Table of Contents

Sisteme de protectie

Ce este un sistem de protectie peste o multime de drepturi? (deniti toate conceptele ce intervin in explicarea conceptului de sistem de protectie)

A protection system over **R** is a finite set C of commands over **R**.

A command over **R** is a construct of the form:

$$\begin{array}{l} command\ \alpha(X_1,\ldots,X_k) \\ \quad op_1,\ldots,op_n \\ end \end{array} \qquad \begin{array}{l} command\ \alpha(X_1,\ldots,X_k) \\ \quad if\ \ r_1\ in\ (X_{s_1},X_{o_1})\ and \\ \quad\quad \ldots \\ \quad\quad r_m\ in\ (X_{s_m},X_{o_m}) \\ \quad then\ op_1,\ldots,op_n \\ end \end{array}$$

where $m,n \geq 1$, $r_1,\ldots,r_m \in R$, $X_1,\ldots,X_k \in \mathscr{V}_{sub} \cup \mathscr{V}_{ob}$, $1 \leq s_1,\ldots,s_m,o_1,\ldots,o_m \leq k$, $X_{s_i} \in \mathscr{V}_{sub}$ and $X_{o_i} \in \mathscr{V}_{ob}$ for all $1 \leq i \leq m$, and $op_1,\ldots,op_n$ are operations over R whose variables are among $X_1,\ldots,X_k$.

A primitive operation over R is a construct of the one of the following types:

1. *enter r into* (X_s,X_o)
2. *delete r from* (X_s,X_o)
3. *create subject* X_s
4. *create object* X_o
5. *destroy subject* X_s
6. *destroy object* X_o

where $r \in R$, $X_s \in \mathscr{V}_{sub}$, and $X_o \in \mathscr{V}_{ob}$.

In ce consta problema sigurantei sistemelor de protectie?

Let $\mathscr{C}$ be a protection system over R, Q a state of $\mathscr{C}$, $r \in R$, and α a command of $\mathscr{C}$. We say that α leaks r from Q if there exists a substitution σ such that:

1. the test of $\sigma(\alpha)$ is satisfied at Q;
2. there exist $Q_0,Q_1,\ldots,Q_i$ such that:
   - $Q_0 = (S_0,O_0,A_0) \Rightarrow_{\sigma(op_1)} Q_1 = (S_1,O_1,A_1) \Rightarrow_{\sigma(op_2)} \cdots \Rightarrow_{\sigma(op_i)} Q_i = (S_i,O_i,A_i)$;
   - $r \in A_i(s,o) - A_{i-1}(s,o)$ for some s and o,

   where $op_1,\ldots,op_i,\ldots,op_n$ is the body of α and $1 \leq i \leq n$.

Let $\mathscr{C}$ be a protection system over R, Q a state of $\mathscr{C}$, and $r \in R$. We say that $\mathscr{C}$ leaks r from Q if there exists a command of $\mathscr{C}$ that leaks r from Q.

Let $\mathscr{C}$ be a protection system over R, Q a state of $\mathscr{C}$, and $r \in R$. We say that Q is unsafe for r if there exists a reachable state Q' from Q such that $\mathscr{C}$ leaks r from Q'.

We say that Q is safe for r if it is not unsafe for r.

The safety problem for protection systems is the problem to decide, given a protection system over some set R of rights, a state Q of $\mathscr{C}$, and a right $r \in R$, whether Q is safe for r.

Ce cunoasteti despre dificultatea rezolvarii algoritmice a problemei sigurantei sistemelor de protectie?

Theorem 19

The safety problem for bi-conditional (i.e., at most two conditions) monotonic (i.e., without delete and destroy operations) protection systems is undecidable.

Theorem 20

The safety problem for mono-conditional protection systems without destroy-operations is decidable.

Most practical systems require multi-conditional commands !

Ce este o lista de control al accesului? Discutati cazul Unix si WinNT.

An access control list (ACL) is a column of the access control matrix(therefore, associated to an object - the ACL associated to o is denoted ACLo, and it is stored along with o).

