sender Authentication. The successful recovery of BEK from the signcrypted cryptogram in the leading RM cell and verification of MAC in the following RM cell together establishes the authenticity of the sender as part of the secure data transmission service. This provides the receiver with a cryptographic evidence for the correct identity of the sender.

The other fundamental security services such as access authorization and service availability are not applicable in the context of secure transfer of a block of data in an ATM VC. Those services need to built at a higher level in the protocol stack using the secure data delivery primitive provided by the proposed secure ABT capability. Furthermore, as our RM cell based secure ATM level data delivery provide only key transport (as against key exchange), both non-repudiation of receipt and authentication of receiver are unavailable. This is a possible shortcoming of the secure delivery technique that should be carefully considered in the context of target applications.

4.1 Effect of Cryptographic Processing on Transmission Latencies

As shown in figure 2, the cryptographic processing of a block of user data can be done at any level above the segmentation and reassembly (SAR) sublayer, either in AAL level or in an application sublayer. The exact placement would depend on the degree of integration between the application security processing and the ATM network protocol stack. The ABT capability is designed for data applications which are mainly bursty. Therefore, latencies introduced at the end-point switching nodes due to additional cryptographic processing before the transmission of a block and after the arrival of a block will not affect the overall *inter-block arrival delay*. This is due to the uniform symmetry of crypto related processing at both sending and receiving nodes.

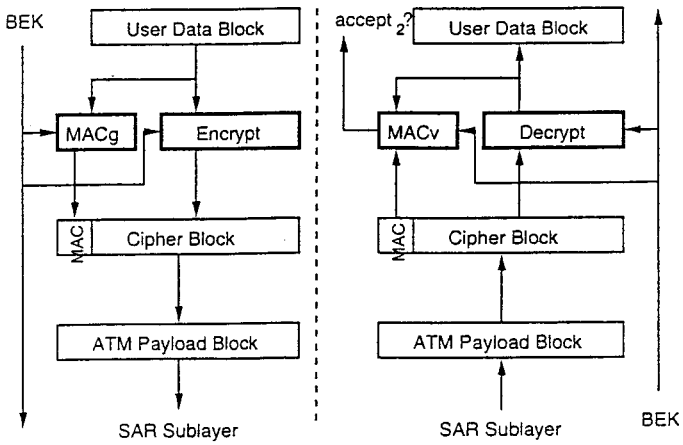

Figure 2: Cryptographic processing of a block of data above SAR or AAL level

Therefore, from an application point-of-view, there will be small delay only at the beginning of an application (a start-up delay). This conclusion is based on the assumption that an adequate *inter-block generation delay* exists at nodes for cryptographic processing without a need for queuing of data blocks. Therefore, we effectively assume that the receiving end-node can complete processing of a block, including cryptographic processing, at the network interface line speed. To justify this assumption we would need a combination of fast algorithms (such as SPEED [22] for block encryption and signcryption [20] for public key cryptography) with hardware implementation to achieve the speeds required for interfaces such as OC-12c (622 Mbps). Delay-sensitive applications will be largely unaffected due to the non-expansion of application detectable latencies beyond start-up as described above. It should be noted that this start-up delay can only be reduced but not completely eliminated using more efficient schemes for bulk encryption and decryption, MAC generation (MACg) and MAC verification (MACv) by faster algorithms or hardware implementations. This is the cost of security on transmission latency. Most importantly, the network security protocol is decoupled from any latency generating action for a data block at the sender or receiver node thus allowing continued improvements for crypto related technology.

From the above discussion it is clear that transmission delay analysis is significant only with regard to the construction and processing of the two leading and following ABT RM cells and not the cell block enclosed by those RM cells. We summarize the conclusions derived from above discussion as follows:

1. The construction of a secure RM cell to carry a signcrypted BEK delays the bandwidth renegotiation and this is an extra ATM layer latency due to security services.

2. The extraction of a signcrypted BEK from the secure RM cell does *not* delay bandwidth renegotiation or subsequent transfer of the data block as these actions can proceed in parallel with receive-end cryptographic processing.

Therefore, analysis of receiver node delay is less significant. As the leading RM cell can be passed up to a signcryption chip or software process for key extraction and authentication while the ATM cells forming the data block pass through the SAR chip for data block reconstruction, the BEK will be available for decryption and MAC verification when the complete data block emerges from SAR sublayer.

In table 4, the standard format of an RM cell used in ABT is shown with individual field names and the number of bits allocated for each field. It also illustrates the modifications (or security specific interpretations) to several fields suggested by us to support secure block transport under both ABT/DT and ABT/IT modes. We make use of only those fields that are either reserved or unused.

4.2 The Secure ABT/DT Protocol Description

Shown below is the complete sequence of message transfers and end-point processing for the proposed secure transport protocol.

