DATA PROTECTION; IDENTIFICATION AND AUTHENTICATION IN APPLICATIONS AND PROTOCOLS

Tedo Vrbanec, B. S.
Teacher training college Čakovec
Dr. Ante Starčevića 55, 40000 Čakovec, Croatia
e-mail: tedo.vrbanec@ck.hinet.hr

Professor Željko Hutinski, Ph. D.
Faculty of organisation and informatics
Pavlinska 2, 42000 Varaždin, Croatia
e-mail: zhutinsk@foi.hr

1. INTRODUCTION

Information security has to respect three contradictory requests: protection of confidentiality, integrity and availability of information. The concept of building security of the information systems distinguishes many approaches. On the level of identification, authentication and authorization of users, users rights, authentication and authorization of users data, many systems of protocols and applications have been developed, which increase the level of security systems.

Cryptography and coding are the basic ways of data protection. Implementation of the protection is carried out by crypto algorithms, which correspond to the demanding level of data security and verification of user's authenticity. All methods of data protection are based on the crypto algorithms by which messages become senseless to those whose are not authorized users. Furthermore, several methods and protocols of key interchange are in use, which have a crucial role in crypting and users authentication.

Defining the security policies and designing the security elements lead to a safe and secure environment for information systems as a main support for open and dynamic business systems.

2. IDENTIFICATION, AUTHENTICATION AND AUTHORISATION

User identification is a procedure during which a potential user identifies him to the information system when logging in. Generally, it is a process during which one entity introduces itself and identifies to another entity (presuming the first entity integrity!). **Authentication** is the authenticity check procedure, i.e. it checks the user's identity, by comparing the data received from the entity with those stored in the base. It should be mentioned that the entity integrity is not necessary. For example, when logging on ISP it is possible to connect more people from the same telephone line and the same computer, under the same user name and password. **Authorisation** is the procedure of a system user access rights definition and most often it is a part of authentication.

The expansion of Internet as a communication system among computers, especially open distributed systems, has resulted in constant exponential transactions growth. It is absolutely logical because a resource, which enables cheaper and faster communication and business transactions, has appeared. However, it has also brought new problems. Independent (but networked) computers and open distributed systems, because of their nature, have become potential targets for the competiton, hostile differently motivated attackers, as well as "passers-by" – onlookers. According to the old saying "Prevention is the surest form of cure" it has become necessary to put the network resources under the kind of protection, which doesn't even allow the unauthorised users enter the system. Initially, user names and keywords were introduced, and later, in smaller extent, (expensive) biometric analyses (of voice, iris, fingerprints etc.) or physical identificators like cards and identity cards. However, the messages remained unprotected and it was still possible to, by analysing the network traffic, block, monitor (and use) or disturb them by removing or modifying particular parts of a message, changing their order, repeating them etc. In order to protect from these or similar kinds of tricks the cryptography i.e. cryptographic algorithms were developed, and then implemented in various applications and protocols. The result and the

aim is the safe message transfer, unambiguous and protected from any modifications, disappearance and unauthorised use.

The payment mode itself – physical form of money –became the obstacle to the new management type, but not for long. Various electronic payment modes have been developed. Yet, electronic cash flow in the form of electronic information between two sides that communicate by means of the Internet enables the third party to monitor it and possibly misuse it. Transaction participant's authenticity check, as well as protection by encryption, has been carried out in order to prevent such undesired activities. Encryption is based on different cryptographic algorithms and mechanisms, as well as on the higher-level protocols developed particularly for electronic data protection and transaction participant's privacy. Privacy and authenticity are the basic characteristics of the potential electronic payment system.

User identification and authentication is a prerequisite for introduction of any protection system. All protection measures we can think of don't make any sense if there is a possibility of false identification and logging in the system. The protected system has to authenticate every user and on that basis allow or not allow certain actions within the system regarding the previously defined rights.

2.1. Ways and methods of identification and authentication

Among many physical ways of authentication let us name only a few: classical way by means of *login & password*, *SmartCard*, fingerprint, etc. Identification and authentication is nowadays usually achieved by a combination of something the user **has** and something the user **knows,** as the optimal ratio of the access level protection and its cost, regardless whether the cost is calculated in currency or time units. It is also possible to apply the authentication method based on user's individual characteristics, but it belongs to (relatively expensive) biometric fingerprint analyses, iris pictures, voice frequency analysis etc.

