BULGARIAN ACADEMY OF SCIENCES

PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS • 2022 • Vol. **78**, pp. 5-22
p-ISSN: 2738-7356; e-ISSN: 2738-7364
https://doi.org/10.7546/PECR.78.22.02

An Approach for Encrypted Exchange of Information in Corporate Networks Based on Tcl/Tk

Ani Boneva, Yordanka Boneva

Institute of Information and Communication Technologies,
Bulgarian Academy of Sciences, Sofia, Bulgaria
Emails: ani.boneva@iict.bas.bg, yordanka.boneva@iict.bas.bg

Abstract: Ensuring information security is crucial for corporations with a complex, geographically distributed structure at many levels: large banks, multinational and state-owned companies. Often the corporate networks of such organizations are built with equipment from different generations and from different manufacturers, which significantly complicates the process of managing the IT system. The article presents the elements of the security system used to protect information in a distributed corporate network. Various elements (firewalls, cryptographic algorithms, anti-virus programs, etc.) and company products, which are the main units in building corporate security systems, are described. Special attention is paid to public key cryptography and a software implementation Coder 1.0 (developed on Tcl/Tk) is proposed, implementing encrypted data exchange inside of corporate networks and between corporate networks.

Keywords: *Tcl/Tk, Corporate networks, Cryptography, TCP/IP, Public Key Encryption, Coder 1.0*

1. Introduction

The use of corporate information systems as an active communication and service environment and a tool for business, services and production, requires security of information transmitted between local computer networks and its protection against destruction and unauthorized access.

Corporate security policy consists of developing and approving general and private security rules and procedures for individual users, departments and/or offices, as well as rules and mechanisms of interaction between them, including [1, 2]:

- Determining the accessible resources and services;
- Access control;
- Rules for local and remote identification and authentication;
- Use of cryptographic methods and devices for data protection and access control;
- Protection against viruses, etc.

The use of cryptographic methods and algorithms for data protection has a key role in building a security information system. New software products for encrypting information (letters, documents, databases, etc.) have been developed [3, 4]. They allow the secure storage of data on corporate computers, as well as data transmission by e-mail, TCP / IP or other transmission media [5, 6]. Various tools are also used to pre-model and simulate different security information systems [7].

The software solutions used are based on various standardized cryptographic algorithms described in detail in [8, 9, 10].

2. Elements of the security system of distributed corporate network

2.1. Identification methods

The identification methods are used to ensure strictly personalized access to the resources of the corporate network. This group includes: Touch memory, Smart cards, passwords, biometric data, etc. [14, 15].

The identification and authorization of a user by his physical characteristics (face, fingerprint, voice or iris) is used in corporate security systems. The advantage of these methods is the impossibility of duplicating the identification elements, but their mass application is still difficult, both due to the high prices of hardware and software and a number of unresolved legal and ethical issues.

2.2. Firewalls

Firewalls are an important element in building a security system in a corporate network. Their main function is to separate PCs from the Internet by monitoring data packets (both inbound and outbound) and determining whether to let them in or block them. They ensure that packets that meet certain security rules (defined by the user or administrator) are admitted to the computer. Firewalls work in both directions, accepting or rejecting all messages based on lists of acceptable and unacceptable sources and ports [2].

2.3. Cryptographic algorithms and products

In general, cryptographic algorithms are divided into symmetric and asymmetric [8, 13].

➢ **Symmetric algorithms:**

- ***DES (Data Encryption Standard)*** - is the name of FIPS (Federal Information Processing Standard), describing the algorithm DEA (Data Encryption Algorithm). It was originally developed by IBM and was called Lucifer, later included NSA and NIST [11, 12, 13].

This algorithm has been the standard for about 20 years. The data is encrypted in 64-bit blocks, the key is 56-bit.

- ***Triple DES*** - Developed as a replacement for DES. Also considered insecure (because 56-bit keys are already vulnerable) [8, 13].
- - ***Blowfish*** - encryption algorithm with variable key length (from 32 to 488 bits). It was developed in 1993 by Bruce Schneier. It is faster than DES and IDEA [8].
- ***IDEA (International Data Encryption Algorithm)***- encrypts 64-bit blocks with a 158-bit key. The decryption process involves eight complex stages [8].

It is designed for embedding in hardware and software. Its hardware application is more common.

- ***AES-(Advanced Encryption Standard) -*** it is based on the Rijndael block algorithm and was completed in October 2000. The key length varies from 128 bits, to 256 and higher. The standard is described in detail in [5, 8, 13].

The described algorithms are developed for embedding in hardware and software.

➢ **Asymmetric algorithms:**

- ***RSA*** (Rivest–Shamir–Adleman)-developed in 1977, it offers not only encryption but also user authentication. Like any asymmetric algorithm, it works with two keys (public and private). This allows any user to send an encrypted message, but only the owner of the correct private key will be able to decrypt it [8].
- ***Elliptic-curve cryptography (ECC)*** – elliptic-curve cryptosystems can be classified according to whether they are analogs of RSA or discrete logarithmic systems. They are analogous to existing public key systems, in which modular arithmetic has been replaced by operations defined on elliptic curves. Today, methods for calculating discrete elliptic curves

are less efficient than those based on discrete logarithms. As a result, shorter key lengths can achieve the security of conventional cryptosystems, leading to lower system requirements and improved performance [8].

Public key cryptography is based on asymmetric algorithms. A detailed comparison of the capabilities of the described algorithms is made in [10].

3. Public key cryptography

3.1. Main characteristics

Public key cryptography and related standards and techniques underlie the protection of many Internet-related products, including signed and encrypted e-mail, form signing, one-time password entry to access all available resources, and the SSL protocol (Secure Sockets Layer) [8].

All communications on the Internet use the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite. This protocol allows information to be sent from one computer to another over multiple intermediate computers and different networks before reaching the recipient. The great flexibility and convenience of TCP / IP leads to its adoption worldwide as a basic protocol for communication on the Internet and Intranet. At the same time, the fact that TCP / IP allows information to pass through intermediate computers allows third parties to interfere with communications in the following ways [2, 9]:

- **Eavesdropping** – the information remains unchanged, but its confidentiality has been violated;
- **Forgery –** the information during the transport is changed or replaced and then sent to the recipient;
- **Pretending –** the information is transmitted to a recipient, who pretends to be the real recipient.

Usually, very sensitive personal and corporate communications on the Internet require caution, which is aimed at the threats mentioned above. Fortunately, many well-established techniques and standards (known as public key cryptography) make such precautions easy and convenient.

