TABLE OF CONTENTS

TABLE OF CONTENTS **1**

THEORETICAL BACKGROUND **2**

GENERAL PRINCIPLES OF EDI SECURING 2
CRYPTOGRAPHY 3
Digital signature *3*
Advantages of digital signature *3*
Certification Authority - meaning and function *4*
EDI AND UN/EDIFACT 4
UN/EDIFACT *5*
GENERAL PRINCIPLES OF SECURING UN/EDIFACT MESSAGES 5

DESCRIPTION OF SECURITY STRUCTURES **7**

SECURITY HEADER AND TRAILER FOR DIGITAL SIGNATURE 7
Principles of digital signature creating and checking *7*
Syntactic and formal rules for the digital signature *9*
Example of a signed message *25*
THE AUTACK MESSAGE 27
The principle of using the AUTACK message *27*
Syntactic rules and formal rules for the AUTACK message *28*
Example of an AUTACK message *37*
CIPHER MESSAGE 39
The principle of using the CIPHER message *39*
Syntactic and formal rules for the CIPHER message *41*
Example of CIPHER message *49*

APPENDIX B - LIST OF ERROR CODES IN THE AUTACK MESSAGE **50**

THEORETICAL BACKGROUND

General principles of EDI securing

When creating systems intended for EDI message transmission, it is necessary to pay special attention to the security of these systems, especially to the data transmission security. For that reason it is necessary to adopt appropriate security policy at the time of system design, which is then implemented when creating and operating the system. The security policy is based on the definition of possible system threats, the identification of appropriate security functions and their effective implementation using security mechanisms.

Possible security threats to the EDI transmission system can be divided into the following categories:

a) Message modification - the message is changed after sent by an approved sender, either deliberately or due to a technical error.
b) Change in the message sequence - the message can be lost during communication or it can be copied and delivered several times, either deliberately or due to a technical error. If the sequence of message receiving is important, then another possible threat is represented by changes in the message sequence.
c) Masquerading - during communication with another party the system user masquerades as some other approved user.
d) Access of a non-approved person to the system - a person not entrusted with system access takes part in the communication and masquerades as an approved system user.
e) Repudiation on the side of the sender - the person sending the message denies having sent it.
f) Repudiation on the side of message receiver - the person intended to receive the message denies its receipt although the message was already delivered.
g) Misuse of confidential data - confidential messages can be obtained by an unauthorized person or misused in other ways.

One of the possibilities for effective prevention of above mentioned threats is the application of the following security functions:

a) Message integrity - this function guarantees that any modification of the message content will be revealed during transmission.
b) Message sequence integrity - this function guarantees that no message can be copied by an unauthorized person and sent again. It also enables the detection of message loss and of any change in the message sequence.
c) Message authentication - this function enables the identification of the person sending the message.
d) Access control - this function enables access limitation to messages and communication only to approved persons.
e) Non- repudiation of the message origin - this function secures that the person sending out the message can not later deny sending it.
f) Non- repudiation of the receipt - this function secures that the person receiving the message can not later deny its receipt.
g) Ciphering of the message content - this function secures data confidentiality during transmission.

The following model of security mechanisms based on efficient cryptographic algorithms has been designed for the implementation of security functions in the program:

1. The security function of the integrity, authentication and non-rejection of origin are secured by the message digital signature and the special cryptographic algorithm based on an asymmetric ciphering algorithm.
2. In order to secure the non-rejection of message receipt an UN/EDIFACT AUTACK message is used, which unambiguously confirms the receipt of a certain message.
3. The sequential integrity is secured by means of message reference numbers which are sequential and unique to each user, and by a security time mark. Any duplicated message can thus be identified on receipt as well as any loss of messages or change in their sequence.
4. EDISEC2 applications are designed to make maximum use of access control mechanisms in UNIX . In addition, access control can be secured by using an access password or a PIN with which the user's secret key is secured so that nobody except the approved user can use it.
5. The confidentiality of the message content is secured by the message encryption using a ciphering algorithm. In EDISEC2 a symmetric DES algorithm is used, which is a globally respected standard for ciphering commercial messages.

Cryptography

The implementation of security functions for the transmission of standard EDI messages is currently achieved in particular via security mechanisms based on effective cryptographic algorithms.
Cryptographic algorithms can be divided into two basic types:

A) symmetric - using one key both for message ciphering and deciphering.

Symmetric algorithms are very old and commonly used. It is practically impossible to decipher their current sophisticated versions without the appropriate key being available.
The disadvantage of these algorithms lies in the fact that communication between a number of users using ciphered messages requires a list of all other users' keys available to each user. So if an unauthorized person obtains this list from one of these persons all users are affected. For a larger group of users, the number of necessary keys extend unfavorably - each pair of users must have their own specific key (a quadratic dependence).

B) asymmetric - using two different keys for ciphering and deciphering the message.

Asymmetric algorithms are relatively new. They were first introduced in the early 1970s.
Each user working with asymmetric algorithms has two **keys**: a **secret** one and a **public** one. The secret one is known only to its holder while the public one is known to all who are communicating with.
The principle of using these algorithms is based on the fact that if you have ciphered a message with one of the keys you can not decipher it without knowing the other one. Also, you can not cipher a message knowing only one key.

Digital signature

One of the most progressive cryptography methods, the **digital signature** is based on asymmetric algorithms.
The digital signature has all **the features of a hand-written signature** and is even **more immune** against falsification.
The digital signature is created by an asymmetric ciphering algorithm, the RSA.
The digital signature is closely connected with key management. Certification Authority is used in the commercial praxis.
The digital signature is a **cryptographic method enabling authentication** of the message and **its integrity checking** (in addition the secret key owner can not later deny his digital signature).
The digital signature is created as follows:

- First the message is provided with a control block of bytes by a special hash function. This control block is unique for each message.
- It is then ciphered by the message sender using his secret key and the ciphered block is added to the message.
- Digital signature checking also starts with the creation of a control block from the message identical to the procedure applied for sending the message.
- The ciphered control block is deciphered by using the public key of the message sender and if the control blocks are:
    - identical, then the digital signature is valid and data are accepted
    - different, then the transmitted data are rejected.

