Considering Security and Quality of Service in SLS to improve Policy-based Management of Multimedia Services

Sandrine Duflos*, Brigitte Kervella
Laboratoire d'Informatique de Paris VI
104, avenue du Président Kennedy – 7th floor, 75016 Paris, France
{sandrine.duflos, brigitte.kervella}@lip6.fr

Valérie C. Gay
UTS Faculty of IT
PO Box 123, Broadway NSW 2007, Sydney, Australia
Valerie.Gay@uts.edu.au

***Abstract*—This paper proposes to improve policy-based management by integrating security parameters into the Service Level Specification (SLS). Integrating those parameters in the QoS part of the Service Level Agreement (SLA) specification is of particular importance for multimedia services requiring security since QoS is negotiated when the multimedia service is deployed. Security mechanisms need to be negotiated at that time when sensible multimedia information is exchanged. In this paper we show that including security parameters in SLA specification improves the negotiation and deployment of security and QoS policies for multimedia services. The parameters this paper proposes to integrate have the advantage to be understandable by end-users and service providers.**

Keywords- SLA; SLS; Security; Security of Service (SoS); QoS; SLA management

I. Introduction

Today, multimedia services are available to end-users over the Internet. They allow the exchange of more or less sensitive information needing different levels of protection. These services have generally Quality of Service (QoS) requirements but also security requirements depending on the type of the service used and the sensibility of exchanged data.

The protection of exchanges is usually achieved using security mechanisms and protocols. However, they have an impact on resources and delay and therefore they can decrease the QoS. The Centre for Information Systems Security Studies and Research (Monterey California) published documents highlighting these issues [1, 2].

To provide the best possible quality of service for multimedia services, security needs to be negotiated and deployed together with QoS since security processing consumes resources from end-user (EU) and provider (e.g.: CPU, throughput, delay) and has therefore an impact on the QoS. We therefore suggest to specify both QoS and security in the Service Level Agreement (SLA) Specification, called Service Level Specification (SLS), for negotiation and deployment of security and quality of service policies. A SLA is a specific contract between a Service Provider (SP) and its customers [3]. It contains general information, to identify the customer and the service to provide, and technical information, corresponding to the SLS, to identify the required quality for those services [3, 4]. The integration of QoS in SLS is the subject of many projects and publications [5, 6, 7, 8, 9, 10, 11]. The SLS used or defined in these projects are not explicitly considering security.

Our objective is to identify the essential parameters to extend current SLS specification for security management. These parameters must express security and also consider information on the impact of security mechanisms on the QoS to improve its management. Integration of such parameters would improve policy-based management of QoS with the generation of network policies that ensure the reservation of adequate amount of resources for both security and QoS needs. In addition, integration of security parameters within SLS would enable SP to propose Security of Service (SoS) to their customers. This allows customers to get the level of security required for their services, without needing to be experts in security and without necessary having the appropriated security mechanisms available at their host. To be optimal, the security parameters need to be understandable by both customer (expert or not) and service

provider. This suggests that at least two levels of interpretation must be included: one abstract level that can be qualified, understandable by non-expert EU and a precise level that can be quantified, interpretable by the expert EU and its SP to negotiate the service configuration and deployment, considering the interactions between security and QoS policies. To do this, we identify a set of security parameters interpretable at both qualitative and quantitative levels, each parameter being possibly specified by a set of sub-parameters to obtain a quantitative guarantee.
To select the SoS parameters we based our work on the ISO 7498-2 Recommendation describing in details the different security services and mechanisms available for each OSI layer [12], and on the IPsec and TLS security protocols commonly used to secure the Internet [13, 14].
This paper provides in Section 2 a state of the art on SLS for QoS and SoS management. Section 3 discusses the SoS parameters that impact the QoS and that should be integrated in the QoS part of an SLS to consider interactions between SoS and QoS. Section 4 describes how to integrate those SoS parameters in an existing SLS. Section 5 presents existing studies and discusses issues on the influence of security mechanisms on network and service performance to improve policy-based management of QoS and SoS. Section 6 concludes on open issues and perspectives.