Public key cryptography has the following characteristics:

- **Encryption and decryption** - allows both communicating parties to hide the information they exchange;
- **Recognition of forged information** – allows the recipient of the information to check that it has not been altered during transport. Any attempt to change or replace data leads to the appearance of a message about the lack of authenticity of the information;

- **Authentication** – allows the recipient of the information to establish its source, i.e. to confirm the identity of the sender;
- **Non-rejection** – makes it impossible for the sender of the information to later claim that the information was not sent.

Encryption is a process of transforming information so that it becomes meaningless to everyone but the recipient for whom it is intended. Decryption is a process of transformation of encrypted information, in which it becomes understandable again. A cryptographic algorithm, also called a cipher, is a mathematical function used to encrypt or decrypt. In most cases, two interrelated functions are used, one for encryption and one for decryption [9, 16, 19].

In modern cryptography, the ability to keep encrypted information secret is based not on a cryptographic algorithm that is widely known, but on a number called a key that must be used in the algorithm to encrypt or decrypt information. Deciphering with the right key is easy and fast. Deciphering without the right key is very difficult and in some cases impossible for any practical purpose.

In cryptography, a message that will be kept secret will be encrypted using an algorithm. Messages that have been encrypted are called plain text, and messages that have been encrypted or are encrypted are called encrypted text. The process of converting plain text to ciphertext is called encryption, and the process of recovering plaintext from ciphertext is called decryption.

Encryption and decryption are functions of transformation between these sets. If the plaintext elements are denoted by M, the elements of the encrypted text are denoted by C, while for encryption the process is denoted by E, deciphered with the notation D. The mathematical notation of this process is [19]:

- Encryption: $E(M) = C$
- Decryption: $D(C) = D(E(M)) = M$

With a block diagram, the encryption and decryption process can be described as follows (Fig. 1):

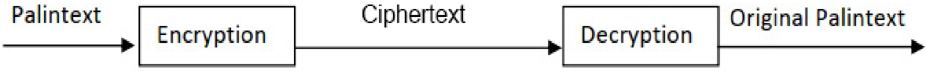


Fig. 1. Block Diagram Encryption and Decryption process [19]

Two types of encryptions are used:

- symmetric encryption,
- public key encryption.

In symmetric encryption, the encryption key can be calculated from the decryption key and vice versa. Most symmetric algorithms use the same encryption and decryption key. It is effective only if the symmetric key is kept hidden by both parties involved in the communication. Detailed information is given in [20].

Public key encryption (also called asymmetric encryption) involves a pair of keys (public and private key) associated with a person who must verify their identity electronically or sign or encrypt data.

The asymmetric algorithm in the encryption and decryption process can be symbolized mathematically and is described as follows [19]:

- Encryption: $EK1(M) = C$
- Decryption: $DK2(C) = DK2\ (EK1\ (C)) = M$

The public key is denoted by $K1$ while the private key is denoted by $K2$. Each public key is published while its corresponding private key is kept hidden from the owner.

Figure 2 shows a simplified view of the operation of public key encryption.

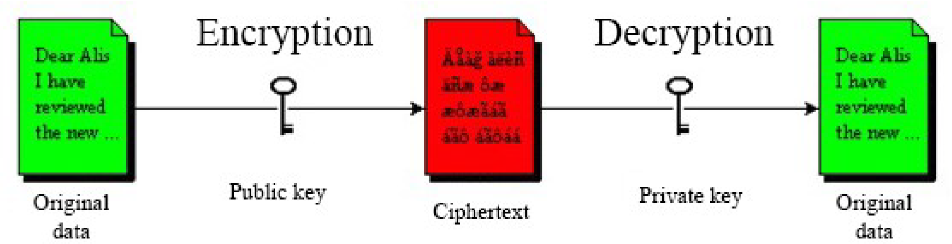


Fig. 2. Public key encryption

As can be seen from the figure, the public key can be known by everyone, but only the owner of the private key will be able to read the data encrypted with it. The most widely used public key encryption implementations are based on algorithms patented by RSA Data Security.

Compared to symmetric encryption, public key encryption requires more calculations and is therefore not always suitable for large amounts of data. However, it is possible to use public key encryption to send the symmetric key, which can be used to encrypt other data [12, 16].

3.2. Main characteristics

3.2.1 Option 1

Recently, experiments have been made in the field of merging software-defined networks (SDN) and Wi-Fi networks into software-defined wireless networks (SDWN). However, none of the proposed architectures allows protection of the wireless connection part of the network. In [4] a specialized EnDeC component that can encrypt and decrypt wireless network traffic in SDWN using the WPA2 security standard is presented. It also communicates with an SDN controller to maintain centralized network management. The EnDeC component downloads an encryption feature and decrypts the wireless access point and improves network performance. Fig. 3 shows an example network.

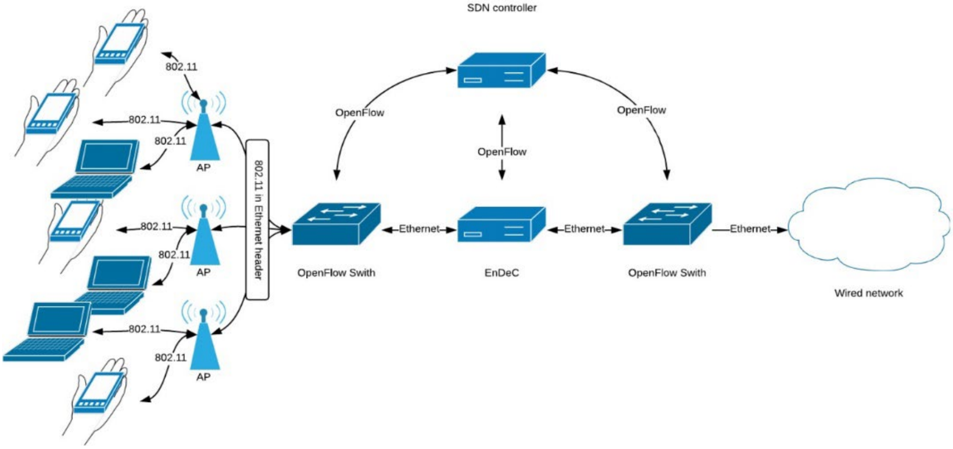


Fig. 3. EnDeC component in SDWN [4]

In order to understand this topic, we need to understand key technologies: SDN and Wi-Fi. SDN architecture enables centralized and agile control of the network by control plane with the SDN controller. This controller is managing the forwarding part of the network where data is transferred. Also connects the application layer via dedicated APIs [10]. Wi-Fi represents a wireless LAN based on the IEEE 802.11 standard protocol, which grants Internet resources to connected devices. It is developed by Wi-Fi alliance [4]. Following solutions of SDWN and their variations are the most popular in the topic of merging SDN and Wi-Fi.

