8 Standard Guide for EDI (HL7)
Communication Security

Bernd BLOBEL, Volker SPIEGEL, Peter PHAROW, Kjeld ENGEL and Rolf KROHN
Otto-von-Guericke University, Medical Informatics Department, Magdeburg, Germany

(approved as an HL7 Informative Document in September 1999)

8.1 Scope

This Standard Guide gives the framework for secure end-to-end communication of EDI messages focusing on HL7. It is based on the common security model that distinguishes the concepts of communication security as rather globally controlled and application security as rather locally controlled. The concepts of quality and safety are not considered. Each of these concepts defines a set of security services, which are provided by sets of security mechanisms based on security algorithms applied to data. The different levels of granularity allow views of different groups of users - including medical users, system administrators, and implementers - within the same specification framework. Additionally, for implementation, the protocol-services-mechanisms relationships with respect to standards and products also have to be considered.

The Standard Guide starts with the specification of internal security services needed for the provision of secure communication between information systems. External security services, like services provided by Trusted Third Parties (TTPs) to facilitate trustworthiness between the principals involved in communication, such as key management, registration services, naming services, certification service, directory services or secure time services, as well as security services for application security, such as authorisation, access control, data element security, data base security and audit, are not considered. These external security services are outside the scope of the present guidance, which only deals with secure communication of EDI messages applying communication security. Communication security includes the assembling and merging of already secured data elements (done by application security services as the integrity of data and the accountability for data and procedures) to complete the security-enhanced EDI messaging. Because the EDI protocols specify only the syntax and semantics of messages exchanged, but not the network infrastructure used, the importance of service availability is not considered here.

Reacting on threats (active users' interactions) or vulnerabilities (systems' behaviour), the security services defined provide the link between the security requirements and objectives as described by security policies, and the security mechanisms and management needed to satisfy these requirements. Each of the security services can be implemented by one or more types of security mechanisms according to different levels of security needed by different policies and applications. The security policy specifies, among other things, the legal, organisational and social business framework, the analysed threats, accepted risks, and intended organisational and technical solutions. If systems of different
organisational and/or policy domains communicate, policy bridging is required. The policy agreed upon defines legal, organisational and technical security issues and the functionality permitted.

Considering the granularity of services and mechanisms, and abstracting from the specific and highly dynamic implementation details - such as using various cryptographic mechanisms of different strengths for implementing security mechanisms; using different security infrastructures available, such as public key or symmetric key infrastructure; or using several communication protocols on different layers, such as, HTTP, SMTP or FTP) - this specification follows a generic and open architectural approach. Thus, it is very flexible in terms of composition of services needed to protect health information systems from the security threats and risks according to the specific Healthcare processes and environments.

8.2 Conformance

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” used in this document have to be interpreted as described in RFC2119.

8.3 Normative References

ECMA-219
Standardising Information and Communication Systems. Authentication and Privilege Attribute Security Application with related Key Distribution Functions – Part 1, 2 and 3.

ISO7498-2

ISO/IEC7816
International Organisation for Standardisation: Information technology, Identification cards, Integrated circuit(s) cards with contacts, multiple Parts (1-11).

ISO/IEC9594-8

ISO/IEC9735
International Organisation for Standardisation: Electronic data interchange for administration, commerce and transport (EDIFACT), Application level syntax rules, multiple Parts (1-10).

ISO/IEC9796

ISO/IEC9797

ISO/IEC9798
International Organisation for Standardisation: Information technology, Security techniques, Entity authentication, multiple Parts (1-5).


8.4 Definitions

The definitions used have been collected from the “HL7 Security Services Framework – Part 3: Glossary” document. To facilitate the reading of this document, these definitions are also provided in the following table.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accountability</td>
<td>Ensures that the actions of an entity may be traced uniquely to that entity (ISO 7498-2).</td>
</tr>
<tr>
<td>Authentication</td>
<td>Provides assurance that an entity is the one claimed (ISO 7498-2).</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>Ensures that information is not made available or disclosed to unauthorised individuals, entities or processes (ISO 7498-2).</td>
</tr>
<tr>
<td>Cryptographic Check Value</td>
<td>Information that is derived by performing a cryptographic transformation on data.</td>
</tr>
<tr>
<td>Data Origin Authentication</td>
<td>Occurs when a principal claiming to be the originator of some data includes its identity along with that data; glued together using the integrity service.</td>
</tr>
</tbody>
</table>
Identification The process of telling a system the identity of a subject using its unique name.

Integrity The property that data has not been altered or destroyed in an unauthorised manner (ISO 7498-2).

Masquerade Occurs when a principal pretends to be a different principal.

Message Authentication Code (MAC) Data derived from a message using symmetric cryptography techniques and a secret key to provide authenticity of integrity and origin.

Non-repudiation of Origin Occurs when the principal receiving data claims to know the data originator so that the sender cannot later falsely deny having sent the data. The integrity service is needed for establishment.

Non-repudiation of Receipt Occurs when the principal sending some data claims to know that this data has successfully reached its intended receiver. The integrity service is needed for establishment.

Policy A set of rules that specifies the procedures and mechanisms required to maintain the security of a system, and the security objects and security subjects under the purview of the policy (ECMA).

Principal A user or programmatic entity with the ability to use the resources of a system.

Strong Authentication Authentication by means of cryptographically derived credentials.

Trusted Third Party A security authority or its agent trusted by other principal with respect to security-related activities.

8.5 Abbreviations

ANSI American National Standards Institute
DN Distinguished Name
EDI Electronic Data Interchange
EDI-MS EDI Messaging System (ITU-T X.435, part of MHS)
ESS Enhanced Security Services (Part of S/MIME Version 3)
FIPS PUB Federal Information Processing Standards Publication (by NIST)
FTAM File Transfer, Access and Management (ISO/IEC 10607)
FTP File Transfer Protocol
HL7 Health Level Seven Application Protocol for Electronic Data Exchange in Healthcare
HLLP Hybrid Lower Layer Protocol
IESG Internet Engineering Steering Group
IETF Internet Engineering Task Force
IPSEC IP Security
ITU International Telecommunication Union
L2F Layer 2 Forwarding
L2TP Layer 2 Tunnelling Protocol
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLP</td>
<td>Lower Layer Protocol</td>
</tr>
<tr>
<td>MLLP</td>
<td>Minimal Lower Layer Protocol</td>
</tr>
<tr>
<td>MIME</td>
<td>Multipurpose Internet Mail Extension</td>
</tr>
<tr>
<td>MOSS</td>
<td>MIME Object Security Services</td>
</tr>
<tr>
<td>MSP</td>
<td>Message Security Protocol (NIST SDNS)</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute for Standards and Technology</td>
</tr>
<tr>
<td>NLSP</td>
<td>Network Layer Security Protocol</td>
</tr>
<tr>
<td>NRD</td>
<td>Non-Repudiation of Delivery</td>
</tr>
<tr>
<td>NRO</td>
<td>Non-Repudiation of Origin</td>
</tr>
<tr>
<td>NRR</td>
<td>Non-Repudiation of Receipt</td>
</tr>
<tr>
<td>NRS</td>
<td>Non-Repudiation of Submission</td>
</tr>
<tr>
<td>NRT</td>
<td>Non-Repudiation of Transport</td>
</tr>
<tr>
<td>PCT</td>
<td>Private Communications Technology Protocol</td>
</tr>
<tr>
<td>PEM</td>
<td>Privacy Enhanced Mail</td>
</tr>
<tr>
<td>PGP</td>
<td>Pretty Good Privacy</td>
</tr>
<tr>
<td>PKCS</td>
<td>Public Key Cryptography Standard</td>
</tr>
<tr>
<td>PPP</td>
<td>Point-to-Point Protocol</td>
</tr>
<tr>
<td>PPTP</td>
<td>Point-to-Point Tunnelling Protocol</td>
</tr>
<tr>
<td>RFC</td>
<td>Request For Comments</td>
</tr>
<tr>
<td>S/MIME</td>
<td>Secure/MIME</td>
</tr>
<tr>
<td>SDE</td>
<td>Secure Data Exchange (IEEE 802.10)</td>
</tr>
<tr>
<td>SDNS</td>
<td>Secure Data Network System</td>
</tr>
<tr>
<td>SFTP</td>
<td>Secure File Transfer Protocol</td>
</tr>
<tr>
<td>SHTTP</td>
<td>Secure HyperText Transfer Protocol</td>
</tr>
<tr>
<td>SILS</td>
<td>Standard for Interoperable LAN Security (IEEE 802.10)</td>
</tr>
<tr>
<td>SMTP</td>
<td>Simple Mail Transfer Protocol</td>
</tr>
<tr>
<td>SOCKS</td>
<td>Sockets Secure Protocol</td>
</tr>
<tr>
<td>SPKM</td>
<td>Simple Public-Key GSS-API Mechanism</td>
</tr>
<tr>
<td>SSH</td>
<td>Secure Shell</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td>SSPI</td>
<td>Microsoft Security Support Provider Interface</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>TLSP</td>
<td>Transport Layer Security Protocol</td>
</tr>
<tr>
<td>TTP</td>
<td>Trusted Third Party</td>
</tr>
<tr>
<td>TVP</td>
<td>Time Variant Parameter</td>
</tr>
</tbody>
</table>
8.6 EDI Communication Security Services