There is physical and logical identification and authentication

a. *Physical identification and authentication* –based on the possession of a certain physical item, e.g. identification card or biometric access control
b. *Logical identification and authentication* – it controls if the user is familiar with the particular data assigned only to him by the system or it was him who entered it into the system

Set of methods used for solving the identification, authentication and authorisation problems include the following:

- defining and implementation of security policy, documentation,
- messages encryption by different kinds of algorithms,
- electronic signature,
- access control and identification of every single access

Previously mentioned methods obey the following *principles*:

- obligatory encryption of the traffic among system nodes,
- use of a security system (like ISO/IEC 17799),
- safe approach to important remote data,
- risk prevention and reduction of unauthorised intrusions into computer systems
- technical measures have to be co-ordinated, planned, managed and included in business processes.

3. INTRANET PROTECTION FROM THE UNAUTHORISED USER ACCESS

Network (of any kind) is a communication medium, which is more or less accessible to unauthorised users as well. Because of that it **has to be considered as unsafe**. Intranet protection from unauthorised access to data contents is usually carried out by means of so called f*irewall* and *network intrusion detection devices*.

Firewall devices represent the passage through which all the data exchanged between two networks should pass through. In that way they control the access by creating a path between two networks through which all the data have to pass. There are two basic kinds: the first kind that functions as a filter, which checks the addresses of every single package received. In that way the

packages coming from the known resource will be let through, while for the others the access will be denied. The access control is carried out in the following ways:

- incoming message filtration;
- control of all communications towards intranet – permission to establish communication granted only to particular computers;
- hiding of the complete intranet, so the outgoing messages are given different addresses.

Proxy firewall devices are mediators between two devices, which communicate through the *firewall.* They interrupt the flow on one side, protect *proxy* services and open the flow to the other side of the *firewall* device.

Network intrusion detection device

These devices are relatively unknown. Their task is to detect network intrusion and to alarm the administrator in case of the suspicious actions within the network. There are two kinds: ***Anomaly detection devices*** use statistical methods for anomaly detection within the network. In case an anomaly is detected, the system administrator is alarmed and all activities recorded in the log. ***Misuse detection devices*** use patterns for detection. They take network traffic patterns and compare the suspicious patterns with the already stored common dangerous activities. Since new intrusion kinds appear now and then, frequent upgrades are necessary. So, it can be compared to anti virus program activities.

4. DATA CONTENTS SECRECY PROTECTION

Network environment security depends on the cryptographic techniques used for protection of consistency and confidentiality of the transmitted information. Consistency protection ensures that the transmitted information cannot be changed, while confidentiality protection ensures that unauthorised people cannot read the transmitted information. Cryptography and encryption are the fundamental ways of data contents protection. Messages of vital importance have to be encrypted so the data like a credit card number or PIN are protected from unauthorised use or misuse. Protection is carried out by means of various encryption algorithms and different authenticity checks of interlocutors. The base of such procedures is a cryptographic algorithm, which makes a message unintelligible to everyone who doesn't possess the information for its decryption. Beside such algorithms there are other protocols, which enable the exchange of keys essential for encryption and authentication of interlocutors, in order to keep their privacy.

Symmetric system uses the same secret key both for message encryption and decryption. The sender and the receiver use the same algorithm and the same encryption and decryption key. The key that is used should be secret, so there is a danger of interception if communication channels are unsafe.

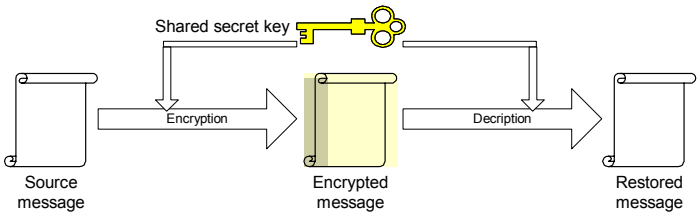

Fig.1. Symmetric encryption system

Asymmetric system introduces a public key for encryption. The message sender has to be familiar with the receiver's public key, which is generated on the base of the secret key, so it is impossible from a public key to make its secret complement. Electronic certificates issued by the third party in order to avoid impersonation check if the public key belongs to a particular person. The message receiver possesses a secret key, which needn't be distributed to other users, i.e. message senders, and it is only him who can, by using it, decrypt the message. This system enables another kind of protection - authenticity check certification by electronic signature.