3.2.2. Option 2

A Wireless Sensor Network (WSN) plays an important part in the growth of various applications such as healthcare, the military, industrial surveillance, etc. In this selforganized network, Sensor Nodes (SNs) with limited energy, storage and computational capabilities are randomly distributed. The SNs monitor different factors, which are wind, humidity, temperature, etc., and then forward the data to the Base Stations (BSs).

A blockchain-based encryption and trust assessment model is proposed in which the identities of aggregator nodes (ANs) and sensor nodes (SNs) are preserved. The authentication of AN and SN is performed in public and private blockchains, respectively.

Fig. 4 shows a model of Option 2. The steps included in the proposed model are initialization, registration, authentication and trust evaluation of the nodes.

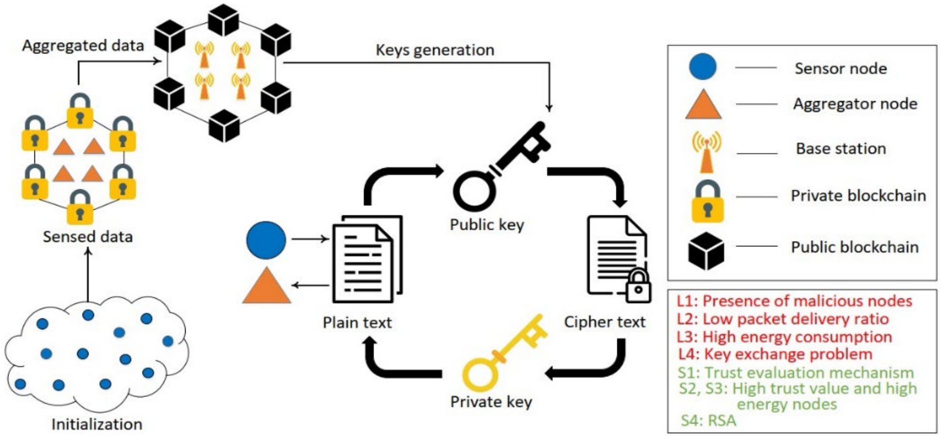


Fig. 4. Proposed system model [5]

The proposed secure blockchain routing mechanism for WSN is RSA-based and is used to encrypt and decrypt data packets for secure routing. A more detailed description of the technology is given in [5].

4. Software package for coded information exchange Coder 1.0

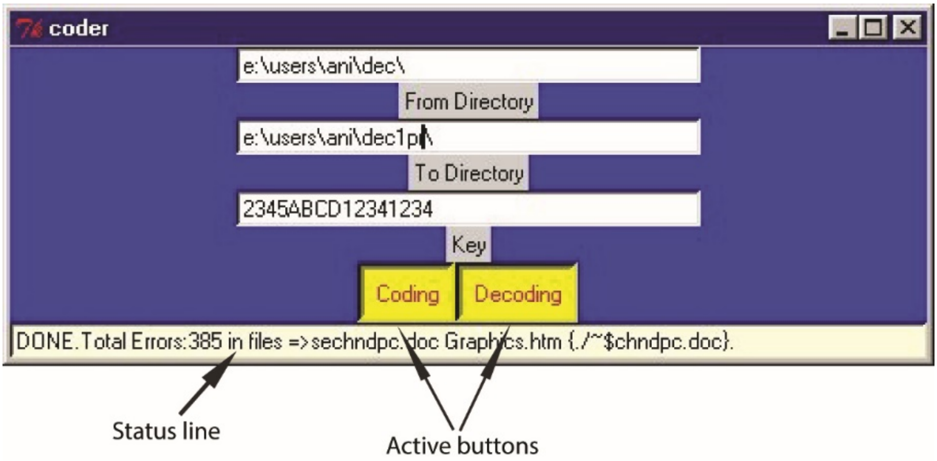
As part of research in the field of encrypted information exchange in corporate and sensor networks, a software package (Coder 1.0) was developed, with which experiments were performed within developed systems in the field of energy, laparoscopic surgery and data collection system sensors, which are described in [23, 24, 25].

For its development a scripting language was used, Tcl/Tk. The package is designed to encrypt / decrypt information contained in a user directory [20, 21, 22]. It supports the DES algorithm with a 64-bit encryption key and has built-in control to verify the integrity of the information [1]. The package can run under Operating Systems: Windows and Linux.

The internal presentation of the information is in hexadecimal format, which makes the output files convenient for transfer via TCP / IP. It supports user-friendly interface.

To transfer encrypted information, the program must be installed on the computers of the sender and recipient. It is necessary for the parties exchanging information in advance to know the encryption / decryption key (which can be sent by e-mail or otherwise).

The encryption key is a 16-digit hexadecimal number (any combination from 0 to F). The control panel of the program contains three fields for entering information (From Directory, To Directory, Key) and two active buttons (Coding and Decoding). In addition to the three fields, the program also shows the decoding result, as shown in Fig. 5.



e:\users\ani\dec\

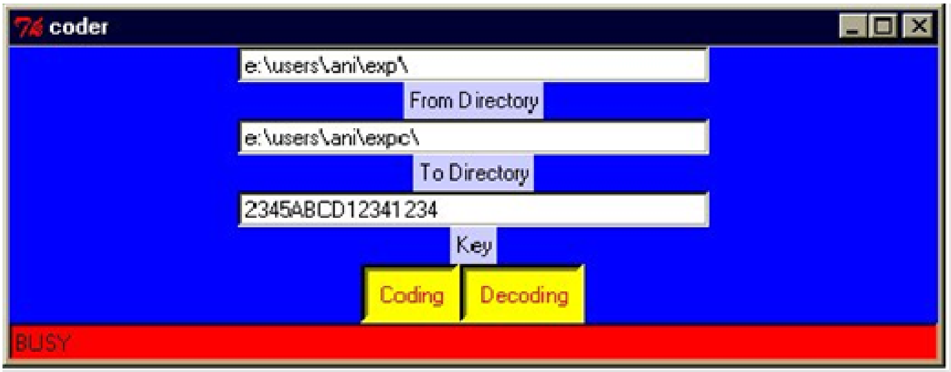
e:\users\ani\dec1p\

2345ABCD12341234

DONE.Total Errors:385 in files =>sechndpc.doc Graphics.htm {./~$chndpc.doc}.