Following the concept of communication security, a set of basic security services is required for the security enhancement of EDI messages using wrapping techniques. The services needed to protect health information systems from security threats and risks should be selected and composed according to specific Healthcare processes and environments. Some of the security services, such as principal authentication and confidentiality, prevent security breaches, while other services, like integrity or accountability, give only the evidence that an attack has taken place without technically preventing it.

8.6.1 Threats and Security Services

In the EDI environment, the threat model consists of at least two principals that are authorised to perform message transmissions to each other using several communication protocols over various infrastructures. Threats are active user (attacker) interactions that cause system vulnerability. According to the security policy, threats, vulnerabilities and accepted risks determine the security requirements that are fulfilled by appropriate security services. The following consideration is based on the common security model, which distinguishes the concept of communication security as rather globally controlled and the concept of application security as rather locally controlled. Each of these concepts defines a set of security services, which are provided by sets of security mechanisms based on security algorithms applied to data. The different levels of granularity allow views of different groups of users, including medical users, system administrators, and implementers, within the same specification framework. Additionally, for implementation, the protocol-services-mechanisms relationships with respect to standards and products have to be considered as well.

An unauthorised principal may try to attack the communication system using passive techniques, such as monitoring, listening and sniffing of data system exploration, or traffic analysis, or active techniques, such as creation, insertion, deletion and replay of data. This may enable the intruder to perform masquerading. A short summary of threats and security services is given in Table 30 below.

Table 30: Threats and Security Services

<table>
<thead>
<tr>
<th>Threats</th>
<th>Security Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>masquerading (unauthorised use of authorised services)</td>
<td>principal identification and authentication</td>
</tr>
<tr>
<td>data manipulation</td>
<td>integrity</td>
</tr>
<tr>
<td>concealment or manipulation of data origin</td>
<td>accountability in the sense of non-repudiation of origin</td>
</tr>
<tr>
<td>repudiation of receipt</td>
<td>accountability in the sense of non-repudiation of receipt</td>
</tr>
<tr>
<td>disclosure of data</td>
<td>confidentiality</td>
</tr>
</tbody>
</table>
8.6.2 Security Services and Security Mechanisms

Reacting on threats (active users' interaction) or vulnerabilities (systems' behavior), the security services defined provide the link between the security requirements and objectives as described by security policies, and the security mechanisms and management needed to satisfy these requirements. Each of the security services can be implemented by one or more types of security mechanisms (the multiplicity is 1:n) according to different levels of security needed by different policies and applications. The security policy specifies, among other things, the legal, organisational and social business framework, the analysed threats, accepted risks and intended organisational and technical solutions. If systems of different organisational and/or policy domains communicate, policy bridging is required. The policy agreed upon defines legal, organisational and technical security issues and the functionality permitted. In general, security services are independent of special scenarios and implementations as they define a set of security functions.

For health information systems, internal and external security services can be distinguished from each other. Internal security services describe functions provided by communicating and co-operating information systems for the provision of communication security.

External security services are services provided by Trusted Third Parties (TTPs) to facilitate trustworthiness between the principals involved in communication and co-operation. These services (e.g. key management, registration services, naming services, certification services, directory services or secure time services) as well as application security services (such as authorisation, access control, integrity and confidentiality of data, accountability for data and procedures, audit) are not discussed here. However, sometimes data secured by such services needs to be communicated to facilitate the security management or to verify accountabilities.

**Table 31: Security Services and Their Enforcing Security Mechanisms**

<table>
<thead>
<tr>
<th>Security Services</th>
<th>Security Mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Asymmetric Techniques</td>
</tr>
<tr>
<td>Principal Identification and</td>
<td>Digital Signature, TVPs</td>
</tr>
<tr>
<td>Authentication</td>
<td></td>
</tr>
<tr>
<td>Data Origin Authentication</td>
<td>Digital Signature, cryptographic check value, DN</td>
</tr>
<tr>
<td>Integrity</td>
<td>Digital Signature, cryptographic check value</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>Encryption</td>
</tr>
<tr>
<td>Accountability</td>
<td>Security Audit (using reports, log files, receipts, time</td>
</tr>
<tr>
<td></td>
<td>stamps and distinguished names)</td>
</tr>
<tr>
<td>Non-repudiation(^2) (of origin</td>
<td>Digital Signature, cryptographic check value, time stamps, DN</td>
</tr>
<tr>
<td>and receipt)</td>
<td></td>
</tr>
</tbody>
</table>

\(^2\) Non-Repudiation is a part of the accountability service.
Table 31 lists the internal security services that EDI SHOULD offer for the secure communication of messages including the security mechanisms used to enforce them. Because the EDI protocols specify only the syntax and semantics of messages exchanged, but not the network infrastructure used, the importance of service availability is not considered here.

The client/server-based communication protocols used for transmission of EDI messages MUST at least provide principal authentication and data integrity (including data origin authentication). This set of required services is called minimal set of required security services.

The implementation of the security mechanisms by specifications, algorithms and products is dependent on the state of current technology, the development of new technologies, and the availability of these technologies to potential attackers. Therefore, it is a highly dynamic procedure. The Internet environment especially raises various possibilities of new attacks, challenges, and countermeasures. The result is new security techniques whereas the security services and mechanisms themselves remain rather stable. Thus, a correct implementation of security mechanisms is REQUIRED (i.e. that the mechanisms are covered completely by adequate security techniques such as cryptographic algorithms for software and technical means for hardware)).

8.6.3 Architectural Placement of Security Services and Security Protocols

The protocol stack reflects the fact that the communication functions are complex and usually divided into independent layers or levels. A stack is a collection of protocol layers that implement network communication. The protocol associated with each layer communicates only with the layers immediately above and below it, and assumes the support of underlying layers. Lower layers are closer to the hardware and higher layers are closer to the user. The number of layers and tasks that each layer performs depends on which stack is used (e.g. OSI, TCP/IP).

The protocol stack is a set of operations that work together to create a seamless path across the layers. As data proceeds from one station to another, the data first begins to travel down the protocol stack until it reaches the physical layer. At the physical layer, the data is placed on a medium (i.e. copper, fibre, wireless). The data then traverses the network until it reaches the receiving station where the data travels up the protocol stack. As the data traverses the network, it may be reformatted (protocol adapted) to the protocol being used in a particular network segment [5].

