# ATM Cell based Security Implementation

**Chandana Gamage, Jussipekka Leiwo and Yuliang Zheng**

Peninsula School of Computing and Information Technology
Monash University
McMahons Road, Frankston, VIC 3199, Australia
{chandag,skylark,yzheng}@fcit.monash.edu.au

**Abstract**

A secure network achieves integrity and privacy in communication by employing a shared private key for generation of a MAC and for payload encryption respectively for its messages. A public key cipher method is used for authentication and secret key exchange among remote nodes. Lower layer security mechanisms currently available for ATM networks operate either at frame level or use a combined frame and cell approach that result in inflexible and inefficient schemes. In this paper, we investigate network and cryptographic technology requirements and limitations encountered in designing an exclusively cell–based secure ATM network. We also present a design for a cryptographic security device that can be transparently dropped–in between an ATM user device and network switch to provide secure virtual connections.

**Keywords:** ATM cell encryption, Network security, Secure call setup, Cryptographic key lengths

## 1 Introduction

At its inception, asynchronous transfer mode (ATM) was a fresh attempt to redefine digital data transmission free of constraints imposed by an installed base of technology. A new collection of standards was developed for every aspect of data transmission from layered protocol architecture and cell formats to ATM–switch fabric and call control and management [18, 19]. ATM, also known as cell relay, uses fixed length cells and provide faster cell switching, multiplexing of large number of logical connections over a single physical interface, minimum link level transmission overhead, bandwidth on demand and was developed to fully utilize the advances in digital switching and transmission technologies.

A vast majority of distributed applications that can greatly benefit from the high communication throughput provided by ATM networks also require their message transmission to be secure. Some examples of these enterprise–level applications would be online

medical diagnosis by specialist physicians through multimedia conferencing, research collaboration by scientists in various national laboratories for data exchange, result analysis and discussion, industrial product design, development and manufacturing by globally distributed units of a multinational company, etc. Cell level security in ATM will enable the creation of very high speed secure *virtual private networks* (VPNs) [10] over public ATM infrastructure that is largely transparent to upper level enterprise applications. The secure VPN can operate over both metropolitan and wide area ATM services while allowing local cell level fine–grained security controls.

In a secure network, a message transfer unit (e.g. frame, packet or cell) can be encrypted using a private key shared by communication end–points to maintain data privacy. The integrity of a message transfer unit can be established by associating with each such unit a *message authentication code* (MAC) generated using the shared key. Key management is the set of functions involved in generating and exchanging these private keys among communicating nodes securely and is usually accomplished through a public key based method that also provide end system authentication. Private key (symmetric) ciphers are used to secure session connection as they are much faster than the public key (asymmetric) ciphers. Outlined below are two of the major solutions currently available for secure ATM networking.

## 1.1 A Secure ATM Virtual Tunnel using a Single Key

An ATM cell encryption/decryption unit can be used to build a secure and permanent ATM virtual tunnel for transmission of ATM cells over a public internetwork (figure 1). The encryption unit removes idle cell and encrypts a complete cell, including header and user payload with added control information, using a symmetric encryption algorithm (e.g. DES [8]) and a single private key. Thereafter, this encryptor output is segmented into a new cell stream with a new predefined cell header containing a single virtual path identifier/virtual channel identifier (VPI/VCI) address and is transmitted across public ATM based WAN infrastructure in a single point–to–point *permanent virtual circuit* (PVC). At the receiver–end the cells are striped of the headers, reassembled and fed into a decryption unit for the reverse transform with the same private key. The private key exchange between remote encryption/decryption units is based on Diffie–Hellman [6] type authenticated key exchange mechanism and is part of the automated key management service integral to the device. The InfoGuard100 is an example ATM cell encryptor/decryptor device from GTE and Cylink [13].

This approach to secure ATM networking while being very efficient has several drawbacks

- A higher layer (ATM frame level) static pre–configuration phase is required for each secure connection in order to establish the PVC and perform key exchange and this prevents the device from being exclusively cell–based. Key exchange needs to be

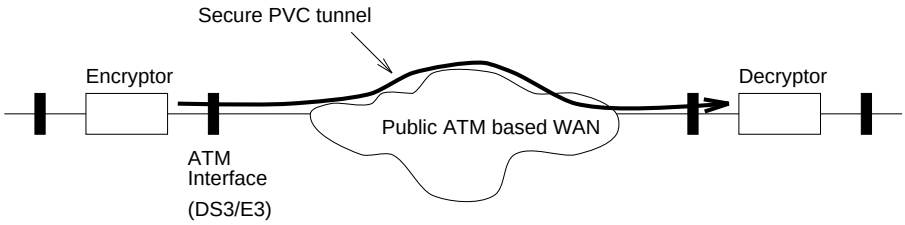


Figure 1: A secure virtual tunnel

done at ATM frame level due to longer data transfer units involved that cannot be accommodated in a single ATM cell (see section 2.1).