Fig. 5. Working screen of Coder 1.0

When encrypting information in a directory, the sender enters the name of the encryption directory in the From Directory field, and the name of the directory containing the encrypted files in the To Directory field. Enter the encryption key in the Key field and press the Coding key (Fig. 6). In this operation, each line of a file is encoded separately (up to CR).



e:\users\ani\exp\

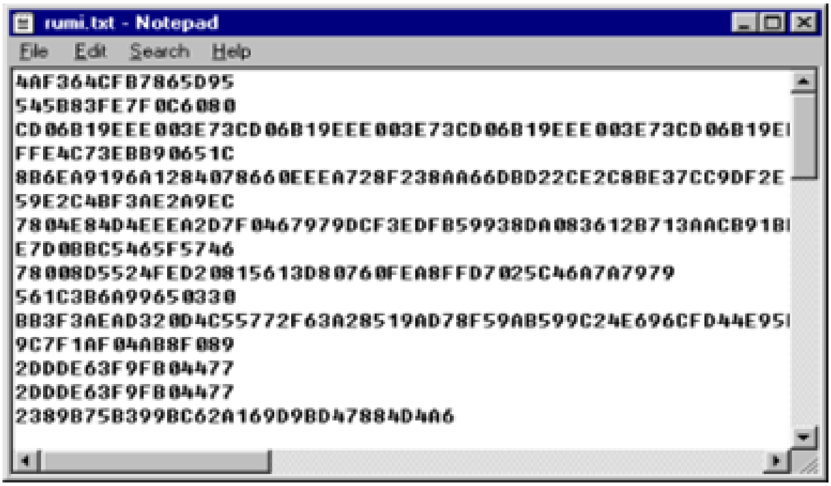
e:\users\ani\expc\

2345ABCD12341234

BUSY

Fig. 6. Encryption operation

Fig. 7 shows the type of encrypted file with Coder 1.0. The sender's information encrypted in this way can be sent to the recipient by e-mail or directly via TCP/IP.



```
4AF364CFB7865D95
545B83FE7F0C6080
CD06B19EEE003E73CD06B19EEE003E73CD06B19EEE003E73CD06B19E
FFE4C73EBB90651C
8B6EA9196A1284078660EEEA728F238AA66DBD22CE2C8BE37CC9DF2E
59E2C4BF3AE2A9EC
7804E84D4EEEA2D7F0467979DCF3EDFB59938DA083612B713AACB91B
E7D0BBC5465F5746
78008D5524FED20815613D80760FEA8FFD7025C46A7A7979
561C3B6A99650330
BB3F3AEAD320D4C55772F63A28519AD78F59AB599C24E696CFD44E95
9C7F1AF04AB8F089
2DDDE63F9FB04477
2DDDE63F9FB04477
2389B75B399BC62A169D9BD47884D4A6
```

Fig. 7. Internal presentation of information encrypted with Coder 1.0

Upon receiving the encrypted information, the recipient enters the name of the received directory in the From directory field, and in the To Directory field indicates the directory in which it will be decoded (Fig. 6). Then in the Key field it is necessary to enter a valid key and activate the Decoding button. After decoding, information about the result of the performed operation appears in the status line (Fig. 6). If the key is invalid or an attempt is made to manipulate the information externally, a message is displayed in this field about the manipulated files and the number of incorrectly decoded lines (for each of them). An error message (CSError) is written to each incorrectly decoded line (on a given file).

An option for encoding / decoding a string (arbitrarily long line) as a Tcl command (SCRIPT) has also been implemented. This option is convenient for use in user programs (for encrypting texts with arbitrary content). The SCRIPT command can be used to encrypt files with * .txt, * .doc, * extensions. html or databases, with no limit on the size of the file. When decoding the information, if there is a discrepancy with the coding key (or an attempt is made to manipulate the data), a Tcl error message is generated, which can be worked out with specialized operators (CATCH, etc.) in the language [18]. In this way, the integrity of the data is controlled.

5. Conducted experiments with the software package Coder 1.0

Two types of experiments were performed: with the Coder1.0 program and with the SCRIPT command (built into the software products described in [23, 24]).

Fig. 8 shows the experimental setup for testing the capabilities of Coder1.0.

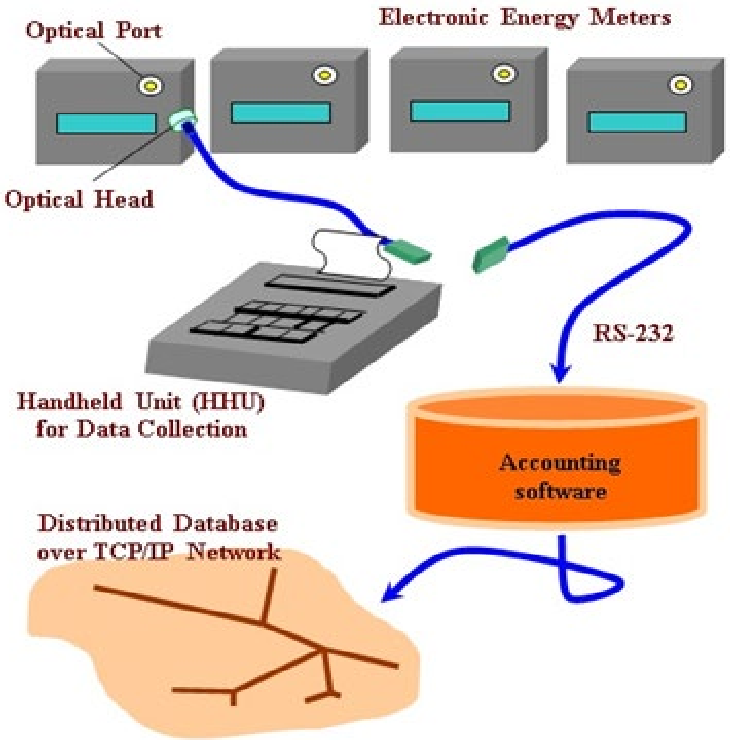


Fig. 8. Block diagram of the experimental setup [24]

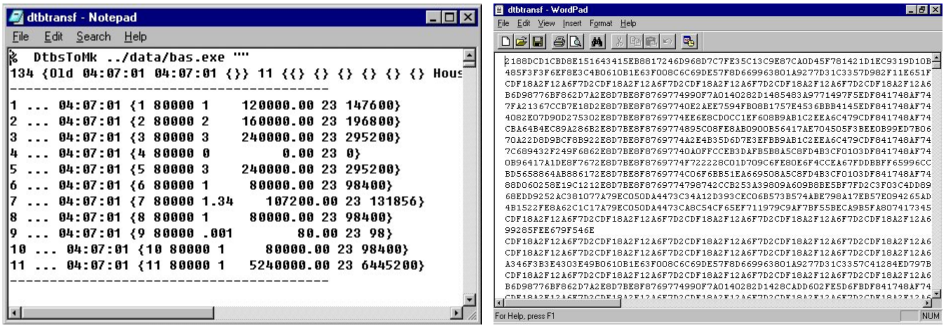
The experiments were conducted within the "Information system for accounting, visualization and analysis of energy consumption" described in detail in [24].