Advantages of digital signature

A message signed with a digital signature can not be modified, nor can the sequence of messages be changed because in both cases the control block would be found defective and the transmission rejected.
It is also impossible to masquerade as a regular user because a digital signature enables the identification of the person sending the message.
Sending or receiving a message can not later be denied by the sender or recipient because by entering the secret key the message is authorized while sending and when confirming correct receipt the secret key is also used by the recipient which is evidence of receipt.
The advantage of the digital signature thus rests upon its almost absolute immunity to falsification. Additionally, the digital signature performs various security functions: **integrity**, **authentication** and both **non-repudiation of the message origin** and **non-repudiation of receipt** (when using confirming message provided with a digital signature).

Certification Authority - meaning and function

To assure proper functioning of the digital signature every approved user of EDI must know the public keys of other approved users with whom they communicate. Distribution of public keys to users and their management is provided by the Certification Authority (CA). However, the secret key must be known only by the owner.

The Certification Authority provides users with certificates which essentially certificate public key possession. The certificates are in electronic form, signed using a secret CA key. The public CA key is commonly known so anyone can check the validity of any certificate. The certificate includes: certificate number, owner's identification, dates of validity and the public key of the owner. The certificate (or its reference number only) is sent together with the signed message and the validity of the digital signature is checked by using the public key contained in it. It is hence impossible to use digital signatures without valid certificates.

The UN/EDIFACT standard defines the exact form of the certificate. Such uniformity enables interconnection of different systems based on this standard. The users' certificates and the digital signatures of messages can be mutually accepted.

In addition to assigning the certificates, the Certification Authority maintains a list of valid certificates, archives the certificates with expired effective dates and maintains a so-called Black List (or Certificate Revocation List), which includes formally valid certificates, the validity of which was effectively voided. CA also distributes the lists of valid certificates and the Black List among the users.

In addition to these basic functions, CA may also function in the field of system security. CA can act as a Trusted Third Party whose technical and "social" credibility entrust it to serve in a number of important areas in EDI systems, e.g. as an Electronic Notary. Furthermore, it certifies functionality, establishes contacts and provides cross-certification, witnessing, etc.

EDI and UN/EDIFACT

EDI (Electronic Data Interchange) - is the electronic interchange of structured standard messages between two applications of two independent subjects.

This underlined definition may be the shortest and most truthful description of the substance of what EDI is. To achieve better understanding of these terms see the following explanation:

- **Electronic data interchange** - data interchange via electronic transmissions (ON LINE). Networks such as Internet, VDS Nextel, IBM IMNS, phone lines and both radio and satellite connections can be used for communication. The electronic data interchange is mostly non-interactive, i.e. first a complete block of data is created which is then sent in one batch. Sending does not take place parallel with data production.
- **Structured data interchange** - structured data are data defined by syntactic rules. These rules make up a common language for all interconnected applications. The syntactic rules define distributors or lengths of items. Structured data include data in the database format, data of a fixed length, data in CDF format (Comma Delimited Format) etc. The syntax of structured messages is defined by the Czech and international standards **ÈSN ISO 9735 (UN/EDIFACT).** Using syntax is very important for automatic processing. Using an internationally valid standard is important for compatibility with other EDI systems. It is necessary that all the systems „speak" the same „language".
- **Standard messages** - a standard message is a predefined type of message where each item has its own location.

  There are hundreds of messages defined for state administration, trade, transport, as well as for health systems, building industry, etc., within UN/EDIFACT (United Nation/ Electronic Data Interchange for Administration, Commerce and Transport). In addition to UN/EDIFACT there is a wide range of branch and national standards (ANSI X12, ODETTE, VDA, SEDAS). These standards are currently being replaced by the truly international standard, the UN/EDIFACT.
- **Between two applications** - unlike systems such as „homebanking" which are structured as single-purpose applications, EDI is intended for data interchange between two general applications. One EDI application transfers data to another fully automatically, ideally without any human intervention. Though some simplified applications with a simple interface for hand operation are used within the load time, mutual communication between applications remains the ultimate goal.
- **Between two independent subjects** - with regard to the design of EDI systems it is necessary to state whether EDI for dependent or independent subjects is in question.

UN/EDIFACT

UN/EDIFACT (United Nations /Electronic Data Interchange For Administration, Commerce and Transport) is an international standard for electronic data transmission **(standards ISO 9735 and ÈSN ISO 9735).**
The UN/EDIFACT standard and other associated recommendations have been developed by the **UN/ECE** (United Nations Economic Commission for Europe) and its working group **WP 4** for international commerce rationalization. The standards created by the UN/ECE/WP.4 are being adopted by the International Standardization Organization (ISO) and issued also as ISO standards.
The UN/EDIFACT standard defines the exact syntax of electronic documents:

- standard set of characters for electronic documents (so-called syntactic levels) are defined; within these sets special characters are defined - separators and release characters.
- fundamental elements of the documents are defined:
    - DATA ELEMENT (DE) is a basic data unit carrying certain information. It is represented by four digits.
    - COMPOSITE DATA ELEMENT (CDE) is a set of data, parts of which (partial data elements) are in relevant or logical relation to each other. The individual elements within one composite data element are separated by a special separator (standard is ´:´). A CDE can be further divided as follows: service represented by S and three digits, and user represented by C and three digits.
    - SEGMENT (SEG) is a connection of logically related elements. Segments are represented by a three character letter cipher; the service segments are always represented by UN or US and by a further character. Individual segments are made by their identification and single or composite data elements. The individual segment parts are separated by a special separator (standard is ´+´). The segments can be further merged into SEGMENT GROUPS (SG); it is possible to define the number of segment repetitions as well as their status for SGs within a message. The status of individual SEGs, CDEs or DEs is defined as follows: M- Mandatory or O- Optional.
    - MESSAGE can be created by connecting all the segments or groups of segments needed to represent a certain operation. The messages contain service and user segments. The segments are separated by a special separator (standard is ´´´).
    - FUNCTIONAL GROUP is a set of related messages. The functional group consists of service segments UNG and UNE, and of individual messages.
    - INTERCHANGE is a set of messages sent by one EDI user to another. The interchange set consists of service segments UNA, UNB and UNZ and of individual messages or functional groups.