- Require effectively dual *segmentation and reassembly* (SAR) phases in the ATM protocol stack that sharply reduce efficiency and throughput at ATM end–nodes.
- All cells over the tunnel are encrypted and thus signaling and control cells can not be processed by intermediate nodes. These control cells are simply discarded at end–points and the tunneling scheme prevents establishment of *switched virtual circuits* (SVCs).
- Only those applications that map to a point–to–point PVC can use a secure ATM channel. Thus, most of the emerging multicast based distributed group applications are excluded from using this scheme due to lack of point–to–multipoint secure transmission capability.

## 1.2 A Secure ATM Virtual Network using Key Agility

A key agile encryption unit placed at the point of interface of a secure ATM subnet and an insecure public ATM WAN can establish a secure virtual network over the public infrastructure (figure 2). Key agility refers to the ability to dynamically use different keys for each *virtual connection* (VC) and to dynamically change the key of an active VC [26]. In key agile systems only the cell payload is encrypted leaving the header unaffected thus, making this scheme highly transparent to the ATM protocol stack. The encryption uses a symmetric algorithm with a unique private key for each VC and the cell payloads are decrypted at the receiver ends by a similar key agile unit. The CellCase key agile encryption system from Secant Network Technologies [24] is a commercial unit that implements this solution. Another implementation of a key agile cryptosystem for ATM networks is Enigma II from MCNC [26]. The key agile method has several favorable features

- As only the cell payload is encrypted, intermediate nodes can process the cell header. This allows routing and control information to remain accessible to the switches permitting the establishment of both SVCs and PVCs.
- As each VC is secured using a separate key, the key agile scheme can be used in different types of secure distributed applications (both unicast and multicast) and VCs can be subjected to a fine–grained access control policy (per VC or VC–group).

- Nodes can simultaneously process mixed traffic streams (encrypted and unencrypted) as individual streams have their own key.

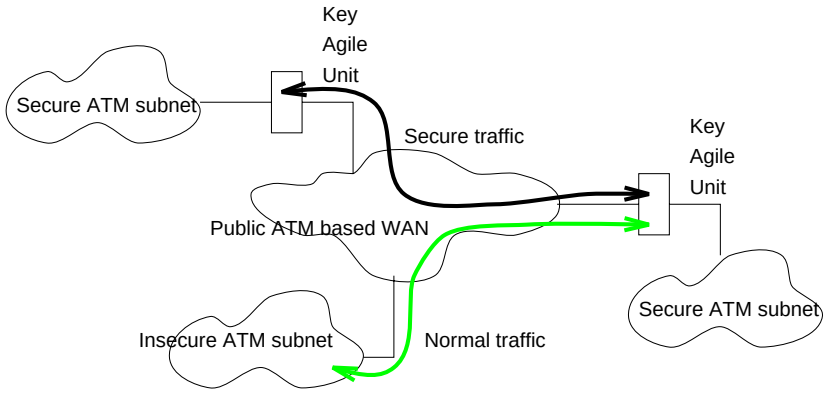


Figure 2: A secure virtual network

However, the scheme from Secant requires encryption key exchange to be performed at a higher layer due to constraints (noted earlier) imposed by the public key cryptosystems and key certificates used for authenticated key exchange (see section 2.1). Although it is possible to dynamically change the key of an on–going VC, this is usually not done due to the overhead involved in the resynchronization of crypto context for the channel. The key agile system from MCNC performs key exchange as part of the ATM VC setup procedure and the technique is based on public key certificates.

# 2 Security Implementation and Cryptography

An ATM cell's payload can be protected against unauthorized access by encrypting the contents using a strong symmetric key algorithm such as DES. Then the strength of the encryption against a possible attack (e.g. a brute force search of the entire key–space) and thus the level of privacy is largely dependent on the length of the private key. Any predictions on a suitable key length, beyond the immediate future, are largely meaningless due to technical advances in computer engineering and manufacturing continually increasing the available processing power and scientific advances in algorithmic and program design causing the required processing power to shrink. However, for practical systems, based on the current state–of–the–art and reasonable assumptions for technological developments, following values can be considered (from chap.7 of [22])

| **Information lifetime** | **Secret key length** |
|---|---|
| minutes/hours | 56–64 bits |
| days/weeks | 64 bits |
| years | 64 bits |

A brute force attack on a cipher system is a worst case scenario giving the upper bound on that cryptosystems strength. However, analytical attacks based on certain properties of the cryptographic algorithm itself or its key schedule are far more effective in breaking a system. Availability of ciphertext, plaintext or matching pairs provides other methods of attacks to find a private key. Given that computational cost increases for stronger cipher systems (i.e. longer key length, etc.) is smaller, use of a fairly long key would seem attractive. In a more recent survey [4] on key length requirements, it is suggested that private keys of length between 75 bits to 90 bits be used. Another approach is to use more than one short key, such as in triple–DES where three (or two) 56 bit keys are used in the key schedule to give an effective key length of 112 bits [11] at extra computational cost. However, it is not possible to make direct comparisons between different cipher systems, based solely on their effective key lengths (i.e. resistance to a brute force attack) as there could be significant reductions in the effective key length for certain cryptanalytic attacks or for time–space trade–offs.

