11

Secure IN internetworking

Alexander Herrigel and Xuejia Lai
R³ Security Engineering AG[1]
Zurichstrasse 151, CH-8607 Aathal, Switzerland.
Fax: +41 1 932 66 60
Email: herrigel@r3.ch

Abstract

This paper presents a new approach for secure IN internetworking. Based on a threat analysis, an adequate cryptographic protocol is proposed to address the derived security concerns. The cryptographic protocol presented is based on the recently published standardization framework ISO/IEC CD11770-3.

1. INTRODUCTION

Confronted with the growing complexity of the public networks and the need for rapid development and deployment of enhanced services the telecommunications community developed a new concept, called the Intelligent Network (IN) architecture. This architecture has a modular structure, separates the call processing logic from the service intelligence, and is based on a centralized service control [ITU-T]. With respect to the network operators requirements for vendor and network independent solutions, the telecommunications community has developed for the IN architecture two standards, called the AIN and the ITU-T Q.1200 recommendations. The ITU-T standard is actually evolving to support service transparency in a multi-vendor environment (Figure 1).

[1]This work has been funded by the Swiss National Science Foundation under the SPP program.

J. Harju et al. (eds.), *Intelligent Networks*

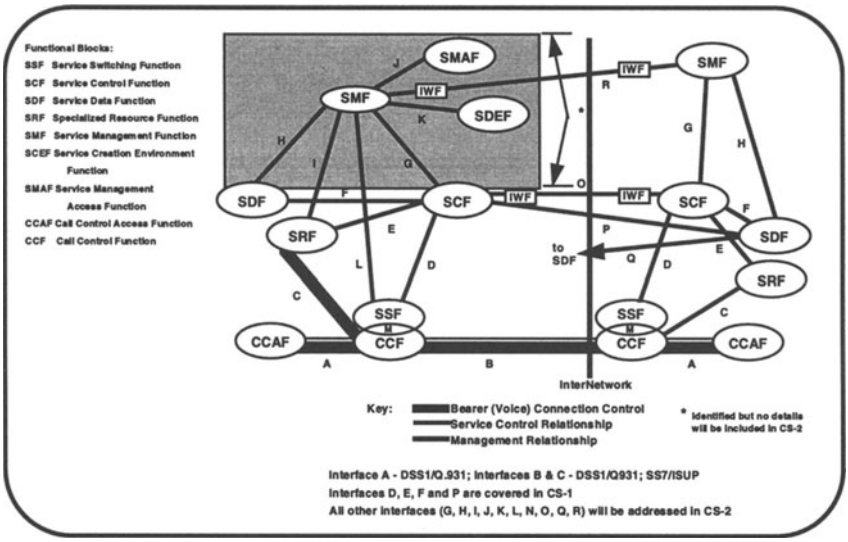

Figure 1: The modified distributed functional plane for CS 2 [Nagaraj].

The IEEE communications society and the ITU standardization body (SG11) are currently investigating an enhanced Capability Set, called CS 2, for the IN architecture. Due to the market demand and deregulation, IN Internetworking aspects between different Network Operators are addressed. Internetworking will provide third party access to an IN and enable the exchange of control and management information between different Network Operators running different Intelligent Networks. From a business perspective, IN internetworking is important in Europe for the Network Operators, since it provides the basis for offering European-wide IN services to the service users. Because the management and control data influence the service processing and billing, the messages transferred to exchange these data are very sensitive and security is a very critical aspect in this domain. In addition, some national regulatory bodies require to protect the privacy of service subscribers by additional means.

The following interfaces are actually investigated [Chatras, Nagaraj, ETSI]:

1. SCF - SDF,
2. SCF - SCF,
3. SDF - SDF and
4. SMF - SMF.

The importance of security issues for IN Internetworking has also been recently noticed by the EURESCOM project group P230 [Hulzebos]. Our paper addresses the Internetworking security aspects for the following interfaces:

1. SCF - SCF, and
2. SMF - SMF.

The paper is organized in 5 sections. The threat analysis for the Internetworking is described in section 2 to identify the security concerns. In section 3, notations and definitions are introduced to provide the adequate basis for a precise formal description of the cryptographic protocols. The derived protocols are presented in section 4. Finally, results and conclusions are summarized and discussed.

