Message authentication and Hash Functions

Prof. (Dr.) K.R. Chowdhary
Email: kr.chowdhary@iitj.ac.in

Campus Director,
JIET College of Engineering, Jodhpur

Tuesday 10th October, 2017

Securing the commercial Internet

- **Integrity.** The original Internet was designed for research, not as a commercial environment.
  - As the Internet grew, the community expanded, and the existing security framework was found lacking.
  - More sophisticated attacks can take advantage of basic flaws in the Internet's infrastructure. For example, the TCP/IP protocol suite used by all computers connected to the Internet is fundamentally lacking in security services.
- In the commercial world, these problems manifest themselves in a number of ways:
  - **Eavesdropping.** Eavesdropping attacks on a network can result in the theft of account information,
  - **Password "sniffing."**
  - **Data modification.** Data modification attacks can be used to modify the contents of certain transactions
  - **Spoofing.** Spoofing attacks can be used to enable one party to masquerade as another party.
  - **Repudiation.** Repudiation of transactions can cause major problems with billing systems and transaction processing agreements.

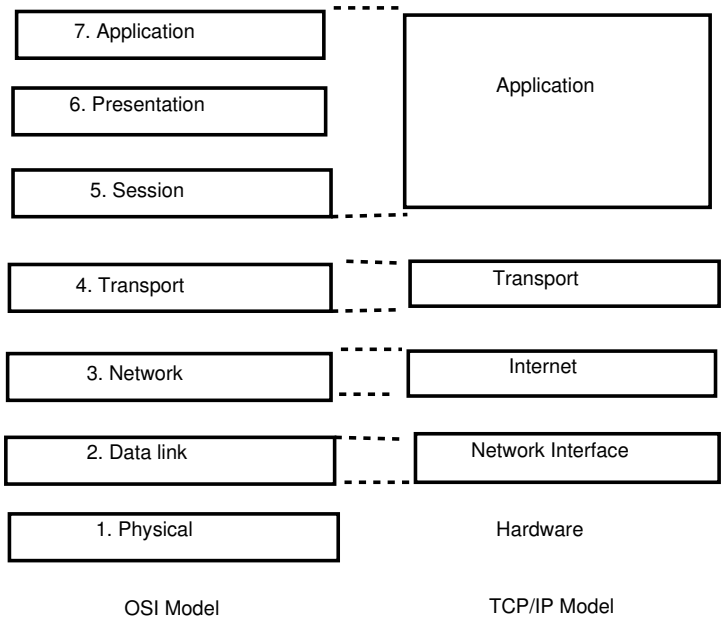
TCP/IP Stack

- TCP / IP is a suite of protocols. The acronym TCP / IP stands for "Transmission Control Protocol / Internet Protocol".
- From its military legacy it has retain the following features:
  1. Splitting messages into packets;
  2. The use of an address system;
  3. The routing of network traffic (routing);
  4. Error checking of data transmission.

  How TCP/IP stack works:
- In order to implement the TCP / IP model to any machine, regardless of the operating system, the TCP / IP protocols was divided into several modules,
- The data (information packets) that run on the network are processed sequentially by each layer, just add a piece of information (called a header) and are then transmitted to the next layer.

TCP/IP Stack...

- The TCP / IP model is very close to the OSI model (with 7 layers), which was developed by the ISO.



- Network Layer(OSI): The only constraint of this layer is to allow a host to send IP packets over the network.
- Internet layer(TCP/IP): This layer performs the interconnection of networks (heterogeneous) without remote connection.
- Due to the imminent role of this layer in the packet routing, the critical point of this layer is routing.
- The internet layer has an official implementation: IP (Internet Protocol).

TCP/IP Stack...

Transport:

- Its role is the same as the transport layer OSI model: allow peer entities to conduct a conversation. Officially, this layer has two implementations: TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).
- UDP, however, is simpler than TCP protocol, is unreliable and connectionless.

Application layer:

- Unlike the OSI model, the next higher layer to the transport layer, simply because the presentation and session layers appeared unnecessary.
- This layer contains all the high-level protocols, such as Telnet, TFTP (Trivial File Transfer Protocol) SMTP (Simple Mail Transfer Protocol), HTTP (HyperText Transfer Protocol).

TCP/IP Stack...