Thus, a 64 bit symmetric key length seems reasonable for a wide range of applications and security requirements. The key length to be used influences the choice of a particular private key cipher by defining the encryption block size that needs to match with the ATM cell size. As an example, the current ATM payload of 48 bytes (384 bits) can be broken to 6 blocks of 64 bit length. A cryptosystem based on fast encryption algorithm (FEAL) [25] that use a 64 bit key and a 64 bit data block would be an appropriate choice. Although it is possible to use algorithms that have key lengths that are not integral factors of the payload by padding data blocks with zeros, such extra work will add an unnecessary processing overhead. Payload encryption does not introduce any extra overhead on the transmission bandwidth. It should be noted that as we concentrate on the use of ATM for computer data transmission, the focus will be on the raw ATM cell as provided by ATM adaptation layer type 5 (AAL5). The AAL5 cell format provides the full 48 byte payload as it has no extra overhead due to sequence numbering, data length or a cell level CRC as in other AAL types.

Furthermore, it is possible to protect an ATM cell's payload against unauthorized modification by attaching a *keyed message digest code* (i.e. a MAC) generated using a strong one–way hash function such as SHS [9]. The same private key used for encryption can be used to generate the MAC which in addition to integrity will provide cell origin authenticity. Practical one–way hash functions that can withstand attacks generate hash values that are at least 128 bits long but are able to process messages in variable lengths such as, 128 bits (N–HASH [14]), 512 bits (SHS, MD5 [20]), 1024 bits (HAVAL [27]) or longer blocks. SHS is particularly strong against cryptanalytic attacks as it generates a hash of 160 bits. Also, it is possible to use only part of the generated hash value (e.g. leftmost 64 bits out of a 128 bit hash) without any serious impact on security for comparatively short lived private keys thus lessening the overhead in an ATM cell for security. However, considering that a MAC of any length would still take part of the cell payload space and a message of any length generates only a fixed length hash value, for transmission efficiency and implementation reasons, an ATM cell level integrity and au-

thentication scheme is undesirable. Such a mechanism is most appropriately implemented at ATM frame level that could be up to 64 Kbyte long. Thus, at cell level it is infeasible to provide a combined secure delivery incorporating privacy, integrity and authenticity for user data payloads.

## 2.1 The Key Material Exchange Problem

As explained in the preceding discussion, to provide privacy for ATM cell payloads a private session key must be shared by the two communicating end–nodes. Thus, to allow decryption of received cell payloads the nodes must already posses the private key. In practice, this session key is transmitted by the session originator to the other node using an authenticated key exchange mechanism based on schemes such as

- Symmetric cryptography and a trusted third party key distribution center (KDC). An example is the Needham–Schroeder method [16] (and improved versions such as [2]).
- Asymmetric cryptography based methods like Diffie–Hellman [6].
- KryptoKnight : This is a two–way authentication and key distribution protocol suite providing a family of protocols useable under different operational requirements (bandwidth availability, packet size, processing delays, etc.). It is designed for deployment particularly at the network level [3].
- Kerberos and X.509 [17, 5] : These are authentication services for distributed application environments and operate at a higher level of abstraction providing a full set of functionality for secure operation.

From a cryptographic point of view, a KDC based mechanism is an attractive scheme for implementation at ATM cell level. As key material (i.e. a private key, generally of length 64 bits) exchange through a KDC is based on symmetric cryptography, such a scheme can quite easily operate within the cell size constraints imposed by ATM (384 bit fixed size payload). Another positive feature is that unlike user data payloads, a key exchange payload can be provided with comprehensive transmission security including privacy, authentication and integrity. This is possible as adequate number of bits are available for the private key (64 bits), a time–stamp and MAC (128 bits) along with other control data. However, this method is practical only for fixed network structures that contain trusted nodes holding pre–distributed secret keys for node–to–KDC communication. Also, strict timing requirements specified in ATM call control standards may render any third party based key distribution mechanisms too slow to operate within ATM signaling schemes. A more generic approach for securing arbitrary ATM networks would be to use public key based asymmetric cryptography techniques that encrypt session keys using public keys for transmission. However, message sizes of this method are much longer than a standard ATM cell can accommodate as it is necessary to use long asymmetric keys.