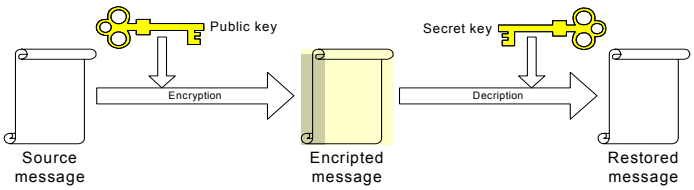

Fig.2. Asymmetric encryption

There are three kinds of cryptographic algorithms:

- **Symmetric algorithms**
  - the same public key is used for encryption and decryption of data/messages or seals
  - it is used for data transfer because of its speed
- **Asymmetric algorithms** (public key algorithms)
  - two different keys, public and secret key, are used for encryption and decryption
  - public key can be known to everyone, it is used for key, certificate and permit exchange.
- **Irretrievable (*hash*) algorithms**
  - there is only the encryption key, while there is no decryption and it is impossible to perform it in real time.
  - it is mainly used when changing base keys into dialogue ones.

4.1. The most common algorithms and encrypting protection functions

4.1.1. Symmetric algorithms

DES algorithm (Data Encryption Standard)

The algorithm is based on a secret key owned by both sides and which is used for both encryption and decryption. Encryption is performed by dividing messages into 64 bit binary blocks, which are encrypted individually in 16 steps and for each of them a different part of the key is used. It is a 56 bit key plus 8 parity bits, so it is possible to make 2^{56} different keys. The main advantage of this system is the speed, but the key distribution, (which has to stay secret at any cost), isn't as simple as with the asymmetric ones. Therefore, in final systems both the asymmetric (for the key distribution) and the symmetric one (for secret data transfer) are used. Although in the USA the algorithm has become standard, because of the 56-bit key, which is considered too small for today's standards it isn't considered to be enough reliable. As the answer to it a *Triple-DES (3DES)* is made, at which the DES algorithm is used three times (encrypt-decrypt-encrypt) with three different keys.

IDEA algorithm (International Data Encryption Algorithm)

A symmetric algorithm type similar to the DES algorithm, except it uses a 128-bit key and the blocks are encrypted in 8 steps. After that follows a bit shift which form a key. It is developed in Switzerland and it is considered to be very safe. Its use for non-commercial purposes is unlimited, and is subject to patent rights.

Blowfish algorithm

It was developed by Bruce Schneier. The algorithm encrypts 64 bit long data blocks. The variable key length is from 64 to 448 bits. It has been more and more used in applications. It is very difficult to break it.

RC4 algorithm

The algorithm is developed by the *RSA Data Security* company with the intention of keeping it secret. However, the source code "leaked". The keys can be of arbitrary lengths, which makes the breaking more difficult. Yet, its reliability is still unknown. It is very fast. RC4 is in fact a pseudo-random numbers generator. It brings the data to the generator exit where XOR (exclusively – or) logical operation is performed and encrypts them. Therefore, the same key cannot be used to encrypt different data.

4.1.2. Asymmetric algorithms

RSA algorithm (Rivest, Shamir, Adelman)

The name of this algorithm was originated from the initial letters of its makers. The asymmetric way of encryption is used both for encryption and electronic signature. The asymmetry arises from the fact that there are two keys needed in communication. One key is used for message encryption and the other one for decryption. The keys can be either public or secret. One of the keys cannot be generated from the other, more precisely, the secret key cannot be created from the public one.

Only one side in communication knows the secret key, which makes the encrypted message very safe. The whole process is based on the factorisation of **big** numbers because the time needed by an unauthorised user for message encryption by means of the "trial and error" method is unacceptable. Simplified, the algorithm for generating public and secret keys is performed in the following way: Two big prim numbers A and B are selected, each of them of more than 100 digits. N is the product of their multiplication. A relatively simple number E is selected in relation to (A-1)(B-1) and number D=(E-1)mod ((A-1)(B-1)) is calculated. The pair (E,N) represents the public key, while (D,N) is the secret one. Numbers A and B are not important for the encryption process itself, but it is essential for them to stay secret because number D is based on them. The source message is then divided into a sequence of whole numbers m_i from the interval 0 to (n-1). A certain number of messages are encrypted according to the formula $c_i=m_{ie}$mod n, and decrypted according to $m_i=c_{id}$mod n.

PGP algorithm (Pretty Good Privacy)

It is the asymmetric encryption, which uses a public and a secret 64-bit key. Because of the security level it offers it is used in a lot of Internet business applications among which electronic trade, electronic banking, electronic stock markets etc. Its disadvantage is slowness of encryption and decryption, so it is used for sending smaller data quantities.

