Secure Error Signalling for Packet-Switched Networks - The Future Core Networks System Error Protocol

Theodore Stergiou[1] and Dimitrios L. Delivasilis[2]
(Corresponding author: Theodore Stergiou)
Isecure-e Ltd. Arahovis 44, 12136 Peristeri, Athens, Greece[1]
(Email: t.stergiou@isecure-e.com)
Department of Information and Communication Systems Engineering, University of the Aegean[2]
Karlovasi, Samos, 83200, Greece

(Received Dec. 20, 2005; revised and accepted June 20 & July 21, 2006)

Abstract

In this paper a secure error-signalling scheme for packet-switched network architectures is presented. Current solutions are based on the Internet Control Message Protocol to deliver information regarding congestion control. The disadvantages of ICMP are given, regarding its limitations and dependence on network protocol structures. We then move into presenting the Future Core Network System, followed by an analysis of the FCNS Error Protocol and its comparison against ICMP. We also present measurements taken to observe the performance of the FCNS in cases where the FCNSEP implementation has been imperative and reveal applicability issues for the FCNSEP in network protocol systems.

Keywords: Network security, secure error-signalling, secure communication protocols

1 Introduction

Error and control protocol architectures are essential to the operation of a set of communication rules, whereby conditions that could prevent the normal communication flow are identified and signalled for correction. Their implementation usually follows the unsuccessful attempts of the system entity and/or process to recover from such a situation, resulting in the necessity of external mechanisms to support and provide the required functions. Cases include congestion build up and notification of the reachability of a particular network host. Error signalling protocols are widely used in computer and telecommunication systems as a means of maintaining the required Quality of Service (QoS) levels for a connection. They provide peers with information regarding the communication well being and recovery from situations that could affect its lifespan.

The deployment of such mechanisms can be further supported by the implementation of error correction techniques. Methods such as Forward Error Control (FEC) [16] enhance the capabilities of the system in detecting and recovering from erroneous situations. Redundant information is included in the user data messages, providing the receiver with the means of detecting and correcting bit errors in the message pattern. These techniques form the final step of the error control system signalling procedures and their use falls outside the scope of the paper, so readers interested in this area can reference the respective bibliography.

Error signalling protocols are dependent on the set of communication rules supporting the connection in a given topology. Of most importance is the Internet Protocol (IP) [8] of the TCP/IP suite [20], supporting the Internet Control Message Protocol (ICMP) [7]. The following section presents the architectural view of ICMP providing an analysis of the protocol functionality and applicability.

The paper is consequently organized in the following sections. Section 3 presents the motivation for this work, based on the disadvantages and implementation pitfalls of the already proposed and under use solutions. Section 4 depicts the FCNS Error Protocol where a thorough analysis is given regarding its architecture and the interlayer and peer error-signalling procedures. We then move into identifying the security considerations of our proposal (Section 5) followed by the evaluation environments and measurements obtained to test its performance (Section 6). Finally, Section 7 consists of conclusions of the research work presented, together with directions for future work on the specific research area.

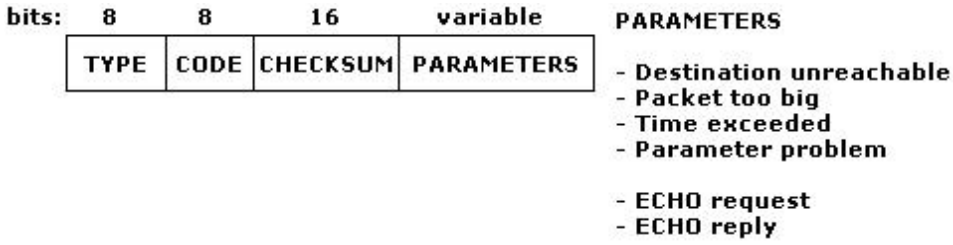

Figure 1: ICMP message header structure

2 IP Error Signalling Protocols (ICMP) [7]

ICMP is one of the most commonly used error signalling mechanisms, designed to support the notification of peer entities of conditions that might affect their communication. Its header message structure is depicted in Figure 1, together with the notification options supported. Depending on the fault situation, the *PARAMETERS* field is updated accordingly, including an amount of the data sent when the problem has been detected, to enable the message recipient to identify the location of the fault condition in the data stream and the upper layer(s) involved in the situation.

ICMP can be used either as an error signalling or informational protocol. In the former case, the nodes communicate information regarding situations that may interrupt or disrupt the communication process. In the latter case, the *echo* messages are used to obtain information regarding the status of the system. Their difference is identifiable by the *type* field of the ICMP header, which will have the high order bit set to '0' for the error communication procedures and to '1' when the message acts as an informational element.

ICMP is used in the TCP/IP protocol stack as the architecture responsible for ensuring that the data messages conform to the parameters set during the connection establishment phase, and also that their routing towards the intended destination is successful. The protocol is not responsible for end-to-end connectivity issues, usually handled by the TCP flow control mechanisms, and hence can provide notification for only a limited number of error conditions.

Consequently, the responsibilities of ICMP are somewhat limited with respect to congestion handling, either as a congestion recovery or congestion avoidance mechanism. In the first case the purpose of its initiation is the prevention of a zero throughput network appearance. The second case involves issues of congestion recovery even when such measures have already been enforced on the particular connection. A typical method realising the congestion monitoring and maintenance services is the *source quench* process, whereby the source is instructed to reduce the rate at which information is forwarded onto the network topology. Unfortunately, ICMP does not inform the end system of the actual reason for which the request has been launched, and hence the indication may not always be specific to the particular system, in the sense that it may not imply that the signalled system is the cause of congestion.

ICMP also provides a measure for informing the data source of any inconsistencies in the network topology that may deny the transmission of its messages towards the receiver. The *destination unreachable* message includes information as to whether a route to the peer actually exists and/or details about the legitimacy and capabilities of the node directly connected to the sending instance. Further notification procedures include the signalling of conditions related to excessive packet sizes, timer expirations and parameter problems of the IP implementation. These techniques attempt to provide the means of regulating the data flow in relation to the QoS levels negotiated for the connection on a link basis.

Since most of the network architectures are based on the TCP/IP stack, ICMP currently forms the only error signalling procedure available. Despite this fact, we have identified several disadvantages of the ICMP family, forming the motivation behind our work.

3 Motivation for the Work

The drawbacks recognized for the ICMP architecture fall into several categories, entailing performance and security issues.