### 2.1.1 Key Length Requirements in Public Key Ciphers

Selection of key lengths is influenced by factors such as information lifetime, level of security required and the target crypto system. The lengths of the keys for symmetric key and asymmetric key crypto systems chosen in a given application must be of comparable strength against possible attacks. That is, it is not sensible to use a weak public key of short bit length to transfer a strong session key of very long bit length. This holds true for the opposite case also. Shown below are corresponding key length estimates based on current technical capabilities to mount attacks on cipher systems.

| **Symmetric key** | **Asymmetric key** |
|---|---|
| 56 bits | 512 bits |
| 64 bits | 768 bits |
| 80 bits | 1024 bits |
| 112 bits | 1536 bits |
| 128 bits | 2048 bits |

We generally use asymmetric keys to protect data with either a long lifetime (e.g. signing of public key certificates by a trusted KDC) or data that protect other data (encryption of session keys for secure exchange). Therefore, it is important to select key lengths that are perceived to be strong enough to withstand attacks for at least a decade into the future. In this context, a key length of 768 bits is suitable for the short–term while a 2048 bit key should be used for the long–term.

### 2.1.2 Methods for Key Material Transmission

A short session key, when encrypted using a public key cipher for transmission over an insecure link, expands to a much larger bit sequence that cannot be put in a single ATM cell and constitutes a difficult problem to solve as explained below.

- A larger ATM cell – This would require a new standard setting cycle for ATM to define modified cell length parameters and as such is not a viable solution. Furthermore, such a solution has no merit as the number of bits required in cryptographic primitives used to implement security is highly unstable (and increasing) due to technological and algorithmic advances in cryptanalysis. It is not practicable to hypothesize a minimum cell length for an acceptable level of security as there are far too many inter–related factors to consider such as the secured information's lifetime and attack capabilities of existing technologies and techniques.

- Cross–cell key embedding – Key material transfer using cell linkage is also not an appropriate solution as ATM layer only provides an unreliable cell transport service susceptible to cell loss and bit errors. This is significant as cell loss can occur frequently due to cell discard in congestion control and the need to treat secure connections and congestion control/avoidance mechanisms as separate issues.

Therefore, a least cost solution for key material transfer problem would be to perform it at the reliable *service data unit* (SDU) transmission provided at the *signaling AAL* (SAAL) layer of the ATM control plane.

## 2.2 Symmetric Key Cipher for use at Cell Level

The encryption of a VC's contents is done at the fixed size ATM cell level after segmentation and similarly decryption is done before reassembly of cells to SDUs. This places several restrictions on the type of encryption algorithm that can be used.

**Type of Cipher** As cryptographic operations are performed on fixed size cells, use of faster block cipher schemes is more appropriate. However, a block cipher can be operated as a more secure stream cipher without any loss in performance by selecting a suitable feedback mode. The electronic code book (ECB) mode of a block cipher is generally not used as it is quite vulnerable to many types of attacks, specially when used without a MAC. An improved scheme that is resistant to cryptanalytic attacks is the cipher block chaining (CBC) mode that also allows detection of block insertion or deletion. Two methods to efficiently operate block ciphers as stream ciphers are the cipher feedback (CFB) mode and the output feedback (OFB) mode.

**Block Expansion** Encryption should not cause block expansion as no re–segmentation facility is available in the standard ATM protocol stack. As the complete cell payload can be divided into full–size blocks, no padding is required and every block cipher mode will generate a ciphertext equal in size to it's plaintext. However, CBC, CFB and OFB modes require an extra initialization vector (IV) which can be a fixed value block.

**Error Extension Property** A single bit error in the plaintext before encryption will possibly affect the entire ciphertext stream generated thereafter. But, after decryption the process is reversed and the result is the same plaintext with a single bit error. However, if the single bit error occurs in the ciphertext after encryption, the effect on decrypted plaintext is much more severe. In CBC and CFB modes, for a single ciphertext bit error, two plaintext blocks are affected. If the blocks in error are chained/fedback across cell boundaries, this would mean discarding of two cells for a bit error in a single cell. Thus, error extension is an important property to consider in selecting a cipher mode for efficient implementation of cell level security.

**Synchronization Error Recovery** An addition or loss of bits in the ciphertext results in a synchronization error that garbles the subsequently extracted plaintext. For example, CBC mode cannot recover from synchronization errors.

**Execution Parallelizability** While encryption can not be done in parallel on blocks when using either CBC or CFB mode, decryption is parallalizable. Thus plaintext recovery from received cells (consisting of six blocks) can be implemented to be faster than ciphertext generation for improved performance.

In general, CBC mode of DES would provide both strong security and efficient processing for cell level encryption and decryption of an ATM VC.