In UNIX:

every file or folder has associated access permissions. There are three types of permissions: read access, write access, execute access. Permissions are defined for three types of users: the owner of the file, the group that the owner belongs to, anyone else (world). Each permission type has exactly two values, allowed or denied, specified by a bit. Ex: drwxrwxrwx Alice Accounts;

In WinNT:

In Windows NT, the access control is richer than Unix, but not fundamentally different: There are six types of permissions: read access, write access, execute access, delete, change permissions (i.e., modify the ACL), take ownership (make current account the new owner), permissions are defined for

users and groups. Each permission type has three values, Access denied, Access allowed, and System audit.

ACLs are associated to items (i.e., files or directors), and each ACL is a list of entries of the form: (user/group,permissions)

Ce este o lista de capacitati?

An capability list (C-list) is a row of the access control matrix (therefore, associated to a subject - the C-list associated to s is denoted Cs, and it is stored along with s).

In practice, it is more convenient to store a C-list Cs as a list of pairs (o; r), where o is an object and r is a right (permission). Such a pair will be called a capability; then, Cs becomes a list of capabilities. Each capability acts like a ticket for s to access o with permission r. Therefore, capabilities are authentication tags.

Ce intelegeti prin acces discretionar si acces mandatar?

Discretionary (DAC) – enforce access control on the basis of the identity of the requester and explicit access rules that establish who can or cannot execute which actions on which resources.

- DAC models enforce access control on the basis of the identity of requesters
- DAC models are called “discretionary” as users can be given the ability of passing on their privileges to other users
- DAC mechanisms usually include a concept of object ownership
- DAC models: take-grant model, access-matrix model, schematic model

Mandatory (MAC) – enforce access control on the basis of regulations mandated by a central authority.

- MAC enforces access control on the basis of regulations mandated by a central authority
- No concept of ownership in MAC
- MAC makes distinction between users and subjects
- MAC models: Bell-LaPadula model, Biba model, Chinese-wall model

IPsec

Ce este o asociere de Securitate in IPsec si care sunt mecanismele de Securitate fundamentale din IPsec?

A security association (SA) is a unidirectional logical connection between two IP systems, uniquely identified by a triple (SPI, IP destination address, security protocol) where

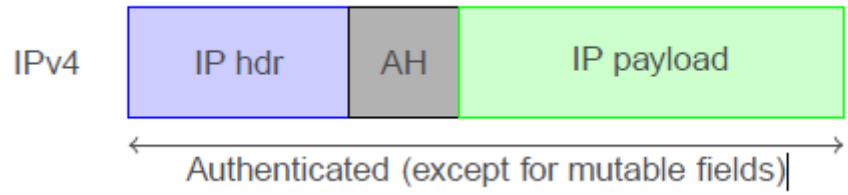
- SPI (security parameter index) is a 32-bit value used to identify different
- SAs with the same destination address and the same security protocol
- IP destination address can be unicast, broadcast, or multicast
- security protocol – this can be either AH or ESP

IPsec operates in two modes: transport (for end-to-end), tunnel (for VPN).

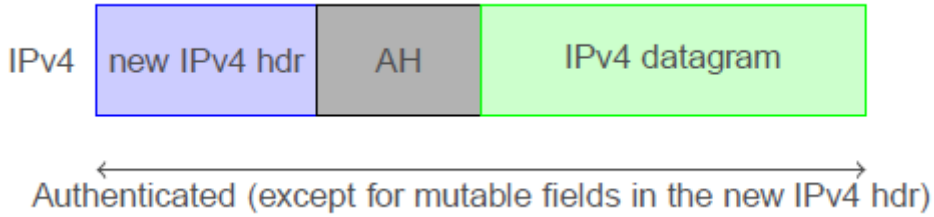
Descrieti, succint dar clar, protocolul AH in cele doua moduri de utilizare pentru datagrame IPv4.