The ICMP protocol forms an integral part of IP, resulting in its use being prohibited in architectures based on another protocol stack. ICMP is used as a peer error signalling technique, meaning that notification is only available for the communicating parties. Signalling of fault conditions to the protocol stack layers is left to mechanisms implicit in the network architecture supporting the connection. However, the dependence of these measures on the implementation of the TCP/IP stack means that different networks may exhibit independent measures. Vendor-specific hardware possesses distinct features reducing compatibility with other topologies, implying the modification of the ICMP implementation depending on the network operator and system running IP.

Until the design of the IP security architecture (IPsec) [14], all ICMP messages were sent in cleartext format,

leaving the network susceptible to various attacks by unauthorised parties [3]. Although IPsec provides for the protection of the protocol's messages against modification and Denial of Service (DoS) attacks, the functions of the architecture closely bind any security considerations to the IPsec structure for which security is not always imperative. This implies that ICMP messages could still be sent unencrypted, if an association was to be initiated without any protection mechanisms enabled. The provision of the IPsec services for the ICMP messages should account for the adequate protection of the protocol data against unauthorised modification and information disclosure attacks. This notion does not imply any protection of the system against attacks aiming at the IPsec architecture itself [4, 5, 9, 10, 11, 15, 18, 19, 25], or by exploiting vulnerabilities of the system running the IPv6 protocol [6, 17, 21].

For the ICMPv6, a number of additional vulnerabilities have been identified, with respect to attacks launched by manipulating the protocol error messages [13]. Of importance are the forcing of the communication to a less secure mode and the routing of data through illicit networks. In the first case, the attacker manipulates a *destination unreachable* message received in response to a key management protocol request, forcing the source to fall back to a scheme or operational mode that is less secure than that requested. In the second attack category, the ICMP *redirect* message is used to force the routing of the user data via a network where the adversary possesses direct access privileges. Finally, implementations of the Microsoft Windows operating systems family are susceptible to ICMPv6 flooding attacks, due to the inability of the connection firewalls to block IPv6 traffic [12].

The most significant disadvantage of the ICMPv6 implementation is however that authentication and encryption are recommended actions. Message integrity and confidentiality should form an implementation prerequisite to ensure the safe passage of the error signalling messages especially for unknown or suspicious networks. Yet, the scheme is left at the discretion of network operators, who may choose to send the messages unprotected to minimize the amount of network resources and processing it would take to authenticate and verify their validity.

The vulnerabilities of the IP error control signalling protocol and its dependence to specific network architectures have led to the development of the FCNS Error Protocol (FCNSEP), implemented for the FCNS reference architecture [22, 23, 24]. To our knowledge, the provision of error signalling structures for secure reference architectures has not been accounted for standardised models, such as the OSI security architecture [1] or the TCP/IP [20]. In particular, the support of an interlayer-signalling scheme independent of the individual layered protocol mechanisms is a notion previous research has not dealt with.

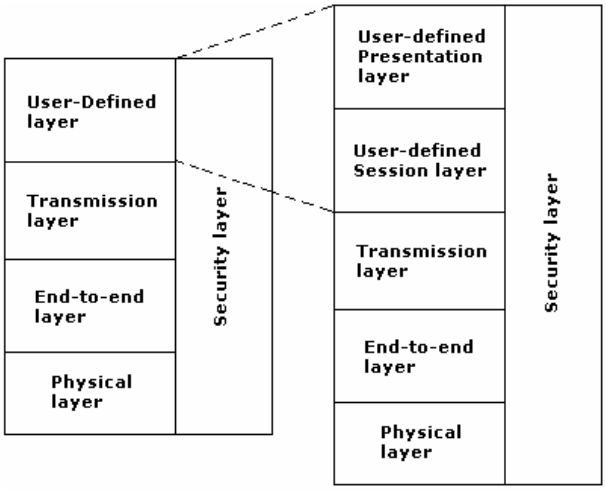

Figure 2: FCNS architecture in relation to the OSI 7-layer model

4 FCNS Error Protocol (FCNSEP)

The FCNSEP has been designed as the means of providing a secure error signalling solution for architectures based on the FCNS protocol stack model. In the following sections a brief overview of the FCNS is given, followed by an analysis of the FCNSEP operation.

4.1 FCNS Architecture

FCNS is a secure reference architecture for packet-switched environments, where emphasis has been given to the protection of both the internal and external messages of the stack. Its conceptual view is given in Figure 2 with respect to the OSI 7-layer model, whereas Figure 3 provides information as to the communication between two network peers. Specific details on the FCNS security mechanisms and functions can be found in [22, 23, 24] and will consequently not be analysed in this paper.

The Security Layer (SL) initiates the functions that encrypt/decrypt a message, verify its validity, secure a communication channel both on an end-to-end and link basis and protect the FCNS error protocol. Furthermore, it governs the functions that setup and maintain the FCNS keystream generator, which provides the secret keys used for the inter-layer messages [22, 23]. Security mechanisms, such as authentication, integrity, confidentiality and non-repudiation are applied to each one of the layers independently, whenever those are requested by the respective protocol.

Management and monitoring of the protection mechanisms on a single layer such as the SL, enhances flexibility and portability issues, whereby possible updates of the security mechanisms would not affect the operation of the FCNS layers.

The secret keys and algorithms used for the protection

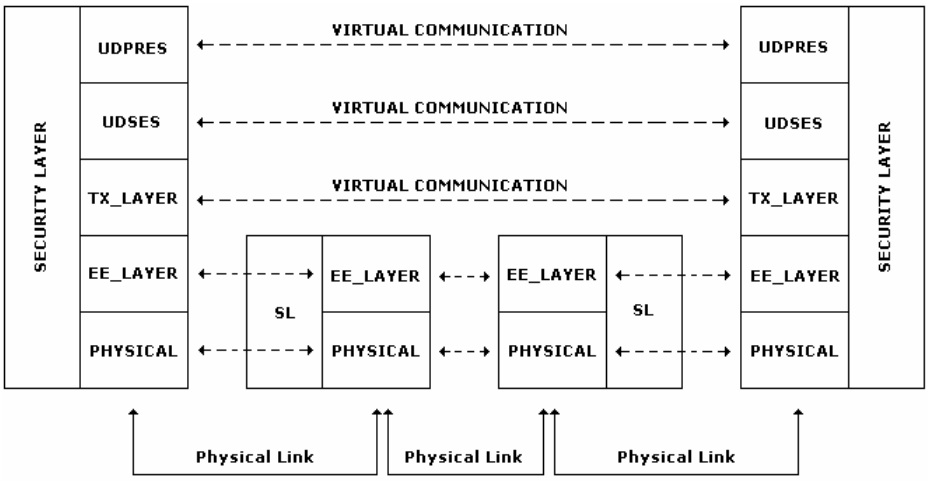

Figure 3: FCNS peer-entity communication

of the FCNS inter-layer messages and those intended for the user data are exchanged prior to the connection establishment phase, in the form of security contexts independent for each layer of the architecture. By this method it is ensured that a compromise at a given layer of the communication would not jeopardize the operability of rest of the architecture, providing at the same time a degree of measure in identifying and explicitly locate implementation pitfalls for recovery purposes.