# 3 A Secure ATM Call Setup Proposal

We propose a secure ATM call setup with authentication and key exchange based on a modification of a two–way authentication protocol appearing in [3]. In this mechanism for key agile VCs with per–connection key exchange, all security related processing is done at the crypto units located between an ATM user device and the public ATM network. The location of security functionality with respect to the ATM protocol stack is shown in figure 3. The correct placement of security and cryptographic processing in the stack has major implications for the transparency of the protocol with the ATM signaling scheme.

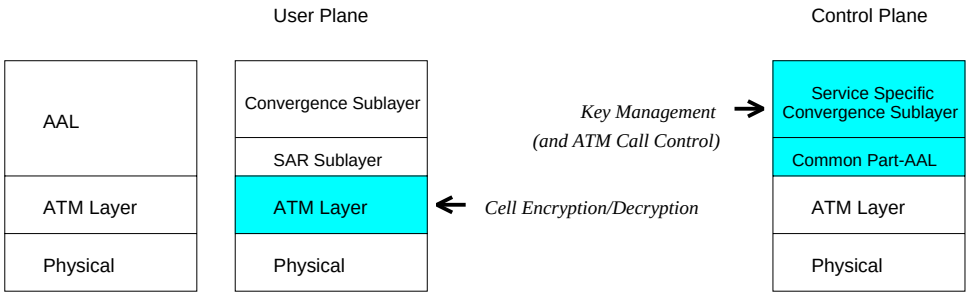


Figure 3: Equivalent function layering for the crypto unit

- No security functions are specified for implementation at the physical layer as it is only a point–to–point bit stream transmission link between network devices and is largely unrelated to the particular upper level networking technology used.
- Cell level encryption/decryption is done in the ATM layer just below the SAR sublayer. As this allows the SAR engine to be fed at nearly the non–secure level of bandwidth, it is possible to maintain performance and QoS at standard levels with only a minimum degradation.
- Key management functions that require larger message blocks are done at the SAAL layer of the control plane using AAL5 SDUs. The actual functionality is placed in the service specific convergence sublayer (SSCS) above the SAR sublayer to allow expansion of transfer data unit (i.e. SDU). This is required for piggy–back sending of security specific data elements in the standard call control SDUs.
- Reliable transfer of key material is an integral function of the common part–AAL (CP–AAL) sublayer and the service specific connection oriented protocol (SSCOP) sublayer. The CP–AAL sublayer that does unreliable SDU transfer, has methods to detect corrupted SDUs and the SSCOP sublayer at SSCS level is capable of recovery from lost or corrupted SDUs.
- The chosen function placement does not permit the transparent insertion of cryptographic checksums at cell level. As cell based security is located below SAR sublayer,

it is not possible to allow transfer data unit (i.e. cell) expansion that would require re–segmentation.

- The convergence sublayer (CS) of the data plane has no security extensions implemented in it. Thus, there is no effect on the standard functions done by CS for managing user data flow and QoS control.

## 3.1 Definition of Terms

| | |
|---|---|
| $\mathbf{C}_X$ | Public key certificate of crypto unit X |
| $\mathbf{N}_X$ | A nonce (one–time random number) generated by X |
| $\mathbf{K}_s$ | Session key shared between two end–point crypto units |
| $\mathbf{MAC}_{Key}(\dots)$ | Message authentication code under $Key$ |
| $\mathbf{E}_{Key}(\dots)$ | Encryption of a message using $Key$ |
| $\mathbf{S}_{Key}(\dots)$ | Signature for a message using $Key$ |

## 3.2 Peer Authentication and Session Key Exchange

The following three steps transparently extend the standard ATM call setup message sequence to provide dynamic key exchange (see figure 4).