(1) A : Signcryption of the BEK at the Sender
(using shortened digital signature standard
version 1 - SDSS1)
SP = Signcrypt(BEK) signcrypted payload

ABT RM cell payload fields	Size	SABT/DT	SABT/IT
Protocol identifier (ID)	8	ID	ID
Direction (DIR)	1	DIR	DIR
Traffic management cell (TM)	1	TM	TM
Congestion indication (CI)	1	CI	CI
Maintenance (M)	1	M	M
Request/acknowledge (R/A)	1	R/A	R/A
Elastic/rigid (E/R)	1	E/R	E/R
Reserved (RES)	2	RES	RES
Block cell rate (BCR)	16	BCR	BCR
OAM block cell rate (O/BCR)	16	O/BCR	O/BCR
Minimum cell rate (MCR)	16	SA (16)	SA (16)
Block size (BS)	32		BS
Sequence number (SN)	32	SP (310)	SN
Reserved (RES)	246		SP(246)
Cyclic redundancy check	10	CRC	CRC
Note: SA - Security association SP - Signcrypted payload (cryptogram)			

Table 4: RM cell payload formats for secure ABT service

SA = encoded 16 bit status word (cryptogram) security association (clear record)

Note: Please refer to section 2.2 for details of the following signcryption operation.

$c = E_{k_1}(BEK)$ $|BEK \oplus fold(k_1)| = 70\ bits$
$r = KH_{k_2}(BEK)$ $|KH.(\cdot)| = 80\ bits$
$s = x/(r + x_a)\ mod\ q$ $|q| = 160\ bits$
SP = (c, r, s)

(2) A → B : RM-CELL[...,SA, SP]
(bandwidth allocating forward RM cell)

(3) B : Unsigncryption of the BEK at the Receiver
BEK = Unsigncrypt(SP)

Note: Please refer to section 2.2 for details of the following unsigncryption operation.

$k = (y_a \cdot g^r)^{s \cdot x_b}\ mod\ p$
split k into k_1 and k_2
$BEK = D_{k_1}(c)$
$if\ KH_{k_2}(BEK) = r\ then\ accept_1 = True$

(4) A ← B : RM-CELL[...] (bandwidth confirmation backward RM cell)

(5) A → B : Transmission of ATM-CELL stream with encrypted payloads
$E_{BEK}(datablock)$ symmetric block encryption

(6) A: Construction of the time variant code (TVC) at the Sender
ts = time-stamp *up to* 182 *bits*
$h = hash$(data block) $|h| = 128\ bits$

TVC = E_{BEK}(ts,h)

(7) A → B : RM-CELL[...,TVC]
(bandwidth deallocating RM cell)

(8) B : Verification of the integrity of data transfer at the Receiver

$ts, h = D_{BEK}(TVC)$

$if\ hash$(data block) $= h\ then\ continue$
$if\ 0 \leq (ts_{now} - ts) \leq$ acceptance-time-window
$then\ accept_2 = True$

The three messages sent from A to B (in steps 2, 5 and 7) are cryptographically related by the common BEK. As the 70 bit crypto-data space available in the signcrypted cryptogram is inadequate to carry both a time-stamp and a symmetric key of reasonable length to provide adequate security, only the BEK is included in the leading RM cell. Therefore, the cryptogram lacks any time variant property (TVP) and an attacker could mount replay attacks using passively recorded RM cells. Similarly, the data block received from an upper layer of the protocol stack is not time-stamped to prevent block expansion and ensuing buffer re-allocation activity. Thus, the encrypted data block is also without any TVP, although the data content itself may have some semantically inherent TVP. The last message (in following RM cell) is used to send a time variant code (TVC) to counter possible replay attacks and give the data transfer a timeliness property.

ABT/DT is more suitable for implementation of security as the BEK can be extracted and verified during the time it takes for the backward RM cell to confirm bandwidth availability and the actual arrival of data block.

The bandwidth confirming backward RM cell can be used to renegotiate the initial SA suggested by the sender. However, in a practical high speed networking environment it is most likely that all participating switches would use the same SA. This will prevent delays associated with loading of program core images from memory to cache for a particular mix of SA parameters and building up the working set for the processor unit of an ATM switch for efficient execution of cryptography related code.

In section 5 we discuss the assumptions made in designing the above secure ATM protocol and also evaluate it against alternative schemes.

5 PROTOCOL DESIGN EVALUATION

5.1 Security Related Assumptions

Availability of public key certificates. The certificates are required for public key cryptography. The switching nodes may obtain them through any number of methods such as out-of-band transmission, manual pre-configuration, online directory access or as part of the VC call setup. As key management involves highly time intensive processes for authenticity validation, revocation list checking, storage and retrieval, it is neither practicable nor meaningful to integrate the public key certificate infrastructure functionality to a protocol for secure high speed data block delivery.