The User-Defined layer is responsible for the semantics and the session establishment of the connection. In particular, the User-Defined Presentation layer (UDPRES) is involved with the message encoding procedures ensuring the secure negotiation of the transfer syntax for a particular connection. Additionally, the User-Defined Session layer (UDSES) has the task of synchronizing a session and enforce the address verification procedures, which can support the FCNS authentication procedures.

The Transmission layer (TX_LAYER) enforces the necessary handshake mechanisms and reliably transfers the data between the ultimate end-nodes. The End-to-end layer (EE_LAYER) is responsible for the routing of the FCNS packets between various subnetworks, error detection and correction techniques, as well as for enforcing the link-based security features of the FCNS. Finally, the Physical layer (PHYS) resembles the interface between the FCNS and the physical medium protocols, supporting the operation of traffic padding mechanisms countering traffic analysis attacks.

FCNS is designed to account for vulnerabilities and implementation pitfalls of the OSI security model and the TCP/IP protocol suite. The major points behind its development are the protection of the messages that are internal to the stack and the placement of the respective functionality into a single layer. The specification of the SL ensures the simplicity of the architecture and the provision of a simple managed solution for network protocol topologies. The removal of the redundant functions from the communication layers enables the implicit error allocation, and the increase of the FCNS flexibility in updates reflecting cryptographic advances.

The FCNS communication layers account for the communication establishment, maintenance and release, in a fashion similar to the OSI 7-layer model. The lack of distinctive Application and Physical layer OSI-type protocols in our work follows from the fact that their simulation would not have provided any valuable information as to the security of the service primitives exchanged. Consequently, details specific to the conversion of the bit stream into electrical and/or digital signals, as well as the specification of implicit applications were outside the scope of the work.

4.2 FCNSEP Operation

The FCNSEP has therefore been developed to support the signalling of a wide range of erroneous conditions, in relation to the stack operation and the communication between peer nodes. It forms part of the overall FCNS stack architecture and is initiated by the communication instances whenever a fault condition arises, to enable its correction and the continuation of the connection. It can also be used as a stand-alone error signalling and control system in packet-switched network architectures, provided that its messages are secured prior to their transmission via the communications channel.

Its operation is based on three main modes, defined as *Interlayer error signalling, Peer error signalling and System signalling for unrecoverable and unidentified errors.* These functions form the very essence of the FCNSEP and are used throughout the various phases of the communication of the FCNS. No matter the approach, FCNSEP is always initiated via the SL upon reception of the error notification by either an FCNS layered protocol or the communicating peer, as shown in Figure 4.

The *NODE_ALERT* message signals the error condi-

tion to the SL, which should reply by providing the appropriate action required for the recovery and correction of the fault situation. If such an action cannot be offered, then the FCNSEP is used to advise the system of the circumstance and inform the protocol instances of the steps to be taken. On the other hand, if the necessary mechanisms can be made available, then FCNSEP is realised to provide the FCNS communication layers and/or the peer nodes with the respective indication.

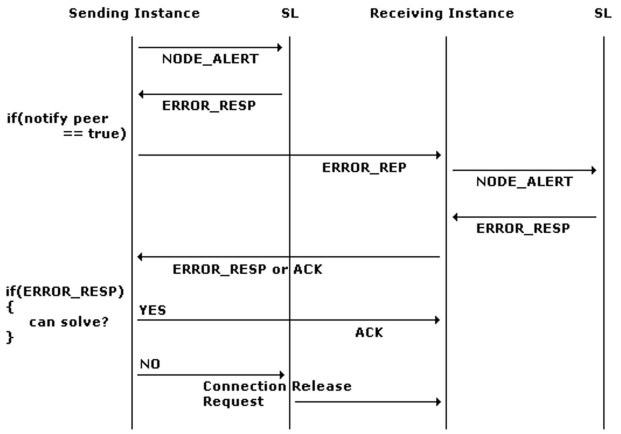

Figure 4: FCNSEP realisation in the FCNS architecture

The *ERROR_REP* and *ERROR_RESP* messages depicted in Figures 5 and 6 respectively, form part of the exchange procedure of Figure 4 and are responsible for the transmission of the information representing the error condition and the action suggested for its recovery.

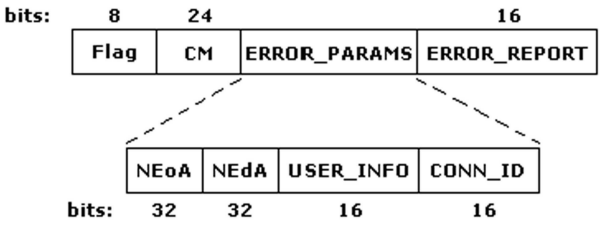

Figure 5: *FCNSEP ERROR_REP* message

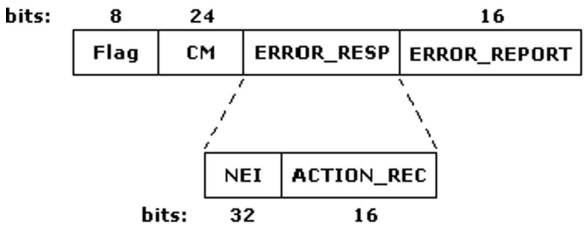

Figure 6: FCNSEP ERROR_RESP message

For the *ERROR_REP* message, the following fields contain the necessary signalling information for the fault condition:

- *ERROR_PARAMS* field, which includes parameters such as the sending node and intended destination of the FCNSEP message (*NEoA* and *NEdA* fields), as well as information about the user and the connection on which the error has been identified. If the destination is not directly adjacent to the transmitted instance, then the message should be mapped onto an FCNS packet and be encrypted with link-based mechanisms, to ensure its secured traversing via the intermediate nodes. The *USER_INFO* and *CONN_ID* fields include details on the QoS parameters that should be used to identify the priority the SL should give the message upon its reception, as well as on the network connection upon which communication is based. The latter field is essential in packet-switched topologies where various messages flow due to numerous simultaneous connections. By this method, the end-systems will identify the appropriate ERROR_REP message forwarding or discarding those intended for other elements.