From the basic data to be sent, the TCP / IP stack adds all the layers needed to send this data over the network. These layers can be summarized as headers present upstream of the data

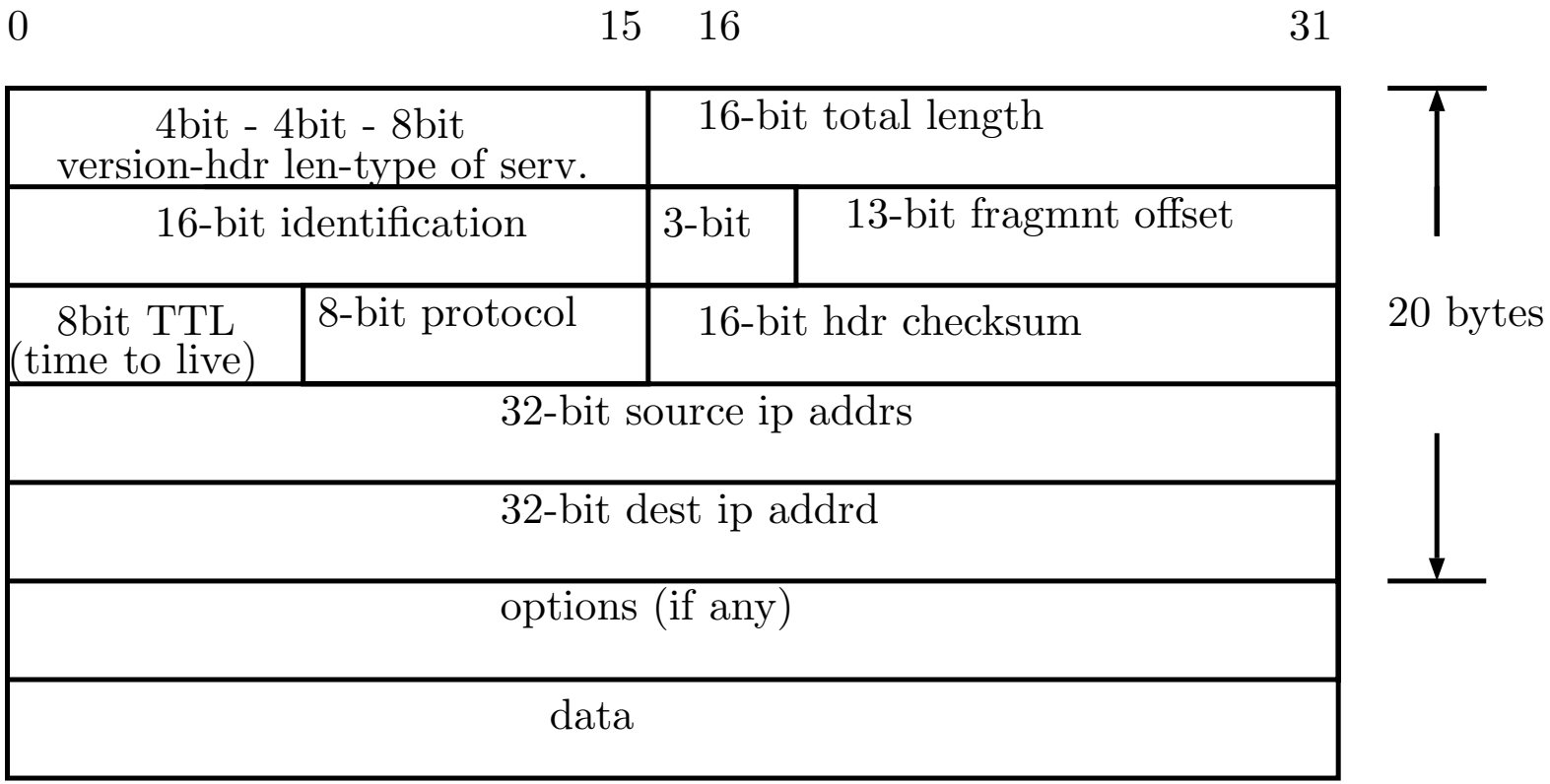


Figure 1: IP header with data.

TCP/IP Stack...

