Formal validation of an ATM security context negotiation protocol

Ahmed Bouabdallah[1], Maryline Laurent-Maknavicius[2]
[1]ENST Bretagne / Rennes - bouabdal@rennes.enst-bretagne.fr
[2]INT / Evry - maknavic@hugo.int-evry.fr
France

Abstract

In February 1999, the ATM Forum international consortium approved the first version of its security specifications aiming to protect communications over Asynchronous Transfer Mode (ATM) networks by offering data confidentiality, partners authentication, etc. In order to improve those specifications, a new approach to negotiate some security services has been presented at the ATM Forum and will be integrated into the version 1.1 of the specifications. Contrary to the ATM Forum approaches, this new approach allows security partners to negotiate some new security parameters at any time during a connection in progress. The idea rests on fixed-size ATM management cells used to transfer security information. To cope with cell losses during transfer, a cell-loss recovery protocol is defined. This article describes this new approach, the cell-loss recovery protocol and the validation of this protocol using SDL tools.

1. Introduction

The Asynchronous Transfer Mode (ATM) technology success is due to its ability to support multimedia application needs offering high bit rates and real time guarantees. As a consequence, ATM is expected to transport more and more data including sensitive data. To prevent ATM traffic from being eavesdropped, tampered, and spoofed, the ATM Forum studied since 1995 the possibility to integrate the confidentiality, integrity and authentication services into the ATM technology [Tar98, Lau98]. The ATM Forum also studied the possibility for security partners (intermediate equipments or stations) to negotiate appropriate services, encryption algorithms and authentication methods in order to protect data depending on their sensitivity level and national legislation. All the security parameters necessary for ensuring data protection are referred to as a **security association** and two approaches are defined by the ATM Forum [AF99] to negotiate security associations. Since they depend on the ATM technology features, it is useful first to describe some ATM basic features.

ATM is connection-oriented meaning that two signaling messages are exchanged to setup a connection prior to any data exchange. ATM is a cell-switching technology. That is, all the information transmitted over the ATM network is in the form of 53-byte length cells. ATM is also provided with **some Operation And Maintenance (OAM)** cells usually employed to carry some alarms and ATM traffic statistics. As such, what can be noted is that the ATM traffic divides into three ATM flow types: signaling, data, and management (OAM) flows. In the ATM Forum security specifications of February 1999 [AF99], all of them are used to carry some security information, but the two first ones are the only ones used for negotiation.

The first approach to negotiate a security association consists in appending a **Security Information Element (SIE)** to both setup signaling messages. This SIE exchange allows the connection partners to negotiate the security services, and mechanisms, and to exchange the encryption keys to be used to protect their data transfer. For the encryption (or decryption) keys to remain confidential, and for the connection partners to be sure of their respective identity, existing two-way **Security Message Exchange (SME)** protocols are used.

An alternative to the signaling approach is to block the data traffic as soon as the connection is established, to realize the security negotiation through the data flow, and finally to unblock the data transfer. The negotiation is done encapsulating the same SIE than in the signaling approach, into simplified signaling messages. One advantage over the signaling approach is that the SIE length is not limited in the in-band approach, thus allowing long-length information such as keys certificates to be included into the SIE. The in-band approach is expected to use three-way SME protocols.

This paper is organized as follows. Section 2 presents a new solution to the security association negotiation which is based on a cell-loss recovery protocol. Section 3 details the cell-loss recovery protocol, and section 4 the results of its validation. Section 5 gives some conclusions, and section 6 a list of useful acronyms along with additional explanations.

2. A new approach for the security association negotiation

A new approach was elaborated in 1999 by the European project SCAN (Secure Communications in ATM Networks) [Lei99, Lau99]. It consists in encapsulating security parameters into newly defined OAM cells dedicated to security. This approach is interesting if permanent connections are considered as no security parameters negotiation through setup signaling messages is allowed. Negotiation through OAM cells is also interesting when the security policy is modified during a connection in progress.