- *ERROR_REPORT* field, which is used to indicate the condition that triggered the FCNSEP message.

Additionally, for the FCNSEP ERROR_RESP message, the fields containing parameters as to the action that the peer or layer should follow for the error correction are as follows:

- *ERROR_RESP* or *ERROR_RESPONSE* field that includes information about the network element or peer for which the action has been suggested (the *NEI* field), as well as the *ACTION_REC* field containing that particular action.

- *ERROR_REPORT* field, which is similar to the respective one of the ERROR_REP message and whose inclusion denotes the error for which the FCNSEP messages have been sent. Failure to include such data in the message results in the discarding of the respective response and the indication of the error to the peer entity or the SL protocol.

The following sections provide information regarding the three FCNSEP operational modes, in relation to the procedure depicted in Figure 4. Prerequisites and requirements for ensuring the protocol functionality are also given, pending the security of the FCNSEP messages.

4.2.1 FCNSEP Interlayer Error Signalling

The FCNSEP implementation is subject to the reception of a message in error more than three consecutive times, or the failure to establish certain connection parameters after an equal amount of attempts. This feature is used to minimise the possibility that FCNSEP functions come into effect for situations where error recovery may be achieved at no extra cost by other FCNS mechanisms.

The FCNSEP is implemented only if this is regarded as an absolute necessity by the system or the node process, to enable the suitable use of the available link resources for the data transfer phase only.

When error signalling takes place, the protocol detecting the error informs the SL of the fault condition, to provide the suggested for recovery action. If the SL protocol can offer the required action supporting the error correction and recovery procedure, then this is indicated to the layer via the *ERROR_RESP* message, including the reason for which the message has been sent to distinguish between any other *NODE_ALERT* requests made. Depending on the severity of the error, the SL protocol decides upon the action to be taken, including the release of the connection in cases the layer cannot recover from the error condition (FATAL situation). At the same time, it also decides whether the user of the layer protocol should be notified of the condition or even the signalling of the situation to the peer.

If the suggested action can be supported then the necessary functions are enforced to correct the error that has occurred and to proceed in supporting the particular connection. If this is not possible, then the *NODE_ALERT* message is sent again, indicating the protocol's inability to conform to the SL dictations. The node should keep a record of the number of error indications sent to the SL and if that number exceeds the predetermined threshold for the particular protocol (usually three tries), then the connection is released and the system is notified of the situation following the signalling of the condition to the peer.

Finally, it may be the case that the SL is unable to provide details about the correction and recovery from the fault condition. In such situations, the NODE_ALERT message is replied with a *NAK* message, signalling the notification of the system of the condition that has arisen. Care is taken in dismissing any error indication by the layers for fear of an active attack by an adversary, whereby the attacker is inserting illicit error requests messages to disrupt the connection service or obtain information about the operation of the SL protocol. It is therefore imperative that all FCNSEP related messages are secured using the parameters exchanged via the Security Contexts (SC) during the connection establishment phase.

4.2.2 FCNSEP Peer Error Signalling

FCNSEP peer error signalling is a process initiated upon indication from the SL protocol that the peer should be notified of the error condition. The indication is included in the ACTION_REC field of the ERROR_RESP message, as well as providing the appropriate NEI identifier in the respective field. The process involves the transmission of the ERROR_REP message towards the peer entity involved in the error condition. The message can be initiated by either the data messages intended destination or an intermediate router responsible for the forwarding of the messages to their recipient. Typical examples include the indication of an ARQ or Transmission layer flow control failure, the notification of an unreachable host, breach of the QoS connection levels or the detection of a congested and/or faulty transmission link.

If the layer entity can, in conjunction with the SL protocol, identify and correct the error then an acknowledgment is sent back to the peer to indicate that communication can proceed. Any suggested actions related to the receiving instance are mapped onto the data messages, reducing the overhead that would be produced by a further ERROR_RESP message transmission. In contrast, if further negotiation of the necessary parameters needs to take place, then this is signalled via the ERROR_RESP message to the appropriate node or the system.

Upon reception of the ERROR_REP response, the node enters a final checking routine phase, wherein the layer attempts to overcome the problem with the action proposed by the SL. If the fault cannot be corrected, then the node initiates the connection release phase. The SL notifies the system of the condition that led to the termination of the association, so that appropriate actions can be found and uploaded to the node instances accordingly.

4.2.3 FCNSEP System Error Signalling

System error signalling procedures involve the explicit notification of the system network elements in cases where error recovery services cannot be offered to a particular FCNS instance. It usually follows the unsuccessful attempts of the SL protocol to negotiate with the FCNS communication layer and/or the peer entity the appropriate mechanisms that can be used for the error correction phase of a particular connection.

The error notification procedure is initiated by the SL protocol instance in the form of the ERROR_REP message, where an indication of the fault condition is given, together with identification of the particular instance that experienced the situation. If the network operator can provide the necessary functions for the error correction and recovery procedures, then these are included in the ERROR_RESP message sent back to the SL. If not, the system informs the SL protocol that it is initiating the connection release process. An entry of the condition caused the termination of the association is kept in a file, to enable the monitoring of the situation and the development of an adequate function that could be used if such conditions ever arise.

The principles governing the detection and signalling of error conditions to the system entity follow those presented for the peer error notification case, since they involve the communication of a particular FCNS instance with an external network element. The only difference observed in the two processes concerns the uploading of the information obtained to all nodes present on the given topology, increasing the network elements' awareness of any fault conditions that might affect their operation. In contrast, peer error signalling is a locally based solution where mechanisms are addressed only to the nodes in-

volved in the data transfer communication process. It is also important that the mechanisms used to secure the system signalling messages be independent of those used for the interlayer and peer error notification functions to protect the network against possible active attacks.

The purpose of the FCNSEP is the signalling of error situations to peers and the negotiation of the proposed schemes for their correction. The possibilities and fault conditions that may occur in a network vary from software to hardware faults. FCNSEP implementation does not dictate the mandatory correction of and recovery from such situations, though it provides an adequate and secured method in responding to system calls regarding the maintenance of the requested connection QoS.