2. THREAT ANALYSIS

2.1. Involved parties

From our perspective, the following entities have to be considered for the IN scenario:

IN-Service: An IN-service is considered as a stand-alone commercial telecommunication service that is executed in an IN environment. This environment can be partitioned into several domains.

Network Operator (NO): A network operator is a commercial organization which runs and maintains an IN infrastructure. During the execution of an IN service, such an infrastructure can be involved for the origination, transfer or terminating call processing.

Service Provider (SP): A service provider is a organization that offers an IN service for subscription. The SP has the legal responsibility for the correct operation of the offered services. The NO itself may also act as a service provider.

Service Subscriber (SS): A service subscriber is a person or an organization that has a contract with a service provider for the supply of a service.

Line Subscriber (LS): A line subscriber is a person or an organization that has rented or bought an IN access device.

Service User (SU): A service user is the end user of a typically IN-service. In a business environment, it can be an employee of the service subscriber.

Regulatory Body (RB): The regulatory body imposes the legal rules for the different parties in a commercial IN environment to ensure fair competition.

2.2. Characterization of the IN architecture

The IN architecture is based on three different layers, namely the service creation and management layer, the service control layer, and the switching and network resource layer. These layers have different resource entities which are shortly described for an adequate threat analysis.

The Service Management System (SMS) represents the service creation and management layer. It supports the specification, the introduction and maintenance of all IN services. It holds the master copy of all network databases and maintains the SCP's tables and service logic. In addition, it maintains the service and control information in the IPs. A Service Creation Environment (SCE) is a an integral part of the SMS. Typically, new IN services are introduced via a file transfer to the SCPs. The SMS updates and retrieves management data from the Service Control Point (SCP), the Adjuncts, and the Intelligent Peripheral (IP).

The network service control intelligence is positioned at centralized nodes (SCP). The SCPs execute the service logic for call control. They support on-line transaction processing with high

transaction rates and volumes. In addition, these nodes hold on-line local databases with respect to the IN services they support.

The SSPs provide the access to the service users. They trigger the calls which have been generated by an IN access device and send the imposed service requests to the SCP. The IN service is executed by the SSP on the basis of instruction guidance from the associated SCP within the framework of a Basic Call State Model (BCSM). The SSP is connected to other local or trunk exchanges on the basis of the bearer connection control and a non-IN call control protocol. In addition, the SSP is connected to the associated SCP on the basis of the SS7 protocol.

The Adjunct offers the same functionality as an SCP. It is, however, connected to a local exchange with high speed connections and supports specific logic programs of a service provider.

The IPs supports enhanced voice/data services like announcements, digit collection, voice messaging, speech response. An IP is shared between different SSPs. It is typically accessed by an ISDN interface.

There are two modes of interactions in this environment:

- Transaction messages for providing updates of management, control, and service information.
- File transfers, typically via the X.25 network, for the installation and modification of the service logic and management data.

IN Internetworking is defined as the process of executing a requested IN service across at least two different autonomous domains. Every domain represents a separate IN network with the above described involved parties and IN resource entities. The IN service can only be executed successfully, if the two different domains cooperate by exchanging management, control, and service data. Since the different domains are typically run by different Network Operators in competition, there is only a very limited trust relationship between these domains. Each domain enforces its own mechanisms to provide integrity , availability, and service user privacy. Depending on the number of different domains involved in a IN service processing, the following different types of domains are identified: 1. The origination domain, 2. the interconnection domain, and 3. the destination domain. From our perspective, the Interworking Function (IWF) of the enhanced distributed functional plane provides the necessary addition means to support secure IN Internetworking during the call processing or exchange of management information.

2.3. Derived threats and security services

The following threats have been identified:

Threat 1: Modification (replication, insertion, deletion) of service, control, and management data during the transmission.

Threat 2: Replay of old messages by unauthorized parties.

Threat 3: Impersonation of network entities (SMS, SCP) to penetrate the NOs databases and service (denial of service).

Threat 4: Illegitimate use of system resources (SMS, SCP) from authorized entities.

Threat 5: Repudiation of management information flow. If management data modification is requested, it should not be possible to deny later having requested the modification.

Threat 6: Disclosure of information from the transmitted data to gain competitive advantages.

Only threat 4 is covered by the access control policy of the operating systems (Unix, etc.) which are typically applied in a specific domain.