Contrary to the ATM Forum in-band approach supporting only the three-way SME protocols (cf. section 1), the OAM cell negotiation approach enables both the two and three-way SME protocols to be used. Another advantage is that the negotiation can occur at any time during the connection, and not only when establishing a connection. As such, security partners are allowed to renegotiate security parameters as often as they need.

Since the negotiation OAM cells aim is similar to what is done in the ATM Forum security specifications 1.0 at connection setup through the SIE, the negotiation OAM approach consists in reusing the same SIE and encapsulating the SIE within negotiation OAM cells. Since the SIE may be bigger than the 46-byte payload offered in OAM cells, SIE fragmentation should take place before its encapsulation into negotiation OAM cells. Since cell losses may occur during OAM cell transfer, a **cell-loss recovery protocol** based on sequence numbers and acknowledgment OAM cells is defined.

Once the negotiation completed and the new security association is ready to be activated, both partners should send some **Security Association Changeover** (SAC) cells to change from the old security association to the new one. That is, once partners are ready for security association changeover, one partner should block its data traffic, sends a group of SAC cells, and unblock its data traffic which is then protected with the new security association. The security association changeover is unidirectional meaning that each partner can at any time realize the changeover.

What should be noted is that this new negotiation approach is based on the principle adopted by the ATM Forum for the data encryption keys exchange. This principle consists in sending the same OAM cell several times (at least 3 times) to be sure that at least one cell arrives at the destination. To preclude possible bursts from erasing all the transmitted OAM cells flows in the network, cells are sent with a delay between each transmission. This principle applies to the transmission of acknowledgment and SAC cells. This principle may appear as unreliable for people familiar with the validation of protocols, however this principle was especially defined to cope with the ATM multicast traffic where connections are unidirectional and as such no feedback is allowed from the receiving partner.

What we propose next is to describe the cell-loss recovery protocol and a validation of it. More precisely, section 3 details the protocol, and section 4 presents the results of the validation for the two-way SME protocol.

3. Informal description of the cell-loss recovery protocol

The purpose of this protocol is to ensure that a SIE being exchanged over the network encapsulated into negotiation OAM cells is received with no losses and correctly ordered. As such, when fragmenting a SIE, each SIE fragment is numbered with a sequence number and each negotiation OAM cell includes the sequence number associated to the SIE fragment it carries. A group of negotiation OAM cells are acknowledged at the same time by an acknowledgment OAM cell which includes the sequence number of the last negotiation OAM cell received.

The complexity of the protocol is due to possible losses occurring in the network, and the SIE processing time being undefined since dependent on the SME protocol security mechanisms used. The difficulty is that SAC cells should only be sent at the end of the SIE exchanges, and should only take place after the last SIE is processed for the data decryption key to be retrieved and ready to be used by the receiving partner. Because of it, and because of the specific role of the acknowledgment cell in the last SME flow, three kinds of acknowledgment cells are defined:

- Intermediate acknowledgment cells are used to acknowledge negotiation OAM cells which are not the last one in the last SME flow. For bandwidth optimization purpose, their transmission is done when a cell loss is detected, or when no negotiation OAM cells have been received for a long time while the SIE is not fully received yet.

- The **Message Exchange Complete (MEC)** cells are sent to acknowledge the last negotiation OAM cell of the last SME flow, and to indicate that no other SIE should be exchanged to specify that a partner is ready for a data encryption key changeover processing.
- The **Ready for Key Changeover (RKC)** cells are sent to indicate that a partner is ready for the data encryption key changeover processing using SAC cells.

To be sure that at least one acknowledgment cell arrives to the destination, a number of similar acknowledgment OAM cells are sent with a delay between their transmission (cf. section 1).