II. Service Level Specifications for QoS and Security Management

In this section, we first describe related work on SLS for QoS management, then related work on security for SLS and finally we justify our decision to integrate security in SLS.

A. Service Level Specification for QoS Management

A lot of research projects deal with SLS for QoS management. We focus here on Aquila [5, 6], Cadenus [7], Mescal [8], Sequin [9] and Tequila [10, 11].
The Aquila, Cadenus and Tequila consortia provide IP Premium services over the Internet [15]. These three projects worked together to define an SLS template tailored to IP networks. The resulting SLS consists of the following units:

- The **common unit** contains general information to identify the SLA context (about the provider, the customer, the service type, the time and the period of SLA applicability).
  - The **topology unit** gives information on the points to access the provider domain, and the relationship of traffic generation and consumption relationship these points.
  - The **QoS unit** describes the traffic streams subject to the SLA and the nature and extent of service differentiation provided to them.
- The **monitoring unit** defines a set of parameters needing to be collected and reported to the customer in order to be compared with the SLA ones.

This SLS template is a de facto standard and a lot of IETF drafts have been produced [16, 17, 18, 19]. Furthermore, it is used in other projects such as Sequin, where the Tequila work is handled to provide an SLS template for the IP Premium service between National Research and Education Networks and the trans-European research backbone GEANT [20]; or Mescal, where the Tequila SLS is used for inter-domain interactions aiming at negotiating the QoS between Customer and SP and between two SPs, while the Tequila project focused mainly on Customer-SP interactions [21].

B. Service Level Specification for SoS management

Little work has been conducted on security integration in SLS. The Arcade Project is an exception [22, 23]. It proposes security parameters by succinctly defining a network level security SLS specific to a Linux implementation of the IPsec protocol in order to map the SLS onto the IETF/DMTF IPsec Configuration Policy Information Model [24]. Two categories of parameters are distinguished in this SLS: the *SLA-dependent parameters*, inherent to the SLA; and the *SLA-independent* parameters, that gather the parameters reusable in others SLAs, where a similar service is required.

C. Our choices to integrate security in SLS

Of the studied projects none is considering both quality and security of service. We use Arcade and Tequila projects as basis for our work. The SLS defined in the Tequila project represents a complete specification for the IP service and is a de facto standard. However, it is specific to QoS management and does not include security parameters despite the impact of security processing on the quality of the service. It is a good base to add security parameters. The Arcade project is interesting for its security SLS but it does not consider QoS parameters and the security ones are specific to a Linux implementation of IPsec.

III. Identification of SoS parameters for SLS extension

In this section, we present our solution. We take selected security parameters into account within an SLS to improve QoS negotiation and management.

An SLS consists of the technical terms of an SLA, a contract negotiated between a customer and a SP. These terms are generally expressed at a low level and are not necessarily understandable by non-expert EU. The introduction of high level parameters will guarantee that the negotiated service corresponds to the EU's expectations. However more expert users would prefer be more precise for their SoS negotiation and would wish to handle specific parameters.

To solve this problem, we propose parameters suitable for both expert and non expert users. Then we adapt to security the principle of qualitative and quantitative guarantees introduced in the Tequila work for QoS. Therefore we must identify a set of security parameters that can be interpreted at both qualitative and quantitative levels, each parameter being possibly specified by a set of sub-parameters to obtain a quantitative guarantee.

The following sections identify the parameters associated to each interpretation level.

A. Qualitative SoS Parameters

Qualitative SoS parameters allow non-expert EU to express their security expectations in a simple manner. They are based on the common security services described in the ISO 7498-2 Recommendation [12]. They represent the objectives to achieve and are invoked to satisfy security policy or user requirements. They can be associated to the qualitative guarantees and are understandable by non-expert EUs. These services are confidentiality, integrity, authentication, and non-repudiation. The non-repudiation service provided by network security protocols corresponds to the 'proof of data emission'.

The selection of a service cannot be limited to a 'security/no security' choice. A security service can be provided by several algorithms having different properties. A classification of these confidentiality algorithms can be done in relation to the service itself. Consequently the strength of a security service varies according to the employed algorithm. Therefore we introduce different levels for the *confidentiality*, *integrity* and *authentication* services. Four qualitative parameter values are then identified: 'high', 'medium', 'default' and 'no'. They indicate the appropriate algorithms to include in the quantitative SLS parameters. *Proof of data emission*, which corresponds to the non-repudiation service, is identified with an 'on/off' choice.