Data (optional)

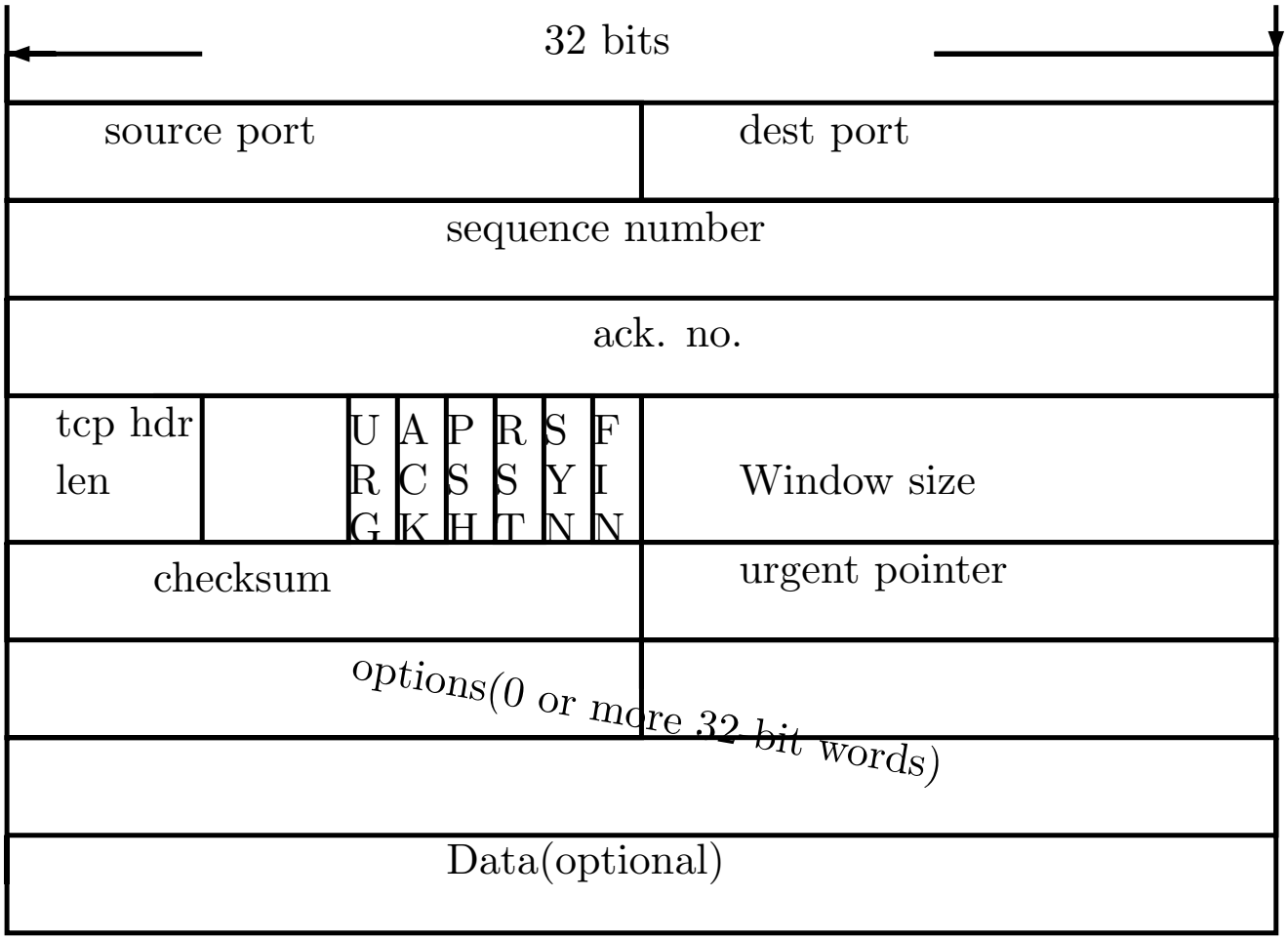


Figure 2: TCP header (data is IP layer)

TCP/IP Stack...

These headers are added by the TCP / IP stack to be able to properly orient the data on the network, as the addressee collects all the data, all without errors.

Sending data:

1. The TCP protocol breaks data into packets
2. The packets travel from router over the Internet according to IP protocol
3. TCP protocol reassembles the packets into the original value.