In order to preclude that both partners wait indefinitely for negotiation OAM cells or acknowledgment cells because of possible losses during transfer, two timers are introduced: T103 and T104. T103 (respectively T104) is the maximum delay between a full SIE transmission and the acknowledgment cell reception (resp. intermediate acknowledgment cell transmission and SIE reception). When T103 elapsed, the SIE is fully or partly transmitted again and T103 is armed again. Likewise, when T104 expires, intermediate acknowledgment cells are transmitted again and T104 is rearmed. To avoid infinite SIE (resp. intermediate acknowledgment cells) transmissions, a maximum number of SIE transmissions MAX-SIE-RETRY (resp. MAX-ACK-RETRY for the maximum of intermediate acknowledgment cells retransmissions) is defined and should be fixed by the constructor or by the users as part of the local security policy. One additional timer T105 is defined to avoid deadlocks due to the SIE processing time remaining unknown. T105 value is chosen great enough so that its expiration occurs only if the partner or the line is down, thus resulting in the connection release.

Once the negotiation is completed and the corresponding acknowledgment cells are sent, the new data encryption keys exchanged by the SME protocol is activated by sending a group of SAC cells.

We focus our approach in what follows on the two-way SME protocol. Partners involved are the initiator which initiates the security association negotiation, and the responder which are represented by nodes A and B indifferently as depicted in figure 1. Nodes A and B correspond to remote processes exchanging some security information through the cell-loss recovery protocol. They may be either the negotiation initiator or the responder. The communication channel is a cell-loss bidirectionnal channel interconnecting nodes A and B. Environments of A and B are local processes interacting with A and B to ask for a new security association negotiation, or to indicate that the processing of the SIE encapsulated into the OAM cells is over. The nodes should also forward to their environments the SIE received or the indication that the negotiation is over and that the security association changeover can be processed.

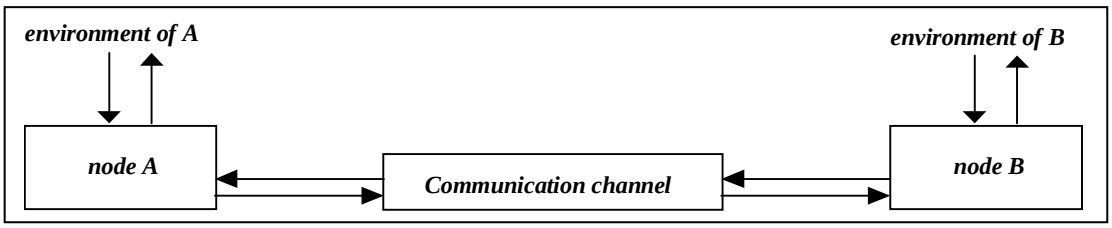

Figure 1: The basic structure of the system

The signals exchanged between nodes and between a node and its environment are described below.

Signals	Meaning
Init_Negotiation	The environment initiates a security context negotiation by sending this message to the initiator node
OK_for_sending_SAC	A node notifies the environment that the renegotiation is over so that it is now possible to send a group of SAC cells
SIE	A group of cells including the fragments of a Security Information Element
ACK	A group of intermediate acknowledgment cells
MEC	A group of acknowledgment cells indicating that the Message Exchange is Complete
RKC	A group of acknowledgment cells indicating that a node is Ready to Key Changeover

Table 1 : Signals

We describe in what follows a scenario, where node A, on the demand of its environment, initiates a negotiation by sending the first flow of security information cells (3 cells IE1); the node B responds by sending its security information element (2 cells IE2). In order to understand the basic principle of the protocol, we make the assumption that in this particular scenario, messages are not lost.