Three extra optional parameters are provided. They are derived from protocols abilities and are proposed to more expert users. These parameters are:

- *Protection against the replay* (or no-replay) discards data already received. It is useful for example when an access code is sent. This prevents an intruder to reemit packet to access unauthorised data. Its value can be 'on' or 'off' and its default value is 'on'. Only IPsec allows 'off' value.
    - *Tunnelling* corresponds to the creation of a tunnel between two points of the network route in order to mask source and/or destination addresses. It is specific to IPsec. It is an 'on/off' parameter with 'off' as a default value.
- *Security protocol* allows the EU to select a specific protocol. If this parameter is specified, and the two other optional parameters are not, they get their default value.

B. Quantitative SoS Parameters

Quantitative SoS parameters are used to configure the chosen security protocol as precisely as possible. They are based on ISO 7498-2 Recommendation security mechanisms, used to implement combinations of security services. These mechanisms represent the underlying algorithms and functions useful to achieve the security objectives. They relate to the quantitative guarantees understandable by expert EU and SP.

1) Parameter for the selected protocol: This parameter identifies precisely the selected protocol(s) for the security.

2) Parameters for the confidentiality service: The objective of confidentiality parameters is to configure the chosen confidentiality mechanism, generally a symmetric cipher algorithm, as precisely as possible. Six parameters are used:

- The *name of the selected algorithm*.
    - The *algorithm category* (per block or per flow). In per block algorithms, data are divided in several blocks to be ciphered.
    - The *block size* depends on the algorithm. It is expressed in bits and can be fixed or variable. In per flow algorithms, data are ciphered as the flow goes along. Per flow algorithm can be considered as a per block algorithm with a block size equal to one.

- The *algorithm operation mode* determines algorithm operation process, for example if the process starts with an initialisation vector or not. Four standardised modes are defined in [25]: the Electronic Code Book (ECB), the Cipher Block Chaining (CBC), the Cipher FeedBack (CFB) and the Output FeedBack (OFB). Other modes can also be specified like CTR (counter) of the NIST [26].
- The *key length* can be fixed or modifiable according to the chosen algorithm. It is expressed in bits.

- The *round number* can be fixed or modifiable. It corresponds to the number of times the cipher transformations are applied.

Except for the first one, these parameters are necessary to configure and determine the strength of a cipher algorithm.

3) Parameters for authentication, integrity and non-repudiation services: Integrity and authentication are generally handled together and are not differentiated in the security protocols we consider in this paper.

The joint provision of authentication and integrity is done through a MAC (Message Authentication Code) or a digital signature. The MAC uses an unkeyed hash function to produce a digest according to entered key and message. The digital signature uses the public key principle to authenticate the message.
The non-repudiation is generally provided by digital signature and data integrity: the sender cannot deny the transmission of signed data. Therefore the guarantee of this service implies the use of a digital signature algorithm for authentication.
We identify five extra parameters to provide integrity, authentication and non-repudiation:

- The *authentication type* determines the use of a MAC or a digital signature.
  - The *authentication algorithm* is the name of the selected algorithm: HMAC (Hashed Message Authentication Code), DSA (Digital Signature Algorithm), etc.
  - The *key length*.
  - The *hash function* is processed to produce the digest.
- The Non-Repudiation selection (on/off).

4) Parameters for protocols options: The no-replay option uses a sequence number different for each packet or record. This number is a 64 bits number in TLS and currently a 32 bits one in IPsec. However, the last advances on the protocol permits to extend this number to 64 bits [27, 28]. Therefore the parameter of no-replay is summed up in the choice of the size in bits of the sequence number. The default value for the sequence number length is 32 bits for IPsec and 64 bits for TLS.

The IPsec protocol provides, in addition to the common security services and the no-replay protection, the possibility to create secure tunnels. The selection of this option implies the use of the IPsec protocol to secure the communications.
The creation of secure tunnels requires the knowledge of the gateways or the hosts where the tunnel begins and ends. So parameters for tunnelling consist of *IP address* specification for the tunnel start and end.