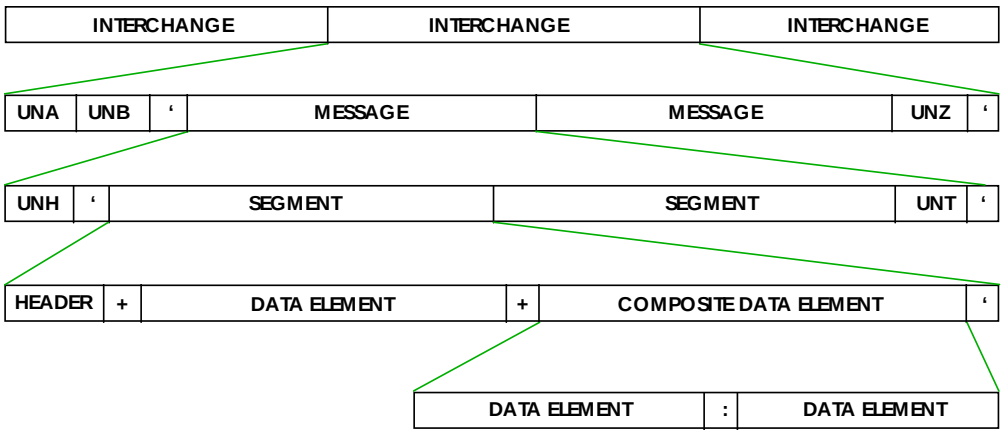

Fig. 1 (Structure of UN/EDIFACT)

General principles of securing UN/EDIFACT messages

There are standards defined within UN/EDIFACT, which describe possible ways of securing UN/EDIFACT structures using various security mechanisms. The security functions provided within the UN/EDIFACT standard guarantee end-to-end securing, i.e. from one end user to another independently on the form of data transmission (the messages can be

transmitted via unsecured public communication networks). The security mechanisms applied become a part of the UN/EDIFACT message structure and if these messages are archived their integrity and authenticity can be checked during archiving at any time.
The basic elements of the UN/EDIFACT security standard used for securing EDI messages are as follows:
There are standard structures defined within the UN/EDIFACT standard which enable the use of digital signature directly in UN/EDIFACT messages. Each message provided with a digital signature includes a so-called Security Header. This Security Header contains the algorithms used for the digital signature, the digital signature creation and a certificate of the user and also a so-called Security Trailer which contains the message authentication - the digital signature itself. Creating a digital signature: first the message is provided with a control block of bytes by a special hash function. This control block is unique to every message. The control block is then ciphered by the secret key of the sender and the ciphered block is added to the message.
The UN/EDIFACT standard defines the CIPHER message which enables the transmission of ciphered data. The UN/EDIFACT message is ciphered from the beginning to the end segment and is inserted into the body of the CIPHER message. In this way the confidentiality of the transmitted data is guaranteed.
The UN/EDIFACT standard describes the AUTACK message. This message responds to the transmitted message and contains a reference and a calculated control block of bytes unique to every transmitted message. The message is provided with a digital signature of the recipient who can not later deny the fact of message receipt.
The UN/EDIFACT also defines the KEYMAN message which enables transmission of keys and certificates between different applications.

DESCRIPTION OF SECURITY STRUCTURES

Security Header and Trailer for digital signature

Principles of digital signature creating and checking

The digital message signature is used for one message securing. If the interchange file contains more messages, each message is secured alone.
The digital signature creating for the one message is stated at the following chart:

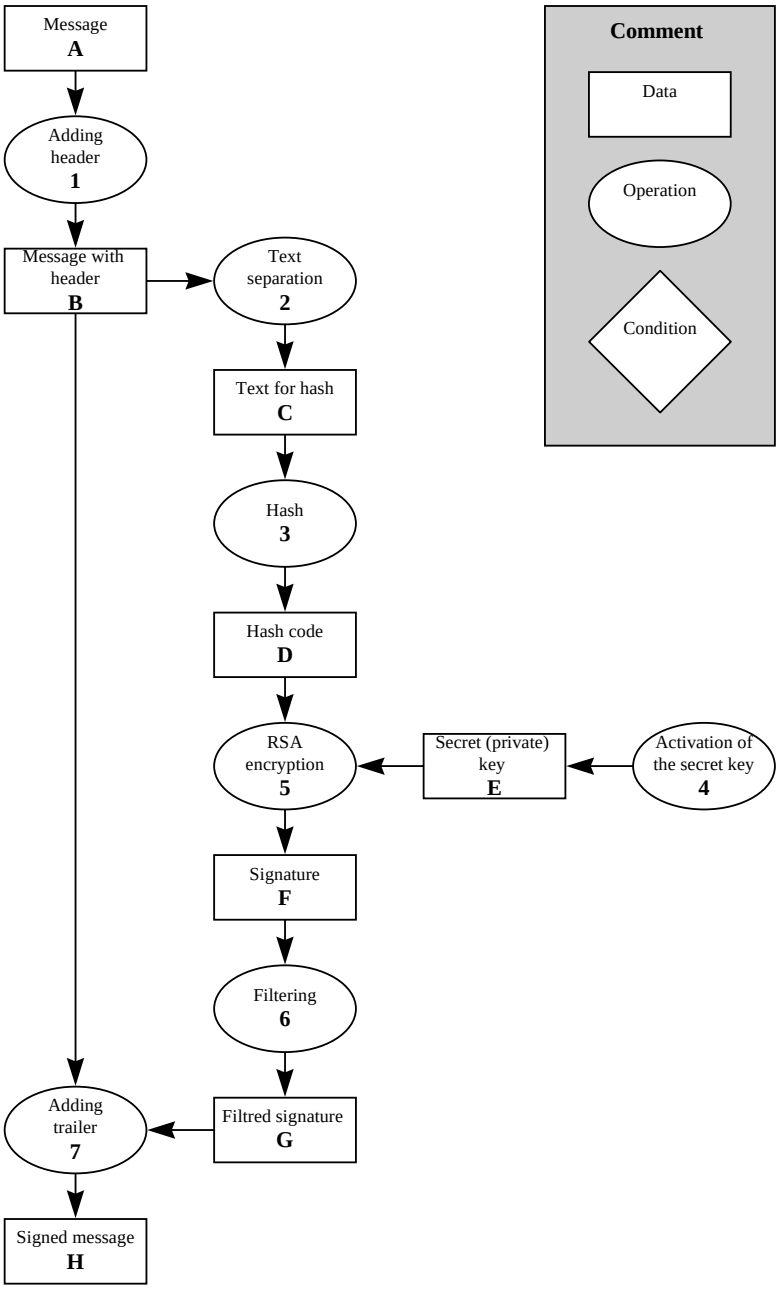

Fig. 2 (The digital signature creating for one message.)