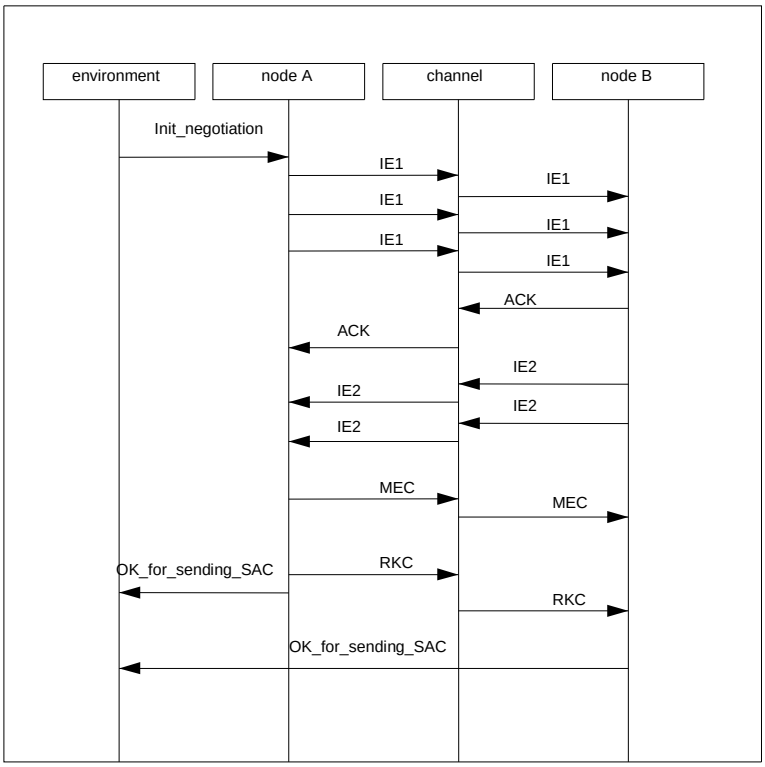

Figure 2: The two-way SME protocol without loss of message

As usual in protocol design,we introduce timers (T103, T104, T105) and acknowledgements (ACK, MEC, RKC) to cope with the possible loss of messages. We have no place in this paper to present scenarios illustrating the loss of the various messages and the solutions we provide, but the reader can easily derive them from the automaton provided in the next part of the paper. Collision is another interesting problem arising when both nodes decide in nearly the same time to initiate a negociation. When a collision is detected, the current negotiations are stopped and a new one is randomly reinitiated.

4. Formal specification of the cell-loss recovery protocol

The protocol is completely symmetrical : the behavior of node A and node B are identical. Figures 2 gives an abstract representation of the behavior of a node using a finite states machine based approach. The initial bold-written state is *idle*. An arrow represents a transition between two states. Transitions are labelled with at least one communication event : the name of the message preceded by **?** for a receiving event and by **!** for a sending event. A timer expiration is considered as a receiving event. When an arrow is labelled by several events, the order of their occurrences can be deduced from the direction of the arrow. Sometimes the occurrence of different events can produce the same transition, in such a case one arrow labelled with the disjunction of the different events is used. This automaton synthesizes precisely the careful examination we proceeded to allow the protocol to cope with the various losses of messages or collision detections. The automaton concerning the behavior of the lossy communication channel is trivial and then omitted. From the automaton of figure 2, we can easily derive an SDL system representing the complete specification of the protocol.