IV. Extension of the Tequila SLS template with security parameters

This section describes how we integrate the SoS parameters identified in Section 3 in the Tequila SLS template to improve QoS policy-based management of secure services.
Supplying SoS is a quality guarantee for multimedia services. It is essential to consider security as a parameter for quality to provide a good quality to the service. Also, security processing acts on the quality of the service by increasing resource consumption, induced delay and traffic load. Considering security as part of QoS makes it possible to take into account the impact of security on QoS. This is also a logical placeholder since security and QoS are applied to the same traffic. The traffic descriptor sub-unit contains combination of DiffServ Information, Source information, Destination Information and Application Information [19]. The Source, Destination and Application Information is necessary for security protocol configuration [13, 14]. Also, since traffic is already described in this sub-unit of Tequila SLS QoS unit, useless repetition of traffic description is avoided.
To introduce SoS parameters in the SLS, we choose to add a new sub-unit to the QoS unit of the Tequila SLS template, the SoS parameters sub-unit, rather than adding a specific security unit. This sub-unit contains the common parameters plus the selected security protocol and the protocol options described in Section 3.
Table I presents the extension of the Tequila SLS QoS unit for security with quantitative guarantees. Only the two sub-units useful for SoS management are shown. The other QoS sub-units are outside the scope of this paper. The additional parameters are in bold. The first column presents the sub-unit. The second and third ones correspond

respectively to the qualitative and associated quantitative parameters, and the fourth contains examples of associated selected values.

In **bold:** proposed SoS parameters structure and example of quantitative SoS parameters

Sub-Unit	Qualitative Parameters	Quantitative Parameters	Value
Traffic descriptor	Diffserv Information	DSCP	11101
	Address	Type	IPV4 Address
	Address	Type	IPV4 Address
	Protocol number	6	
SoS parameters	**Security protocol**	**Value**	ESP (or 50)
	Alg Name	DES	
Authentication	**Alg Type**	MAC	
Integrity	**Hash function**	MD5	
Tunnelling	**Source address**	**Type**	IPV4 Address
		IPV4 Address	
	Sequence Number length	32 bits	

The negotiated values associated to SoS parameters can be either qualitative or quantitative depending on the EU expertise. In the first case, a level, an on/off choice or a default value can be attributed to the parameters. In the second case, a subset of specific parameters is associated to the common ones except for the non-repudiation parameter which is ‘on or off’ depending on the type of authentication algorithm. Therefore, if non-repudiation is selected, the authentication algorithm must be a digital signature.

During the negotiation, it is possible not to select any of the security parameters or to use only part of it. For example, the required SoS can be confidentiality only. In this case, the common and optional parameters that are not selected can be qualitatively specified with the ‘no’, ‘on’ or ‘off’ value, or not specified at all. In the last case, the options default values are attributed according to the security protocol selected.
In case quantitative values are attributed, as presented in Table I, the SP can directly consider the SLS to configure security of its network. However, in case of qualitative agreements, the SP must interpret the values. This interpretation is done through mapping tables such as Table II and Table III, where a level corresponds to a set of algorithms to choose from. This choice is also possible with quantitative guarantees. Several alternatives may be associated to a particular SoS parameter.
Tables II and III present respectively some examples of quantitative parameters for confidentiality, authentication, integrity and non-repudiation. The ‘no’ qualitative level corresponds to the ‘NULL’ algorithm. This one is used for confidentiality, authentication and integrity. It specifies no encryption or no authentication and integrity appliance. This NULL value is derived from IPsec and TLS protocols RFC [13, 27, 28, 14]. The ‘Security Protocol’ column allows the identification of the protocol(s) supporting the algorithms.