Security Requirements for Commercial Transactions

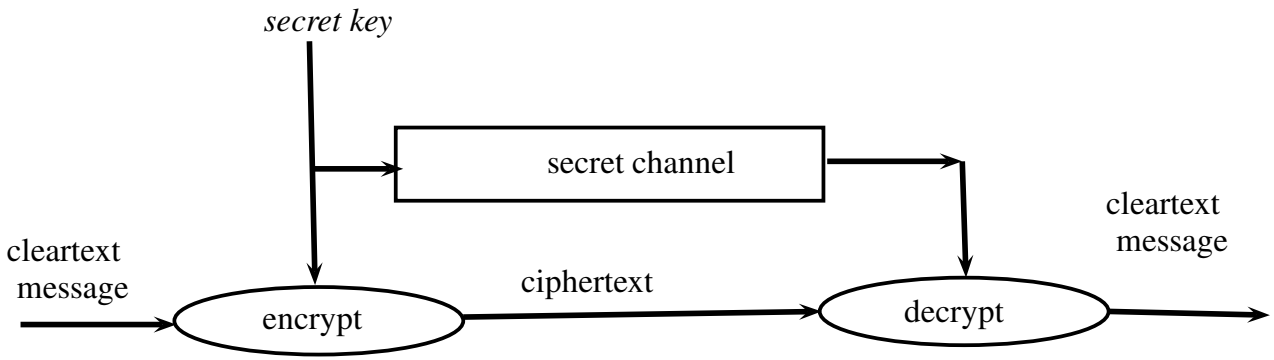
While firewalls serve a valuable purse in securing Internet-connected networks, they do not provide end-to-end transaction security

- **Confidentiality.** All communications between parties are restricted to the parties involved in the transaction.
- **Authentication.** Authentication is usually provided through digital signatures and certificates
- **Data integrity.** Data sent as part of a transaction should not be modifiable in transit. Similarly, it should not be possible to modify data while in storage.
- **Non-repudiation.** Neither party should be able to deny having participated in a transaction after the fact.
- **Selective application of services.** It may be desirable for part of a transaction to be hidden from view while the remainder of the same transaction is not.

Confidentiality is usually provided through encryption. Authentication, data integrity, and nonrepudiation are usually provided through digital signatures and publickey certificates

Cryptographic concepts

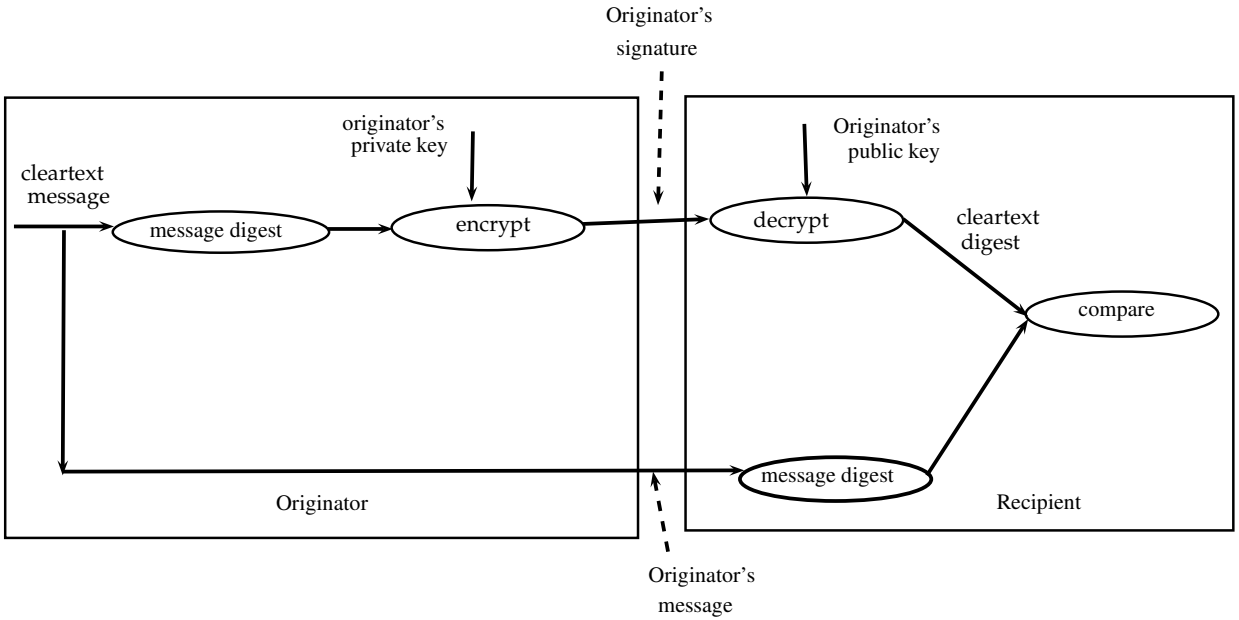
- Traditional cryptography made use of symmetric algorithms
- Such process provide confidentiality of information but
  1. do little to authenticate parties to each other, or
  2. to validate the integrity of the data transmitted.