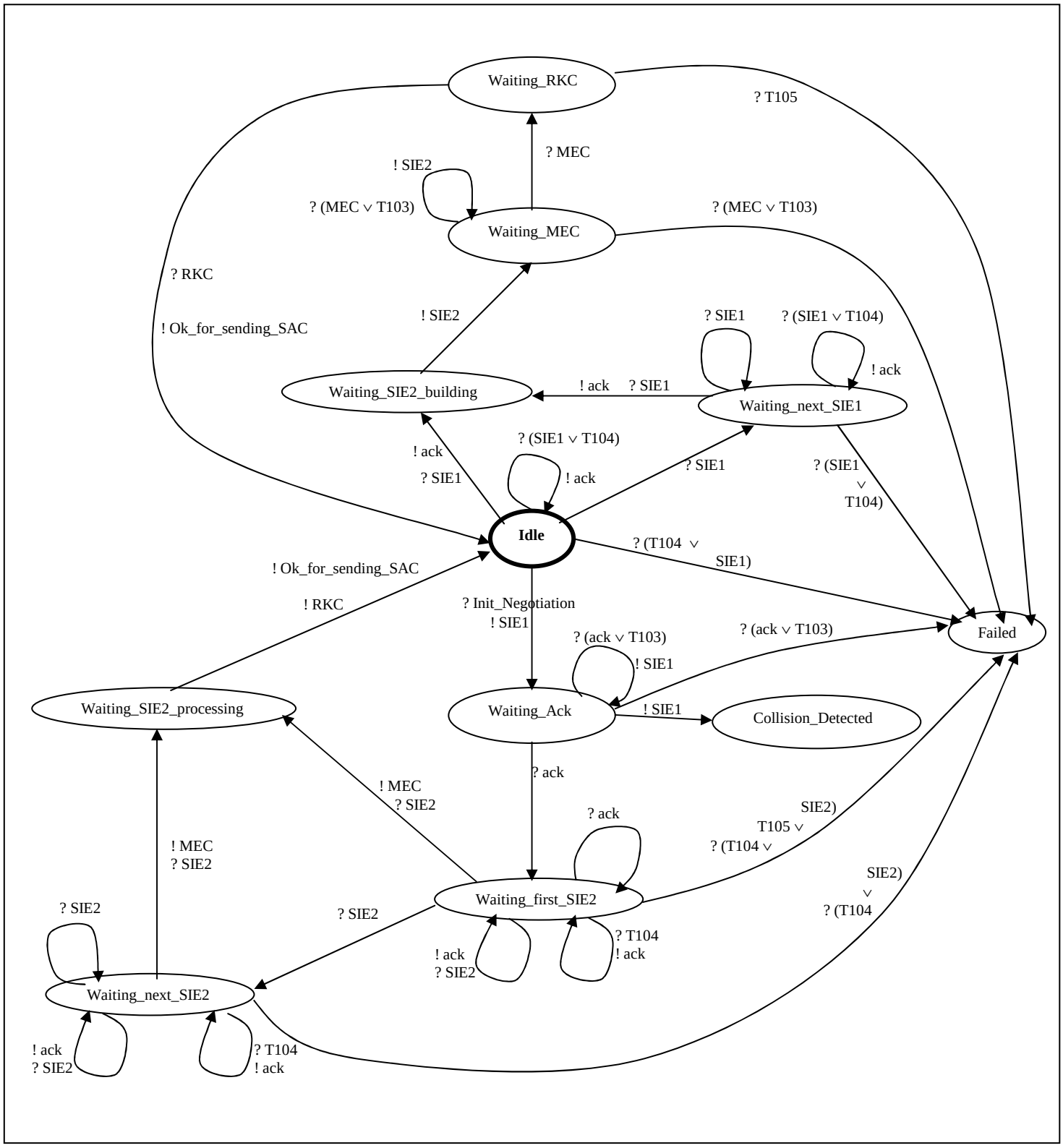

Figure 2 : Finite states machine abstracting the behavior of a node

5. Protocol validation

The principle of our validation rests on an exploration of the state space of the specified system [Hol91]. A state of the system is a collection of the control and local state (local variables, procedure calls, active timers) of each process plus the state of their respective input queues. The exploration of the states space consists in trying to reach every possible state starting from an initial state and analysing them. A state graph containing all reachable states and execution sequences is generated recursively by applying depth-first-search or breadth-first-search algorithms. Our validation has been conducted using the Telelogic tool SDT [Tel00]. This tool offers the possibilities to edit specifications in SDL96 [ITU-T97][Ells97], to simulate them, to produce code in various programming languages, to generate test sequence, etc. The validation capabilities of SDT consists in exploring the system states space randomly or following the depth-first bit-state algorithm [Hol91]. With this algorithm, already analyzed system states are stored in a hashing-table. Whenever a system state is analyzed by comparing its hash-value with already existing hash values, a collision risk exists. The collision risk which must be very low, is measured in percent. It is inversely proportional to the size of the hash table, which must be therefore very important for non trivial systems. The analysis of a system state during an exploration consists in evaluating predefined and/or user-defined properties. Predefined properties provided by SDT are :

- Deadlock,
- Livelock,
- Non specified signal consumption,
- The maximal user-defined length of the input queues, is exceeded,
- Static semantics errors (range, index, decision … errors).