The used algorithm DES (at the request of the applicant) implemented in Coder1.0 and built into the software package Accounting (developed on Tcl / Tk). Encrypted exchange is used when transferring TCP / IP information from local to the central server, as well as when storing it in local databases. The terminal has the ability to read and accumulate data for up to 2000 electric meters. The collected information is accumulated in local databases (in the local servers of the system) and transferred to a central server [24].

With Coder1.0, various experiments were performed to encrypt / decrypt without manipulation (with manipulation) of files with extensions * .txt and *

.html. Files in such formats have been selected, as the program was developed in order to protect the service and service information to the information systems Fig. [23, 24]. The experiments were performed under the Windows operating system.

Figures 9 (a, b) and 10 (a, b) show the results of an encryption operation on * .txt and * .html files. When decoding the files shown in Fig. 9 (b) and Fig. 10 (b) the original files are obtained (Fig. 9a and Fig. 10a).



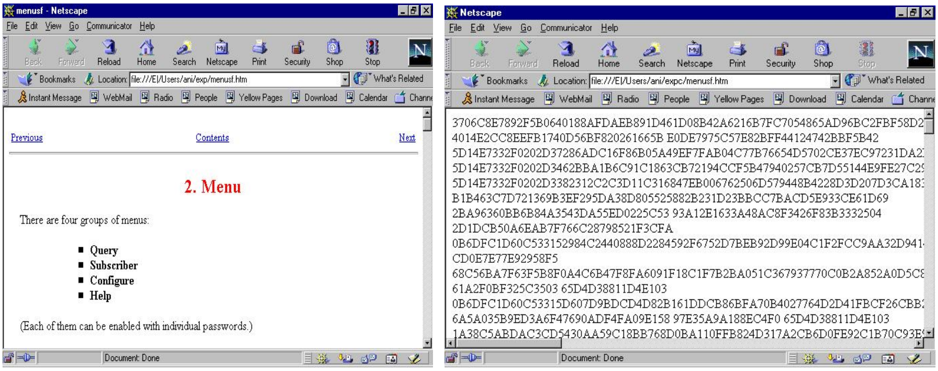
```
%  DtbsToMk ../data/bas.exe ""
134 {Old 04:07:01 04:07:01 {}} 11 {{} {} {} {} {} {} Hous
----------------------------------------

1 ... 04:07:01 {1 80000 1     120000.00 23 147600}
2 ... 04:07:01 {2 80000 2     160000.00 23 196800}
3 ... 04:07:01 {3 80000 3     240000.00 23 295200}
4 ... 04:07:01 {4 80000 0          0.00 23 0}
5 ... 04:07:01 {5 80000 3     240000.00 23 295200}
6 ... 04:07:01 {6 80000 1      80000.00 23 98400}
7 ... 04:07:01 {7 80000 1.34    107200.00 23 131856}
8 ... 04:07:01 {8 80000 1      80000.00 23 98400}
9 ... 04:07:01 {9 80000 .001        80.00 23 98}
10 ... 04:07:01 {10 80000 1      80000.00 23 98400}
11 ... 04:07:01 {11 80000 1   5240000.00 23 6445200}
----------------------------------------
```

```
2188DCD1CBD8E151643415EB8817246D968D7C7FE35C13C9E87CA0D45F781421D1EC9319D10B
485F3F3F6EF8E3C4B0610B1E63F008C6C69DE57F8D669963801A9277D31C3357D982F11E651F
CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6
B6D98776BF862D7A2E8D7BE8F8769774990F7A0140282D1485483A9771497F5EDF841748AF74
7FA21367CCB7E18D2E8D7BE8F87697740E2AEE7594FB08B1757E4536BBB4145EDF841748AF74
4082E07D90D275302E8D7BE8F8769774EE6E8CD0CC1EF608B9AB1C2EEA6C479CDF841748AF74
CBA64B4EC89A286B2E8D7BE8F8769774895C08FE8AB0900B56417AE704505F3BEE0B99ED7B06
70A22D8D9BCF8B922E8D7BE8F8769774A2E4B35D6D7E3EFBB9AB1C2EEA6C479CDF841748AF74
7C689432F249F6862E8D7BE8F87697740A0FFCCEB3DAFB5B8A5C8FD4B3CF0103DF841748AF74
0B96417A1DE8F7672E8D7BE8F8769774F722228C01D709C6FE80E6F4CCEA67FDDBBFF65996CC
BD5658864AB886172E8D7BE8F8769774C06F6BB51EA669508A5C8FD4B3CF0103DF841748AF74
88D060258E19C1212E8D7BE8F8769774798742CCB253A39809A609B8BE5BF7FD2C3F03C4DD89
68EDD9252AC381077A79EC050DA4473C34A12D393CEC06B573B574ABE798A17EB57E094265AD
4B1522FE8A62C1C17A79EC050DA4473CA8C54CF65EF711979C9AF7BF55BECA9B5FA807417345
CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6
99285FEE679F546E
CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6
CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6
A346F3B3E4303E49B0610B1E63F008C6C69DE57F8D669963801A9277D31C3357C41284ED797B
CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6
B6D98776BF862D7A2E8D7BE8F8769774990F7A0140282D1428CADD602FE5D6FBDF841748AF74
```

Fig. 9. Coder 1.0 software package:
a) Original text file (* .txt); b) Encrypted version of the same file

These files are encrypted and decrypted with the same key. Encrypted text files retain their extension and increase their size minimally. When encryption / decoding operations are performed on a personal computer, this must be done in different directories (due to the fact that the encrypted files retain their names and extensions).

Experiments have been conducted to encrypt / decrypt * .html files (due to the fact that part of the service documentation of the above information systems is in this format)


menusf - Netscape
Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: What's Related
Instant Message WebMail Radio People Yellow Pages Download Calendar Channe
Document: Done

Location: file:///E|/Users/ani/exp/menusf.htm

Previous Contents Next

2. Menu

There are four groups of menus:

- Query
- Subscriber
- Configure
- Help

(Each of them can be enabled with individual passwords.)