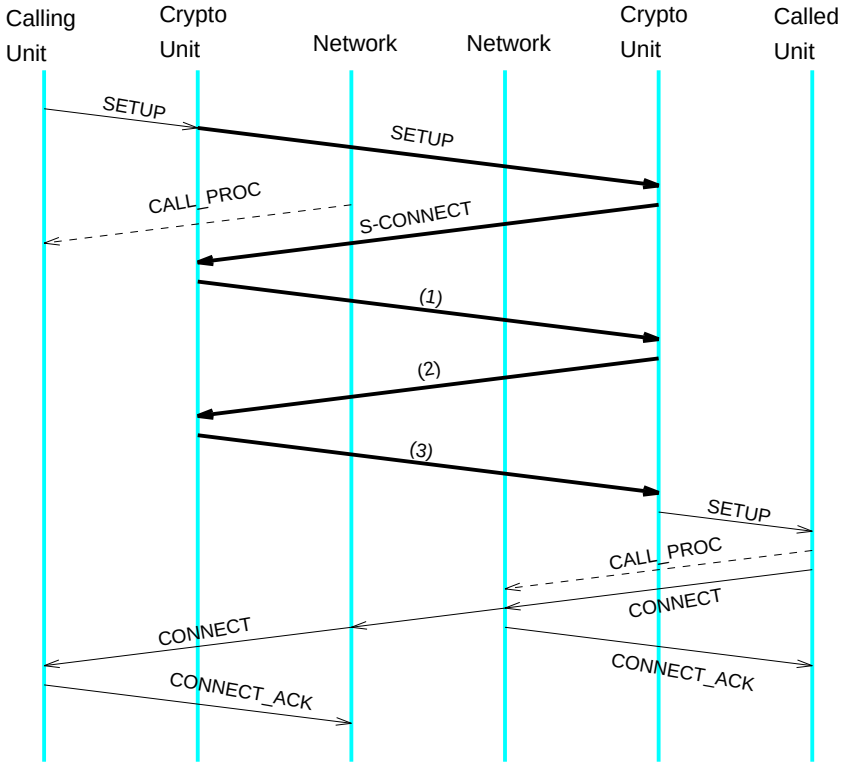


Figure 4: A secure virtual connection setup

1. The calling–crypto unit ($A$) sends a message containing it's public key certificate and a nonce to the called–crypto unit ($B$).
   $[A, B, N_A, C_A]$

2. A reply message from $B$ contains its public key certificate and a nonce together with an encrypted session key generated by $B$ for subsequent use in the end–to–end VC. It also contains a signed hash value that provides message integrity and is part of the authentication scheme.
   $[B, A, N_B, C_B, E_{public_A}(K_s), S_{private_B}(H)]$
   where $H = HASH(N_A, N_B, B, E_{public_A}(K_s))$
   The contents and the format of this message are somewhat generic as presented above. In particular, the specific public key digital signature scheme used in an implementation may allow optimization of both message size and cryptographic computations. The most widely available digital signature schemes are based on either *discrete logarithm problem* (DSA [15], ElGamal [7], Schnorr [23], etc.) or the *factoring of large numbers problem* (RSA [21]).

3. The final message from $A$ to $B$ contains a $MAC$ generated under the shared key that complete the mutual authentication of peer crypto units. The exchange of nonces prevents replay–attacks on the protocol in addition to authenticating communicating parties.
   $[A, B, MAC_{K_s}(N_A, N_B)]$

## 3.3 Methods to Reduce Security Related Call Setup Delays

The extended signaling is limited to the two end–point crypto units and the only effect on ATM user equipment is an extra delay in call setup time that should be within the timing constraints enforced by the user–to–network interface (UNI) timers T303, T310 and T313 [1]. The proposed secure signaling extension has several features that aim to reduce the time required for protocol processing

1. Number of message exchanges – limited to three messages for the complete key exchange and mutual authentication.

2. Number of asymmetric key operations - used twice for encryption and signing of the shared key before exchange and twice for verification of received public key certificates.

3. Use of hashing instead of encryption - second step of the security protocol signs a hashed value of critical elements (that gives a fixed size block) rather than the potentially multi–block complete message resulting in increased computational efficiency. Also, last message in the mutual authentication scheme uses a MAC generated using keyed hashing that is much faster than public key encryption.

4. Use of nonces instead of time–stamps - nonces are used to ensure message freshness and to prevent replay attacks instead of time–stamps that require additional clock

synchronization between distributed crypto units. Also, as the lifetime of a SVC is implicit and does not require pre–negotiation, it is not necessary to include any timing data in the messages.

5. No dynamic public key certificate directory accesses are required - as the required key certificates are dynamically exchanged, the crypto unit is only needed to be preconfigured with its own key pair and the public key of the certifying authority to validate received key certificates.

## 3.4 Call Control Message Processing by the Crypto Unit

The calling side crypto unit intercepts the `SETUP` message that contains a call reference and lists it as pending an end–to–end security association. This message interception is aided by the fact that ATM connection setup is always done using the permanent signaling virtual channel connection (VCC) with $VPI = 0$ and $VCI = 5$. The `SETUP` message is modified by appending a security association identifier (SAID) value to it and changing the message length parameter accordingly. ATM standard allows messages to carry additional information that general user or network devices may not understand. A SAID value with end–to–end significance is required to uniquely identify a session key for a particular VCC during call establishment as call reference values have no end–to–end significance. Thereafter, the message is passed through to the network. On the called side crypto unit, the `SETUP` message is temporarily buffered and a special `S-CONNECT` message is returned using the VC data obtained from the received message. Upon receipt of this message the called side crypto unit initiates the authentication and session key exchange procedure explained in section 3.2. The `S-CONNECT` message (and the key exchange messages) needs to be formatted with a protocol discriminator value from the unassigned set and also carries the SAID value.