Diffie-Hellman algorithm

This asymmetric algorithm is most commonly used for key exchange. If the keys are long enough as well as a good random number algorithm it can be considered as safe. It is based on the logarithm function features.

4.1.3. One-way *hash* functions and its associated algorithms

One-way *hash* functions are also called compression functions, contraction functions, message summary, imprint, cryptographic data, message integrity check … Out of a variable length sequence they create a fixed length sequence. The procedure is, in general, irreversible, i.e. there are no inversion functions, although it is likely for two different input sequences to result in the same output sequence. The quality of these functions is the fact that it happens very rarely, i.e. 1:1 copy is less possible to happen. Because of their features, they are used for electronic signature generation.

MD5-hash algorithm and RIPEMD –160

MD5 belongs to the *hash* algorithm group. It was developed in *RSA Data Security*. It is the comparison of encryption results; it is not the message decryption itself (inverse messages don't exist). The result of implementation of this algorithm is a 128-bit sequence, always of the same length called m*essage digest*. To establish its authenticity one side sends a coded message and the digest. The other side lets the message go through the same algorithm and the given digest is compared with the received one. The identity of the digests confirms the authenticity of the message. RIPEMD –160 is a more recent algorithm designed to replace the MD5. It comprises the sequences of arbitrary lengths into 20-byte sequences.

SHA (Secure Hash Algorithm)

The US government has recently issued this algorithm. The arbitrary length sequences are comprised into a 160 bit data. It is considered to be a considerably good algorithm.

4.2. Electronic signature

Electronic signature enables

- To prove the authenticity of documents and people who sign the documents,
- Identification of the person who signed the document, message or data, and
- Identification of what has been signed.

The electronic signature mechanism includes the electronic signature creation algorithm and check algorithm that is used for authenticity check of both the message and the sender. The electronic signature is created by source message transformation into a final data by means of an algorithm (*Message Digest Algorithm*), and it is additionally asymmetrically encrypted. The message receiver separates the message itself from the signature, creates the contents of the message, decrypts the signature and compares it with the received message digest. If the receiver succeeds to decrypt

the electronic signature, and if the messages he compares are identical, the receiver can be sure that the message hasn't been altered during the transmission. In the world of computers electronic signature has the same role as in the everyday world – to confirm the authenticity and accuracy of the message. If we want the signature to be really authentic it has to identify only one particular person. It can be achieved if it is encrypted and illegible to the third party in communication, and it is possible by using a sequence or a union of cryptoagorithms. The most common form is the combination of the MD5 *hash* algorithms and RSA asymmetric cryptosystem. The sent message is encrypted with a *hash* algorithm to get a digest. The digest is further encrypted with the RSA algorithm by using a secret key and the result you get is an electronic signature. When receiving a message the signature is separated from the message. A public key decrypts the signature and the message goes through the same *hash* algorithm as before it had been sent. The digest you get is compared with the decrypted signature and confirms the authenticity if they are identical.

According to what is needed to achieve, there are three kinds of electronic signatures: for contents authentication, for user identification, or for both. Different combinations of cryptographic algorithms and functions are used at it.

Contents authentication

In this case electronic signature is needed because it will help the message receiver to establish if the message has or hasn't been changed. At this point we are not interested in its secrecy. Regarding this, as well as the authentication speed of a large number of messages, simple one-way *hash* functions are used and they function as a unique digest. A secret key encrypts the digest.

User identification

Previously described kind of electronic signature often is not satisfactory, because it doesn't keep the message secrecy, and there is still a possibility for the user to deny his sending the message. Provided the requirements are higher, a symmetrical algorithm encrypts the complete message, which mean that both sides included in a conversation are familiar with the unique secret key. When the receiver decrypts the signature, he will get the sender's identification. It is important to mention that there is also a server, trusted by everyone, which contains a secret key base.

Contents authentication and user identification

For the complete data content protection, both message and user authentication and user identification are needed. In that case a combination of an asymmetric algorithm and a one-way algorithm, i.e. compression and usage of a public and secret key. A server is not necessary. The sender uses a one-way function to add the digest to the end of a message. The newly created message is encrypted by a secret sender's key which makes sure that only the receiver will be able to read the sent message. On the other hand, the receiver decrypts the message by his own secret key, sends the message through the same one-way algorithm, and checks the identity of the digest.