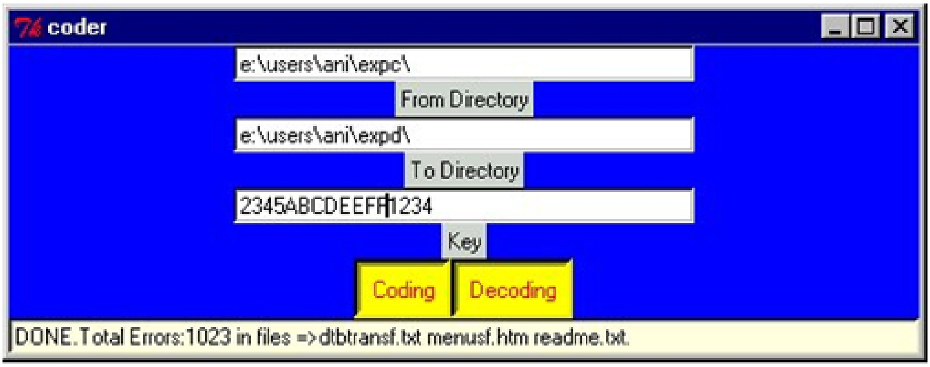
Location: file:///E|/Users/ani/expc/menusf.htm

```
3706C8E7892F5B0640188AFDAEB891D461D08B42A6216B7FC7054865AD96BC2FBF58D2
4014E2CC8EEFB1740D56BF820261665B E0DE7975C57E82BFF44124742BBF5B42
5D14E7332F0202D37286ADC16F86B05A49EF7FAB04C77B76654D5702CE37EC97231DA2
5D14E7332F0202D3462BBA1B6C91C1863CB72194CCF5B47940257CB7D55144E9FE27C2
5D14E7332F0202D3382312C2C3D11C316847EB006762506D579448B4228D3D207D3CA18
B1B463C7D721369B3EF295DA38D805525882B231D23BBCC7BACD5E933CE61D69
2BA96360BB6B84A3543DA55ED0225C53 93A12E1633A48AC8F3426F83B3332504
2D1DCB50A6EAB7F766C28798521F3CFA
0B6DFC1D60C533152984C2440888D2284592F6752D7BEB92D99E04C1F2FCC9AA32D941
CD0E7E77E92958F5
68C56BA7F63F5B8F0A4C6B47F8FA6091F18C1F7B2BA051C367937770C0B2A852A0D5C
61A2F0BF325C3503 65D4D38811D4E103
0B6DFC1D60C53315D607D9BDCD4D82B161DDCB86BFA70B4027764D2D41FBCF26CBB
6A5A035B9ED3A6F47690ADF4FA09E158 97E35A9A188EC4F0 65D4D38811D4E103
1A38C5ABDAC3CD5430AA59C18BB768D0BA110FFB824D317A2CB6D0FE92C1B70C93E
```

Fig. 10. Coder 1.0 software package:
a) Original HTML file (* .html); b) Encrypted version of the same file

When encoding * .html files, it should be borne in mind that the tested program Coder 1.0 does not support image encryption. If there is an image in an encrypted file (of this format), the links are saved when decrypting, and the image space remains empty as the program puts a symbol indicating the lack of expression. This type of file (in encrypted form) increases in size more (compared to text). With the Coder 1.0 program, experiments were performed to encrypt files (contained in a certain directory), with encryption performed with one key and decryption with another.

Fig. 11 shows the result of decrypting a directory (containing three files) with a key different from the one used in its encryption.



e:\users\ani\expc\

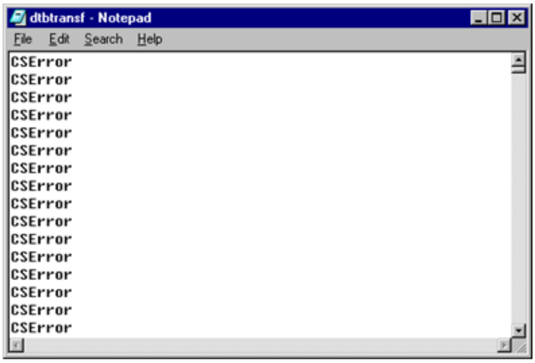
e:\users\ani\expd\

2345ABCDEEFF1234

DONE.Total Errors:1023 in files =>dtbtransf.txt menusf.htm readme.txt.

Fig. 11. Decoding with an invalid key

As can be seen from the figure, in this case during the decoding operation in the status line the message “Total Errors: 1023 in files” appears. Because the encryption is done line by line with this message, the program reports errors on all decoded lines. After this message, the names of the incorrectly decoded files are displayed in the same line. When open any file (from those decoded with an invalid key), sequential CSError messages appear on the screen, as shown in Fig. 12.

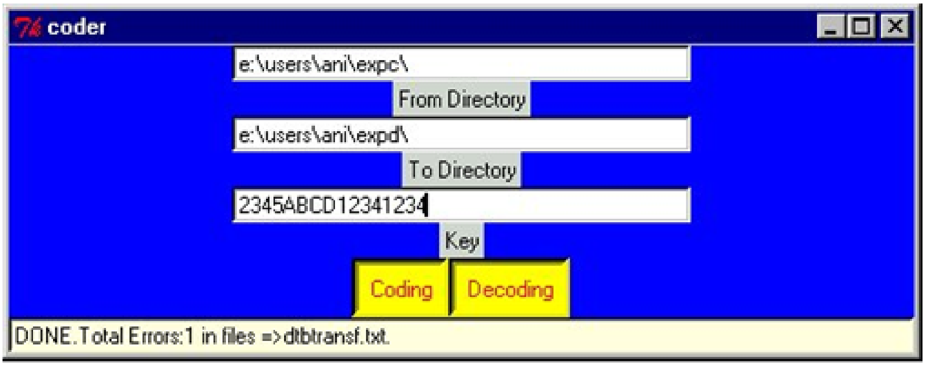


```
CSError
CSError
CSError
CSError
CSError
CSError
CSError
CSError
CSError
CSError
CSError
CSError
CSError
CSError
CSError
CSError
```

Fig. 12. Internal presentation of information in a file decoded with an invalid key

As can be seen from Fig. 11 and Fig. 12, the software package Coder 1.0 does not allow encryption of information with one key and decryption with another. This indicates that information encrypted in a program directory is only available to those who use a valid key to decrypt it. It is unreadable to everyone else.

Files decoded with an incorrect key are 1K in size. Experiments were performed with the presence of manipulated information in an encrypted directory with Coder 1.0. The purpose of the experiment is to prove the ability of the program to locate and declare the presence of files with manipulated information in a secure service or user directory. To this end, an additional line of information has been added to the dtbtransf.txt encrypted file. The result of its decryption is shown in Fig. 13 and Fig. 14