Authentication Header (AH): piece of information associated to an IP datagram in order to authenticate certain fields of the datagram

In the transport mode, AH authenticates the IP payload and selected portions of the IP header (e.g., mutable and unpredictable fields are not authenticated)



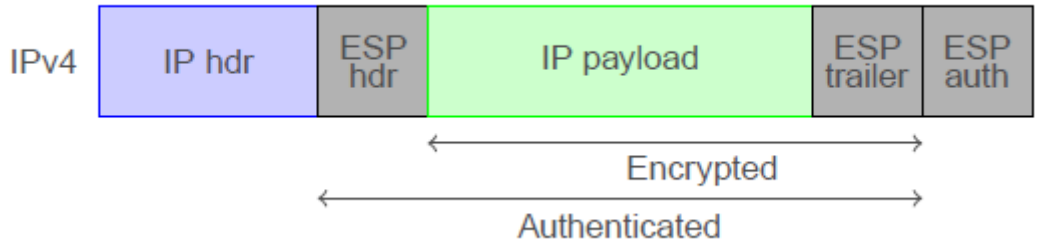
In the tunnel mode, AH authenticates the entire inner IP packet plus selected portions of the outer IP header and outer IP extension headers



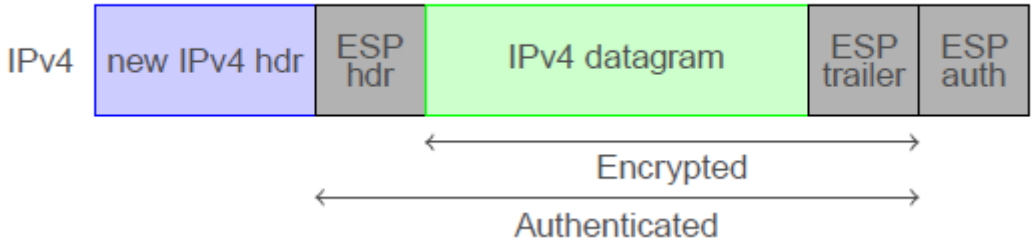
Descrieti, succint dar clar, protocolul ESP in cele doua moduri de utilizare pentru datagrame IPv4.

Encapsulating Security Payload (ESP): obtained from an IP datagram by encrypting, and optionally authenticating, certain fields of the datagram.

In the transport mode, ESP encrypts and optionally authenticates the IP payload (but not the IP header)



In the tunnel mode, ESP (with authentication) encrypts (and authenticates) the inner IP packet



Descrieti cateva combinatii de asocieri de securitate in IPsec (end-to-end, VPN, end-to-end cu VPN).

End-to-end security:

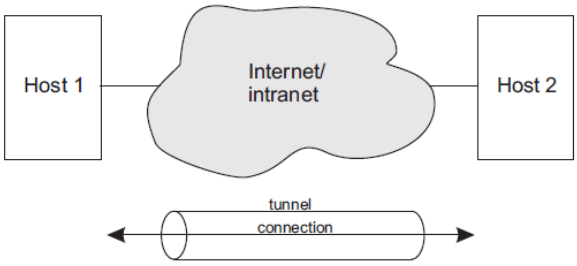


Figure: End-to-end security

Two hosts are connected through the Internet or an intranet without any security gateway between them. They can use ESP, AH, or both. Either transport or tunnel mode can be applied

VPN security:

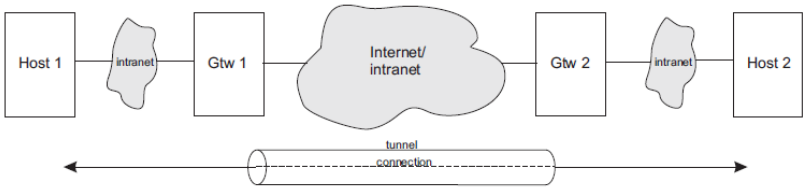


Figure: Basic VPN support

The hosts in the intranets are not required to support IPsec, but the gateways are required to run IPsec and support tunnel mode (either with AH or ESP)

End-to-end Security with VPN Support:

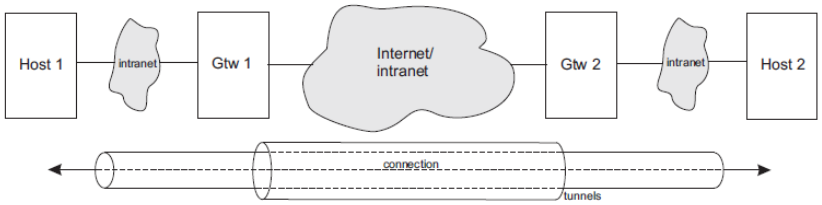


Figure: End-to-end security with VPN support

This is a combination of the previous two cases. For instance, the gateways may use AH in tunnel mode, while the hosts use ESP in transport mode

SSL/TLS

Care este scopul de baza a protocolului SSL/TLS?