The signature creating procedure is following:

1. The filled starting security segments (the security header) are added to the UN/EDIFACT message [A].
2. The text, which will be entered to the hash function, is obtained from the message with the security header[B] (it means the starting security segment first and then the message body). The result is continuous text [C], which is represented as byte sequence.
3. The text [C] is processed by MD5 hash function and the result is the 16 byte code [D].
4. The secret user key[E], which is necessary for signature creating, is obtained.
5. The Hash code [D] is ciphered by the RSA algorithm and the secret key [E]. The result is the digital signature [F]. Its length is corresponding to the key module.
6. The digital signature [F] is filtered to the text shape to be transferred in the UN/EDIFACT message.
7. To the message are added the finishing security segments (the security trailer), which contain the filtered signature [G]. The result is the complete secured message[H].

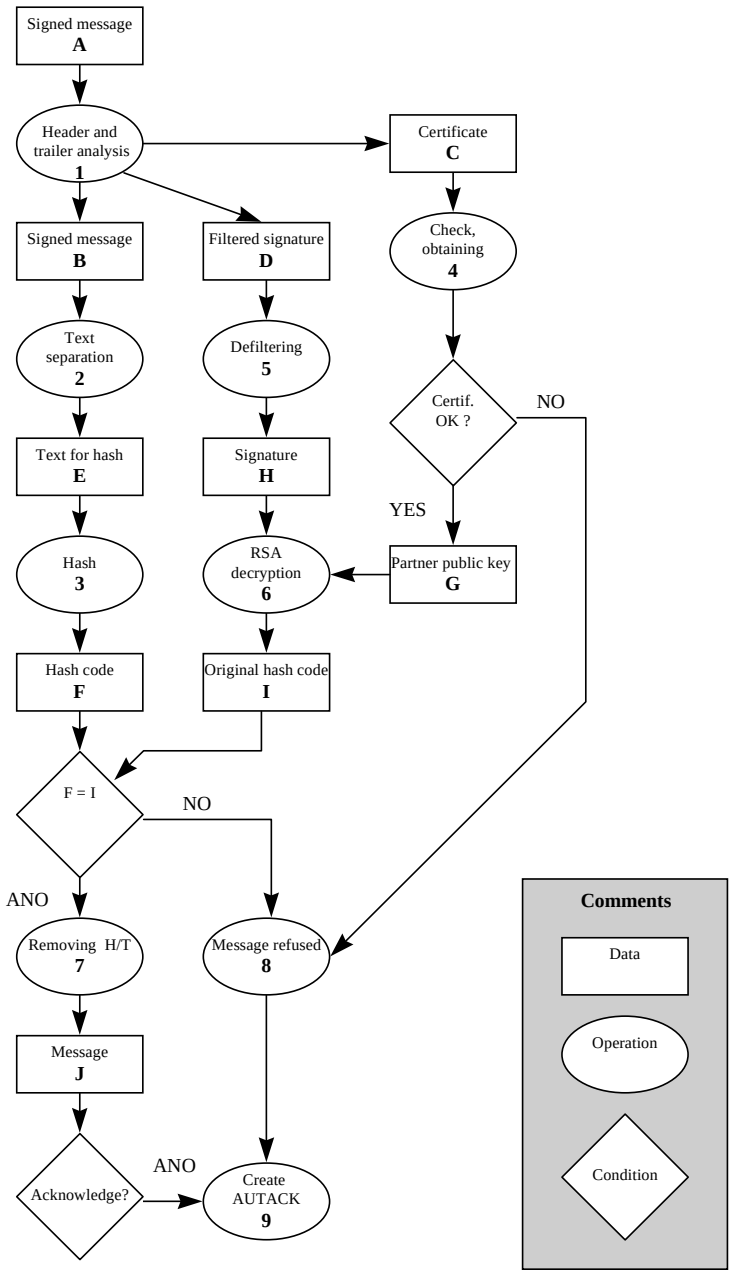

Fig. 3 (The signature checking for one message chart.)

The check procedure is following:

1. The data, specified in the security header and trailer segments, are loaded from the signed message [A].
2. The text, which will be entered to the hash function (it means the security headers segments first and then the message body) is obtained from the signed message [B]. The result is the continuous text [E], which is represented by the byte sequence.
3. The text [E] is processed by the MD5 hash function and the result is the 16 bytes hash code [F].
4. The certificate number [C] was loaded from the header, the certificate must then be loaded from the local database. The certificate can be checked and the partner public key [G], necessary for message check, is obtained from the certificate. The complete certificate [C] could be stated in the header and the certificate is checked, in this case, and the partner public key[G] is obtained from it.
5. The filtered digital signature [D] is loaded from the trailer. It must be filtered back to the binary shape.
6. The digital signature [H] is deciphered by RSA algorithm by the partner public key [G]. The result is the original message hash code [I] and its length is 16bytes.
7. If the signature checking was successful (i.e. the computed hash code is equal to the original hash code), the header and trailer segments are deleted from the message and the „clear“ message will arise [J].
8. If the signature checking was not successful (i.e. the computed hash code is not equal to the original) or the certificate was not successfully checked (bad certificate signature, non valid certificate, the certificate is not available) the message must be refused and must not be processed more.
9. If the message should be acknowledged or refused, the acknowledging message AUTACK is created for original message sender (details see chapter The AUTACK).

Syntactic and formal rules for the digital signature

The message securing mechanism UN/EDIFACT by the digital signature is designed on the recommendation of the UN/TRADE/WP.4/R.1026/Add.2 and ISO/CD 9735-5. The formal digital signature implementation is done by the security segments header and trailer. The special segments for each UN/EDIFACT message are added, the securing message structure is on the Fig. 2. These segments allows up to 9 digital signatures creating for one message. The one signature is used in the SUD implementation. If an error is detected during the message receiving in the digital signature, the AUTACK message (see next paragraph) is generated, which is sent to the receiving message originator. The receiver/application receiver must be
informed too and the event must be recorded to the receiver log file.