5. AUTHENTICATION PROTOCOLS

The need for safe authentication within open distributed computer systems has brought to creation of authentication standards and systems.

Authentication protocols should enable the introduction of a potential user to a system. Messages, which are exchanged between the user and the system, should have the following features: authenticity[1], coherence and integrity as well as uniqueness[2].

Within the user network in which services from a large number of separate servers are required, there are three ways to check the access to services:

1. The computer (to which a user logs in) itself protects the unauthorised access,
2. The user has to confirm his identity, and the access computer believes him
3. The user has to prove his identity for every required service.

[1] The message, which has already been sent, cannot be pronounced invalid afterwards.

[2] The same message cannot be repeatedly sent through the communication channel

A lot of authentication protocols use the authentication server whose primary task is to generate keys and function as a key server (which performs the server personal data exchange).

SSL (Secure Sockets Layer)

SSL is a secure sockets layer, which ensures data encryption, data integrity and authenticity of a server and a client. It supports three ways of authenticity definition:

- mutual authenticity definition
- server authenticity definition,
- server/client anonymity

Authenticity can be achieved by certificates.

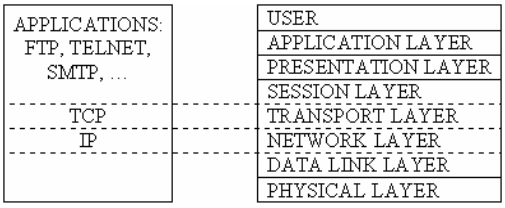

Fig 3. Relation of the *ISO/OSI* model and the *Internet* data communication model

In the ISO/OSI model and in the Internet data communication level model SSL is situated between the application and transfer layer, forming a new layer, independent, but at the same time transparent to other layers.

It consists of the *Handshake protocol,* which enables identification of both the server and the client as well as exchange of the encryption algorithm and cryptographic keys, and of the *Record Layer* protocol which is used for encryption and message transmission. SSL uses the RSA algorithm; more precisely it uses the double asymmetric encryption and decryption keys technique. To protect the communication from a possible external modification MAC (*Message Authentication Code)* is included in the protocol. It is a data generated from the secret key and message. Namely, it is difficult to generate a new message with the same MAC, and it is difficult to apply the inversion method for getting the encrypted message back.

To enable the communication between two sides (client and server) they have to support SSL (Secure Socket Layers) protocol i.e. they have to possess certain certificates issued by a certificate authority. The certificate is a set of essential information about the user and the organisation it belongs to. Essential information includes public keys, too.

The information exchange is performed in the following way: interlocutors use a symmetric key in their communication for faster encryption and decryption of the messages they use. Asymmetric protocol is used for the symmetric key distribution. Each side electronically signs the encrypted information, which additionally protects its authenticity and integrity. This protocol is mainly used in web communication.

SET (Secure Electronic Transaction)

SET is an open protocol, which enables transactions through Internet by using credit cards like Visa and MasterCard. It was originated as a joint project of Netscape, Microsoft, Visa and MasterCard. To be able to use the SET, corresponding programs on both sides are required, i.e. on both the server and the client side.

SSH (Secure Shell)

SSH is a protocol, which when used on *Unix,* enables safe logging in and performing instructions on another network and file transfer between computers. It establishes and keeps encrypted connection between the SSH user programme and the SSH server, it checks both user and server authenticity using some of the available encrypting algorithms, such as RSA. What's more, the protocol protects from some forms of masking and transformation, such as IP address modification, which unauthorised users use for unauthorised access to remote computers.

Kerberos protocol

It would be more convenient for the user to log in and authenticate to the system only once, and then within the protected system at more servers if necessary, for the tasks he is authorised for. To make it possible the user has to use his authentication data all the time. Kerberos technology/system/protocol has successfully solved the problem. What's more, it is possible to use it with all operating systems. It has become a standard for establishing authenticity in distributed client/server environments. It is designed for TCP/IP protocol based networks.

Kerberos offers three different protection levels: Application programmer defines which one is the most suitable one, regarding application requirements:

- applications require authentication immediately after the connection is established

- applications require authentication of every message, no matter if the content of the message is exposed or not,
- applications require safe messages.

Higher security level is achieved by private messages, where each message is authenticated and encrypted. Kerberos server itself uses private messages for sending passwords through the network. Encryption is in Kerberos based on the DES algorithm. There is a possibility of several encrypting methods, with a possibility to choose between their speed and security.