Example of a mapping table for confidentiality

Level	Name	Category	Mode	Block size	Key length	Round number	Security protocol
High	AES	Block	CBC	128	128	9	ESP,TLS
	Block	CBC	64	192	48	ESP,TLS	
	IDEA	Block	CBC	64	128	8	ESP,TLS
Medium	RC5	Block	CBC	64	128	16	ESP
	Block	CBC	64	128	16	ESP	
Default	DES	Block	CBC	64	56	16	ESP,TLS
	Block	EBC	64	40	18	TLS	
		Block	EBC	64	40	16	TLS

	DES						
No	NULL						

Example of a mapping table for authentication, integrity and non-repudiation

Level	N-R value	Auth Type	Auth Name	Auth Key length	Hash function	Security protocol
High	Off	MAC	HMAC	128	SHA-1	AH, ESP,TLS
	MAC	HMAC	128	RIPEMD_160	AH, ESP	
Medium	Off	MAC	HMAC	128	MD5	AH, ESP,TLS
Default	Off	MAC	HMAC	128	MD5	AH, ESP,TLS
No	Off		NULL		NULL	

The SLS we propose is negotiated between an end-user and its SP. The quantitative parameters, derived from SLS or obtained from mapping tables, are used by SP to configure their networks. To do this, the SP translates the SLS into policies to provide the required security. Nevertheless, mapping tables and policies offer a choice among several SoS solutions, each having a different impact on QoS. This impact will need to be taken into account while negotiating QoS.

V. Security influence on network and service performance

This section discusses the influence of security on network and service performance in the context of our SLS parameters for Security of Service. We base our study on work conducted at the Centre for Information Systems Security Studies and Research (Monterey, California). A study on the impact of security on resource consumption is proposed in [1, 2]. The studied resources are CPU, memory and bandwidth. For each resource, two types of costs are distinguished: initialisation and streaming costs. The initialisation represents the initialisation phase of the security mechanism process, and the streaming represents the data packet emission. We consider the resources (CPU, memory and bandwidth) and distinguished costs with each SoS parameter specified for security SLS to identify how they influence network and service performance.

No-replay is the parameter consuming less resource. It acts on CPU and memory during initialisation phase for the sequence number initialisation to zero. While streaming, CPU and memory are again consumed due to sequence number increment and verification. Bandwidth is also necessary for sequence number transfer, since it is sent in the protocol header with IPsec and in the authentication data with TLS.

The **tunnelling** has also a few impacts on CPU and memory during streaming phase. It consists of generating a new IP header for each packet. On the other hand, the addition of a new header raises the bandwidth streaming cost. Actually, adding a header increases at least 20 bytes the size of the IPv4 packet, and at least 40 bytes for IPv6.

Authentication, integrity and non-repudiation are considered at the same time because they are provided together by IPsec and TLS. Generally the HMAC authentication algorithm is used in association with a hash function to provide a digest. This has an influence on CPU and memory consumption in the initialisation phase as well as while streaming. Bandwidth is also consumed with the sending of the digest. The quantity of consumed bandwidth depends on its size, which can be 0, 96, 160 or 192 bits for IPsec and 0, 128 or 160 bits for TLS.

Confidentiality has on the one hand an important impact on CPU and memory. Encryption algorithms have generally an initialisation phase where the key is used to initialise specific tables (key expansion) and where an initialisation vector used to start the encryption, is produced. While streaming, CPU and memory are used for enciphering and deciphering data. On the other hand, confidentiality itself consumes little bandwidth during the streaming phase, only for the initialisation vector sending that is generally equal to the size of a block. However, padding can be added in IPsec, and it can stretch to 255 bytes. Therefore the size of sent data and consequently bandwidth consumption increases significantly.

Finally, the **protocol** itself has mainly an influence on resources during initialisation phase with the security context establishment (key generation, negotiation of used algorithms, etc.), which consumes CPU, memory and bandwidth as well. Then the protocol processing influences CPU and memory streaming costs and the specific protocol data (IPsec/TLS header) increases cost in bandwidth.

Figure 1 summarises the results with a down/top classification of resource consumption for our SoS parameters.