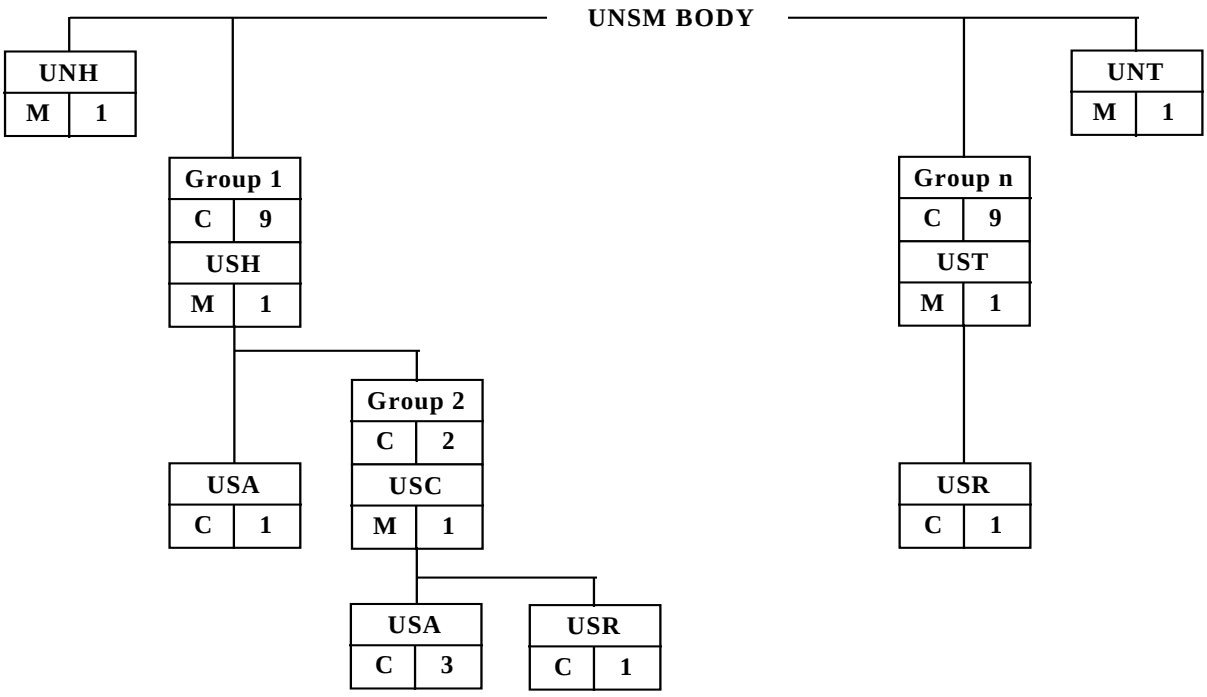

Fig. 4 (The structure of a secured UN/EDIFACT message.)

The UNH and UNT segments are standard functional segments of the securing message (Message Header, Message Trailer). UNSM Body is the securing message body, starting by BGM segment and all next segments.

The certain security segments (USR,USA) contain data, which are the results of ciphering functions or they are used as the input of these functions. This data may be generally binary (i.e. each data byte can be in the 0..255 range). The data must be first processed by filtering function (sometimes called ciphering too) to be stated in the EDIFACT segment. The filtering function converts the binary data to other representation, when the data are represented by displayable characters and therefore can be stated in the EDIFACT form. On the contrary if the data are loaded from the EDIFACT, the filtering function must be designed, which will create the binary data from the character representation. The simplest filter example is the hexadecimal filter, which each byte of binary data represents like pair of characters ('0'-'F'). It means e. g. the number 125163 is represented like character sequence '1', 'E', '8', 'E', 'B'.

The message securing scope (i.e. hash function input) is following: Security Header (all segments from group 1 and 2), from the USH segment first character, i.e. 'U' till the separator last segment character of the Security Header (i.e. ' ' ') including and the message body (follows after Security Header) to the last segment of body separator included (segment immediately in front of UST).

M/C - **mandatory (M)** - it is an element or a segment, a group, which are defined as mandatory by the standard and therefore they are mandatory for the security segment implementation too.
- **used non mandatory (C)** - it is an element or a segment, a group, which are defined in the standard as non mandatory, but they are used in the implementation and their using is mandatory.
- **obviously non used (O)** - it is an element or a segment, a group, which are defined in the standard as non mandatory and they are not used in the implementation now. They are bypassed or they are defined for the future using or for other standard systems compatibility. Their possible using do not affect the basic security functions of the system and partial implementation can use them for their specific purposes.

Op. - **number of repetition**, the max. number allowed by standard is stated in the brackets. The repetition, which will not be used is stated in () and the repetition, which can be used in the implementation is stated in [].

Group SEGMENT	M/C	Op.	DESCRIPTION
1	C	1[9]	This segment group identifies the used security functions
USH	M	1	Defines the special security services used for the given message. It contains the time stamp and the data about sites, which offers the security services.
USA	C	1	Algorithm used for message hashing
2	C	1(2)	The group of 2 segments specifies the sending site certificate.
USC	M	1	It contains the certificate number, the certificate owner identification, period of validity, the certificate issue date and other data.
USA	C	1[3]	It contains the algorithm data, which owner uses for the digital signature and the owner public key.
USR	C	1	The certificate signature created by the CA secret key.
n	C	1[9]	The result of message authentication.
UST	M	1	The segment is used for marking of the message part, which the given security mechanism is applied on, and connecting of the authentication result with the given USH segment.
USR	C	1	The digital message signature by the secret key using.

Tab. 1 (The security segments groups.)

Note. The group numbering is not dependent on the group numbering in the message body.

Comp. elem. - the composite element number in the UN/EDIFACT Standard Directory
Element - the element number in the UN/EDIFACT Standard Directory
M. - mandatory (M), used non mandatory (C), obviously non used (O) segment, element
Format - the format specification by the UN/EDIFACT convection
Content - in ' ' are stated constants, text identifiers refer to the variable values given by the security application

Note.: If the repeated segments or repeated composite elements are occurring, their meaning (and content too) is not assigned by the occurrence order, but by the qualifiers, which are in the composite element or segment.