The following section identifies the security considerations of the FCNSEP. The information exchanged via its messages it too vital to be sent in plaintext format, since an attacker could use these to discontinue the communication process. Consequently, details are also given with respect to possible attacks that could be launched against the system, in the absence of the necessary protection mechanisms. The discussion also includes comparison with the ICMP to further support our claims for the superiority of our designed in relation to the ICMP architecture, given the issues presented in Sections 2 and 3.

5 FCNSEP Security Considerations

The SL protocol provides the required security services for the FCNSEP signalling messages. The service of the FCNSEP can be therefore regarded as connectionless, in the sense that the messages can be transmitted by an FCNS communication layer towards the SL at any time, no matter the connection which the layer might be involved at that time. Similarly, a peer entity might be supporting more than one communication processes, yet an FCNSEP request can be launched at any time irrespective of the connection and the phase association resides at that time. Although the appropriate connection parameters are included in the FCNSEP messages to facilitate the identification and correction of the fault condition, the end-to-end connectivity issues are left to the FCNS communication layers responsible for the actual transmission of the error signalling information.

The security contexts agreed during connection establishment, provide for the peer entity authentication, integrity and confidentiality services to be used throughout the communication process, securing the data transfer on a connection basis rather than on a message-based foundation. Consequently, FCNSEP is secured on a connectionless service basis, to enable the protection of the error signalling data irrespective of the connection in question, since the peers have already been afforded the required services during association set up. However, for the FCNSEP peer error signalling process, the network operator is able to request peer-based security services for the connection, given that the appropriate certification authority can provide the necessary parameters for the FCNSEP protection.

One of the first security measures applied for the FCNSEP messages is the authentication of the data transferred either between the FCNS layers or the peer entities. The mechanisms, other than the secret keys, could be the same for both cases. The message digest is included in the data, where an identifier is also contained for protection against possible replay attacks. The identifier must be specific for a connection or error-signalling request and never be used for another FCNSEP initiation. For the FCNSEP, the SHA-512 algorithm [2] has been chosen to generate the necessary message hash, as it currently form one of the most powerful one-way hash functions, since SHA-1 has been broken [27].

Message authentication follows the data origin verification principles of the security functions offered by the SL protocol. The peer entities confirm the validity of only the sender of the FCNSEP message, relying on the peer-entity authentication functions to confirm the legitimacy of the nodes themselves. The source and data recipients verify that the message can indeed traverse the network and is not the product of an illicit node attempting to manipulate the connection.

Since the validity of the message can only ensure the legitimacy of the transmitter, FCNSEP messages are afforded the appropriate integrity functions to certify that their contents have not been tampered with, either by an adversary or a faulty transmission link. Encryption of the message is mandatory even in the interlayer signalling cases, to counter an active protocol attack. In this particular security service, the SL is able to provide the network operator and/or the host with an additional measure against replay attacks. For each FCNSEP message a timestamp value is calculated, which is then encrypted together with the message, given that authentication services have successfully been applied. By this method, the FCNS further enhances the protection of the end-systems by minimising the possibilities that any replayed messages can be accepted as valid ones.

If an attacker were able to alter the information contained in the FCNSEP data, then that could cause serious problems possibly leading to disconnection. A typical example of such a case is the repeated manipulation of the ERROR_REP and ERROR_RESP messages, whereby the adversary alters the contents of the ERROR_REPORT field issuing a FATAL error request to the peer entity. If the node accepts the request as being a legitimate one, then that would lead to the termination of the connection. Therefore, a successful DoS attack could be mounted if the messages are not sent securely over the communications channel.

Furthermore, since the messages are intended for a particular destination, the FCNSEP security mechanisms can additionally ensure the confidentiality of the data in transit, together with verifying its integrity. Message

integrity functions may be coupled with a public key scheme, whereby only the legitimate destination is able to decipher the message.

The application of functions such as non-repudiation and access control is not of importance for the FCNSEP architecture. The former is rendered redundant by the monitoring and maintenance capabilities FCNS can provide to an already established connection. The latter is made unnecessary by the assumption in this paper that all nodes are peers. The security implications of the FCNSEP architecture form one of the essential elements of the protocol and any unsecured messages are discarded.

The importance of the SL and the FCNS security functionality can also be validated via the examples given in Section 2. The protection of the FCNSEP messages is mandatory in contrast to the ICMP ones, where security is optional. Even when a functional IPsec implementation is provided, ICMP security is not set by default. This means that vital messages could be sent un-encrypted over an unsecured public network, resulting in their manipulation by an adversary. For example, the Source Quench messages could repeatedly be sent towards a host, essentially minimising its sending rate to zero, forcing a successful Denial of Service (DoS) attack. Similarly, the Destination Unreachable messages, if not validated, could cause the denial of transmission towards a specific network element, and hence cause connection discontinuity.

In contrast, the FCNSEP messages are always authenticated and secured prior to their transmission, even in the interlayer signalling procedure case, avoiding problems associated with the ICMP. Considering the case of the failed link error situation, if the ERROR_REP message was sent in plaintext format, an adversary could gain knowledge of the fault condition and force illicit messages in the network. That would result in the network element receiving the message to recalculate the specific route without any legitimate reason. If the messages were to be continuously sent, then the FCNS implementation could enter a livelock, whereby the particular implementation would always calculate the specific route, until the operator was made aware of the situation. Even worse, the manipulation of the ERROR_RESP messages could force a host to follow an alternative route that could match the needs of an attacker, causing a Redirect attack for the data and/or signalling messages. Given that FCNS has been designed to support connectivity in 3G core network systems [22, 23, 24], the attack could imply the discontinuity of a connection for several subscribers.

For the timer recalculation error signalling procedure, successful manipulation of the ERROR_REP and ERROR_RESP messages could cause a DoS attack, given that an end-system could be forced to wait indefinitely for the acknowledgment messages. In both cases, it is obvious that the support provided by the SL is vital for the successful operation of the FCNSEP. Given that the application of the FCNS security mechanisms does not affect the operation of the hosts running the stack [24], then it can be concluded that the FCNSEP provides a more complete solution than the ICMP, both in the area of error signalling notification capabilities, as well as of the system security.

Following the identification of the FCNSEP functionality and the motivation behind this proposal, we move into providing performance measurements of the error protocol and its effects on the communication procedure.

6 Evaluation Environments

The evaluation of the FCNSEP operation has taken place using the OMNET++ simulator [26], running on a Windows 2000 machine, under the Microsoft Visual C++ environment. The FCNS simulation environments are depicted in Figures 7 and 8. In the former topology, the functionality of the FCNSEP running on the interlayer error-signalling mode has been measured. In the latter model, the FCNSEP peer error signalling procedures have been put under test.