Figure 1 (a) and (b) show the initialisation and streaming costs for CPU and memory. These resources are considered together since their consumption has the same cause. During the initialisation, CPU and memory costs are due to initialisation of the no-replay sequence number and of the authentication and confidentiality algorithms. During streaming phase the sequence number increment and checking, the creation of a new (tunnel) header for each packet and the processes of authentication/integrity and confidentiality algorithms consume also these two resources. Figure 2 (c) presents bandwidth costs while streaming. Our classification depends on the amount of data transferred for each specific SoS parameter. For example, the sequence number exchanged to ensure no-replay is a 32 bits value, whereas

the size of the added header for tunnelling is at least 20 or 40 bytes for respectively IPv4 and IPv6, or more, the size of data when padding is added to enciphered data can reach 255 bytes. Initialisation bandwidth cost is not shown here. Only the protocol has an impact on it for its security context establishment (key generation, algorithms negotiation, etc.).

Figure 1. Classification of SoS parameters resource consumption

To determine the precise impact on bandwidth of the chosen protocol, we ran some tests applying the IPsec protocols for different security levels. We used the Ethereal tool [29], a network protocol analyser, to value bandwidth costs for a MPEG video and a DVD sequence. Multimedia sequences are read with VLC (Video LAN Client) on a laptop from a desktop running on Windows OS and are secured with the Windows OS IPsec Policy Tool. The data are exchanged over a LAN.

The Windows IPsec Policy Tool provides confidentiality using 3DES or DES algorithms. The SHA-1 and MD5 algorithms associated with HMAC are available for authentication and integrity services. To measure the bandwidth costs, we made tests twice for both multimedia sequences (MPEG and DVD) with all possible combinations of security protocols and algorithms (i.e. AH with SHA-1, AH with MD5, ESP with SHA-1, ESP with MD5, ESP with 3DES, ESP with DES, ESP with SHA-1 and 3DES, ESP with SHA-1 and DES, ESP with MD5 and 3DES and ESP with MD5 and DES). We can notice that the multimedia sequence quality, the confidentiality level and the authentication and integrity level do not impact bandwidth costs. Only the choices of the security services and of the protocol have an impact on it.

Table IV depicts the increase bandwidth costs before and after security inclusion. Bandwidth cost during the initialisation phase is expressed in bytes since it consists in the security context establishment (key generation, negotiation of used algorithms, etc.) and the number of exchanged packets is limited (10 for IPsec). While streaming, it is expressed in bytes per packets because it corresponds to the protocol processing, which depends on the multimedia file. Table IV shows that the bandwidth initialisation cost depends only on the protocol, ESP consuming more resources than AH. During the streaming phase the bandwidth consumption varies according to the chosen security services except for the protocol. Confidentiality consumes less bandwidth than authentication and integrity, which consume fewer resources than confidentiality, authentication and integrity. This confirms our classification in Figure 1 (c).

Bandwidth costs for UDP and IPsec protocols

Protocol	**Bandwidth cost during the initialisation (bytes)**	**Bandwidth cost while streaming (bytes/packet)**	
UDP	*not relevant*	1358	
AH	Authentication and integrity	1688	1382
ESP	Authentication and integrity	1712	1382
	1712	1378	
	Confidentiality, authentication and integrity	1712	1390

We are now extending our tests to the other resources (CPU and memory), and for each SoS parameter.

VI. Conclusion and future work

This paper presented our research work on including security parameters in Service Level Agreements specification to improve policy-based SoS and QoS management for multimedia services. We used the Tequila project SLS definition as a basis and extended it with SoS parameters.

We identified the essential SoS parameters to integrate in the QoS part of an SLS. They consist of a set of network specific parameters useful for network security protocols configuration and to evaluate the impact on resource consumption and consequently on QoS. We also highlighted the necessity for EUs to provide higher-level parameters to the SLS in order to express their SoS requirements in terms they understand. Finally we discussed the influence of security on the performance of services and networks. It is essential to consider it to improve the QoS management. An incorrect valuation of resource consumption reduces the negotiated service quality. Therefore it is essential for security to be considered while negotiating QoS. Also the choice of security protocol and of algorithms has to be in accordance with the SP supported performances.

Currently, we continue our tests on the other resources consumptions for each SoS parameter. The objective is to determine and add parameters representative of the resource consumption into mapping tables. It can be useful to

choose the most suitable algorithm and security protocol. It will improve QoS management by adapting and optimising the resource consumption for security.

References

Irvine, C., et al., "Security as a Dimension of Quality of Security Service", In Proc. of the Active Middleware Services Workshop, San Francisco, CA, Aug 2001, pp 87-93.