The applicability of security services is not bound to a specific layer of the OSI protocol stack (ISO7498-2, ISO10181), but from the communication protocol perspective, only four levels need to be distinguished: application layer, transport layer, network layer, and data link layer ([2]). Abstracting from the placement on different communication layers and choosing different sets of security services, each existing and upcoming security protocol can be described completely. For example confidentiality, integrity, and principal authentication are selected and placed on the transport layer, and the security protocol TLS is specified that provides these security services for the session layer and beyond. Table 32 below gives an overview of existing protocols usable for secure communication purposes.
Table 32: Placement of Security Services

<table>
<thead>
<tr>
<th>Security Services</th>
<th>Confidentiality</th>
<th>Integrity</th>
<th>Entity Authentication</th>
<th>Data Origin Authentication</th>
<th>Non-Repudiation of Origin</th>
<th>Non-Repudiation of Receipt</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSI Layers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Link</td>
<td>SILS/SDE, PPTP(^2), L2TP</td>
<td>SILS/SDE, L2TP</td>
<td>PPTP, L2TP, L2F</td>
<td>SILS/SDE, L2TP</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Network</td>
<td>IPSEC, NLSP</td>
<td>IPSEC, NLSP</td>
<td>IPSEC, NLSP</td>
<td>IPSEC, NLSP</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Transport</td>
<td>SOCKS, TLSP, SSL, TLS, PCT, SSH</td>
<td>SOCKS, TLSP, SSL, TLS, PCT, SSH</td>
<td>SOCKS, TLSP, SSL, TLS, PCT, SSH</td>
<td>TLSP</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Application</td>
<td>SHTTP, SPKM, MHS, MSP, PEM, SFTP, PGP/MIME, MOSS, S/MIME, PKCS#7</td>
<td>SHTTP, SPKM, MHS, MSP, PEM, SFTP, PGP/MIME, MOSS, S/MIME, PKCS#7</td>
<td>SHTTP, SPKM, MHS, MSP, PEM, SFTP, PGP/MIME, MOSS, S/MIME, PKCS#7</td>
<td>SHTTP, SPKM, MHS, MSP, PEM, SFTP, S/MIME, ESS</td>
<td>SPKM, MHS, MSP, SFTP, S/MIME, ESS</td>
<td></td>
</tr>
</tbody>
</table>

Following the HL7 client/server network architecture using communication servers for end-to-end communication where applications meet and exchange their messages, the security services providing HL7 communication security MUST be placed on the transport layer or application layer of each principal. Additional protection MAY be applied using security services provided by protocols located at the lower layers (placed on the network layer or data link layer). Placing the services on the application layer allows security protocol elements that are dependent of the application (e.g. HTTP or FTP). Security protocol elements on the transport layer provide protection on an end-system basis. Establishing end-to-end security REQUIRES that the end systems are trusted, but all underlying communication network(s) MAY be non-trusted.

HL7 communication security (i.e. wrapping HL7 messages by security envelopes when in transit as shown in Figure 24) can be achieved by using HL7 external communication protocols with security already implemented (as SHTTP or SFTP) or by securing the HL7 lower layer protocols located at the session layer (based on the sockets interface) and beyond.

\(^2\) PPTP does not address any security issues in the current version, but end-to-end security is addressed by PPP, which is tunnelled by PPTP through an IP network.

\(^4\) Only the client is authenticated to the server by showing that he is able to apply message enhancement according to the security requirements of the server.
For interoperability reasons, only standard documents available as ISO Standards, IETF/IESG Internet Standards (RFCs), IETF Internet Drafts (IDs), NIST publications (NIST FIPS PUB) or similar MUST be used for the security enhancement of existing protocols (standards conformance).

In the next sections, the basic protocols are described that are usable for secure communication regarding the remarks mentioned above.

8.6.3.1 IPv6

TCP/IP, the protocol suite on which the global Internet and corporate intranets are based, is decades old and therefore breaking. IPv4, the current version of the Internet Protocol, is reaching the end of its life. The main reasons are:

- limitations in the number of devices it can address
- growing demand for new functionality
- lack of essential security features

IP lives in end computer systems and in the routers that connect them. When an application on one end system wants to send data, it encapsulates the message in an IP protocol data unit (PDU) which traverses a path of networks connected by routers to reach its intended target.

The key services provided by IP in this exchange are as follows:

- Addressing: the PDU must inform each router it encounters of its destination.
- Packetising: physical networks specify a Maximum Transmission Unit (MTU), or packet size, which PDU's must observe.
- Service class: specifies treatment of PDU relative to other traffic as regards priority, reliability and delay.
- Security: PDUs can be encrypted and contain signature and authentication data.
The design of the new protocol IPv6 (also called IP next generation or IPng) pays special attention to addressing and security services. It also improves overall network performance and provides enhanced service class options.

The address field size in IPv6 is increasing from 32 bits to 128 bits and is therefore 2^96 times bigger than today's IP address space. IPv6 also describes the rules for three modes of addressing: uni-cast, which is one host to one other host, any-cast, which is one host to the nearest of multiple hosts and multicast, which is one host to multiple hosts. Uni-cast addresses target individual hosts. Several variants of uni-cast are allowed, including an IPv4-compatibility mode intended to provide a smooth migration path. Any-cast addressing is a refinement of uni-cast that streamlines routing. The address provides the possibility of sending a message to the nearest of several possible gateway hosts with the idea that any one of them can manage the forwarding of the packet to others. Multicast allows messages to be sent to a predefined group of uni-cast addresses with a single multicast address.

IPv6 provides improved performance in three ways.

- Reduced number of header fields. The so-called packet header in IPv6 is, at 40 bytes, actually longer than the IPv4 header (20 bytes minimum), but it contains fewer fields. This expedites processing by the router.
- Fixed-length packet header. IPv4 allows a number of options in the packet header that can change its size. IPv6 has a 40 bytes header, which again streamlines the work done by routers.
- The IPv6 header now includes extensions that allow a packet to specify a mechanism for authenticating its origin in order to ensure data integrity / privacy.
- No fragmentation allowed. IPv6 accommodates the MTU requirements of intervening networks at the source end, using an algorithm to discover the transmission path and lowest MTU. This saves the overhead of fragmentation and re-assembly.

Finally, an impressive array of security features has been built into IPv6. In spite of the given abundance of proven application-level mechanisms like S/MIME, Privacy Enhanced Mail (PEM), S-HTTP and SSL this is necessary because of the following: IP-level security works for all applications, whether aware or ignorant of security concerns. IPv6 supports two security functions: authentication and privacy. The authentication mechanism ensures that a received packet was in fact transmitted by the source identified in the packet header, and not by a forger or interloper. As a corollary, authentication ensures that the message has not been tampered within transit. Privacy, the assurance that a message can be seen only by authorised parties, is implemented by strong encryption.

The set of security services offered includes access control, connectionless integrity, data origin authentication, protection against replays (a form of partial sequence integrity), and confidentiality. These services are provided at the IP layer, offering protection for IP and/or upper layer protocols. Therefore, they can be used by any higher layer protocol (e.g. TCP, UDP, ICMP, and BGP).

These objectives are met by the use of two traffic security protocols - the Authentication Header (AH) and the Encapsulating Security Payload (ESP) - and by using cryptographic key management procedures and protocols. The set of IPsec protocols employed in any context, and the ways in which they are employed, will be determined by the security and system requirements of users, applications, and / or sites / organisations.
When these mechanisms are correctly implemented and deployed, they ought not to adversely affect users, hosts, and other Internet components that do not employ these security mechanisms for protection of their traffic. These mechanisms are also designed to be algorithm independent to allow easy integration with new, more powerful algorithms once they are available. This modularity also permits selection of different sets of algorithms without affecting the other parts of the implementation. For example, different user communities may select different sets of algorithms if required.

A standard set of default algorithms is specified to facilitate interoperability in the global Internet. The use of these algorithms, in conjunction with IPsec traffic protection and key management protocols, is intended to permit system and application developers to deploy high-quality, Internet layer, cryptographic security technology [3].

8.6.3.2 SSL and TLS

The Secure Sockets Layer (SSL) was developed by Netscape Communications Corporation to provide security and privacy over the Internet. The protocol supports server and client authentication and is application independent; allowing protocols like HTTP, FTP and Telnet to be layered on top of it transparently. The SSL protocol is able to negotiate encryption keys as well as authenticate the server before data is exchanged by the higher-level application. It maintains the security and integrity of the transmission channel by using encryption, authentication and message authentication codes. The protocol basically consists of the following components: Record Protocol, Handshake Protocol and Alert Protocol. The Handshake Protocol consists of two phases: server authentication and client authentication, with the second phase being optional. In the first phase, the server, in response to a client’s request, sends its certificate and its cipher preferences. The client then generates a master key, which it encrypts with the server’s public key, and transmits the encrypted master key to the server. The server recovers the master key and authenticates itself to the client by returning a message encrypted with the master key. Subsequent data is encrypted with keys derived from this master key. In the optional second phase, the server sends a challenge to the client. The client authenticates itself to the server by returning the client’s digital signature on the challenge, as well as its public-key certificate. A variety of cryptographic algorithms are supported by SSL. During the “handshaking” process, the RSA public-key crypto-system is used. After the exchange of keys, a number of ciphers are used. These include RC2, RC4, IDEA, DES, and triple-DES. The MD5 message-digest algorithm is also used. The public-key certificates follow the X.509 syntax [6].