To observe the response of the system by the initiation and implementation of the FCNSEP, we have measured the FCNS overall message loss ratio, as well as the FCNS packet throughput response throughout the duration of the simulation runs (in kilobits per simulation time second). For both environments, the datarate has been set to 10 Mbps to emulate the bandwidth offered by a typical Ethernet network, whilst the message and service primitives processing delay has been set to 2 msec. We have also applied a 5 msec processing time for a message to traverse the FCNS architecture and hence reach the Physical layer pending its transformation onto the FCNS frame. The size of the FCNS packet and frame are 12232 bits and 12368 bits respectively, when full security measures are applied. Given the 10Mbps datarate, then the optimum transmission time for a single frame will be 1.24 msec.

For the interlayer error-signalling procedures, the environmental variables correspond to a Round Trip Time (RTT) of 60 msec for the link between the input and output FCNS instances (Figure 7). The example erroneous case, involved an initial 25 msec message timer that had to be changed to adapt to the network environment of the propagation delay of 30 msec. We have implemented an ARQ Go Back N flow control mechanism, with the Physical layer constructing the FCNS frames as they arrive by the upper layer, that is, without storing them first into a buffer before creating the message block. The block has been assumed to consist of 32 FCNS frames. All measurements have been taken with respect to the Bit Error Rate (BER) probability, for the constant RTT of 60 msec and FCNS packet and frame sizes.

The effects of the FCNSEP implementation on the system's loss ratio are depicted in Figure 9. On a theoretical level, the minimum effect of the FCNSEP realisation will be as follows. For the receiving instance to adjust its timer, the NODE_ALERT message is sent after three

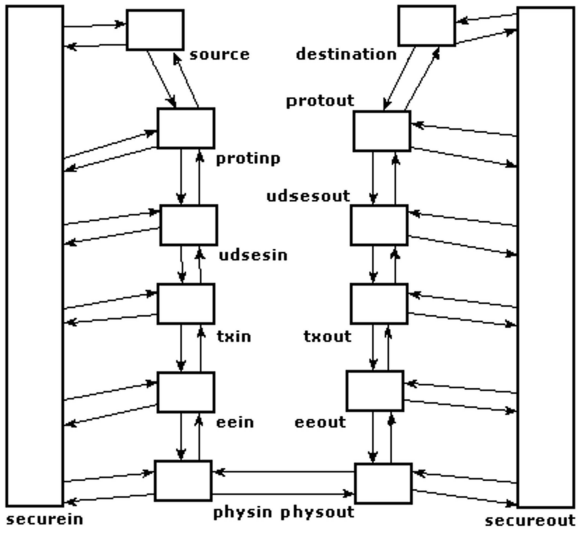

Figure 7: FCNS stack OMNET++ simulation environment

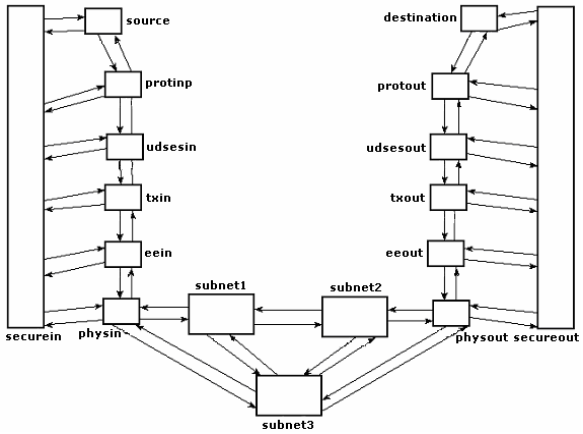

Figure 8: FCNS generic packet-switched OMNET++ simulation environment

frames have been discarded, or the receiving timer has expired three times. The time it will take for the SL to issue the appropriate action and the receiver to adjust the new value is 8 msec and hence an overall 13 msec for the Physical layer protocol to be notified of the alteration. For the optimum transmission time of 1.24 msec, this should imply a loss of 10 frames, due to rejection by the receiving protocol instance.

Overall, the discarded messages should have been 13 from just the alteration process, plus 7 additional frames due to the timer expiration at the Physical layer before its adjustment ($32 message \times 1.24 msec = 39.68 msec, 9.68 msec \rightarrow 7 frames$), making up for an overall loss of 20 frames. However, due to the discrete-event nature of the simulator, the loss induced on the system has only been 5 frames (3 that were lost before the FCNSEP application, 1 after the ERROR_RESP re-

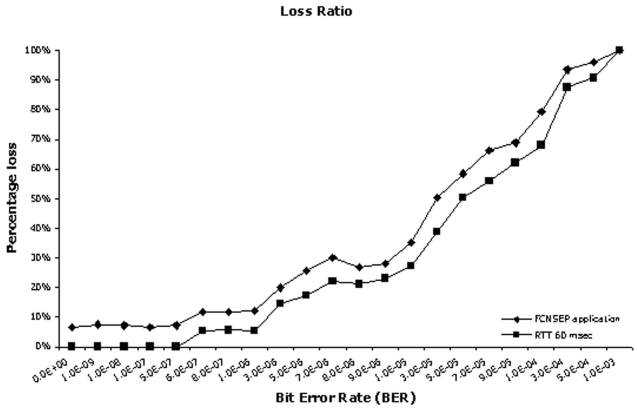

Figure 9: FCNS loss ratio - Variable BER, constant data size and RTT of 60 msec

ception and 1 after the adjustment of the timer). This is due to the fact that throughout the FCNSEP signalling procedures, the sender has sent no messages as would have been expected in a real-network situation.

These effects reflected upon the FCNS packet throughput response as is depicted in Figure 10. The decrease observed is in the range of 5-7 frames, depending on the FCNS instance status and the BER value. As the latter is increased, more frames and FCNSEP messages may be discarded at the receiver due to unrecoverable bit pattern errors, resulting in the response of Figure 10.