Availability of loosely synchronized clocks. A local clock is required by the switching nodes to generate time stamps for the TVC. To keep the *acceptance time window* to a suitably small value, these local clocks need to be loosely synchronized. A secure distributed clock synchronization protocol could be implemented using RM cells, nonces and signcryption.

Block discard on cell error. ATM switches discard entire blocks on cell loss. However, a full block

may arrive at the receiver with cells containing bit errors. If blocks are discarded on both cell loss and cell error, then it is not necessary to implement any crypto-context resynchronization mechanism. Therefore, we assume that on error detection blocks are discarded and upper layer protocols request retransmission. As the high speed optical links used in ATM networks have very low bit error rates (BER), this would be a reasonable assumption to make.

Generating cryptographic data. The SAR sublayer of a sender-side switch receives the tuple (BEK,MAC,cipher) from an upper layer, where MAC is a pre-calculated TVP and cipher is the encrypted data block. The symmetric key BEK should be generated using a suitably seeded cryptographically strong pseudo-random number generator. This assumption regarding construction of cryptographic data elements at a higher layer is made in accordance with the design principle that computations not directly related to actual block transfer is not done at the cell level operations of a switch (i.e., SAR and below).

Multilevel verification. The receiver-side switch performs the verification shown in step (8) of the protocol. The switch will abort the data transfer if condition $accept_1$ is $False$. However, we suggest that the more application oriented condition $accept_2$ should be checked at a higher layer to prevent limiting the bandwidth of a link due to anomalies related to clock synchronization.

5.2 The Secure ABT/IT Protocol Design Problem

The development of a secure protocol for ABT/IT poses a major challenge. That is the space available for a cryptogram through unused and reserved fields in the ABT/IT RM cell is only 246 bits long. Therefore, in a single cell key transport mechanism only 6 bits will be available to carry data, which renders secure key transportation impractical. A possible solution would be to send the encrypted BEK part in the following RM cell. This can be achieved by reducing the number of bits allocated for the hash value and the time-stamp.

In ABT/IT, the RM cell carrying the cryptogram (which contains the BEK) and the ATM cell stream (which contains the encrypted data block) arrives almost simultaneously. Therefore, unlike in ABT/DT, we do not have a time delay to recover the BEK. However, key extraction and verification can still be done by a crypto-processor in parallel with the block reconstruction in which the SAR sublayer processes the received ATM cells.

5.3 A Secure ABT Protocol using ATM Cells for Key Transport

An alternative method for cryptographic key material transport which does not use RM cells is shown in figure 3. In this scheme we use the first ATM cell to carry a cryptogram containing a signcrypted BEK which encrypts the payloads of remaining ATM cells belonging to the block. The sequenced delivery of cells in an ATM VC coupled with the EPD form of congestion control used by ABT, guarantees that if a block of data is successfully received at an end-point, then the crypto ATM cell would be the first to arrive.

The advantage of this method over the RM cell based scheme is that now we have the full 384 bit cell

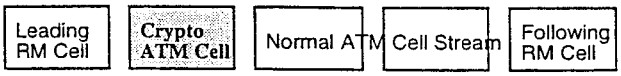

Figure 3: ABT cell stream with key in ATM cell payload

payload to transport the cryptographic key material securely. Possible disadvantages are the loss of control and data flow independence by using a data cell to carry control information (i.e. BEK and TVC), lack of a time delay to pre-process the cryptogram before the arrival of cell stream containing encrypted data block and the small increase in overhead of one extra ATM cell per block in ABT. However, this single cell overhead is most likely to be insignificant if the size of a *block* in ABT is determined according the principles of application layer framing (ALF) for efficient implementation of protocol stacks designed on the model of layer independence [3]. For example, the default maximum transfer unit (MTU) for IP over ATM using AAL5 is set at 9180 octets [1]. This comparatively large MTU is suggested to support multiple higher layer protocol data units (PDU) that have varying MTUs and to prevent unnecessary fragmentation of IP datagrams that significantly reduce performance [12]. For cell level data transmission, the overhead is insignificant for block sizes above 20 ATM cells or approximately 1 KB data segment. Our assumption of insignificant overhead is justified by the fact, that the dominant WWW traffic over Internet consists of short data transfers in the range of 1 to 2 KB.

5.4 Comparison

Finally, let us summarize the comparative security related ATM cell overheads for the above discussed secure protocol methods. The secure ABT/DT technique has zero cell overhead due to use of RM cells for crypto services. The secure ABT using a standard cell for crypto services has a single cell overhead per block. It should be remembered that the ABT capability itself has a 2 cell protocol overhead per block of user cells compared to native ATM modes.

In section 6 we conclude by summarizing the main advantages of the secure ATM protocol described in this paper.