Administration server (KDBM server) is connected with the database by a network interface. The client can start the program form any network computer. The administrator server can be started from the computer which contains the Kerberos database to be able to make database changes. In case a change in the Kerberos database is needed, a new login and authentication to KDBM server is needed.

Authentication server or Kerberos server performs read only operations on the Kerberos database. This server doesn't modify the Kerberos database, so it can be started from the computer with the Kerberos database replica.

Kerberos maintains the database of its clients and their private keys. A private key is a big number known only to the Kerberos server and to the client it belongs to (it is an encrypted password). Kerberos can create messages to confirm the identity of a client. It also generates temporary secret keys called *session keys*, for two clients talking to each other, to encrypt messages between two sides.

There are two confirmation types in the Kerberos authentication model: tickets and authenticators. They are both generated by private key encryption, but they are encrypted my means of different keys. A **ticket** is used for a safe passage between the authentication server and end server. The ticket also delivers information, which can be used to check whether the person who uses the ticket is the same person to whom the ticket was issued. **Authenticator** contains additional information, which, when compared with those on the ticket, prove that the client who uses the ticket is the same person to whom the ticket was issued. Authenticator can be used only once and must be generated every time when the client wants a certain service. Authenticator is encrypted by a session key, which is a part of a ticket.

The authentication process is performed in the following way[1]:

- the user logs in using his user name,
- the user name is transmitted to the AS authentication server,
- if there is an AS user in the base, a session key is assigned to him, and the user will use it for communication with the server in order to get a ticket encrypted by a private user key from the base,
- the user enters the password which transforms into a DES key and becomes a private user key,
- the answer from the AS is decrypted by the generated private key,
- a ticket, session key and some data about the user are stored, while the password and the DES key are erased from the memory,
- the user (sooner or later) needs a service,
- he asks for the application server ticket from the TGT (T*icket-Granting Ticket*), so the authenticator and server name are encrypted by a session key and proceeded to the TGT,
- TGT generates a new session key for the application server the access to which was demanded by the user.

SESAME protocol

SESAME is the acronym of the *Secure European System for Applications in a Multi-vendor Environment* i.e. European security system for applications in multi-user environment. At the same time, it is the European research project financed by the European committee, and the name for the technology generated from that project, which enables authentication, access control, and cryptographic data protection exchange in network (unsafe) environment. SESAME is in fact an architecture which consists of finished components which can be built into application wherever it is needed..

[1] All communication among different servers is encrypted by secret keys familiar only to them

The project owes its existence to American laws, which ban the export of high cryptographic technology. In order to be able to spread the Kerberos technology to Europe, the banned parts had to be replaced. Yet, the compatibility with the Kerberos has been retained. Moreover, the protocol has some additional security parts. E.g. it also supports the D*irectory Name* standard which was originated because SESAME supports public keys and certificates, uses all three kinds of cryptographic (symmetric, asymmetric and irretrievable) algorithms which can be intensified or weakened if needed, according to the user's needs and according to the current legal regulations. Furthermore, the protocol uses two types of keys (base and dialog), and supports two single log in methods:

- Kerberos authentication mechanism based on passwords
- Authentication method which uses asymmetric cryptography.

SESAME uses up to seven servers: Authentication Server (AS), Privilege Attribute Server (PAS), Key Distribution Server (KDS), Domain Security Server (DSS), Certification Authorisation (CA), Certification Authority Agent (CAA), and Local Registration Authority (LRA).

Identity is a real world feature used by a user to log in a server, to access protected services, to show data ownership, etc. SESAME architecture supports authenticated identity (*Authenticated*), access identity (*Access*) and responsibility identity (*Audit*). The identity supports the Kerberos syntax.

6. ELECTRONIC CASH PROTOCOLS

When thinking about the problem of paying by electronic cash we can come to the conclusion that there are three parties involved: a buyer, a seller and a bank. There are various payment modes: electronic cheques, debit cards, credit cards, cards with stored value, electronic cash ... In the future all these payment modes can be replaced by a so-called smart card. Every kind of payment should consist of the following main components: *authenticity* (impersonation protection), message *integrity* (unauthorised change protection) and *impossibility to deny* the transaction, which has already been performed, and sometimes, privacy (unauthorised reading protection).