Although the crypto unit is placed between ATM user device and the network, returning of the VPCI/VCI value (there is a one–to–one mapping between virtual path connection identifier (VPCI) and VCI) selected by the network to the called user is not affected. It is possible to do this transparently as the optional `CALL_PROCEEDING` message (which extends time for connection setup) or the `CONNECT` message is simply tunneled through the crypto unit back to the ATM call originating unit. As the crypto unit does not send these messages in advance of the network issuing them, it is not necessary to generate pseudo VPCI/VCI values and remap them across the crypto unit on a per cell basis.

Once the security association is established and the session key for the VCC is cached by the two crypto units, the called side crypto unit forwards the buffered SETUP message to the called ATM user. Then the standard call setup process continues with `CONNECT` and `CONNECT_ACKNOWLEDGE` messages being sent as appropriate. The crypto units can maintain the relationship between a SAID and the associated call reference values at the two end points as they are located in between ATM user and network.

The proposed secure connection extension does not affect normal processing of call setup failures and call clearing. If the calling side network rejects a call setup request with a `RELEASE_COMPLETE` message due to reasons such as, when a VPCI/VCI cannot be allocated, requested QoS is unavailable, etc., the crypto units simply passes through the message and clears the pending security association from their lists. Similarly, call clearing due to setup failure at the called network side is handled by removing pending SAID entries at the called side crypto unit. When an ATM call is terminated the SAID value can be reused in a new secure connection after an appropriate time delay to prevent race conditions and to allow purging of cached values at crypto units.

While normal ATM connection establishment has no safeguards against an attack by masquerading network or user devices, authentication phase in secure VC setup prevents such attacks. Most importantly, a call setup rejection by the called user can occur only after a full end–to–end SAID is established. Thus, the protocol processing can be modified to validate a `RELEASE` message received from called side network due to a `RELEASE_COMPLETE` message from the called user. This has the potential to effectively prevent possible denial–of–service attacks. ATM call control messages with a *global* significance (particularly `CONNECT` and `RELEASE`) are vulnerable to this type of attack and should possibly be transmitted between the crypto units as encrypted cell streams using a looped secure tunnel between crypto unit and ATM network. However, this would require closer integration of crypto units and the interfacing ATM network switch as these messages are processed and acted upon by those network devices.

Finally, the proposed security extensions are limited for implementation in point–to–point ATM connections with non–zero return path bandwidth to support bi–directional message transfer for secure VC establishment without a separate security signaling channel.

# 4 Conclusion

For efficient processing of ATM cells and SDUs, SAR level of the ATM protocol stack needs to be implemented in hardware. Thus, the throughput of cipher systems used in crypto units must be able to satisfy the operating requirements of appropriate WAN ATM interfaces such as DS3 (45 Mbps) and E3 (34 Mbps) by providing bit streams of comparable speeds. Software implementations of symmetric key ciphers (e.g. DES, IDEA [12]) can not provide such high bit rates even if executed on very fast software. Therefore, crypto units will need to be equipped with hardware based implementations of symmetric block ciphers for cell encryption and decryption. There are other performance enhancement techniques adaptable in a crypto unit.

- Due to high bandwidth availability of ATM networks, *connection lifetimes* are correspondingly much smaller and sessions last for relatively short periods and session keys (private keys) undergo frequent changing. Although, a generic correspondence cannot be assumed among the elements in the tuple key lifetime, cell content life-

time and connection lifetime, for certain realtime applications (e.g. video streams in multimedia conferences) short key lengths may be used as an attacker will have only a short period to be active.

- If the RSA public key algorithm is used for generation of key certificates, the trusted KDC can use a relatively smaller public key exponent (say, an 8 bit number of value less than 100) to improve efficiency of the certificate verification steps. Signing of certificates using the private key exponent is a one time operation performed off–line by the KDC and therefore large exponent values can be used.

- It is possible to use a separate key for each direction in a bi–directional ATM connection by modifying the third step of the proposed security extension so that $A$ sends a session key to $B$. Although this involves an extra encryption operation for secure key transfer, by using the already received key and using a symmetric key cipher, it can be done very efficiently.

We have discussed several major issues that need careful attention in building secure ATM networks at the cell level including suitable encryption methods, secret key lengths, types of secure connections and key exchange mechanisms. We presented several approaches to constructing secure ATM networks and proposed a key agile secure SVC setup protocol that transparently extends the existing ATM call setup signaling procedure. Specific techniques were discussed to prevent reductions in QoS, reduce the use of new control message types and preserve transparency so that crypto units can be implemented as *drop–in modules* between ATM user devices and ATM network nodes.