The following security services must be supported for a secure IN Internetworking. The reader is referred to [ISO] for a detailed definition of these security services.

1. Message Content Authentication
2. Message Origin Authentication
3. Non-Repudiation of Origin
4. Confidentiality

The co-operation of the different domains is only possible if specific legal contracts have been negotiated and agreed between the Network Operators. These contracts define a low level trust relationship with respect to the messages, that have to be exchanged.

3. NOTATIONS AND DEFINITIONS

The following definitions are introduced to distinguish the different cryptographic approaches which are applied in the protocols.

Cryptographic algorithm: A function f: (A x B) -> C from the Cartesian product set (A x B) to a set C is called a cryptographic algorithm if

1. the computational complexity CC to find for a given c = f(a, b) in C and a in A without the knowledge of b in B such that f(a, b) = c is infeasible,
2. it is easy to compute a for a given c from C and b from B.

The set A is identified as the plain text space and the set B is identified as the key space. The set C is called the ciphered message space.

Asymmetric cryptosystem: An asymmetric cryptoystem is based on a cryptographic algorithm that uses two related transformations, a public transformation (public key) and its inverse, a private transformation (private key). The two transformations have the property that, given the public transformations, it is computationally infeasable to derive the private transformation.

Symmetric cryptosystem: An symmetric cryptosystem is based on a cryptographic algorithm that uses two transformations t and l. They have the following properties:

1. Given one transformation it is possible to derive the other.
2. Given an a in A, a b in B, and a c in C the following relation holds: l(t(a, b), b) = l(c, b) = a.
3. It is computationally infeasable to find for a given c in C an a in A without the knowledge of the b in B such that t(a, b) = c.
4. It is computationally infeasable to find for a given a in A a c in C without the knowledge of the b in B such that l(c, b) = a.

Collision resistant hash function: A function f: A -> B from a set A to a set B is called a collision resistant hash function, if it fulfills the following conditions:

1. The argument of f can be of arbitrary length and the image of f has a fixed length of n bits.

2. It is computationally infeasable to find for a given b in B and a in A such that t(a) = b.
3. It is computationally infeasable to find a and b in A, $a \neq b$, such that f(a) = f(b).

Message authentication code (MAC): A function f: (A x B) -> C from the Cartesian product set (A x B) to a set C is called a message authentication code, if the following conditions are fulfilled:

1. The argument of f can be of arbitrary length and the image of f has a fixed length of n bits.
2. For a given c = f(a, b) in C, it is computationally infeasable to find an a in A without the knowledge of the given b in B such that f(a, b) = c.

Token: A token is a message which is sent from one entity to another entity during the execution of a cryptographic protocol.

Ticket: A ticket is a message that is partitioned by a token and the associated certificate.

Certificate: A certificate C is a specific cryptographic transformation M of a token T. M is defined as the composition of a collision resistant hash function f_{crh} and a private transformation t_{pr} of an asymmetric cryptosystem ($C = M(T) = t_{pr}(f_{crh}(T))$).

The following notation is introduced to specify precisely the proposed cryptographic protocols. The protocols are described by relations which identify the objectives (goals), the different starting assumptions, and the communication or computation phases.

$R_{operation}$:= { < x, y > | The process x executes the operation y.}

$R_{condoperation}$:= { < x, y, (z) > | The process x executes operation y, if the condition z is fulfilled.}

$R_{condmessage}$:= { < x, y, (m_1, m_2, ..m_n), (z) > | The process x sends the message m = (m_1, m_2, m_3, ..., m_n) to process y, if the condition z is fulfilled. The considered communication channel is untrusted.}

$R_{assumption}$:= { < x, y > | The process x executes the operation y for initialization purposes.}

R_{goal} := { < x , y> | x is the name of the cryptographic protocol, y is the goal of the cryptographic protocol}

$R_{constraint}$:= { < x , z> | Imposed constraint z for the cryptographic protocol x.}

4. THE APPROACH

4.1. Motivation

The proposed cryptographic protocol includes an asymmetric one-pass key transport scheme which supports a mutual authentication with key confirmation (ISO/IEC CD 11770-3). The scheme establishes in first phase a shared secret key between two entities. A ticket of the management data is generated in the second phase to support message content authentication, message origin authentication, and non-repudiation of origin. The ticket is then ciphered to support confidentiality.