We have been used to consider the process of shopping in the following way: A customer needs a certain product/service. He finds a seller who sells him exactly what he needs. After some additional questions regarding the price, the customer withdraws money from his bank account, pays the seller who puts it to his bank account. Such information flow through Internet is not safe if it is not protected. Some options, which arise from such transactions, are the following:

- the customer doesn't receive the electronic banknote although the amount is reduced from his bank account,
- the received banknote is invalid,
- the value of the received banknote has been changed,
- the customer himself changes the value of the banknote,
- the seller doesn't receive the banknote
- the seller receives the banknote from the wrong person,
- the seller receives a false banknote,
- the banknote is copied by the third party,
- the bank doesn't receive the banknote from the seller ,
- the value of the received banknote has been changed …

The main characteristics every electronic payment system should have should be as similar to those in real life as possible:

- Easy transmission,
- Easy to change,
- Directness (when buying/selling the third party is not needed),
- Easy to recognise (visual identity),
- Customer anonymity
- It's hard to monitor the cash flow

By authentication i.e. by implementation of authentication protocols, all except the last two characteristics can be found at electronic payment, too. The validity of an electronic banknote has to be confirmed by the electronic signature of the institution which issued it. In order to avoid a multiple usage of the same banknote, its usage has to be restricted to only one transaction.

Theoretically there are two protocol groups: with and without anonymity. The shopping procedure if the **protocol without anonymity** is used can be divided into a few stages: Firstly, the communication between a buyer and a seller is established. The buyer chooses desired articles and asks for a kind of a pro forma invoice in order to get the information about the amount needed to withdraw from the bank account. The bank forms an electronic banknote of a certain value and put its electronic signature on it, according to which its authenticity will be confirmed later. The banknote serial number, which will be later used to check its authenticity, is stored in its database. Such a "signed" banknote is sent to the buyer and the amount on his account is reduced. The buyer sends the same banknote to the seller who forwards it to the bank. The bank confirms its authenticity and sends the seller a receipt of its validity. The amount on his account is increased and the seller can send the goods to the buyer.

The customer anonymity in such a system is not guaranteed. It was necessary to build a protocol to guarantee it. The situation with **protocols with anonymity** is different. In order to avoid the possibility of monitoring the transaction from more sides, the buyer uses the method of masked identity. This complicated protocol is an enhanced version of the base protocol and requires more dialogues from both sides. The protocol with anonymity starts in the same way as the base protocol. More important differences can be noticed when it comes to payment requests. Namely, a buyer sends N encrypted banknotes of the same value to the bank for authentication. The bank sends the request for encryption keys (N-1) of randomly picked banknotes to check their value. It puts the signature on one of the intact banknotes according to which they will later check its authenticity. In that way the buyer's anonymity is guaranteed. Although the real banknote value is not known, the chances for fraud are (1/N) very small because N is a big number. After the amount of the seller's bank account has been reduced, the bank sends the signed banknote back to the buyer. The buyer forwards it to the seller who checks the bank signature. After the authenticity has been confirmed (the banknote is not forged), the buyer sends the sequences to confirm his identity. The sequences give information about the buyer. The seller compares the sequence digests obtained after the hash function has been performed and those received with the banknote. If they are identical, the data are then sent to the bank. The bank checks the banknote serial number. If it has been already used the halves of the buyer's sequences are matched and the identity is recovered, not to forget that one half is sent by the seller together with the banknote, and the other one is currently kept in the bank base. If the sequence halves matched it would directly show to the fact that the seller has been tried to use the same banknote twice.

6.1. Commercial protocols

CyberCash

This protocol is based on a 768-bit RSA cryptographic algorithm, which ensures the undisturbed transaction flow. To use it, a buyer has to own *The Wallet* program and a seller has to have an account with a credit card company and his own bank ID. The protocol itself can be described in the following steps: Firstly, the communication between a seller and a buyer has to be established, when the buyer chooses articles, and the seller sends a receipt. After that, the buyer sends this encrypted message to the *CyberCash,* which decrypts the message and sends it to the seller's bank, any other way except Internet can be used. When the transaction is successfully processed, the bank returns the information to *CyberCash,* which informs the buyer about the successfully performed transaction.