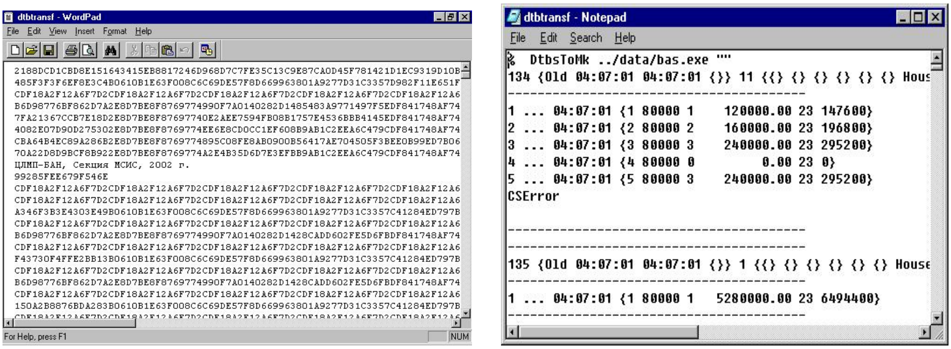
e:\users\ani\expc\

e:\users\ani\expd\

2345ABCD12341234

DONE.Total Errors:1 in files =>dtbtransf.txt

Fig. 13. Decoding a coded directory in the presence of a manipulated file in it



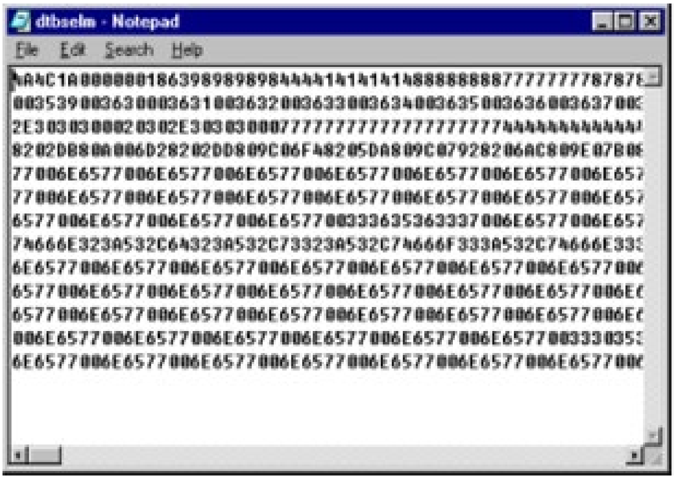
```
2188DCD1CBD8E151643415EB8817246D968D7C7FE35C13C9E87CA0D45F781421D1EC9319D10B
485F3F3F6EF8E3C4B0610B1E63F008C6C69DE57F8D669963801A9277D31C3357D982F11E651F
CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6
B6D98776BF862D7A2E8D7BE8F8769774990F7A0140282D1485483A9771497F5EDF841748AF74
7FA21367CCB7E18D2E8D7BE8F87697740E2AEE7594FB08B1757E4536BBB4145EDF841748AF74
4082E07D90D275302E8D7BE8F8769774EE6E8CD0CC1EF608B9AB1C2EEA6C479CDF841748AF74
CBA64B4EC89A286B2E8D7BE8F8769774895C08FE8AB0900B56417AE704505F3BEE0B99ED7B06
70A22D8D9BCF8B922E8D7BE8F8769774A2E4B35D6D7E3EFBB9AB1C2EEA6C479CDF841748AF74
ЦЛМП-БАН, Секция МСИС, 2002 г.
99285FEE679F546E
CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6
CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6
A346F3B3E4303E49B0610B1E63F008C6C69DE57F8D669963801A9277D31C3357C41284ED797B
CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6
B6D98776BF862D7A2E8D7BE8F8769774990F7A0140282D1428CADD602FE5D6FBDF841748AF74
CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6
F43730F4FFE2BB13B0610B1E63F008C6C69DE57F8D669963801A9277D31C3357C41284ED797B
CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6
B6D98776BF862D7A2E8D7BE8F8769774990F7A0140282D1428CADD602FE5D6FBDF841748AF74
CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6F7D2CDF18A2F12A6
150A2B8876BDA283B0610B1E63F008C6C69DE57F8D669963801A9277D31C3357C41284ED797B
```

```
%  DtbsToMk ../data/bas.exe ""
134 {Old 04:07:01 04:07:01 {}} 11 {{} {} {} {} {} {} Hous
----------------------------------------

1 ... 04:07:01 {1 80000 1    120000.00 23 147600}
2 ... 04:07:01 {2 80000 2    160000.00 23 196800}
3 ... 04:07:01 {3 80000 3    240000.00 23 295200}
4 ... 04:07:01 {4 80000 0         0.00 23 0}
5 ... 04:07:01 {5 80000 3    240000.00 23 295200}
CSError

----------------------------------------
----------------------------------------
135 {Old 04:07:01 04:07:01 {}} 1 {{} {} {} {} {} {} House
----------------------------------------
1 ... 04:07:01 {1 80000 1   5280000.00 23 6494400}
----------------------------------------
```

Fig. 14. Internal presentation of the information in a manipulated file
a) Encrypted file with additional text entered; b) Contents of the manipulated file after decoding

As can be seen from the figures when performing an operation to decode the pre-manipulated security file in the status line of the program appears the message “Total errors: 1 in files => dtbtransf.txt”. In this way, the program reports an error in one line of the file. When you open it with a text reader, you can see that the message “CSError” is displayed in place of the added line (Fig. 14b). This message is a warning about the breach of confidentiality of protected information, i.e. the information in this file becomes invalid. In this case, it is necessary to perform a check to determine whether an attempt has been made to manipulate (by external or internal malicious persons) the protected file. The messages in Fig. 13 and Fig. 14 show that the developed software product has a built-in algorithm for monitoring possible attempts at malicious interference.

With the developed additional option SCRIPT (built into the software product described in [24]) experiments were performed to protect databases of the type MK4Tcl. The type of encrypted database is shown in Fig. 15.



```
4A4C1A000000018639898989898444414141414888888887777777787878
0035390036300003631003632003633003634003635003636003637000
2E3030300002030 2E303030007777777777777777777777444444444444
8202DB80A006D28202DD809C06F48205DA809C079282 06AC809E07B00
77006E6577006E6577006E6577006E6577006E6577006E6577006E657
77006E6577006E6577006E6577006E6577006E6577006E6577006E657
6577006E6577006E6577006E6577003336353633337006E6577006E657
74666E323A532C64323A532C73323A532C74666F333A532C74666E333
6E6577006E6577006E6577006E6577006E6577006E6577006E657700
6577006E6577006E6577006E6577006E6577006E6577006E6577006E
6577006E6577006E6577006E6577006E6577006E6577006E6577006E
006E6577006E6577006E6577006E6577006E6577006E65770033303 5
6E6577006E6577006E6577006E6577006E6577006E6577006E657700
```

Fig. 15. Encrypted database of [24]

The implemented SCRIPT command is convenient for embedding in user programs, for encrypting files of the type: * .txt, * .doc, * .html, databases, etc.

The ability of the Tcl/Tk language to replace commands, i.e. each command can be replaced by a sequence of Tcl commands, which allows changing its operation under program control, creates opportunities for dynamic change of encryption algorithms. The EVAL command of the language allows the interpretation of any text as a Tcl program. It combined with SCRIPT gives the opportunity to encrypt not only data but also executable Tcl programs.

6. Conclusion

Information encryption tools are an important element in building a corporate security system. This fact is a consequence of the requirements set by today’s organizations and companies, their data to be protected both on their own machines and on the way to another destination.