SSL is available as a programming interface on top of, or augmenting, the sockets programming interface, which in turn is an interface to TCP. It is possible to see SSL as a layer between the application and TCP, and not as a replacement for TCP. SSL offers the following benefits:

- Eavesdroppers cannot read the data.
- Either side can verify the identity of the other side. This is accomplished by one side presenting a “Certificate” to the other.
- Data integrity is assured. Any change to a byte will invalidate the checksum on each SSL chunk.

SSL is useful for standard HL7 connections that use TCP today. A crucial piece of technology is the digital certificate. The digital certificate proves that you know a per-user secret key. Because this secret key is never transmitted, it is operationally easier to keep
the secret key secret. Thus, a recipient can place high reliance on the assumption that the holder of the certificate, is who the certificate says it is and the secret key is correct. It is difficult to forge certificates or steal meaningful private keys. Therefore, we allocate trust to the certificate authority. SSL appears to be a good choice to solve the problem of authentication and privacy between two sites using TCP. However, SSL is unsuitable for “store and forward” environments. Once the data is read off the wire, all knowledge/proof of its origin is lost. When message routers are involved, SSL authentication only has the ability to authenticate the last link. Without changes on the Hub, it is not possible to verify that the sender is who the HL7 message says it is. The message router would have to check the MSH segment against the SSL certificate.

The TLS protocol was developed based on the SSL protocol. The differences between TLS and SSL are not dramatic, but they are significant enough that they do not inter-operate.

The TLS protocol provides communications privacy over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery.

The primary goals of the TLS protocol are to provide privacy and data integrity between two communicating applications and:

- Cryptographic security: TLS should be used to establish a secure connection between two parties.
- Interoperability: Independent programmers should be able to develop applications utilising TLS, which will then be able to successfully exchange cryptographic parameters without knowledge of one another's code.
- Extensibility: TLS seeks to provide a framework into which new public key and bulk encryption methods can be incorporated as necessary. This will also accomplish two sub-goals: to prevent the need to create a new protocol, and risk the introduction of possible new weaknesses, and to avoid the need to implement an entire new security library.
- Relative efficiency: Cryptographic operations tend to be highly CPU intensive, particularly public key operations. For this reason, the TLS protocol has incorporated an optional session caching scheme to reduce the number of connections that need to be established from scratch. Additionally, care has been taken to reduce network activity.

The protocol is composed of two layers: the TLS Record Protocol and the TLS Handshake Protocol. At the lowest level, layered on top of some reliable transport protocol (e.g. TCP), is the TLS Record Protocol. It provides connection security that has two basic properties:

- The connection is private. Symmetric cryptography is used for data encryption (e.g. DES, RC4). The keys for this symmetric encryption are generated uniquely for each connection and are based on a secret negotiated by another protocol. The Record Protocol can also be used without encryption.
- The connection is reliable. Message transport includes a message integrity check using a keyed MAC. Secure hash functions (e.g. SHA, MD5) are used for MAC computations. The Record Protocol can operate without a MAC, but is generally only use in this mode while another protocol is using the Record Protocol as a transport for negotiating security parameters.
The TLS Record Protocol is used for encapsulation of various higher level protocols. One such encapsulated protocol, the TLS Handshake Protocol, allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data. The TLS Handshake Protocol provides connection security that has three basic properties:

- The peer's identity can be authenticated using asymmetric or public key cryptography (e.g. RSA, DSS, etc.). This authentication can be made optional, but is generally required for at least one of the peers.
- The negotiation of a shared secret is secure: the negotiated secret is unavailable to eavesdroppers, and, for any authenticated connection, the secret cannot be obtained, even by an attacker who can place himself in the middle of the connection.
- The negotiation is reliable: no attacker can modify the negotiation communication without being detected by the parties to the communication.

One advantage of TLS is that it is application protocol independent. Higher level protocols can layer on top of the TLS Protocol transparently. The TLS standard, however, does not specify how protocols add security with TLS. The decisions on how to initiate TLS handshaking and how to interpret the authentication certificates exchanged are left up to the judgement of the designers and implementers of protocols which run on top of TLS [7].

8.6.3.3 HL7 Interfaces and Lower Layer Protocols

HL7 interfaces exist at the application layer of the OSI model and are usually located at the application, but sometimes also at the communication server. Thus, they usually require the support of some lower level protocol (LLP) that provides an interface between HL7 and the network. It is important to select an LLP that meets the needs of the interface and fits into the overall telecommunications strategy and architecture. The LLPs are built up assembling various modules (e.g. initiating module, accepting module, encoding rules module, communication module). The communication module delivers a message from the source to its destination using the network socket interface. If this delivery mechanism is used for message transmission instead of external communication protocols, it MUST be secured by the security services defined to gain communication security.

In Appendix C, the “HL7 Implementation Support Guide for HL7 Version 2.3” (final version 6/98) defines lower layer protocols usable for unsecured communication of HL7 messages. In this service, the minimal lower layer protocol (MLLP), the hybrid lower layer protocol (HLLP), and an ANSI X3.28 based data link protocol are specified. Moreover, the HL7 sequence number protocol is explained and pseudo code for a TCP-based implementation of LLPs is given. More information about the LLPs of HL7 can be found in the “HL7 Implementation Support Guide” - especially section 1.5, 2.5, 3.4, and 3.6 as well as Appendix C.

8.6.4 Communication Protocol Security Requirements

For secure end-to-end communication between two principals in an HL7 environment, the following security services are needed:
Chapter 8

- principal authentication (applications and systems)
- data origin authentication
- confidentiality
- integrity
- non-repudiation of origin
- non-repudiation of receipt

In general, communication protocols distinguish control data from message data. Control data is used to emit protocol specific commands from the sender to the receiving principal and to reply codes back to the originator for status report. More structured and differentiated protocols, such as FTP, separate these kinds of data and use two connections, called control connection and data connection, simultaneously.

8.6.4.1 Control Data

This Standard Guide REQUIRES the control data to be secured by integrity. Additionally, the control data SHOULD be protected by the security services non-repudiation of origin and non-repudiation of receipt. Since the control data is a well-known and often small set of commands, confidentiality SHOULD NOT be applied. Furthermore, the control data MUST be armed against loss of data bits in environments not capable of full binary transport. This may be achieved by various conversion techniques such as Base64-Encoding or Quote-Printable-Encoding. Finally, before applying the integrity service, the data MUST be canonicalised after the encoding process to prevent system dependency (like different end of line codes for PC and Unix environments; system independence) leading to invalidation of the integrity code (digital signature or MAC).

8.6.4.2 Message Data

The data connection delivering the message data MUST be secured by the integrity service. Confidentiality, non-repudiation of origin, and non-repudiation of receipt SHOULD be applied as well to gain more security. Switching between different operation modes such as plain text, encrypted-only, signed-only, or signed-and-encrypted, SHOULD be possible according to the security policy given (operation modes independence). Moreover, the message data MUST be character converted and canonicalised to prevent loss or manipulation of certain EDI characters (as the HL7 segment terminator) leading to invalidation of the integrity code (digital signature or MAC).

For the correct handling of received data concerning the operation modes, content encoding, and other parameters, insertion of the message data into a cryptographic syntax capable of wrapping and feature negotiation is REQUIRED. For this purpose, an encapsulation scheme using MIME entities consisting of headers and bodies MAY be used (e.g. PGP/MIME, S/MIME, or MOSS). The recipient MUST be able to recognise the data received as an EDI interchange.

When transporting signed data by Internet (HTTP, SMTP) or end-to-end in non-MIME environments, gateways are generally not aware of security encapsulation schemes and, therefore, mistreating the data or even applying conversions to the structure and its contents according to the local format. Thus, either the original message could not be reconstructed and the signature could not be verified, or the signature verification fails.
Additional measures SHOULD be applied to avoid this behaviour (e.g. using a special wrapping mechanism as defined by S/MIME).