Primary role of SSL(Secure Sockets Layer): to provide a private channel between communicating applications to ensure privacy of data authentication of the parties integrity.

When SSL was standardized by the IEFT, it was renamed to TLS.

Descrieti, succint dar clar, metodele de schimb de cheie RSA si DH in SSL.

The pre-master secret is established between parties, in the handshake protocol, by one of the following methods:

RSA: the client generates a pre-master secret that is encrypted with the server's public key, and sends it to the server

DH: the pre-master secret is the DH value obtained from the client's and server's DH public arameters. There are three variants:

- **Fixed DH:** the server must have a certificate which should include his DH public parameters. The client provides its DH public parameters either in a certificate or in a key exchange message
- **Ephemeral (temporary, one-time) DH:** DH public parameters are exchanged and signed using sender's private RSA or DSS key. Certificates are needed to authenticate the public keys
- **Anonymous DH:** this is DH with no authentication

Care sunt pasii de baza ai protocolului "SSL record"?

SSL Record Layer protocol has got the below functions to fulfill:

- Breaking Down the Data from Application layers, with fixed length.
- Compress the Data
- Add Message Authentication Code, Which is calculated with the help of Integrity Key.
- Encrypt the packets(which was broked down with fixed length).
- Add SSL header's in the packets with fixed length. Which consists the following headers, which combinely form a 5byte header.

Received data are processed in reverse order.

DNS si DNSsec

Descrieti, succint dar clar, modul de functionare a protocolului DNS.

Domain Name Servers (DNS) are the Internet's equivalent of a phone book. They maintain a directory of domain names and translate them to Internet Protocol (IP) addresses.

The IP address is retrieved by following these steps:

1. I log onto my Internet Service Provider (ISP) to use the Web.
2. I open my web browsing software (i.e. Internet Explorer or Netscape Navigator) and type http://www.dashsystems.com into the location bar.
3. My computer asks my ISP's DNS server(s) for the IP address of www.dashsystems.com.
4. My ISP's equipment first checks its memory cache to find out if it has fulfilled a request for this same address recently.
5. If it has not, my ISP's equipment will communicate with InterNIC's conglomeration of root servers that make up the Domain Name System to find out which DNS server holds the IP address of the domain name.
6. My ISP's equipment takes the address provided and sends a query to the authoritative DNS server for that domain.
7. The authoritative DNS server responds with the IP address of the desired server.
8. My ISP's equipment updates its memory cache with the address so that it respond to future requests without all the steps above.
9. My ISP's equipment responds to my computer with the IP address of the server for which I am looking.
10. My computer updates its memory cache so that it doesn't have to look up the address for a while.
11. My computer hands the address to my browser, which opens a connection to the server (using the specified IP address) and retrieves the first page from the site I requested.
12. My browser displays the requested page on my screen.

Descrieti, succint dar clar, modul de functionare a protocolului DNSsec.

DNSsec adds a layer of trust on top of DNS by providing authentication.

When DNSSEC is used, each answer to a DNS lookup will contain an RRSIG DNS record, in addition to the record types that were requested. The RRSIG record is a digital signature of the answer DNS resource record set.

From the results, a security-aware DNS resolver can then determine if the answer it received was correct (secure), whether the authoritative name server for the domain being queried doesn't support DNSSEC (insecure), or if there is some sort of error. The correct DNSKEY record is found via an Authentication Chain, starting with a known good public key for a Trust Anchor, preferably at the DNS root. This public key can then be used to verify a delegation signer (DS) record.

Prezentati si discutati 2 argumente pentru care credeti ca DNSsec asigura securitate.

When properly maintained, DNSSEC signed zones provide extra security by preventing man-in-the-middle attacks.

Any customer with DNSSEC-aware resolver will not be at risk from DNS spoofing.

https://technet.microsoft.com/en-us/library/dn593670(v=ws.11).aspx

RFC 822 si MIME

Care sunt principalele dezavantaje ale formatului de e-mail RFC 822?