First Virtual

Today's payment modes are mostly credit cards based. This protocol is developed in order to perform such transactions via the network. But, unlike other protocols, which use the Internet as a data transfer medium, this protocol uses safe and solid bank network. Before starting the transaction a buyer has to contact his *First Virtual* bank to get his own *VirtualPIN* which will function as his identificator. The buyer, if he wants to log in or ask for certain services, has to present his *VirtualPIN* to the server for identification and check. The server contacts the *First Virtual* bank and sends all transaction information to it. The buyer receives an e-mail to confirm if he wants this transaction to be performed. This e-mail has its additional role: additional authenticity check of the buyer. After that the money is transferred to the *First Virtual* server account and it debits the buyer's credit card. The server will be granted the access to it only after some time because of the possible further complaints by credit card companies or buyers.

e-Cash

This protocol concept is based on so-called "electronic coins" i. e. a sequence of characters with its nominal value, serial number given by the bank and its electronic signature. They are used as base payment units in transactions. In case there aren't enough smaller coins, the buyer demands from the bank to change a bigger one into two smaller ones, and the value of one of them will be identical to the amount which should be paid. The protocol starts with confirming the buyer's request for a product. The final result is reduction of the number of coins on the buyer's disk for the paid amount. The coins can be stored or withdrawn from the account at any time, and all the transactions are recorded in order to make keeping files easier.

Secure Pay

This method uses cheques as a payment mode, and not credit cards. For this protocol everything you need is the buyer's account in the bank which accepts cheques payable in US$. The buyer chooses articles and enters his *Secure Pay ID* and the previously given code. This information is forwarded to *Redi-Check company,* where this protocol was created. After the authorisation, a cheque with the buyer's data and the spent amount is printed and sent back to the server by regular mail. The server gets the money 24 hours after the purchase.

7. SUMMARY

Identification, authentication and authorisation are the main prerogatives for the data contents security and for the safe communication among the users of the open distributed systems, as well. These measures are performed on the physical and logical level, and special attention is paid to communication between the Internet and intranet. The implementation itself, without adhering to security principles important for their introduction and use, does not guarantee security – neither of the data contents in database, or the one in communication.

Data contents should be protected by means of cryptographic algorithms, which complexity and speed depend on the required security level. The example of this is an electronic signature, the means of accuracy confirmation and message credibility. Authenticity protocols, even complete authentication systems were generated as the means of cryptographic algorithm implementation, security methods and principles (of data contents) in dynamic and distributed systems, which becomes very important in the case of electronic financial transactions, i.e. electronic cash payment protocols.

BIBLIOGRAPHY

1. *Elektronički oblici plaćanja*, group of authors, available on the URL: www.rasip.fer.hr/ecash, January 2002.
2. ftp://ftp.rsasecurity.com/pub/labsfaq/rsalabs_faq41.pdf
3. *Sigurnost i šifriranje*, dostupno na Internet adresama: http://www.carnet.hr/vodic/cro-gnrt/security/encryption.html, http://www.carnet.hr/vodic/cro-gnrt/security/pgp.html, January 2002.
4. eCash Technologies, Inc. http://www.digicash.com/Solutions/, January 2002.
5. Aida-inženjering, *Pametne kartice i sigurnost*, available on the URL: http://www.aidia-i.ba/bih_smartcard/bos_smartcard3.html, January 2002.
6. Luka Baranović: *Protokoli plaćanja elektroničkim novcem*, available on the URL: http://sigurnost.zemris.fer.hr/emoney/baranovic/, January 2002.
7. Dubravko Gorupić: *Zaštita podataka prilikom plaćanja elektroničkim novcem*, available on the URL: http://sigurnost.zemris.fer.hr/emoney/gorupic/, September 2000.
8. Miroslav Alković, *Sigurnosni bankarski sustavi*, available on the URL: http://sigurnost.zemris.fer.hr/emoney/alkovic/, January 2002.
9. Biljana Nekić: *Prilagodba programa sigurnosnom sustavu KERBEROS*, available on the URL: http://sigurnost.zemris.fer.hr/protokoli/KERBEROS/nekic/, February 2000.
10. Ivo Lukač: *SESAME*, available on the URL: http://sigurnost.zemris.fer.hr/protokoli/SESAME/lukac/, 2000/2001.
11. HD-info home page: http://www.hdinfo.hr/index.html, January 2002.
12. *International PGP Home Page* http://www.pgpi.org/, January 2002.
13. Free Secure Shell Client for Windows 95/NT 4.0, dostupan na Internet adresi: http://www.massconfusion.com/ssh, January 2002.
14. *ISO/IEC 17799 – 2000* (*BS 7799 – 2000 Part 1*) – Međunarodna norma za izgradnju sigurnosti IS-a