User-defined properties can be :

- Predicates defined as boolean expressions,
- Temporal claims expressed with specific automata called observers.

A property satisfaction generates a report containing informations about the corresponding system state. The tool can be configured such that :

- the fulfillment of a property will be reported to the user and the search will be aborted,
- a report will be generated and the behavior tree below this state will not be further examined,
- only a report will be generated without affecting the exploration.

Others parameters of the SDT validator can affect the exploration and therefore be adjusted in several ways. We adopted the initial values of the various parameters of the validator, excepted for the ones concerning :

- the search-depth : it defines the depth at which the state space exploration should be aborted. It was set to 2000 which was strictly greater than the depth of the behavior tree of the system.
- the scheduling discipline : it defines which process instances of the ready-queue are selected for further execution. The ready queue is a queue, which contains all process instances that received a signal and are ready to execute a transition. The ready-queue is sorted by priority and insertion time. The value *All* defines that all process instances in the ready queue can be executed. The value *First* defines that only the first process instance being in the ready-queue is selected for execution.
- the priorities of the various events (internal events, inputs from the environment, timeout events, channel outputs, spontaneous transitions) ranging from 1 the highest to 5 the lowest.
- the assumption concerning the time (zero or undefined) taken for the execution of a SDL symbol. When set to zero, it is assumed that all actions performed by processes are infinitely fast compared to the timer values.

Varying these parameters clearly directly affects the size of the state space we have to explore. Using the bit-state algorithm based exploration [Hol91][Tel00], we get the following results on a HP_UX B132L+ with 768Mbytes of main memory.

Validation 1

Bit state options :
Hash table size : 268435455 bytes
Search depth : 2000

Event priorities : 1 the highest to 5 the lowest
(Internal events | Input from ENV | Timeout events | Channel output | Spontaneous transition) = (1|2|2|1|2)

State space generation options
Scheduling : First
Symbol time : Zero

** Bit state exploration statistics **
No of reports: 0.
Generated states: 63137877.
Truncated paths: 0.
Unique system states: 37778606.
Size of hash table: 2147483640 (268435455 bytes)
No of bits set in hash table: 63184562
Collision risk: 2 %
Max depth: 456
Current depth: -1
Min state size: 260
Max state size: 484
Symbol coverage : 89.81

Validation 2

Bit state options :
Hash table size : 268435455 bytes
Search depth : 2000

Event priorities : 1 the highest to 5 the lowest
(Internal events | Input from ENV | Timeout events | Channel output | Spontaneous transition) = (1|2|2|1|2)

State space generation options
Scheduling : All
Symbol time : Zero

** Bit state exploration statistics **
No of reports: 0.
Generated states: 101705295.
Truncated paths: 0.
Unique system states: 49382515.
Size of hash table: 2147483640 (268435455 bytes)
No of bits set in hash table: 82447811
Collision risk: 3 %
Max depth: 481
Current depth: -1
Min state size: 260
Max state size: 468
Symbol coverage : 89.81

Validation 3

Bit state options :
Hash table size : 268435455 bytes
Search depth : 2000

Event priorities : 1 the highest to 5 the lowest
(Internal events | Input from ENV | Timeout events | Channel output | Spontaneous transition) = (1|2|2|1|2)

State space generation options
Scheduling : All
Symbol time : Undefined

** Bit state exploration statistics **
No of reports: 0.
Generated states: 229195034.
Truncated paths: 0.
Unique system states: 111451698.
Size of hash table: 2147483640 (268435455 bytes)
No of bits set in hash table: 165312110
Collision risk: 7 %
Max depth: 501
Current depth: -1
Min state size: 260
Max state size: 468
Symbol coverage : 90.43

Validation 4

Bit state options :
Hash table size : 268435455 bytes
Search depth : 2000

Event priorities : 1 the highest to 5 the lowest
(Internal events | Input from ENV | Timeout events | Channel output | Spontaneous transition) = (1|1|1|1|1)