The proposed software package for coded information exchange – Coder 1.0 has the following advantages:

- is designed to work under different operating environments;
- hardware requirements are minimal;
- encrypted information is in hexadecimal format, which makes the output files convenient for transfer via TCP / IP;
- the SCRIPT option implemented as a command is convenient for embedding in user programs, for encrypting files of the type: * .txt, * .doc, * .html, databases, etc.;
- can be used both as a standalone product and as an element, within a common security system of a distributed corporate network;
- the integration of the encryption algorithm in a user Tcl program provides additional opportunities for building an encrypted protocol that extends the network mechanisms (implemented in the language).

In connection with increasing the requirements for the reliability of information on corporate networks, work will continue with the study and use of other algorithms for encryption and development of software products for encrypting information in multimedia objects (images and sounds) using the capabilities of Tcl/Tk.

Acknowledgments

This research is supported by the Bulgarian FNI fund through the project "Modeling and Research of Intelligent Educational Systems and Sensor Networks (ISOSeM)", contract КП-06- Н47/4 from 26.11.2020

References

1. Boneva, A.: Means of information protection in a distributed corporate network. In: Proc. of "Scientific reports" October 2002, CLMI-BAS, Conference "ROBOTICS and MECHATRONICS'2002", Drjanovo, Bulgaria, pp. 4.11–4.16, (2002) (in Bulgarian)
2. Беляев, А.В.: Методы и средства защиты информации (курс лекций), Авторские права: Череповецкий филиал Санкт-Петербургского государственного технического Университета, (2000), Available: http://www.citforum.ru/internet/infsecure/index.shtml
3. Utimaco Safeware AG, www.utimaco.com (last accessed 2022/02/07)
4. Prochazka, M., Bencel, R., Kostal, K., Ries, M.: Encryption and decryption of wireless traffic in software-defined wireless networking. In: Software-Defined Wireless Networking, International Symposium ELMAR, 2019, pp. 77–80, https://doi.org/10.1109/ELMAR.2019.8918662 (2019)

5. Awan, S., Javaid, N., Ullah, S., Khan, A.U., Qamar, A.M., Choi, J.-G.: Blockchain based secure routing and trust management in wireless sensor networks. Sensors 22, 411, 1–24, https://doi.org/10.3390/s22020411 (2022)

6. Agrawal, E., Pal, P.R.: A secure and fast approach for encryption and decryption of message communication. IJESC 7(5), 11481–11485 (2017)

7. Gaydarski, I., Minchev, Z., Andreev, R.: Model driven architectural design of information security system. Advances in Intelligent Systems and Computing, Madureira A., Abraham A.,Gandhi N., Silva C., Antunes M. (eds) Proc. of the Tenth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2018), 492, pp. 349–359, https://link.springer.com/chapter/10.1007/978-3-030-17065-3_35 (2019)

8. Antonov, P., Malchev, S.: Cryptography in Computer Communications. TU-Varna, (2000) (in Bulgarian)

9. Menezes, J., an Oorshot, P.C., Vanstone, V.S.A.: Handbook of Applied Cryptography, CRC Press, (1996)

10. Anuraj, C.K, Joseph, S.: Analytical study on encryption techniques and challenges in network security. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) 6(6), 153–162, (2017), Available: https://ijettcs.org/Volume6Issue6/IJETTCS-2017-12-23-48.pdf.

11. AES: data encryption, https://bg.srimathumitha.com/kompyutery/61515-aes-shifrovanie-dannyh.html (last accessed 2022/02/07)

12. Abdullah, A. M.: Advanced encryption standard (AES) algorithm to encrypt and decrypt data. Cryptography and Network Security, 1–13, (2017), Available: https://www.researchgate.net/publication/317615794_Advanced_Encryption_Standard_AES_Algorithm_to_Encrypt_and_Decrypt_Data.

13. Singh, G., Supriya: A study of encryption algorithms (RSA, DES, 3DES and AES) for information security. International Journal of Computer Applications 67(19), 33–38, https://research.ijcaonline.org/volume67/number19/pxc3887224.pdf (2013)

14. Pirbhulal, S., Pombo, N., Felizardo, V., Garcia, N., Sodhro, A.H., Mukhopadhyay, S.C.: Towards machine learning enabled security framework for IoT-based healthcare. In: 13th International Conference on Sensing Technology (ICST), pp. 1–6, https://doi.org/10.1109/ICST46873.2019.9047745 (2019)

15. Indu I., Rubesh Anand P.M., Vidhyacharan Bhaskar.: Encrypted token based authentication with adapted SAML technology for cloud web services, Journal of Network and Computer Applications 99, 131–145, https://doi.org/10.1016/j.jnca.2017.10.001 (2017)

16. Dodis, Y., Karthikeyan, H., Wichs, D.: Updatable public key encryption in the standard model. In: Nissim K., Waters B. (eds) Theory of Cryptography. TCC 2021. Lecture Notes in Computer Science 13044, pp. 254–285, https://doi.org/10.1007/978-3-030-90456-2_9 (2021)

17. Терехов, А. Н, Тискин, А. В.: Криптография с открытым ключом: от теории к стандарту. Программирование РАН, 5, 17–22, (1994), Available: https://www.math.spbu.ru/user/ant/All_articles/043_Terekhov_Tiskin_OpenKey_Cryptography.pdf

18. Welch, B.: Practical Programming in Tcl and TK, part3 – TclHttpd Web Server, Ajuba Solutions, (1998–2000).

19. Liestyowati, D.: Public Key Cryptography, ICComSET 2019, Journal of Physics: Conference Series, IOP Publishing 1477, 052062, 1–7 (2020)

20. Wippler, Jean-Claude. Scripted Documents, 7th USENIX Tcl/Tk Conference – Tcl/Tk, Austin, Texas, USA, (February 14–18, 2000)

21. Tcl/Tk program, https://www.tcl.tk/ (last accessed 2022/02/07)

22. Hipp, R., Mktclapp. A Toll For Mixing C/C++ with Tcl/Tk, Charlotte, NC, (1999).

23. Ivanova, V.: Laparoscopic Tools and Smart Instruments to Robots. Technical University Publishing House, Sofia, (2018) (in Bulgarian)

24. Batchvarov, D., Belov, K., Angelov, S., Calikoglu, F., Boneva, A., Krasteva, R.: A technology for electronic energy meters intelligent accounting using distributed database over Tcp/Ip Network. Problems of Engineering Cybernetics and Robotics 54, 48–53 (2004)

25. Ilchev, S., Andreev, R., Ilcheva, Z., HybridNET management and sensor data acquisition system. In: IoT '17 Proc. of the Seventh International Conference on the Internet of Things, Linz, Austria, DOI: 10.1145/3131542.3140268 (2017)