6 CONCLUSION

We have described an efficient technique for secure transmission of blocks of cells in an ATM network using the ABT capability using the bandwidth management RM cells. The native ABT protocol is enhanced by modifying the payload of controlling RM cells to carry security related data such as a security association and a block encryption key.

The protocol presented in this paper is one of the first secure network transmission protocols to use the new signcryption public key cryptographic primitive. The major significance of signcryption is its ability to compress a signed and encrypted cryptographic payload to allow secure authenticated key transport within a single ATM cell payload. This cryptographic key in turn is used to provide confidentiality and integrity of the block of cells being transmitted.

As the major portion of cell losses in an ATM network is due to cell discard at ATM switches on buffer overflow, the EPD style operation of ABT ensures optimum bandwidth utilization. Furthermore, as the complete block of cells is discarded on any single cell loss, use of ABT in the described secure transmission protocol obviates the need for expensive cryptographic resynchronizations on error conditions.

The final result of the research described in this paper is a secure network protocol for key agile block transfer in a high speed ATM network.

References

[1] R. J. Atkinson. *Default IP MTU for use over ATM AAL5, RFC-1626.* Naval Research Laboratory, Washington, DC, May 1994.

[2] T. M. Chen, S. S. Liu, D. Wang, V. K. Samalam, M. J. Procanik, and D. Kavouspour. Monitoring and control of ATM networks using special cells. *IEEE Network Magazine*, pages 28–38, September/October 1996.

[3] D. D. Clark and D. L. Tennenhouse. Architectural consideration for a new generation of protocols. In *ACM SIGCOMM-1990 Symposium*, pages 200–208, September 1990.

[4] S. E. Deering and R. M. Hinden. *Internet Protocol, Version 6 (IPv6) Specification.* IETF, December 1995.

[5] R. H. Deng, L. Gong, and A. A. Lazar. Securing data transfer in asynchronous transfer mode networks. In *Proceedings of the IEEE Globecom'95*, Singapore, November 1995.

[6] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.

[7] M. Ehrlich. Encrypting ATM traffic over the ACTS ATM internetwork. *IEEE Communications Magazine*, 35(8):144–148, August 1997.

[8] T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31(4):469–472, July 1985.

[9] S. Fotedar, M. Gerla, P. Crocetti, and L. Fratta. ATM virtual private networks. *Communications of the ACM*, 38(2):101–109, February 1995.

[10] C. Gamage, J. Leiwo, and Y. Zheng. ATM cell based security implementation. In *Proceedings of the New Zealand ATM and Broadband Workshop (ATMWORKS'97)*, pages 67–82, University of Waikato, Hamilton, New Zealand, February 1997.

[11] ITU, Perth, Australia. *ITU-T Recommendation I.371, Traffic Control and Congestion Control in B-ISDN*, November 1995.

[12] C. Kent and J. Mogul. Fragmentation considered harmful. In *Proceedings of the ACM SIGCOMM'87 Workshop on Frontiers in Computer Communication Technology*, August 1987.

[13] S. Lane. ATM information security - InfoGuard100, January 1996. GTE Government Systems, Needham, MA 02194, USA. Available at `http://www.gte.com`.

[14] National Bureau of Standards, U.S. Department of Commerce. *Data Encryption Standard. Federal Information Processing Standards Publications (FIPS PUB) 46*, January 1977.

[15] National Institute of Standards and Technology, U.S. Department of Commerce. *Secure Hash Standard. Federal Information Processing Standards Publications (FIPS PUB) 180-1*, April 1995.

[16] B. C. Neuman and T. Ts'o. Kerberos: An authentication service for computer networks. *IEEE Communications Magazine*, 32(9):33–38, September 1994.

[17] A. Rieke. Link encryption in ATM systems. In S. Katsikas, editor, *Proceedings of the Communications and Multimedia Security Conference - CMS'97*, volume 3, pages 143–154, London, September 1997. Chapman and Hall.

[18] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[19] D. Stevenson, N. Hillery, and G. Byrd. Secure communications in ATM networks. *Communications of the ACM*, 38(2):45–52, February 1995.

[20] Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption). In B. S. Kaliski, editor, *Advances in Cryptology - CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer-Verlag, 1997.

[21] Y. Zheng. Signcryption and its applications in efficient public key solutions. In E. Okamoto, G. Davida, and M. Mambo, editors, *Proceedings of the 1997 Information Security Workshop (ISW'97)*, volume 1396 of *Lecture Notes in Computer Science*, pages 291–312, Ishikawa, Japan, September 1997. Springer-Verlag.

[22] 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.

[23] Y. Zheng and H. Imai. Compact and unforgeable key establishment over an ATM network. In *Proceedings of the IEEE INFOCOM'98*, San Francisco, 1998.