- To provide the security services mandated the electronic commerce, most solutions also use asymmetric,
- However, since asymmetric systems are generally not as computationally as efficient

Digital Signature

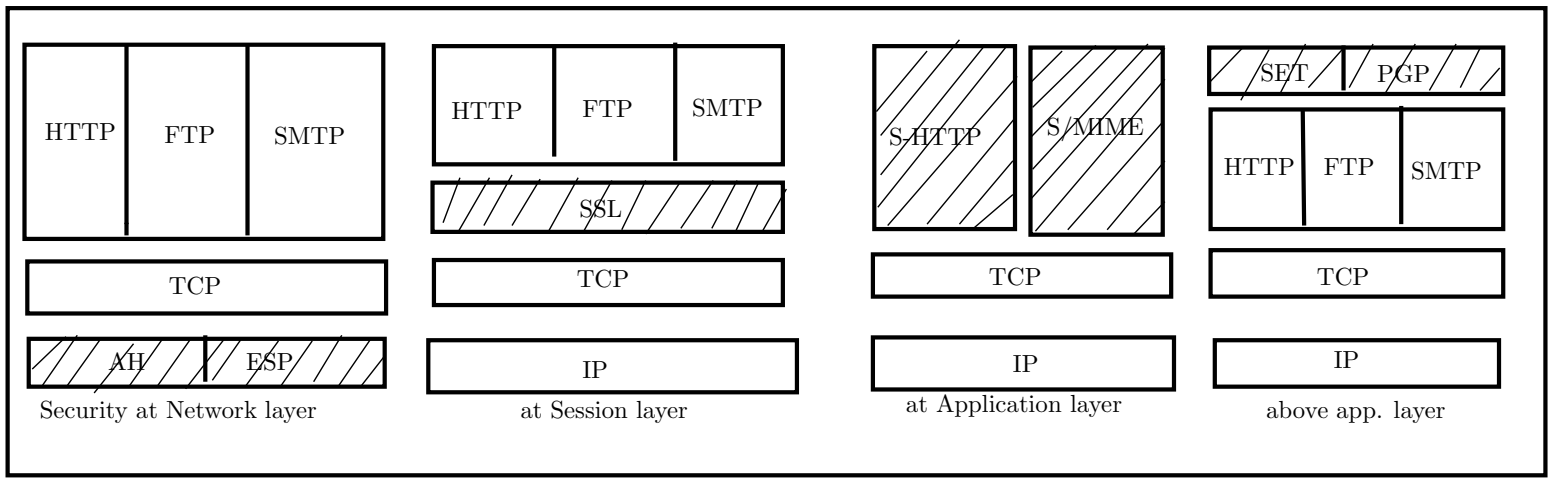
- Digital signature performs a function in the electronic world similar to the function of paper signature in the real world.
- Due to he inefficiency of most public systems, however, signatures are often implemented in conjunction with a message "digest" or "hash" function, as shown in figure.



- Crucial to the proper function of public-key system is the ability to match specific keys to their owners.
- Dig. Sig. provides data integrity, authentication, and non-repudi.

Security Initiatives

- In order to provide these services, a number of cryptographic protocols have been proposed. While these protocols are similar in the services they provide and in the cryptographic algorithms they use, they vary in the manner in which they provide the services and in their locations with respect to the rest of the TCP/IP protocol stack.
- Some initiatives endeavor to implement security at the network IP layer; others, just above TCP, at the session layer; still others, such as ftp, HTTP, and telnet, within specific application protocols,
- The issue of where within the protocol stack to provide security is contentious (shades area in fig. is security layer).



Security Solutions

- **Network-layer Solutions.** It is IP security architecture to provide cryptographic protection for Internet traffic. This includes two mechanisms for providing security services:
  - Authentication header (AH), which provides authenticity and integrity using the MD5 (message digest) algorithm
  - Encapsulating security payload (ESP), which provides confidentiality using the Data Encryption using Standard (DES) algorithm
- **Session-Layer Security Solutions.** The most prevalent session-layer protocol is the Secure Sockets Layer (SSL), which provides security services just above the TCP layer using a combination of public-key and symmetric cryptosystems to provide confidentiality, data integrity, and authentication of the server and (optionally) the client.