RFC 822 does not clearly distinguish the envelope fields from the header fields. A line of text is subjected to the following constraints:

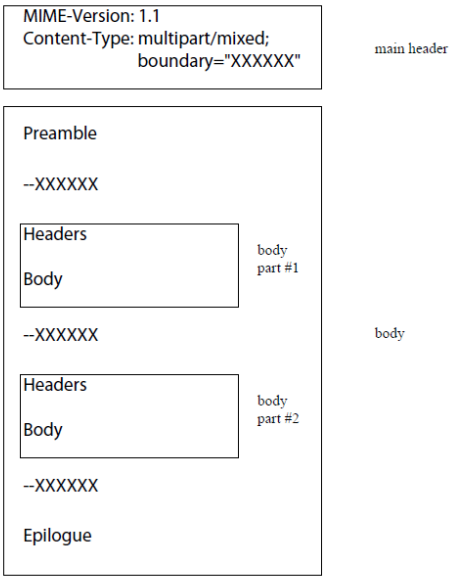
- consists of 7-bit ASCII characters only;
- each line ends with a CR character, followed by an LF character (this combination is called \CRLF");
- each line should be 78 characters or less (not including CRLF), and must not be more than 998 characters (not including CRLF);
- the characters CR and LF must not appear by themselves within the text line.

RFC 822 email format **does not support: (primary disadvantages maybe?)**

- messages in languages with accents (French, Romanian) in non-Latin alphabets (e.g., Russian), or in languages without alphabets (e.g., Chinese);
- non-text information (e.g., graphic files, multimedia);
- arbitrary binary files (including executable programs).

Care este structura de baza a unui format MIME?

Multipurpose Internet Mail Extensions (MIME) came as solution to the above problems.



Care sunt cele 4 metode de codificare a informatiei in MIME?

- **7bit.** This indicated 7-bit ASCII format (RFC 822 standard)
- **8bit/binary.** Encoding using 8-bit characters;
- **Quoted-printable**. This is used when most of the data is ASCII text and only very few characters are non-ASCII
- **Base64**. Encodes data by mapping 6-bit blocks of input to 8-bit blocks of outputs, all of which are printable ASCII characters (also called Radix-64 or ASCII armor)
- **RFC 2046** defines two other encodings, **ietf-token and x-token**, to allow new encoding types to be defined in the future

In ce consta metoda de codificare “quoted-printable"?

The Quoted-Printable encoding is intended to represent data that largely consists of octets that correspond to printable characters in the ASCII character set. It encodes the data in such a way that the resulting octets are unlikely to be modified by mail transport. If the data being encoded are mostly ASCII text, the encoded form of the data remains largely recognizable by humans.

In this encoding, octets are to be represented as determined by the following rules:

A) General 8-bit representation: Any octet, except those indicating a line break may be represented by an "=" followed by a two digit hexadecimal representation of the octet's value. The digits of the hexadecimal alphabet, for this purpose, are "0123456789ABCDEF". Uppercase letters must be used when sending hexadecimal data, though a robust implementation may choose to recognize lowercase letters on receipt. Thus, for example, the value 12 (ASCII form feed) can be represented by "=0C", and the value 61 (ASCII EQUAL SIGN) can be represented by "=3D". Except when the following rules allow an alternative encoding, this rule is mandatory

B) Literal representation: Octets with decimal values of 33 through 60 inclusive, and 62 through 126, inclusive, MAY be represented as the ASCII characters which correspond to those octets (EXCLAMATION POINT through LESS THAN, and GREATER THAN through TILDE, respectively).

C) White space: Octets with values of 9 and 32 MAY be represented as ASCII TAB (HT) and SPACE characters, respectively, but MUST NOT be so represented at the end of an encoded line. Any TAB (HT) or SPACE characters on an encoded line MUST thus be followed on that line by a printable character. In particular, an "=" at the end of an encoded line, indicating a soft line break (see rule #5) may follow one or more TAB (HT) or SPACE characters. It follows that an octet with value 9 or 32 appearing at the end of an encoded line must be represented according to Rule #1.

D) Line breaks: A line break in a text body, independent of what its representation is following the canonical representation of the data being encoded, must be represented by a (RFC 822) line break, which is a CRLF sequence, in the Quoted-Printable encoding.

E) Soft line breaks: The Quoted-Printable encoding REQUIRES that encoded lines be no more than 76 characters long. If longer lines are to be encoded with the Quoted-Printable encoding, 'soft' line breaks must be used. An equal sign as the last character on an encoded line indicates such a non-significant ('soft') line break in the encoded text.