Spyropoulou, E., et al., "Managing Costs and Variability of Security Services", IEEE Symposium on Security and Privacy, Oakland, California, May 2001.

Verma, D., "Service Level Agreements on IP Networks", in Proc. of the IEEE, vol 92, no. 9, Sept. 2004.

Westerinen, A., et al., "Terminology for Policy-Based Management", RFC 3198, Nov. 2001.

Engel, T., et al., "AQUILA: adaptive resource control for QoS using an IP-based layered architecture", IEEE Communications Magazine, Jan. 2003, pp. 46-53.

IST Aquila Project: http://www-st.inf.tu-dresden.de/aquila/ [last access on 15th Jan. 07]

Cortese, G. et al, "Cadenus: creation and deployment of end-user services in premium IP networks", IEEE Communications Magazine, Jan. 2003, pp. 54-60.

IST Mescal Project: http://www.mescal.org [last access on 15th Jan. 07]

IST Sequin Project: http://archive.dante.net/sequin/ [last access on 15th Jan. 07]

Mykoniati, E., et al., "Admission control for providing QoS in DiffServ IP networks: the TEQUILA approach", IEEE Communications Magazine, Jan. 2003, pp 38-44.

IST Tequila Project: http://www.ist-tequila.org [last accessed on 15th Jan. 07]

ISO, "Information Processing System – Open System Interconnection – Basic Reference Model – Part 2: Security Architecture", International Standard ISO 7498-2, ISO, Feb. 1989.

Kent, S., and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, Nov. 1998.

Dierks, T. and E. Rescorla, "The TLS Protocol Version 1.1", IETF Internet Draft, May 2005. <draft-ietf-tls-rfc2246-bis-11.txt>

Koch, B., et al., "IST Premium IP Cluster", IST deliverable, Mar. 2003.

T'Joens, Y., et al., "Service Level Specification and Usage Framework", Internet Draft, Oct. 2000. <draft-manyfolks-sls-framework-00.txt>

Rajan, R., et al., "Service Level Specification for Inter-domain QoS Negotiation", Internet Draft, Nov. 2000. < draft-somefolks-sls-00.txt >

Salsano, S., et al., "Definition and usage of SLSs in the AQUILA consortium", Internet Draft, Nov. 2000. <draft-salsano-aquila-sls-00.txt>

Goderis D., et al., "Attributes of a Service Level Specification (SLS) Template", Internet Draft, Oct. 2003. <draft-tequila-sls-03.txt>

Sevasti, A., and M. Campanella, "Service Level Agreements specification for IP Premium Service", Deliverable D2.1 - Addendum 2, IST Sequin Project, Oct. 2001

Morand, P., et al., "Initial Specification of Protocols and Algorithms for Inter-domain SLS management and Traffic Engineering for QoS-based IP Service Delivery and their Test Requirements", Deliverable D1.2, IST Mescal Project, Nov. 2003.

Arcade Project: http://www-rp.lip6.fr/arcade/ [last access on 15th Jan. 07]

Yilmaz, V., et al., "Gestion et déploiement de services de sécurité dans un réseau basé sur des politiques" (Written in French, title in English: Management and Deployment of Security Services over a Policy-based Network), SAR 2003 Conference, Nancy, France, June 2003.

Jason, J., et al., "IPsec Configuration Policy Information Model", RFC 3585, Aug. 2003.

American National Standard for Information Systems – "Data Encryption Algorithm - Modes of Operation" (ANSI X3.106-1983), ANSI, Approved 16 May 1983.

Dworkin, M., "Recommendation for Block Cipher Modes of Operation: Methods and Techniques", NIST Special Publication 800-38A, Dec. 2001.

Kent S., "IP Authentication Header", IETF Internet Draft, Mar. 2005. <draft-ietf-ipsec-rfc2402bis-11.txt>

Kent S., "IP Encapsulating Security Payload (ESP)", IETF Internet Draft, Mar. 2005. <draft-ietf-ipsec-esp-v3-10.txt>

Ethereal home page: http://www.ethereal.com [last access on 15th Jan. 07].