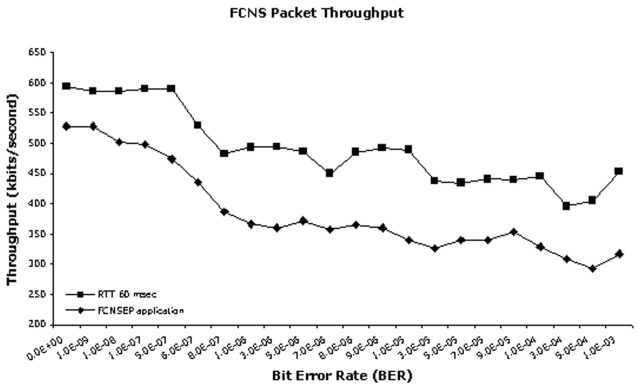

Figure 10: FCNS packet throughput - Variable BER, constant data size and RTT of 60 msec

For the peer error-signalling procedures, the model of Figure 8 has been afforded an overall 260 msec RTT delay, divided as 40 msec transmission delay imposed to the input FCNS instance - subnet1 and subnet 2 - output FCNS instance links, with the remaining 50 msec induced in the subnet1 - subnet 2 one. All links to and from subnet 3 have been assigned a propagation delay of 50 msec, to compensate for an alternative to the primary (input instance to subnet 1 to subnet 2 to output instance) route.

The effects of the FCNSEP application are illustrated in Figure 11, where an indication of the loss ratio is given in relation to the measurement taken for an error-signalling free transmission. The increase of the number of lost messages with respect to the response of Figure 9 is due to the application of various FCNS functions for the establishment of the data transfer process. As the BER increases, the probability that any message, including those of the FCNSEP, is received on error in maximised, irrespective of the phase communication may reside. Additionally, if the primary route fails, then the transmission of the FCNS messages via an alternative path will result in further losses, due to the path and route verification procedures that must take place between the End-to-End layer and the SL at all network nodes. The FCNSEP will be used to signal the route failure to the network nodes, increasing the amount of data frame lost during the process. Furthermore, SC exchanges may need more than two attempts to complete, since BER may affect the contents of the respective primitives at any node of the simulation environment.

Overall, there is a significant increase of the loss ratio when FCNSEP is applied. In contrast to the model of Figure 7, we have introduced in this environment two additional delay parameters, namely the insertion and queuing losses of the subnetworks present in the topology. This approach enabled the simulation of a realistic network model, where message reception timers may be expired due to the queuing of the FCNS frames and FCNSEP messages at the router buffers, or due to a node switching messages slower than the peer sending and receiving rates.

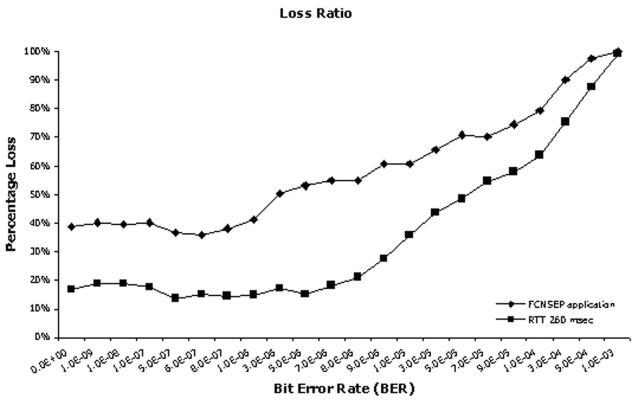

Figure 11: FCNS loss ratio - Variable BER, constant data size and RTT of 260 msec

To enable the identification of the FCNS packet throughput effects, the observation of Figure 12 is provided.

The throughput difference falls into the area of *64,133 bits or 5 packets* for small error probabilities, rising to more than *103,807 bits or 8* packets for larger BER. Given that message and service primitives processing times at

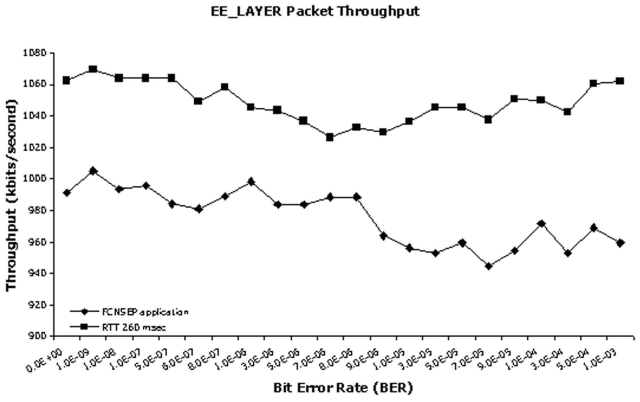

Figure 12: FCNS packet throughput - Variable BER, constant data size and RTT of 260 msec

the FCNS instances are identical to those of the model presented in Figure 7, then the response obtained conforms to the design expectations of the FCNSEP.

7 Conclusions

In this paper, the error protocol of the FCNS architecture has been presented. FCNSEP defines a framework for use within network architectures, where explicit error notification should take place irrespective of the technology supporting the connection. At any given time, an erroneous situation such as a faulty transmission link or excessive congestion could arise in a network topology, as well as procedural errors for the stack protocols. This mandates the need for an architecture that can provide for the signalling of the error parameters throughout the environment nodes.

The ICMP has been described, including details of identified disadvantages and implementation pitfalls of the architecture. The overall security of the ICMP system relies on external to the protocol parameters and set of communication rules, in contrast to the FCNSEP where security services are afforded for the protocol independently of the underlying network structure and FCNS implementation. Moreover, FCNSEP addresses a wide variety of error conditions that could affect the communication flow, reporting any condition to the system and/or the peer entity and not only those specific for the link-based data transmission. Finally, FCNSEP is not bound by the functionality of the FCNS stack and hence can be used in virtually any packet-switched architecture and telecommunication system.

The use of FCNSEP constitutes a last resort in attempting to notify and recover from an error network situation. Parameters such as congestion control can be added at the user frames or packets, to reduce the overhead that could be produced by the FCNSEP messages. However it is imperative that there exists a means of alert-

ing the system administrator of any situation that could endanger the communication process, enabling not only the correction of such a situation but the initiation of the appropriate mechanisms that could afford its future prevention.

FCNSEP can support the notification of the network elements involved in the communication process, irrespective of the nature of the network they reside, thus offering a degree of connection monitoring throughout all phases of the association. Current research focuses on applying a robust and secure error signalling mechanism for adhoc wireless and sensor networks and, hence, the refinement of FCNSEP for use in such environments. The particularities of such topologies lay, in most of the cases, in the absence of a dedicated server that could act as the intermediate between the peer network and the network management system. That interface could enable the transmission of the respective actions inside the ERROR_REP and ERROR_RESP messages in an erroneous situation. However, the independence of such networks signifies the need for a more flexible architecture. In this context, the provision of FCNSEP as the error signalling/notification architecture supporting all packet-switched architectures can enable the secure deployment of virtually any network topology, with applications ranging from the academia to military communications.