In ce consta metoda de codificare Radix64?

Base 64 Encoding takes a stream of bits and converts them to 8 bit characters that belong to the universal ASCII character set. Base 64 Encoding does not care about how many bits (8 or 16) are necessary to make a character as it works at the bit level. Once a stream of bits have been converted to characters that belong to the universal ASCII character set (Base 64 encoded) they can be transported with ease over the Internet using the e-mail protocols.

Base64 Encoding takes three bytes of character data (3 ASCII characters or 1½ UNICODE character) and produces four bytes of data from the universal character set. That is, Base64 Encoding takes 24 bits of input data and converts it to 32 bits of encoded data. (Decimal encoding: 1 to 3 ratio; Hexadecimal encoding: 1 to 2 ratio; Base64 encoding: 3 to 4 ratio)

Base64 Encoding does not care what the data is. If the data contains line feeds, nulls or special characters, Base64 Encoding does not care - it simply sees it as a sting of bytes.

Pretty Good Privacy(PGP)

Ce servicii ofera PGP?

Pretty Good Privacy (PGP) is a computer program that provides cryptographic privacy and authentication for data communication.
PGP offers **two cryptographic services**:

- authentication (by signatures);
- confidentiality (by encryption),

and **three auxiliary services**:

- compression (which is applied after signing a message but before encryption);
- email compatibility (base64 conversion);
- segmentation (messages longer than a given maximum length are broken up into smaller segments).

Cum se realizeaza autentificarea in PGP?

PGP authentication (applied to x):

- $A \to B : Z(x, sig_A(x))$

where $sig_A(x)$ is A's signature on x, and Z denotes Zip-compression. A's signature can be an RSA or DSS signature on $SHA\text{-}1(x)$.

Cum se asigura confidentialitatea in PGP?

PGP confidentiality (applied to x):

- $A \to B : (\{K\})\ \{Z(x)\}_K, \{K\}_{K_B^e}$

where the symmetric encryption is by CAST-128, IDEA, or 3DES, all in the CFB mode, and the asymmetric encryption is by RSA or ElGamal.

Cum se realizeaza autentificarea si confidentialitatea, impreuna, in PGP?

PGP authentication and confidentiality are obtained by applying confidentiality to $x' = (x, sig_A(x))$.

Duh.

Cum se realizeaza compresia informatiei in PGP?

As a default, PGP compresses the message after applying the signature but before encryption. This has the benefit of saving space both for e-mail transmission and for file storage. The compression algorithm used is called **ZIP**.

Explicati modul de formare si utilizare a inelelor de chei in PGP.

PGP key storage is based on associating identifiers to public keys. Then, asymmetric keys are stored as table-like structures, called key rings, that may be indexed by user ID or key ID**.** Using PGP key rings:

- User U signing a message:
  - PGP retrieves U's encrypted private key from the private-key ring using ID(U);
  - PGP prompts U for the password to recover U's private key;
  - the signature is constructed;
- User U encrypting a message:
  - PGP generates a session key and encrypts the message;
  - PGP retrieves the recipient's public key from the public-key ring using his/her ID;
  - the session key is encrypted by the recipient's public key.
- User U verifying a signature:
  - PGP retrieves the sender's public key from the public-key ring using the key ID (which is attached to the message he received);
  - PGP verifies the signature;
- User U decrypting a message:
  - PGP retrieves U's encrypted private key from the public-key ring using his/her ID;
  - PGP prompts U for the password to recover U's private key;
  - PGP decrypts the message.

S/MIME

Cum se realizeaza autentificarea in S/MIME? (atentie: exista doua tipuri de autentificare in S/MIME).

Secure/Multipurpose Internet Mail Extensions (S/MIME) is a standard for public key encryption and signing of MIME data.
S/MIME provides the following functions:

- envelop data. Encrypt data and then base64-encode it;
- sign data. Sign data and then base64-encode it;
- clear-sign data. Sign data and then base64-encode the signature only;
- sign and envelope data. Signed and encrypted data are nested.