GROUP1 (C,1 - 9) SEGMENT USH (M, 1)

Comp. Elem.	Element	M.	Format	Meaning	Content	Comments
	0552	M	an..3	Segment structure version	'94W'	1994 version
	0501	M	an..3	Security function - code	'1'	Non refusal of origin
	0534	M	an..14	Check reference	link	link=01 for one signature
	0541	C	an..3	Security scope - code	'1'	starting sec. Segments + message body
	0503	C	an..3	Answer type, code	ack	ack= '1' - message not to be acknowledged by AUTACK message ack= '2' - message to be acknowledged by AUTACK message
	0505	C	an..3	Filter (function)- code	filter	Filter for binary data
	0507	C	an..3	Charact. coding -code	'2'	ASCII 8 bits
	0509	C	an..3	Sig. Site roll -code	'1'	Document originator
S500		*O*		*Site Identification*		*The sending site identification*
S500	0577	M	an..3	Site qualifier	'1'	Message sender
S500	0538	C	an..35	Key name	key	key = key number (name) used for signing
S500	0511	C	an..17	Site ID	EDI_ID	EDI_ID= EDI application sender identification
S500	0513	O	an..3	Used site list	'1'	Bank list code (bank EDI applications)
S500	0515	O	an..3	List updated agency	'CNB'	The Czech national bank
S500	0586	O	an..35	Site name	org_name	org_name= organisation name
S500	0586	O	an..35	Site name	org_dep	org_dep= department (branch) in the organisation
S500	0586	O	an..35	Site name	org_pers	org_pers= responsible worker
S500		*O*		*Site identification*		*By-passed*
	0520	C	an..35	Reference number	ref_num	ref_num= sequential reference number
S501		*C*		*Date and time*		*Time stamp - signature creating*
S501	0517	M	an..3	Date and time qualifier	'1'	Security time mark
S501	0338	C	n..8	Date	date	date= signature creating date, format YYYYMMDD
S501	0314	C	n..15	Time	time	time= signature creating time, format HHMMSS
S501	0336	C	n4	UTC offset (time offset)	offset	offset = '0100' - offset from UTC is + 1 hour (winter time) offset = '0200' - offset from UTC is + 2 hour (summer time)

Tab. 2 (The security segments structure.)

The segment group 1 defines the parameters for the digital message signature and together with the group n forms the digital message signature. The group 1, together with the group n, is repeated for each digital message signature. The actual application uses only one signature, and only one groups repeating.

Elements description:

0552 - Segment structure version
The value '94W' defines, that the operating security segments, described in the UN/TRADE/WP.4/R.1026 and ISO/CD 9735-5 document are used.

0501 - The security functions
The origin non-rejection function is used for message securing (value '1')

0534 - The check reference
This element is used as unique key for group1 (Security Header) and n (Security trailer) connecting - it means the parameters defined in the group1 are used for group n, which has the same element value 0534. The element value has two digits and is numerical. The application uses only one digital signature - the link value is '01'. The link is incremented for the next group repeating.

0541 - Securing scope
The value '1' defines that the signature is computed from the starting security segment text (group 1 and 2) - from the segment first letter (i.e. 'U') to the segment separator (included), which terminates the segments, and message body text, which is immediately connected - from the first character behind the starting segment finishing separator ('B' form the segment BGM) to the finishing security segments separator (included). In the case of one signature, it means the hash function is applied to the continuous text from the 'U' of USH segment till to '' in front of segment UST. The function is applied only for one actual group of starting security segments (i.e. groups 1 and 2) and the message body in the case of more signatures. It means the signatures are independent and they are not hierarchically ordered.

0503 - Answer type
This element specifies whether the sender requires the receiving non refusal function from the receiver - i.e. message acknowledgement by AUTACK message, ack can have two values:
'1' - the sender does not require acknowledgement by AUTACK message
'2' - the sender requires acknowledgement

0505 - Filter (function)
It defines the function type, which is used for the binary data filtering,. The data are the result of the digital signature before their recording to the message (to the element S508:0560 in the USR segment in the group n).
The hexadecimal filter can be used for filtering, or the filter, which is defined in ISO 9735-5 (in R.1026 too) called UNO-A filter. The both filters meet UN/EDIFACT syntactic level A (they are universal). The selected filter is then used for all binary data in the message (except the certificate, where the filter is defined for the certificate).
The Hexadecimal filter represents one byte in the pair of characters ('0' - '9', 'A' -'F'), the first character represents upper 4 bits, the seconds bottom bits. The left characters represents more important bits in the hexadecimal transcription. No important 0's from the left can be missing.
Filter code has following values:
'2' - hexadecimal filter
'5' - UNO-A filter

0507 - Character ciphering
It defines the character ciphering of EDIFACT message before the digital signature application. The 8 bites ASCII is used (value '2'), it means the message must be in this ciphering where the digital signature is checked or created.

0509 - The role of the signing party (site)
see table Tab. 2

S500 - Site identification (the first appearance)
This is used for unique site identification, which created the digital message signature. It contains the site identity data and the key identification, which was used for the signature. Identical data are stated in the certificate, therefore it does not need to be specified.

S500:0577 - Site qualifier
see the table

S500:0538 - Key name
This contains the user secret key identification, which was used for the digital signature. The value of the key must be unique for all user's keys, for valid and deleted keys (incremental key numbering is recommended). The key value must be identical to the element value in the public key certificate (S500:0538 in segment USC) for the secret key and the specified public key certificate to connect.

S500:0511 - Partner's (Site) ID
This contains the organisation identification for EDI. The CNB assignees the EDI_ID value. EDI_ID identifies the organisation (site) in the EDI communication and can be different (if the organisation uses more EDI applications) from the application identification in the segment UNB (elements S002:0004 and S003:0010). This identification is used for key management purposes. It enables only one key for more application in organisation. The mapping between EDI_ID (which is only for organisation) and the application identification from UNB (when the organisation may have arbitrary number of application, differently identified) must be done in the implementation.

S500:0513 - Used list of partners (sites)

S500:0515 - Agency updating the list
Only one list is used in the actual application. The values do not need to be stated, the values in the table are default. They will be used later when more EDI application will be in the CNB.

S500:0586 Partner's (Site) name
This is designed for detailed site specification. It will not be used in the actual application. The use is assumed when the user will have more keys or run more EDI applications.