Suppose the SMS A wants to send some management data MD to the SMS B over an interconnection domain. Then the protocol is described by the following scheme:

Phase 1:

1. A generates the key agreement token $KATo_A := < SN_t, ID_A, SK_{CIP}, SK_{CIP}[CD_C] >$, with SN_t as the sequence number[2] at time t, ID_A as the ID of the SMS A, SK_{CIP} as the generated secret key for the ciphering, and CD_C as the control data for the key confirmation. Then SN_t is increased by one and stored. If a specific threshold value has been reached, the sequence number is initialized.
2. A retrieves the public key V_B from the SMS B and ciphers the key agreement token with this key ($V_B[KATo_A]$).
3. A generates the token $ATo_A := < V_B[KATo_A], ID_B >$.
4. A hashes the token ATo_A with the collision resistant hash function f_{crh} and signs the resulting output with the private key P_A from the RSA key pair (P_A, V_A). The ticket $Ti_A := < P_A[f_{crh}(ATo_A)], ATo_A >$ is then generated and sent to B.
5. B receives the ticket Ti_A, retrieves the public key V_A from A and verifies the ticket (valid signature). If the signature is valid, B retrieves the key agreement token $KATo_A$ by deciphering the token $V_B[KATo_A]$ with its private key P_B from the RSA key pair (P_B, V_B).
6. B sends the key confirmation token $< B, CD_C >$ to A, if the sequence number SN_t is not minor as the last stored one. SN_t is then increased by one and stored.

Phase 2:

7. A receives the key confirmation token and verifies the data. If the confirmation is valid, A sends the following token to B: $< A, SK_{CIP}[< < MD, SN_t >, P_A[f_{crh}(< MD, SN_t >) >] >$. The sequence number is then increased by 1 and stored. If a specific threshold value has been reached, the sequence number is initialized. B receives the token $< A, SK_{CIP}[< < MD, SN_t >, P_A[f_{crh}(< MD, SN_t >) >] >$. B retrieves the symmetric secret key SK_{CIP} and the public key V_A from A. The management data MD is generated by deciphering the token, i.e. $SK_{CIP}[SK_{CIP}[< < MD, SN_t >, P_A[f_{crh}(< MD, SN_t >) >] >.]$.
8. The management data is accepted, if the sequence number SN_t is not minor as the last stored one and the signature $P_A[f_{crh}(<MD, SN_t >)]$ has been verified. SN_t is then increased by one and stored. If a specific threshold value has been reached, the sequence number is initialized.

4.2. The protocol

4.2.1. Imposed constraints

goal(IN Internetworking, The SMS A wants to send securely some management data MD to the SMS B over an interconnection domain).

constraint(IN Internetworking, the secret key SK_{CIP} has at least a length of 64 bits);

constraint(IN Internetworking, HASH(M) is a collision resistant hash function computation with the message M as the input);

---

[2]A random number based challenge can also be used in this phase.

4.2.2. Initialization

assumption(A, generate asymmetric RSA key pair (P_A, V_A));
assumption(A, distribute authentically V_A to B);
assumption(B, generate asymmetric RSA key pair (P_B, V_B));
assumption(B, distribute authentically V_B to A);

4.2.3. Protocol execution

4.2.3.1. Key transport with implicit authentication

operation(A, generate SK_{CIP});
operation(A, retrieve SN_t);
operation(A, compute $SN_t := SN_t + 1$ and store the result);
operation(A, retrieve ID_A);
operation(A, generate control data CD);
operation(A, compute SK_{CIP} [CD]);
operation(A, generate key agreement token KATo := < SN_t, ID_A, SK_{CIP}, SK_{CIP}[CD_C] >);
operation(A, retrieve public key V_B from B);
operation(A, compute V_B[KATo]);
operation(A, generate token ATo := (V_B[KATo]), ID_B);
operation(A, generate certificate Sig := P_A[HASH(ATo)]);
operation(A, generate ticket Ti_A :=(Sig, Ato));
operation(A, compute $SN_t := SN_t$+1 and store the result);
condoperation(A, initialize SN_t , (Sn_t > range));
message(A, B, Ti_A);

4.2.3.2. Ticket verification

operation(B, retrieve public key V_A from A);
condoperation(B, stop execution, (V_A[Sig] < > HASH(ATo));