S/MIME enveloped data

S/MIME enveloped data

S/MIME envelopes x as follows:

- $A \rightarrow B : (\{K\})\ R64(ID_{PKC}, ID_{EA}, \{K\}_{K_B^e}, \{x\}_K)$

where:

- ID_{PKC} is an identifier of the recipient's public-key certificate (which certifies K_B^e);
- ID_{EA} is an identifier of the encryption algorithm used to encrypt the session key K;
- $R64$ denotes the base64 encoding.

S/MIME signed data

S/MIME signed data

S/MIME signs x as follows:

- $A \rightarrow B : R64(ID_{PKC}, ID_{MDA}, ID_{EA}, sig_{K_A^d}(x), x)$

where:

- ID_{PKC} is an identifier of the signer's public-key certificate (which certifies the public key needed to verify the signature);
- ID_{MDA} is an identifier of the message digest algorithm used to obtain a message digest of x;
- ID_{EA} is an identifier of the encryption algorithm used to encrypt the message digest of x;
- $R64$ denotes the base64 encoding.

S/MIME clear sign data

S/MIME clear signing

S/MIME clear-signs x as follows:

- $A \rightarrow B : R64(ID_{PKC}, ID_{MDA}, ID_{EA}, sig_{K_A^d}(x')), x'$

where:

- x' is obtained by encoding x by base64 or quoted-printable encoding in case that x is not 7-bit ASCII;
- all the other elements are as in the case of S/MIME signed data.

In case of a clear signature, the message can be viewed by anyone who have MIME capabilities, although he cannot check the signature.

Cum se asigura confidentialitatea in S/MIME?

Nu stiu care e legatura intre (sign,envelope) si (authentication, confidentiality) :D

Cum se realizeaza autenticarea si confidentialitatea, impreuna, in S/MIME?

Nu stiu care e legatura intre (sign,envelope) si (authentication, confidentiality) :D

Elemente de criptografie

Ce este un criptosistem simetric?

The same key is used for both encryption and decryption. The key need to be kept as private key, hence the SC can also be called as private key cryptography. The secure distribution of keys is the major challenge that is associated with symmetric key cryptosystems. Data Encryption Standard and Advanced Encryption Standards are the algorithms which uses common cryptosystems.

Ce este un criptosistem asimetric (cu chei publice)?

Both private key and public key are used in Asymmetric cryptosystems. One key is used for data encryption and another for data decryption. Asymmetric cryptography is used in solving the challenge of secure distribution of the secret keys. Asymmetric cryptography solves the challenge of secure distribution of secret keys.

Care este diferenta majora intre un criptosistem simetric si unul cu chei publice?

The key difference between asymmetric and symmetric encryption is that symmetric encryption uses one secret key that has to be shared among the sender and recipient of the message, while asymmetric encryption utilizes a private key and a public key to decrypt and encrypt messages during communication. Asymmetric encryption is a relatively new technique compared to symmetric encryption.

Ce este o functie hash?

A hash function takes a group of characters (called a key) and maps it to a value of a certain length (called a hash value or hash). The hash value is representative of the original string of characters, but is normally smaller than the original.

Ce este o semnatura digitala? Cum se construieste semnatura digitala RSA?

A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software or digital document.

To create RSA signature keys, generate a RSA key pair containing a modulus, N, that is the product of two large primes, along with integers, e and d, such that $e\ d \equiv 1 \pmod{\varphi(N)}$, where φ is the Euler phi-function. The signer's public key consists of N and e, and the signer's secret key contains d.

To sign a message, m, the signer computes a signature, σ, such that $\sigma \equiv m^d \pmod{N}$. To verify, the receiver checks that $\sigma^e \equiv m \pmod{N}$.

Mdaaa

In ce consta metoda de demonstratie challenge-and-response?

Challenge-response authentication is a group or family of protocols characterized by one entity sending a challenge to another entity. The second entity must respond with the appropriate answer to be authenticated.

A simple example of this is password authentication. The challenge is from a server asking the client for a password to authenticate the client's identity so that the client can be served.

Ce se intelege prin zero-knowledge-proof?

In cryptography, a **zero-knowledge proof** or **zero-knowledge protocol** is a method by which one party (the *prover*) can prove to another party (the *verifier*) that a given statement is true, without conveying any information apart from the fact that the statement is indeed true.

https://en.wikipedia.org/wiki/Zero-knowledge_proof