S500 - Site identification (the second appearance)
This element is bypassed

0520 - Reference number
This element contains a reference number which is used for sequence checking of signed messages. The ref_num value is numerical and is unique for the given user and for all sent messages. It is incremented for each sent message. It is recommended (if possible) to use the reference message number here (segment UNH element 0062).

S501 - Date and time
This element defines the date and time of the message signature. This element, together with 0520 element, is used to guarantee the integrity of message sequences.
Note.: The S501 values are controlled by ISO 8601 standard, in UTC offset (0336) the '+' is not stated for positive values (it is an UN/EDIFACT separator).

S501:0517 - Date and time qualifier
see table

S501:0338 - Date
The date value must have the specified format YYYYMMDD (e.g. 19950403).

S501:0314 - Time
The time value must have the specified format HHMMSS (e.g. 182033). The time value is the common time.

S501:0336 - UTC offset
This element is used for distinguishing between standard and daylight savings time. The offset value specifies the local time difference from the Greenwich Mean Time; + 1 hour (value'0100') in winter and +2 hours (value'0200') in the summer.

Note. The bad date data can affect security functions, therefore it is necessary to specify time according to the valid time.

GROUP 1 (C, 1 - 9) SEGMENT USA (C, 1)

Comp. Elem.	Element	M.	Format	Meaning	Content	Comments
S502		*M*		*Security algorithm*		*Hash message algorithm*
S502	0523	M	an..3	Algorithm using - code	'1'	The algorithm is used for the message hashing
S502	0525	C	an..3	Operational mode - code	'0'	There is no meaning for the given algorithm
S502	0533	O	an..3	Operational mode list	'1'	The list defined by UN/EDIFACT SJWG
S502	0527	C	an..3	Algorithm - code	'6'	MD5 algorithm (Rivest, Dusse - RSA Security Inc., 1991)
S502	0529	O	an..3	Algorithm list	'1'	The list defined by UN/EDIFACT SJWG
S503		*O*		*Algorithm parameters*		*Bypassed - there is no meaning for the given algorithm*
S503		*O*		*Algorithm parameters*		*Bypassed - there is no meaning for the given algorithm*
S503		*O*		*Algorithm parameters*		*Bypassed - there is no meaning for the given algorithm*
S503		*O*		*Algorithm parameters*		*Bypassed - there is no meaning for the given algorithm*
S503		*O*		*Algorithm parameters*		*Bypassed - there is no meaning for the given algorithm*

Table 3 (The security segments structure.)

Element description:
S502 - The security algorithm
This element describes the user's algorithm, which is used for message hashing and digital signature creating.

S502:0523 - The Algorithm use
see Table 3

S502:0525 - The Operational mode
see Table 3

S502:0533 - The operational mode list
This element defines the used operational modes list. The list, defined in the 1994 UN/TRADE/WP.4/R.1026 document (value '1'), is used in this case

S502:0527 - The Algorithm
This element defines the used algorithm. The detailed specification of the algorithm and its parameters is in the Used ciphering algorithms parameters chapter.

S502:0529 - The Algorithm list
This element defines the used algorithm list. The list, defined in the 1994 UN/TRADE/WP.4/R.1026 document (value '1'), is used in this case.

S503 - Parameters of algorithm
These elements are not used; the MD5 algorithm does not need any input parameters.

GROUP 2 (C, 1) SEGMENT USC (M, 1)

Comp. Elem.	Element	M.	Format	Meaning	Content	Comments
	0536	C	an..35	Certificate ref. number	ref_num	ref_num= certificate reference number - unique
S500		*C*		*Site identifier*		*Certificate owner identification*
S500	0577	M	an..3	Site qualifier	'3'	Certificate owner
S500	0538	C	an..35	Key name	key1	key1= certificate key number (name)
S500	0511	C	an..17	Site ID	EDI_ID	EDI_ID= EDI identification of key owner organisation
S500	0513	O	an..3	Used site list	'1'	Bank list code (EDI bank applications)
S500	0515	O	an..3	Updated agency list	'CNB'	Czech National Bank
S500	0586	O	an..35	Site name	org_name1	org_name1= organisation name
S500	0586	O	an..35	Site name	org_dep1	org_dep1= department (branch) in organisation
S500	0586	O	an..35	Site name	org_pers1	org_pers1= responsible person
S500		*C*		*Site identification*		*Certificate authority identification*
S500	0577	M	an..3	Site qualifier	'4'	CA, site acknowledging certificate validity
S500	0538	C	an..35	Key name	key2	key2= key number(name) for certificate number
S500	0511	C	an..17	Site ID	CA_ID	CA_ID= CA identification
S500	0513	O	an..3	Used site list	'1'	CA list code
S500	0515	O	an..3	Updated agency list	'CNB'	Czech National Bank
S500	0586	O	an..35	Site name	org_name2	org_name2= CA organisation name
S500	0586	O	an..35	Site name	org_dep2	org_dep2= department(branch) in organisation
S500	0586	O	an..35	Site name	org_pers2	org_pers2= responsible person
	0544	C	an..3	Certificate format version	'94W'	1994 version
	0505	C	an..3	Filter (function)-code	filter	Filter for binary data
	0507	C	an..3	Char. Ciphering-code	'2'	ASCII 8 bites
	0543	C	an..3	Char selection-code	'4'	UN/EDIFACT syntax level D
	0546	O	an..35	Level of rights	rights	rights= status word, defines the owner rights
S505		*O*		*Separators*		*Separators used for certificate signature*
S505	0550	C	an..4	Separator	'27'	Separator ' ' '
S505	0551	C	an..3	Separator qualifier	'1'	Segment separator