To fulfil these requirements, data other than HL7 messages can also be sent securely over the data connection (data-type independence), i.e., other EDI messages (e.g. X12, xDT\textsuperscript{5}) or non-EDI data (arbitrary binary) can be wrapped and transmitted, too. Thus, the secure communication protocol can be used in any desired environment for data delivery. Data-type independence means that the receiving principal MUST be able to recognise the type of data received. For that reason, if inserting an HL7 message into an encapsulation scheme, header information identifying the message content MUST be supplied (like content-type for MIME). This scheme SHOULD be capable of specifying additional parameters to state encoding rules (syntax) or other information (e.g. version number). Another possible solution is to map the HL7 message - including the additional protocol parameters - into a ANSI X12 message, using the standardised mapping rules, and to insert the result into an encapsulation scheme as defined in RFC1767 (MIME encapsulation of X12 and EDIFACT objects using the content-types application/EDI-X12 and application/EDIFACT). When operating in an HL7-environment, data type independence SHOULD NOT be attended to, since the HL7 interface definitely knows that only HL7 message data is sent between applications.

For large file processing, compression of EDI messages MAY be done before encryption, after applying the integrity service if needed. Applying compression before encryption strengthens cryptographic security since repetitious strings are reduced due to their redundancy. If compression is used, additional data MUST be provided to convey compression information.

8.6.4.3 Authentication

Before any control or message data is exchanged, except the control command to request authentication, principal identification and authentication of applications or systems MUST be performed as described in Section 8.6.5. Again, for interoperability reasons, the authentication tokens exchanged MUST be character converted and canonicalised.

In the context of some EDI specific protocols or user-applications interactions based on standardised EDI protocols for enabling open communication independent of application platform and environment, a human user may also occur as a principal instance. Then, other security services, mechanisms and techniques may be used such as the European security infrastructure based on electronic identity cards (smartcards, Health Professional Cards = HPC) and corresponding TTP services.

8.6.4.4 Cryptographic Algorithms

If the cryptographic algorithm to be applied is not an algorithm approved by any national authority or other community authorities, then it SHOULD be an algorithm registered and identified using the procedures described in ISO/IEC9979. The communication protocol used SHOULD be independent of the underlying cryptographic mechanisms (cryptographic mechanism independence) and cryptographic message syntax (cryptographic message syntax independence). It SHOULD allow the negotiation of different algorithms, operation modes, and cryptographic message syntax as well as the

\textsuperscript{5} The German EDI protocol for communications between doctors' office systems as well as between these systems and those of other Healthcare providers.
selection of different technical means, such as smart card, biometric device, directory server, and CRL server.

8.6.4.5 Communication and Networking

Basically, communication of EDI messages SHOULD be carried out by direct link connections. Solutions are protocols based on store-and-forward delivery (asynchronous as SMTP, MHS) or real-time delivery (synchronous as FTP, FTAM). Since the delivery of EDI or cryptographic enhanced data objects are independent of the communication protocol, there are many different protocols and options used for such solutions.

The HL7 Standard is defined in terms of the client/server model (remote operation) and is therefore applicable to file transfers (batch processing). One or more messages MAY be encoded according to the Encoding Rules, grouped in a file and similarly transmitted. General mechanisms for the batch transmittal of HL7 messages are provided in the “HL7 Implementation Support Guide” (Section 1.5) and in the “HL7 Standard” (Section 2.23.3). Following the HL7 paradigm of bi-directional, synchronous communication where applications rendezvous and exchange their messages, security protocols capable of synchronous message transfer featuring the given security services above MAY be preferred. Transfer of the EDI interchange can take place in real time, without any deferred delivery. The data transmission is point-to-point, with no requirement for temporary storage anywhere. The size of interchanges can be very large and the exchange could be initiated by both the sender and the recipient (e.g. for data collection). Modern communication servers support the variety of protocols.

For reliable transmissions, (connection-oriented) TCP/IP based networks are REQUIRED.

8.6.4.6 Protocol Model Implementation

Among the security considerations described, the specification of the protocols used - like FTP, TCP, and IP - contains a number of mechanisms inherent to their protocol model that can be used to compromise network security. Thus, there are many so-called Internet attacks based on infrastructure weakness; for instance DNS spoofing, ICMP bombing, source routing (IP spoofing), TCP sequence guessing/hijacking, TCP splicing, FTP bouncing, racing authentication and denial of service.

Attacks arising from the weakness of the process protocol and the underlying protocols SHOULD be addressed by appropriate countermeasures in the implementation model. For example, the Computer Emergency Response Team (CERT Co-ordination Centre [1]) studies such Internet security vulnerabilities, provides incident response services, publishes a variety of incident reports and security alerts, and develops information to improve security.

Racing authentication, which is based on faster authentication of the attacker than the victim, SHOULD be prevented by strong authentication based on challenge-response protocols. Moreover, a restriction to one simultaneous login of the same principal and to the total number of control connections possible at once SHOULD be carried out.

To protect against FTP bouncing, which is namely the misuse of the PORT command where the attacker is acting as a server, the server SHOULD NOT establish connections to arbitrary machines (for instance to a second FTP server called proxy FTP) and to ports on these machines. The server SHOULD ensure that the received IP address, which specified
in the PORT command, must match the client's source IP address for the control
connection. To prevent this attack from occurring at all, FTP driven protocols SHOULD
use the PASV command instead of the PORT command to establish data connections.

Furthermore, the server SHOULD disallow data connections on TCP-ports that are
well-known ports (port 0 to 1023) or registered ports (1024 to 49151). Only dynamic,
private ports (port 49152 to 65535) SHOULD be allowed.

Hence, a port scan against another site hiding the true source and bypassing access
controls like firewalls (for instance bouncing to a well-known port) cannot be performed.

Random local port (private-) numbers SHOULD be used for the data connection to
address port number guessing. Guessing the next port number is much easier when using
simple increasing algorithms (for example: next port = old port + constant number)
enabling attacks like the denial of a data connection or hijacking a data connection to steal
files or insert forged files.

TCP splicing, which is the hijacking of the connection on the TCP layer, MAY be
prevented by the application of level end-to-end confidentiality since the attacker cannot
generate messages that will decrypt to meaningful data. When confidentiality is applied,
network sniffing does not pay, but from the TCP layer downwards creating a traffic flow
analysis evaluating packet headers and trailers is still possible. Traffic flow confidentiality
(e.g. address-hiding or traffic padding) MAY be provided by applying confidentiality on
the data link layer. In the context of HL7 EDI, this threat may be overestimated and could
be neglected.

In addition to the authentication procedures given in Section 8.6.5, restrictions based on
network addresses MAY be provided. The server accepts only connection requests from
pre-defined IP addresses within authorised organisations and confirms that this address
matches on both the control connection and the data connection. Authentication MUST
NOT rely on IP address authentication only. Relying solely on an IP address authentication
makes an attack like source routing of IP packets (IP spoofing) is possible.

To address DNS spoofing, hostname to IP address resolution or vice versa (DNS)
SHOULD NOT be performed. It is RECOMMENDED that the destination machine be
caught by the IP address directly.

For the detection of compromises like denial of service attacks and other attacks, the
server SHOULD keep reports and log all activities, including connection attempts,
disconnection, command executions and others. Reports and logs SHOULD be integrity
and confidentiality protected.

8.6.5 Authentication Service Requirements

8.6.5.1 Purpose

The authentication service provides assurance of the identity of a principal. When a
principal claims to have a particular identity, the authentication service will provide a
means of confirming that this claim is correct. Generally, a principal to be authenticated
proves its identity by showing its knowledge of a secret. The mechanisms are defined as
exchanges of data between principals, and, where required, with a trusted third party
(TTP).

Authentication is the most important of the security services; all other services depend
upon this assurance. The identity authenticated is used for accountability, data origin
authentication, and access control depending on the assured knowledge of identities.
8.6.5.2 Service Requirements

This service MUST use cryptographic techniques that establish strong authentication; weak authentication (e.g. passwords) SHALL NOT be used. The authentication framework given by ISO/IEC10181-2 is on top of a hierarchy of authentication standards that provide concepts, nomenclature, and a classification for authentication models. Directly below, standards such as ISO/IEC9798 provide a particular set of these methods in more detail. Finally, at the bottom, standards such as ISO/IEC9594-8 (ITU-T Recommendation X.509) use these concepts and methods in the context of a specific application or requirement (ISO/IEC9594-8 was initially developed for use with the directory service). NIST FIPS PUB196 is based on ISO/IEC9798-3 and might be helpful for implementation details like token formatting. ISO/IEC9798 specifies different protocols that address both unilateral and mutual authentication by mechanisms using:

- symmetric encipherment algorithms (ISO/IEC9798-2),
- digital signature techniques (ISO/IEC9798-3, ISO/IEC9594-8, NIST FIPS PUB196),
- a cryptographic check function (ISO/IEC9798-4), or
- asymmetric zero knowledge techniques (ISO/IEC9798-5).