Further work in this area is required to design cryptographic–context resynchronization methods for secure channels under error conditions (such as cell losses and bit errors) and methods to support secure channel establishment using dedicated secure signaling subchannels when a non–zero return path bandwidth is unavailable. Also, it may be possible to extend the management plane of the ATM standard to support secure connections providing standardized methods to negotiate and manage cryptographic contexts (algorithms, key length, etc.) and security policies (access controls, key life times, etc.) using out–of–band transmissions prior to actual secure SVC establishment. A simulation study of the secure protocol segment can aid in determining possible optimizations for an implementation of the protocol with a concrete algorithm.

# References


[1] The ATM Forum. *ATM User–Network Interface Specification, Version 3.1*, Sep 1994.

[2] M. Bellare and P. Rogaway. Provably Secure Session Key Distribution – The Three Party Case. In *Proceedings of the 27th ACM Symposium on the Theory of Computing*, May 1995.

[3] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung. The KryptoKnight Family of Light–Weight Protocols for Authentication and Key Distribution. *IEEE/ACM Transactions on Networking*, 3(1):31–41, Feb 1995.

[4] M. Blaze, W. Diffie, R. L. Rivest, B. Schneier, T. Shimomura, E. Thompson, and M. Wiener. Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security, Jan 1996. Available at `http://www.bsa.org`.

[5] CCITT (Consultative Committee on International Telegraphy and Telephony). *Recommendation X.509: The Directory–Authentication Framework*, 1988.

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

[7] T. ElGamal. A Public–Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *Proceedings of Advances in Cryptology–CRYPTO '84*, pages 10–18, Springer–Verlag, 1985.

[8] Federal Information Processing Standards Publications (FIPS PUB) 46. *Data Encryption Standard*. National Bureau of Standards, Jan 1977.

[9] FIPS. Proposed Federal Information Processing Standard for Secure Hash Standard. *Federal Register*, 57(21):3747–3749, Jan 1992.

[10] S. Fotedar, M. Gerla, P. Crocetti, and L. Fratta. ATM Virtual Private Networks. *Communications of the ACM*, 38(2):101–109, Feb 1995.

[11] J. Kelsey, B. Schneier, and D. Wagner. Key–Schedule Cryptanalysis of IDEA, G–DES, GOST, SAFER, and Triple–DES. In N. Koblitz, editor, *Proceedings of Advances in Cryptology–CRYPTO '96*, number 1109 in LNCS, pages 237–251. Springer, 1996.

[12] X. Lai. On the Design and Security of Block Ciphers. In *ETH Series in Information Processing*, volume 1. Hartung–Gorre Verlag, Konstanz, 1992.

[13] S. Lane. ATM Information Security – InfoGuard100, Jan 1996. GTE Government Systems, Needham, MA 02194, USA. Available at `http://www.gte.com`.

[14] S. Miyaguchi, K. Ohta, and M. Iwata. 128–bit Hash Functions (N–Hash). In *Proceedings of the SECURICOM '90 Conference*, pages 127–137, 1990.

[15] National Institute of Standards and Technology, NIST FIPS PUB 186. Digital Signature Standard, May 1994. U.S. Department of Commerce.

[16] R. M. Needham and M. D. Schroeder. Using Encryption for Authentication in Large Networks of Computers. *Communications of the ACM*, 21(12):993–999, Dec 1978.

[17] B. C. Neuman and T. Ts'o. Kerberos: An Authentication Service for Computer Networks. *IEEE Communications Magazine*, 32(9):33–38, Sep 1994.

[18] C. Partridge. *Gigabit Networking.* Addison–Wesley, Massachusetts, 1993.

[19] P. Reilly. PDH, Broadband ISDN, ATM, and All That: A Guide to Modern WAN Networking, and How it Evolved. White paper, Apr 1994. Silicon Graphics, Inc.

[20] R. L. Rivest. The MD5 Message Digest Algorithm. RFC 1321, Apr 1992.

[21] 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, Feb 1978.

[22] B. Schneier. *Applied Cryptography.* John Wiley & Sons, New York, Second edition, 1996.

[23] C. P. Schnorr. Efficient Signature Generation for Smart Cards. In *Proceedings of Advances in Cryptology–CRYPTO '89*, pages 239–252. Springer–Verlag, 1990.

[24] Secant Network Technologies Inc. CellCase Key Agile Encryption System for ATM, Sep 1996. Research Park Triangle, NC 27709, USA. Available at `http://www.secantnet.com`.

[25] A. Shimizu and S. Miyaguchi. Fast Data Encipherment Algorithm FEAL. In *Proceedings of Advances in Cryptology–EUROCRYPT 87*, pages 267–278. Springer–Verlag, 1988.

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

[27] Y. Zheng, J. Pieprzyk, and J. Seberry. HAVAL – A One–Way Hashing Algorithm with Variable Length of Output. In *Proceedings of Advances in Cryptology–AUSCRYPT '92*, pages 83–104. Springer–Verlag, 1993.