References

[1] Anon, *Open Systems Interconnection - Basic Reference Model part 2: Security Architecture*, International Standards Organisation (ISO), Information processing systems, ISO 7498-2, 1989.

[2] Anon, *Secure Hash Standard*, Federal Information Processing Standards (FIPS), Publication 180-2, 2002.

[3] Anon, CERT Coordination Centre, *CERT/CC Vulnerability Notes VU#471084, VU#104823, VU#221164, VU#471084 and VU#612833*, Software Engineering Institute, Carnegie Mellon University, USA, 2002-2003. (http://www.cert.org)

[4] S. Bellovin, "Problem areas for the IP security protocols," in *Proceedings of the 6th Usenix UNIX Security Symposium*, pp. 1-16, 1996.

[5] S. Bellovin, "Probable plaintext cryptanalysis of the IP security protocols," in *Proceedings of the 1997 Symposium on Network and Distributed Systems Security, Institution of Electrical and Electronic Engineers (IEEE)*, pp. 52-60, 1997.

[6] D. Brumley and D. Boneh, *Cryptographic Libraries and Applications Do not Adequately Defend Against Timing Attacks*, CERT Coordination Centre, Software Engineering Institute, Carnegie Mellon University, USA, CERT Vulnerability note VU #997481, 2003. (http://www.cert.org)

[7] A. Conta and S. Deering, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol version 6 (IPv6) specification*, IETF RFC 2463, 1998.

[8] S. Deering and R. Hinden, *Internet Protocol version 6 (IPv6) specification*, IETF RFC 2460, 1998.

[9] C. Ellison and B. Schneier, "Ten risks of PKI: what you've not being told about public key infrastructure," *Computer and Security Journal*, vol. 16, no. 1, pp. 1-7, 2000.

[10] N. Ferguson and B. Schneier, *A Cryptographic Evaluation of IPsec*, Counterpane Internet Security Inc, 1999. (http://www.counterpane.com)

[11] IETF, *IPv6 Operations (v6ops) documents*, 2003-2005. (http://www.ietf.org)

[12] *Insecure Security Focus Group Mailing Lists.* (http://lists.insecure.org)

[13] *IPng Mailing List*, Western Computers Users Group, Western Washington University, USA. (http://www.wcug.wwu.edu/lists/ipng)

[14] S. Kent and R. Atkinson, *Security Architecture for the IP*, IETF RFC 2401, 1998.

[15] P. Kirstein, *Factors Influencing IPv6 Deployment*, IR6 WINIT, European Commission IPv6 Task Force, 1st phase, 2001. (http://www.ec.ipv6tf.org/in/i-documentosOLD.php)

[16] S. Lin and J. D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, 1983.

[17] A. Manion, *Multiple Vendors IKE Implementations Do not Properly Handle IKE Packets*, CERT Coordination Centre, Software Engineering Institute, Carnegie Mellon University, USA, Vulnerability node VU#287771, 2003. (http://www.cert.org)

[18] P. Nikander, *Denial of Service, Address Ownership and Early Authentication in the IPv6 World*, Ericsson Research, Finland, 2001. (http://www.tcm.hut.fi)

[19] A. Nuopponen and S. Vaarala, "Attacking predictable IPsec ESP Initialisation Vectors," LNCS 2513, pp. 160-172, Springer-Verlag, 2002.

[20] N. Olifer and V. Olifer, *Computer Networks: Principles, Technologies and Protocols for Network Design*, John Wiley and Sons Ltd, 2005.

[21] T. Sabin, *Multiple IPsec Implementations Do not Adequately Validate Authentication Data*, CERT Coordination Centre, Software Engineering Institute, Carnegie Mellon University, USA, CERT Vulnerability note VU #459371, 2002. (http://www.cert.org)

[22] T. Stergiou, D. Delivasilis, R. Green, and M. S. Leeson, "Future core networks system (FCNS) - A secure signalling protocol stack for the UMTS core network," in *Proceedings of the IEE 3G2002 Conference*, vol. 489, pp. 329-333, London, UK, 2002.

[23] T. Stergiou, R. Green, and M. S. Leeson, "Protocol stack design for 3rd generation mobile systems - UMTS core network," in *Proceedings of the Third International Network Conference*, pp. 485-493, Plymouth, UK, 2002.

[24] T. Stergiou, M. S. Leeson, and R. J. Green, "An alternative architectural framework to the OSI security model," *Computers & Security Journal*, vol. 23, no. 2, pp. 137-153, Elsevier, 2004.

[25] M. Szalay, *A Special Attack Against IPsec*, SANS Information Security Reading Room, 2000. (http://rr.sans.org)

[26] A. Vargas 2005, *OMNET++ 2.0 Discrete Event Simulator*, Technical University of Budapest, Hungary. (http://whale.hit.bme.hu/omnetpp)

[27] X. Wang, Y. L. Yin, and H. Yu, *Collision Search Attacks on SHA1*, Shandong University, China, 2005.

Theodore Stergiou has received his PhD in Engineering from the University of Warwick, UK in 2004, for protocol security issues in third generation telecommunication systems. He has been involved since in academic lecturing in the area of computer networks and security, as well as in acting as an independent security consultant. His research interests include secure communication protocols design, wireless and mobile security, sensor network security, clinical information systems security, intrusion detection/prevention and intrusion tolarance mechanisms. Currently, he is an external member of the Business System Integration Team of Vodafone S.A. Hellas. He is an active reviewer for the IEEE Communication Letters, IEEE Transactions on Computers, the International Journal of Network Security, IEEE Globecom 06 and IEEE ICC 07. Dr Stergiou is a member of the Institution of Engineering and Technology, UK (MIET) and a member of the Institution of Electrical and Electronic Engineers (MIEEE). He also serves as a member of the IEEE Communications Society and the Communications and Information Security Technical Committee.

Dimitrios L. Delivasilis was born in Greece in 1976. He holds a B.Eng. in Computer Hardware and Software Engineering, an MSc. in Data Communication Systems and a Ph.D. in Data Security for Third Generation (3G) Telecommunication Systems, from the universities of Coventry, Brunel and Warwick respectively. Prior to the beginning of his academic career he has obtained over four years commercial experience in R&D departments of telecommunication industry, in the countries of England and Greece. His current research interests include wireless security, cryptographic algorithms, intrusion tolerance systems and biometrics. His published scientific work includes several international journals and conferences, as well as, a filed British patent. He is a member of International Association for Cryptologic Research (IACR), IEEE Computer Society and serves as a reviewer for IEE Proceedings.