For a higher level of assurance, mutual authentication carried out by challenge-response protocols, using symmetric or asymmetric security techniques, is RECOMMENDED. In large networks, mechanisms that use symmetric techniques, such as Kerberos, depend upon trusted on-line authentication servers to distribute public key certificates and CRLs. Mechanisms that use asymmetric techniques require off-line servers - which need not to be trusted - for the distribution of public key certificates and CRLs. Due to the inherent and well-known disadvantages of symmetric techniques (secret key cryptography), asymmetric techniques, such as digital signature or zero knowledge techniques, SHOULD be used for open networks like the Internet, where many principals communicate with each other. The application of this technique requires the management of security certificates (e.g. using a directory server) inside a public key infrastructure (PKI) that has to be established. The authentication depends on the successful verification of the digital signature, which is bound on the key pair. Thus, there is a requirement to validate the public key used to verify a claimed identity. The commonly used mechanism for validating a public key is the use of certificates issued by a trusted certification authority (CA). This technique is described in ISO/IEC9594-8 (ITU-T Recommendation X.509).

The authentication service MUST be built up using the services of principal authentication and data origin authentication. This implies that the integrity service MUST be applied. Assurance of origin and receipt (non-repudiation of origin, non-repudiation of receipt) SHOULD be used for a higher level of security. Confidentiality of the authentication tokens is NOT RECOMMENDED in order to avoid unnecessary interactions between security mechanisms, which may result in security flaws.

The principal authentication service assures that a principal, which has a specific communication relationship with the verifier, is the one claimed. Principal authentication techniques generally involve an exchange of cryptographic protected authentication data, which is used to validate a claimed identity.

The data origin authentication service assures that a principal is the source of data as claimed. This is provided by use of digital signature (asymmetrical techniques) or encryption (symmetric techniques). Applying digital signatures, the principal signing the data cannot deny that he or she applied the signature, since the principal is the only one
with knowledge of his or her private key. This method also achieves non-repudiation of origin. Applying encryption, the principal to be authenticated corroborates its identity by demonstrating knowledge of a secret authentication key.

An important factor in authentication exchange techniques is the need to protect against replay of authentication using time variant parameters (unique numbers) as time stamps, sequence numbers, and random numbers. Assurance of continuity of authentication MUST be provided for the exchanges of data in order to prevent cutting in or taking over after authentication has completed. This can be achieved by carrying out the integrity service over the whole connection period (binding of principal authentication and integrity service), and by performing further authentication exchanges from time to time.

Authentication protocol design needs to take into account an array of possible attack scenarios and provide appropriate countermeasures. To cover all the services required for authentication (principal authentication, data origin authentication, non-repudiation of origin and non-repudiation receipt) and all possible threats as described in ISO/IEC9798 and ISO/IEC10181-2, each authentication token exchanged MUST be completely integrity protected and SHOULD consist of:

- a token identifier
- a sequence number
- the IP address of sender and receiver
- the network hardware adapter address (MAC) of sender and receiver
- the DNs of sender and receiver
- a state indicator (authentication request or invitation)
- the role of the principal sending the token (initiator, responder)
- time stamps stating token generation and token expiration (in UTC time)
- a random number challenge
- the MIC of the last authentication token received

The implementation of the authentication service SHOULD give a consistent view on token parameters (the parameters must be distinguishable independent of their position inside the token) and on the token order in order to prevent security flaws. For both, additional information SHOULD be included in the tokens—gaining more efficiency due to unambiguosness, and eliminating security threats. For a consistent view on parameters, the tag-length-value encoding SHOULD be used, and for a consistent view on the token order, sequence numbers and token identifiers SHOULD be applied.

As mentioned before, in the context of some EDI specific protocols or user-applications interactions based on standardised EDI protocols for enabling open communication independent of platform and environment of the application, a human user may also occur as a principal instance. Before trustworthiness between principals can be established by authentication, the human user MUST authenticate himself to a cryptographic module of the local end system (human user authentication), which then acts as an initiator (client) and performs the integrity service on behalf of the human user toward the responder (server) carrying out system authentication. Human user authentication relies on principles of something known (e.g. passwords), something possessed (e.g. chip cards following ISO/IEC7816), characteristics of the individual human user (biometrics), and accepts that an identified TTP has established the human user’s identity, or context (e.g. source address). This Standard Guide RECOMMENDS the usage of chip cards (smart card with cryptographic processor) in combination with biometrics (e.g. fingerprint) and/or PIN
codes. For the authentication services described above, the user must keep and protect his
private asymmetric key (or a secret symmetric key). This SHOULD be done using chip
cards, protected with a PIN code and/or biometrics for reasons of security (the keys are
protected in a physical device carried by the user, which cannot be copied and from which
a readout cannot be displayed) and mobility (the user can authenticate himself in any
environment that has proper devices). Mobility is an essential argument for using chip
cards in the Healthcare sector for health professionals.

8.6.6 Confidentiality Service Requirements

8.6.6.1 Purpose

The confidentiality service protects against information being disclosed or revealed to
principals not authorised to read and interpret message data obtaining the information. This
service does not prevent against the reading of the protected data.

Concerning the granularity of the confidentiality with respect to communication
security, this service applies to all the message data transmitted on the connection. This
technique is called wrapping or enveloping of data. Selective field confidentiality, which
applies only to designated data fields within a data unit is not a matter of communication
security but of application security.

8.6.6.2 Service Requirements

A general framework for provision of confidentiality services is given by
ISO/IEC10181-5. This Standard Guide defines basic concepts of confidentiality, identifies
classes of generic mechanisms and describes confidentiality policies. It neither specifies
nor depends on the use of particular mechanisms and algorithms since ISO does not
standardise cryptographic algorithms, but rather their procedures for registration (see
section 8.6.4.4).

Confidentiality SHOULD only be served to the message data delivered by the data
connection (see section 8.6.4). It MUST NOT be applied on the authentication or control
data. Confidentiality SHOULD be provided by the use of strong cryptographic
mechanisms employing hybrid techniques. For bulk encryption (content encryption), a
strong symmetric session key having at least 112 significant key bits (as IDEA or DES3)
SHOULD be used and for each transfer of message data another key SHOULD be applied.
The session key is protected by asymmetric encryption techniques using at least 1024 key
bits (as RSA, ElGamal or Elliptic Curves = EC, key encryption/transport). Content
encryption by asymmetric techniques MUST NOT be applied. Modes of operation of
symmetric keys SHOULD follow ISO/IEC8372 for 64-bit block ciphers or ISO/IEC10116
for n-bit block ciphers.

The combination of the services of confidentiality and integrity is RECOMMENDED
for the transport of message data. The interaction between these security mechanisms and
their ordering may result in security weakness (ISO/IEC10181-1 chapter 10). In general,
the integrity service MUST be applied first. If three services are desired (triple wrapping),
two integrity services MUST be applied; one before and one after the confidentiality
service. Following this approach, in message handling systems, two different message
integrity codes can be placed on the data; one computed on the encrypted data (applied by
transfer agents) to provide chained non-repudiation, and one on the plain text (applied by
the first sender) to provide data origin authentication.
8.6.7  **Integrity Service Requirements**

8.6.7.1  **Purpose**

Data integrity service *ensures* data consistency while in communication by giving the possibility to *detect* its modification. Changing the value of data includes insertion, deletion, modification, or reordering parts of the data.

This service *does not prevent* the manipulation of data but *allows the detection* of its alteration. Duplication that may be a result of a replay attack *can be neither inhibited nor* recognised without additional, unique tokens as random numbers or time stamps.

Concerning the granularity of the integrity with respect to communication security, this service *applies to all the data transmitted* on the connection. Selective field integrity, which applies only to particular data fields within a data unit is *not a matter* of communication security but of application security. However, integrity and accountability dealing with items could be required for exchange.

8.6.7.2  **Service Requirements**

A general framework for the provision of integrity services is given by ISO/IEC10181-6. This Standard Guide defines basic concepts of integrity, identifies classes of generic mechanisms and describes integrity policies. It neither specifies nor depends on the use of particular mechanisms and algorithms since ISO does not standardise cryptographic algorithms, but rather their procedures for registration (see section 8.6.4.4).