4.2.3.3. Key confirmation

operation(B, retrieve private key P_B);
operation(B, generate KATo by computing P_B[V_B[KATo]]);
operation(B, extract SN_t);
operation(B, extract SK_{CIP});
operation(B, extract ID_A);
condoperation(B, stop execution, (SN_t is minor as the last stored one);
operation(B, compute $SN_t := SN_t$ + 1 and store the result);
condoperation(B, stop execution, (identified party does not match with ID_A));
condoperation(B, stop execution, (SK_{CIP}[SK_{CIP}[CD]] <> CD);
operation(B, store SK_{CIP});
message(B, A, (B,CD));

4.2.3.4. Sending management data

condoperation(A, stop execution, (CD is not valid));
operation(A, retrieve or generate management data MD);
operation(A, retrieve SN_t);
operation(A, compute $SN_t := SN_t$ + 1 and store the result);
operation(A, generate T:= < MD, SN_t >);
operation(A, generate HT := HASH(T));

operation(A, retrieve private key P_A),
operation(A, generate Sig := P_A[HT]);
operation(A, generate MT := < T, Sig >);
operation(A, retrieve SK_{CIP});
operation(A, generate CMT := SK_{CIP}[MT]);
message(A, B, CMT);

4.2.3.4. Receiving management data

operation(B, retrieve SK_{CIP});
operation(B, retrieve public key V_A);
operation(B, generate MT by SK_{CIP}[CMT]);
operation(B, extract SN_t from T);
condoperation(B, stop execution, (SN_t is minor as the last stored one);
operation(B, compute SN_t := SN_t + 1 and store the result);
condoperation(B, initialize SN_t , (Sn_t > range));
condoperation(B, stop execution, (V_A[Sig] < > HASH(HT));
operation(B, extract MD from T);

6. CONCLUSIONS AND FUTURE WORK

We have presented in this paper a new approach for secure IN Internetworking. The approach has the following advantages:

1. The key agreement is based on an asymmetric technique. The imposed trust relationship between the involved parties for this scheme is adequate for the scenario. In addition, such a scheme provides a low key management complexity.
2. The scheme is efficient, since it needs only one pass for the key agreement.
3. The scheme supports key confirmation, i.e. the assurance that the SMS B is in the possession of the correct key.
4. The scheme provides explicit authentication from SMS A to SMS B through the signature and implicit authentication of B to A, since SMS B is the only entity that can compute the session key.
5. Since the session key is generated on a demand basis, there is no need to store any additional keys for the security services.
6. The described protocol uses a sequence number to prevent replay attacks. This sequence number can be generated by a counter. There is no need that both entities have to share a synchronized clock.
7. The same scheme can be applied for the following additional interfaces:
8. SMAF - SMF, and 2. SMF - SCF.

We are actually investigating the Nyberg-Rueppel key agreement protocol [Nyberg]. In contrast to other asymmetric key agreement schemes, this scheme supports also message recovery.

REFERENCES

[ITU-T] "General Recommendations On Telephone Switching And Signalling Intelligent Network, Introduction To Intelligent Network Capability Set 1", ITU-T Recommendation Q.1211, March 1993.

[Chatras] Bruno Chartras and Francois Gallant, "Protocols For Remote Data Management In Intelligent Networks CS1", IEEE Intelligent Network '94 Workshop, Heidelberg, Germany, May 24-26, 1994.

[Hulzebos] Raymond Hulzebos and Steve Reeder, "Pan European IN Reference Architecture ", IEEE Intelligent Network '94 Workshop, Heidelberg, Germany, May 24-26, 1994.

[Nagaraj] Kesavamurthy Nagaraj, "CS-1 Refinements and CS-2 Scope", IEEE Intelligent Network '94 Workshop, Heidelberg, Germany, May 24-26, 1994.

[Nyberg] K. Nyberg and R. A. Rueppel, "Message Recovery for Signature Schemes based on the Discrete Logarithm Problem", Proceedings of Eurocrypt'94, Springer-Verlag, 1994.

[ETSI] Network Aspects (NA), Security Requirements for Global IN Systems, DTR/NA-61201, Version: 1.0.5, date: 03.05.94, ETSI.

[ISO] ISO International Standard 7498-2: Open Systems Interconnection Reference Model - Part 2: Security Architecture, 1988.