State space generation options
Scheduling : First
Symbol time : Zero

** Bit state exploration statistics **
No of reports: 0.
Generated states: 1009726119.
Truncated paths: 0.
Unique system states: 401525949.
Size of hash table: 2147483640 (268435455 bytes)
No of bits set in hash table: 600999602
Collision risk: 27 %
Max depth: 1482
Current depth: -1
Min state size: 260
Max state size: 620
Symbol coverage : 89.81

The SDT validator hasn 't detected any predefined error in the specification. Concerning the symbol coverage which never equals 100%, it appears clearly that some transitions present in the specification, never occur

during the execution of the system and must therefore be simplified. When trying to reduce the collision risk to 0% , we discover some unknown constraint concerning the SDT validator : if we give to the hash table a size greater than 268435455 bytes, the statistics of the exploration become uncommon (negative value for the number of generated states, …). This unexpected limitation of the SDT validator has been pointed out to Telelogic. The non-zero collision risk clearly is an avatar of the famous states explosion problem, which decreases the value of this validation. Trying to avoid it, has led us to combine model-checking with others techniques like abstraction, infortunately SDT doesn't offer a rigorous setting to conduct such experiments, and we must therefore use other tools, merely academic, to proceed to these approaches, which are beyond this paper.

6. Conclusions

In this paper, we show some validation results for a cell-loss recovery protocol designed in the European project SCAN to offer users the possibility to negotiate security services during a connection in progress. Those results are essential since the protocol will be integrated into the ATM security specifications version 1.1. This protocol was presented under contribution [Lau00] at the Vienna (USA) ATM Forum meeting and was welcomed by the security working group. Moreover, the protocol is implemented in a prototype [Lau99][Lei99] which is likely to be commercialized in the beginning of 2001. The validation described in this paper is limited to the two-way SME protocol; the three-way SME protocol has been formally specified but its validation using the same approach unsurprisingly came up again the states explosion problem, so notably decresing the value of the validation. An other approach using rigorous abstraction based techniques, supported by academic tools, is currently experimented.

7. References

[AF99] ATM Forum, "ATM Security Specification Version 1.0", February 1999.

[Ells97] J. Ellsberger, D.Hogrefe, A. Sarma, "SDL : Formal Object-oriented Language for Communicating Systems", Prentice Hall, 1997.

[Hol91] G.J. Holzmann, "Design and Validation of Computer Protocols", Prentice Hall, 1991.

[ITU-T97] ITU-T Recommendation Z.100, "Specification and Description Language SDL", 1997, http://www.itu.ch.

[Lau98] M. Laurent, O. Paul, P. Rolin, "Securing Communications over ATM Networks: The Remote ATM Private Networks Inter connection Example", *Annales des télécommunications*, N°9-10, September-October 1998.

[Lau99] M. Laurent, A. Bouabdallah, C. Delahaye, H. Leitold, R. Posch, E. Areizaga, J.M. Mateos, « Secure Commmunications in ATM Networks », Proc. of "15th Annual Computer Security Applications Conference (ACSAC'99), IEEE Computer Society, 6-10 December 1999, Scottsdale, Arizona (USA).

[Lau00] M. Laurent, T.D. Tarman, "Security services negotiation through OAM cells", ATM Forum/00-0159-R1, 16-20 october 2000, Vienna, Virginia (USA).

[Lei99] H. Leitold, R. Posch, E. Areizaga, A. Bouabdallah, M. Laurent, J.M. Mateos, O. Molino, "Security Services in ATM Networks", Interoperable Communication Networks ICON Journal, Baltzer Science Publishers, 1999.

[Tar98] T.D. Tarman, R.L. Hutchinson, L.G. Pierson, P.E. Sholander, E.L. Witzke, "Algorithm-Agile Encryption in ATM Networks", *IEEE computer*, Vol.31 N°9, pp. 57-64, September 1998.

[Tel00] Telelogic Tau 4.0 "SDL Suite", Avril 2000, http://www.telelogic.com.