It is *REQUIRED* that integrity be applied to the authentication tokens exchanged, the message data transferred (possibly before its confidentiality protected), and the control data transmitted.

Integrity MUST be achieved by the application of cryptographic techniques. In general, a cryptographic check value (i.e. a message digest computed by a hash function) MUST be calculated over the data in order to be integrity protected. Then, this check value has to be shielded by transformation through an encipherment mechanism (sealing, symmetric cryptography) or by combination with the private key to form a digital signature (asymmetric cryptography). The usage of digital signatures for the provision of integrity by asymmetric techniques is RECOMMENDED.

For calculation of the message digest, keyed (message authentication code, MAC) or unkeyed (called modification detection, MDC) hash function can be used. In general, hash function using MDCs SHOULD follow ISO/IEC10118 and those using MACs SHOULD follow ISO/IEC9797. Unkeyed hash function are based upon block ciphers (as MDCx-DES), the MD4-family (as MD5, RIPEMD-x), or modular arithmetic (as MASH-x), whereas keyed hash functions rely on block ciphers (as MAC-DES-CBC, MAC-IDEA-CBC), hash MACs (as HMAC-MD5, HMAC-RIPE-x, HMAC-SHA-1), or steam ciphers (as CRC-based MAC). The difference between these two types lies in the application of keys. Keyed mechanisms require a secret key as input for the MAC algorithm to calculate a MAC, whereas unkeyed mechanisms apply the key (secret or public) on the MDC that has been previously calculated by the MDC algorithm.

For integrity provision based on asymmetric techniques (i.e. digital signatures) the usage of MDCs based on the MD4-family is RECOMMENDED since these functions are specifically designed for the explicit purpose of hashing, with optimised performance (dedicated hash functions).
When symmetric techniques are applied, the concatenation of the text and the appended message integrity code (either MAC or MDC) has to be sealed (encrypted). The usage of MACs offers the advantage that, should the encryption algorithm be defeated, the MAC still provides integrity. A drawback is the requirement of managing both an encryption key and a MAC key, which may lead to security weakness by unwanted algorithm dependencies. Nevertheless, it should be noted that the MDC is a known function of the plain text, while a MAC is an authenticator secret.

The cryptographic check function, the signing algorithms (as RSA, DSA, ElGamal, or EC) as well as the authentication algorithms (as IDEA or DES) SHOULD offer an appropriate strength.


Integrity based upon digital signatures with an appendix (the MDC or MAC is appended to the processed data) SHOULD follow ISO/IEC14888. For digital signature schemes giving message recovery, ISO/IEC9796 SHOULD be obeyed. Applying integrity using digital signatures with an appendix, data origin authentication and non-repudiation of origin and receipt can be provided.

Duplication that may result in replay attacks SHOULD be countered by including additional time variant parameters (TVPs), such as random numbers or time stamps in the signature process.

The combination of the services confidentiality and integrity is RECOMMENDED for the transport of message data. Interaction between these security services and their ordering may result in security weaknesses (ISO/IEC 10181-1 chapter 10). In general, the integrity service MUST be applied first. If three services are desired (triple wrapping), two integrity services MUST be applied; one before and one after the confidentiality service. Following this approach in message handling systems, two different message integrity codes can be placed on the data; one computed on the encrypted data (applied by transfer agents) to provide chained non-repudiation, and one on the plain text (applied by the first sender) to provide data origin authentication.

8.6.8 Data Origin Authentication Service Requirements

8.6.8.1 Purpose

Data origin authentication (message authentication) is used to authenticate the real source of data, in that it provides assurance of the source of data by gluing the originator's identity along with the data using the integrity service. It does not provide protection against duplication, reordering, or loss of data. In contrast to non-repudiation, this service is initiated by the data originator that wants to give proof of source against the recipient.

Concerning the granularity of the data origin authentication with respect to communication security; this service applies to all the data transmitted on the connection. Selective field origin authentication, which applies only to designated data fields within a data unit is not a matter of communication security but of application security.
8.6.8.2 Services Requirements

It is REQUIRED that data origin authentication be applied on the authentication tokens exchanged, the message data transferred (before its confidentiality protection), and the control data transmitted.

The data origin can be authenticated by applying digital signature algorithms, MACs, or sealed authenticators. In contrast to MDCs, MACs are, themselves, secret authenticators. The appended authenticator, MAC, is used along with encryption. See section 8.6.7.2)

Data origin authentication SHOULD be provided by using digital signatures with an appendix. The source of data is assured by signing the concatenation of data and the originator’s DN. Then, the message digest, the DN, and the data are transferred to the recipient.

Data origin authentication based on shared secret keys (as MACs) does not allow a distinction to be made between the parties sharing the key, and thus – in contrast to digital signatures – does not provide non-repudiation of origin. If a resolution is required, either an on-line TTP as a notary authority, or asymmetric techniques may be used. For further requirements see section 8.6.7.

8.6.9 Non-repudiation Service Requirements

8.6.9.1 Purpose

The property ensures that the actions of an entity may be traced uniquely to the entity. All relevant information about actions performed by users or processes acting on their behalf are recorded so that the consequences of those actions can later be linked to the user in question, and the user held accountable for his actions [4]. In the sense of the shared-care paradigm having inter-institutional communication and co-operation, this service is most important for accountability because it provides legal evidence of the responsibility of each principal involved.

In general, the intended usage of non-repudiation is to ensure availability of irrefutable evidence for resolution of any dispute about occurrence or non-occurrence of some event or action, so that a principal cannot falsely deny being responsible. Evidence establishes accountability regarding a particular event or action. This service does not prevent principals to attempt repudiation.

8.6.9.2 Service Requirements

A general framework for provision of non-repudiation is given by ISO/IEC10181-4. This Standard Guide defines basic concepts of non-repudiation, identifies classes of generic mechanisms, and describes non-repudiation policies. It neither specifies nor depends on the use of particular mechanisms and algorithms since ISO does not standardise cryptographic algorithms, but rather their procedures for registration (see section 8.6.4.4).

Non-repudiation service can be separated into non-repudiation of origin (NRO), non-repudiation of submission (NRS), non-repudiation of transport (NRT), and non-repudiation of delivery (NRD). NRO is a combination of non-repudiation of creation and non-repudiation of sending; NRD must be seen as concatenation of non-repudiation of receipt (NRR) and non-repudiation of knowledge. Concerning communication security,
only NRO, NRS, NRT, and NRR have to be addressed, all other kinds of non-repudiation are not matters of communication security but of application security. If delivery authorities (DA) are involved (i.e. operating in store-and-forward systems (e.g. using SMTP)), the support of NRO, NRS, NRT, and NRR is RECOMMENDED. Otherwise, if no DAs are present (e.g. synchronous transfer using FTP), NRO and NRR SHOULD be provided. If NRR has been successfully proven for the latter scenario, NRS and NRT have also been assured. NRR is initiated by the data originator that wants to have proof of reception against the recipient, and NRO is initiated by the intended recipient that wants to give proof of source against the originator. NRS and NRT are used by the originator to protect against the DAs. It is RECOMMENDED that the kinds of non-repudiation mentioned for the authentication tokens exchanged — the message data transferred (before its confidentiality protection), and the control data transmitted — be used.

Non-repudiation mechanisms that provide evidence MUST be based upon cryptographic techniques using symmetric or asymmetric techniques as described by ISO/IEC13888-2 or ISO/IEC13888-3, respectively. It is REQUIRED that the generalities of ISO/IEC13888-1 be followed. The application of asymmetric techniques, using digital signatures, is RECOMMENDED and REQUIRES the involvement of an offline TTP to guarantee the genuineness of keys (public key certificates management including CRLs and directory servers). Symmetric techniques, using secure envelopes, MAY be applied instead and REQUIRE an on-line TTP for generation and validation of the secure envelopes including resolution of origin preventing fraudulent repudiation. Mechanisms using shared secret keys do not allow a distinction to be made between the parties sharing the key, and thus — in contrast to digital signatures — do not provide NRO. The mechanisms have to provide protocols for the exchange of non-repudiation tokens specific to each kind of non-repudiation. These tokens MAY be stored as information by disputing parties for arbitration.