SSL/ TLS Session-Aware authentication

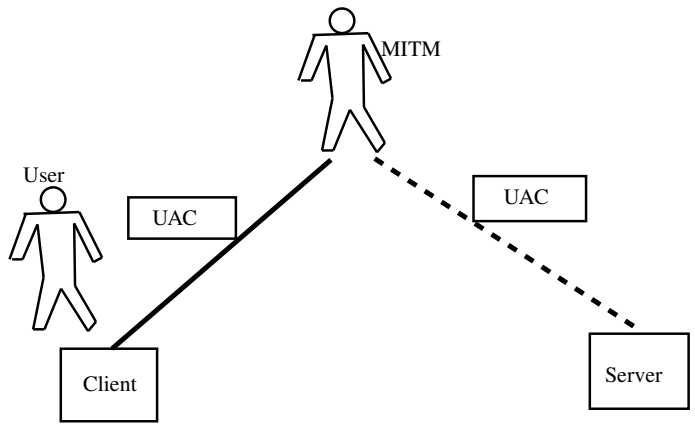
- Most e-commerce applications used today employ the Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocol to authenticate the server and cryptographically protect
- While developers consider the SSL and TLS protocols to be sound and secure in practice, the vast majority of SSL/TLS-based e-commerce applications that employ user authentication at the application layer are vulnerable to phishing, Web spoofing, and–most importantly man-in-the-middle (MITM) attacks.
- A MITM attack refers to - a form of active wiretapping attack
- Think of an MITM attack as an adversary that represents an SSL/TLS proxy server, or relay, between user and server.

Countermeasures against MITM

- Analysts within the security industry often state that only one strong authentication can thwarts MITM attack.
- An MITM may employ many tricks,and gives the connected user the impression of intended website server.
- SSL/TLS protocol has been designed to protect against MITM attacks. These protocol requires that all the clients be equipped with public-key certificate.
- Other approaches to thwart MITM attacks is to use multiple communication channels.

SSL/TSL Session-aware (SA) user authentication

- Effective countermeasure against MITM attacks in SSL/TLS setting must either enforce proper server authentication or combine user authentication with SSL/TLS session establishment.
- In the server authentication, validation of public key certificates is too involved.
- Hence, the second approach is better. It is called TLS-SA (session aware); it focus making authentication depend not only on user's secret credential but also on the state information related to SSL/TLS session
- The user authenticate himself by providing the *user authentication code* (UAC), that depends on both his credentials and SSL/TLS session.
- The key point is that the UAC is bound to a particular SSL/TLS session. Hence, if the UAC is submitted on a different session, the server can detect this



Implementation of TLS-SA

The TLS-SA is not a user authentication mechanism or system, it can be implemented in many different ways.

Basic Solution. In the basic solution, a user U is equipped with an authentication token T with a small display. U employs a client browser C to access an SSL/TLS-based application on server S.

- U is identified with ID_U and holds PIN_U, a PIN shared with S. T is identified by a serial number, SN_T, that might, for example, be imprinted on the back side of the token.
- Further, T is equipped with both a public-key pair (k, k^{-1}) and a secret token key K_T shared with S. The keys k and k^{-1} are the same for all tokens (and hence the token is impersonal), whereas K_T is unique and specific to T.
- Note that K_T is not specific to the user, hence the tokens need not be personalized, which is the main advantage of using impersonal tokens. K_T can be generated randomly, or pseudorandomly, using a master key MK, typically held exclusively by S:

$$K_T = E_{MK}(SN_T) \qquad (1)$$

Implementation of TLS-SA...

1. In the first case, where K_T is generated randomly, the system must store all token keys on the server side. In the second case, where K_T is derived from SN_T the system does not need to store token keys centrally. Instead, the user can generate K_T dynamically from SN_T and MK.
2. To access S, U directs C to S.

C and S then try to establish an SSL/TLS session using the SSL/TLS handshake protocol. As part of this protocol, S authenticates itself using a public-key certificate. S is configured so that it always requires certificate-based client authentication by sending an SSL/TLS `CertificateRequest` message to C.

Implementation of TLS-SA...

3. When C receives this message, it knows it must authenticate itself by returning a Certificate and a properly signed `CertificateVerify` message to S. The `CertificateVerify` message comprises a digitally signed hash value, *Hash*, of all messages previously exchanged during the execution of the SSL/TLS handshake protocol.
4. The server's `Certificate` message constitutes part of these messages, comprising the server's public-key certificate and hence the server's public key, included in this certificate.
5. The `CertificateVerify` message is thus logically bound to the server's public key.