The non-repudiation service involves the generation, verification and recording of evidence, and the subsequent retrieval and re-verification of this evidence in order to resolve disputes. For evidence generation, the TTP MAY act on behalf of a principal involved as a token generation authority (TGA), digital signature generating authority (DSSA), time stamping authority (TSA), notary authority (NA), or monitoring authority (MA). Evidence transfer MAY be carried out by a TTP acting as a delivery authority (DA) or evidence record-keeping authority (ERA). At last, the TTP MAY be in the role of an evidence verification authority (EVA). As NA the TTP SHOULD arbitrate disputes by providing evidence about the properties of the principals involved and of the data stored or communicated using a generic notarisation token (NT) as defined in ISO/IEC13888-1. In providing an evidence recording service, the TTP SHOULD keep records of operations so that they will be available for the resolution of any disputes that may arise at some time in the future. When a trusted time is required and when the clock provided by the token generating party cannot be trusted, it is necessary to rely on a TTP. As a TSA, the TTP SHOULD provide a time stamping service by generating a generic time stamping token (TST) as specified in ISO/IEC13888-1. The non-repudiation tokens NROT, NRDT, NRST, and NRTT are all derived from a generic non-repudiation token (GNRT) given by ISO/IEC13888-1. For provision of NRR, the NRRT SHOULD be also derived from the GNRT. An overview of the non-repudiation tokens and their usage is given in Figure 25 below (after ISO/IEC13888-1 and ISO/IEC13888-3).

In general, the token data (TD) of GRNT, TST, and NT SHOULD consist of DNs, a service indication flag, time stamps, and the imprint of the message data m, which can either be the hash code of m or m itself. Return of content MAY be wasteful of network bandwidth and time. Thus, it is RECOMMENDED that only the hash code, and not the
whole message, be returned. To be more specific, the GRNT contains the DNs of the message originator, of the message recipient, and of any other authority (as TGA, DA, TSA, and MA) involved. Two time stamps are included stating the date and time when the evidence token was generated, and when the message data was processed (e.g. sent, received, submitted, delivered).

When using symmetric cryptographic techniques, an on-line TTP is REQUIRED that takes over the roles of TGA, TSA, NA, and EVA. Additionally, it MAY act as MA, ERA or DA (in-line). The non-repudiation tokens (NRxT) used consist of a cryptographic check value (e.g. MAC) computed on TD by symmetric integrity techniques and the TD itself. Any principal holding that secret key can verify the integrity and origin of TD. For the purpose of generating and verifying evidence, the envelope is constructed and verified by a TTP using a secret key known only to the TTP. The secure envelope MAY also be used for the origin and integrity protected communication between a TTP and any other principal. In that case the envelope is generated and verified with a key known by both the principal and the TTP. For additional assurance, the TSA and TGA SHOULD be different authorities.

For asymmetric mechanisms, an off-line TTP is REQUIRED as TSA and NA. Additionally, it MAY act as MA, ERA or DA (in-line). The Principal that wants to obtain evidence (service requester) MUST generate and verify the evidence on its own. The non-repudiation tokens used consist of a digital signature computed on TD using a private
signing key and the TD itself. Any principal having access to the corresponding public key is able to verify the integrity and origin of TD. A chain of public key certificates or identities MAY have to be verified to obtain the necessary assurance. If confidentiality is needed, the result itself (non-repudiation token) MAY be enveloped using the confidentiality service.

To prevent an endless chain of non-repudiation tokens (i.e. giving NRR for the NRRT received) only two transactions SHOULD be carried out between client and server as follows. The client sends an HL7 request message that includes a request for a signed receipt and the server responds by transmitting the HL7 reply message including the receipt, cryptographic enhanced as described above. The client abandons sending a receipt for the server's response in turn.

8.6.9.3 Non-repudiation of Origin

The NRO service provides the recipient of data with proof that protects against any attempt by the sender to falsely deny sending the data. The evidence (non-repudiation of origin token, NROT) is generated by the originator of the message and sent to the intended recipient. The originator sends both the message and the NROT to the recipient. To provide proof, the identities and the integrity of data must be confirmed, and the time stamps must be within the given time window.

8.6.9.4 Non-repudiation of Receipt

The NRR service provides the sender of data with proof that protects against any attempt by the recipient to falsely deny having received the data. The evidence (non-repudiation of receipt token, NRRT) is generated by the recipient of the message and sent to the originator. The recipient sends both the reply message (if any) and the NRRT to the originator. To provide proof, the identities and the integrity of data must be confirmed, and the time stamps must be within the given time window.

8.6.9.5 Non-repudiation of Submission

The NRS service provides the sender of data, which may be another DA, with proof that protects against any attempt by the DA to falsely deny having accepted the data for transmission. The DA does not care what the content of the message is. The originator, or a preceding DA has sent a message to the next DA that receives this message and sends the NRS token to the originator or the preceding DA — establishing a chain of intermediate NRST tokens providing chained NRS.

To provide proof, the identities and the integrity of data must be confirmed, and the time stamps must be within the given time window.

8.6.9.6 Non-repudiation of Transport

The NRT service provides the sender of data with proof that protects against any attempt by the DA to falsely deny having delivered the data to the intended recipient. The DA does not care what the content of the message is and cannot guarantee that the message is duly received by the recipient. The evidence, in the form of a non-repudiation of transport token or NRTT, is generated by the DA that is delivering the message to the intended recipient (the last DA in the chain of DAs) and sent back to the originator. To
provide proof, the identities and the integrity of data must be confirmed, and the time stamps must be within the given time window.

8.7 Merging Secured Data Elements with EDI Messages

Communication security includes the assembling and merging of already secured data elements to complete EDI messages. The protection of data elements (i.e. application of security services, and their type of storage (e.g. using a database)) is part of application security (data element security, data base security) and is not considered here.

The merging process MUST build a complete EDI message gathering certain secured data elements by retrieving them from the storage device (e.g. database). The way in which the message is constructed (structure) and what element delimiters are used determines the kind of the EDI message generated. Possible kinds of EDI message are HL7, EDIFACT, xDT and others.

If the data element is integrity protected, the integrity code (digital signature or MAC) MUST be character converted (e.g. using Base64-encoding) and canonicalised to prevent loss of characters resulting in invalidation of integrity. Furthermore, the DN is REQUIRED for origin authentication. It MUST be possible to separate these data fields (i.e. original data element, integrity code as well as DN) from each other for further processing (e.g. integrity verification, data base storage). This can be achieved by introducing special delimiters. The approach of using only one new delimiter requires some intelligence of the EDI interface, namely counting the delimiter, to recognise that the first entry separated by the delimiter contains the DN and the second entry separated by the same delimiter contains the integrity code. For that reason, different delimiters may be more applicable — limiting necessary interface modifications for parsing and evaluation.

When the data element has confidentiality protection, it MUST be character converted and canonicalised as well to provide interoperability.

For insertion of protected data fields into EDI messages and to distinguish the different levels of protection - plain, signed-only, encrypted-only and signed-and-encrypted - proper means of identification MUST be provided. Again, appropriate delimiters SHOULD be applied.

The protected data items have to be transmitted (forwarded, processed) by EDI applications together with the original data element as part of the message, regardless of any communication security mechanisms that could be used as envelope.

Construction of EDIFACT messages is specified by ISO9735. Security as specified by ISO9735-5 (authenticity, integrity, and NRO), ISO9735-6 (message authentication and acknowledgement AUTACK, i.e. providing NRO and NRR), ISO9735-7 (confidentiality), ISO9735-9 (key management), and ISO9735-10 (security for interactive EDIFACT) is aimed at EDIFACT structures (segment levels). Data element security for HL7 is currently under construction. Until now, there is no security standard for xDT structured data available.
8.8 References and Bibliography (Informative)

1 CERT Co-ordination Centre (Computer Emergency Response Team): Security Improvement, Tech Tips, Alerts and Advisories;
Http://www.cert.org/nav/securityimprovement.html,
ftp://ftp.cert.org/pub/tech_tips/FTP_PORT_attacks,
http://www.cert.org/advisories/CA-97.27.FTP_bounce.html,
http://www.cert.org/advisories/CA-95.01.IP spoofing.attacks.and.hijacked.terminal.connections.html).


Security Architecture for the Internet Protocol (draft-ietf-ipsec-arch-sec),
http://www.ietf.org/html.charters/ipsec-charter.html


http://www.chips.ibm.com/techlib/micronews/vol3_nol/kellow.htm