S505		*O*		*Separators*		*Separators used for certificate signature*
S505	0550	C	an..4	Separator	'2B'	Separator ' + '
S505	0551	C	an..3	Separator qualifier	'2'	Data element separator
S505		*O*		*Separators*		*Separators used for certificate signature*
S505	0550	C	an..4	Separator	'3A'	Separator ' : '
S505	0551	C	an..3	Separator qualifier	'3'	Composite data element separator
S505		*O*		*Separators*		*Separators used for certificate signature*
S505	0550	C	an..4	Separator	'3F'	Separator ' ? '
S505	0551	C	an..3	Separator qualifier	'1'	Release character
S501		*C*		*Date and time*		*Date and time for certificate*
S501	0517	M	an..3	Date and time qualifier	'2' or '6'	Creating/Deleting of certificate
S501	0338	C	n..8	Date	date1	date1= date, format YYYYMMDD
S501	0314	C	n..15	Time	time1	time1= time, format HHMMSS
S501	0336	C	n..4	UTC offset (Time difference)	offset1	offset1 = '0100' - difference from UTC is + 1 hour (standard time) offset1 = '0200' - difference from UTC is + 2 hour (daylight saving time)
S501		*C*		*Date and time*		*Date and time for certificate*
S501	0517	M	an..3	Date and time qualifier	'3'	Validity beginning from
S501	0338	C	n..8	Date	date2	date2= date, format YYYYMMDD
S501	0314	C	n..15	Time	time2	time2= time, format HHMMSS
S501	0336	C	n4	UTC offset (Time difference)	offset2	offset1 = '0100' - difference from UTC is + 1 hour (standard time) offset1 = '0200' - difference from UTC is + 2 hour (daylight saving time)
S501		*C*		*Date and time*		*Date and time for certificate*
S501	0517	M	an..3	Date and time qualifier	'4'	Valid to
S501	0338	C	n..8	Date	date3	date3= date, format YYYYMMDD
S501	0314	C	n..15	Time	time3	time3= time, format HHMMSS
S501	0336	C	n4	UTC offset (Time difference)	offset3	offset1 = '0100' - difference from UTC is + 1 hour (standard time) offset1 = '0200' - difference from UTC is + 2 hour (daylight saving time)
	0567	C	an..3	Security status	status	status= certificate status

Table 4 (The security segments structure.)

The segment group 2 defines the user's certificate. Group 2 is repeated only once in the message. This certificate is fully created in the CNB during the public key certification. The data given by the user, e.g. user identification, key identification, public user key and the data given during certification in the CNB, for example the certificate reference number and the validity time data, are stated in the certificate. The certificate is not subsequently possible to change; the data inside the certificate are used only for getting digital signature data.

The certificate is equipped with the digital signature of the certification authority. The digital signature of the certificate is similar to the message digital signature. The certificate text is first processed by the hash function (MD5); its result is a short checking byte block. The certificate text scope, which is input to the hash function, is as follows: from the USC

segment first character (‘U’) to the segment separator (character ' ' ') behind the last USA segment repetition (in group 2). The checking byte block is ciphered by RSA algorithm by using the secret CA key, and this result is then filtered and stated in the USR segment of group 2.

The complete certificate can be sent with a message (all of group 2), or with only the reference certificate number (in the case, the second side already has appropriate certificate), i.e. only the USC segment, which contains only one element 0536. The second way will be used for the CNB application. Certificates are exchanged between banks and CNB by the procedure described in the Key management chapter.

The element description

0536 - Certificate reference number
This element contains the certificate reference number. The ref_num value is unique for all certificates in the system, valid and non valid. The element is filled in CNB during certification.

S500 - Site identification (first appearance)
This is used for unique certificate owner identification. It contains owner identity data which are delivered to the CNB with the public key for key certification. It contains the public key identification.

S500:0577 - Site qualifier
see table

S500:0538 - Key name
This contains the user’s public key identification, which is included in the certificate. The key1 value must be unique for all user’s keys, valid and deleted too (the incrementing of key numbering is recommended).

S500:0511 - Site ID
This contains the organisation identification for EDI. The EDI_ID value is assigned by the CNB. EDI_ID identifies organisation (site) in the EDI communication and can be different (in the case, that the organisation uses more EDI applications) from the application identification in the UNB segment (elements S002:0004 and S003:0010). The identification is used for the key management purposes. It allows the use of one key for more EDI applications inside the organisation. The mapping between EDI_ID (only for the organisation) and the application identification from the UNB (when the organisation may have an arbitrary number of differently identified applications) must be done in the implementation.

S500:0513 - Used site list

S500:0515 - Updated list agency
Only one list is used in the actual application. The values do not need to be stated, the values in the table are default. They will be used later when more EDI application will be in the CNB.

S500:0586 Partner’s (Site) name
This is designed for more detailed site specification. Values in the elements are stated only when being delivered by a user. The use is assumed when the owner owns more keys or runs more EDI applications.

S500 - Site name (second appearance)
This element is intended for identification of the Certification Authority, i.e. the partner (site) which made the digital signature of the certificate.

S500:0577 - Site qualifier
see table

S500:0538 - Key name
This contains identification of the CA public key from the pair of keys which was used for certificate digital signature. The key2 together with CA_ID are used for identification of the CA key certification, which shall be used for certificate signature check.

S500:0511 - Partner's (Site) ID
This contains the CA identification. This identification is used for key management purposes. Formally it is equal to the EDI_ID of the organization.

S500:0513 - Used site list
S500:0515 - Updated list agency
Only one list is used in the actual application. The values do not need to be stated; the values in the table are default. They will be used later when more EDI application will be in the CNB.

S500:0586 Partner's (Site) name
It is designed for detailed CA specification. They will not be used for the present.

0544 - Certificate format version
The value '94W' defines that for the certificate the service security segments described in the UN/TRADE/WP.4/R.1026 and ISO/CD 9735 - 5 documents are used. The element is filled in at the certification.

0505 - Filter (function)
This defines the function type which is used for the binary data filtering. The data are the result of the certificate digital signature (element S508:0560, USR segment, group 2) and the binary data representing the key (element S503:0532, USA segment, group 2) before their recording into the certificate.
The hexadecimal filter can be used for filtering, or the filter defined in ISO 9735-5 (in R.1026 too) called UNO-A filter. Both filters fully meet UN/EDIFACT syntactic level A (they are universal). The selected filter is then used for all binary data in the certificate.
The Hexadecimal filter represents one byte in the pair of characters ('0' - '9', 'A' -'F'), the first character represents the upper 4 bits, the second the lower ones. The left characters represent more important bits in the hexadecimal transcription. The unimportant 0's from the left can be omitted.
Filter code has the following values:
'2' - hexadecimal filter
'5' - UNO-A filter
The element is filled in at the certification.

0507 - Character ciphering
This defines the character ciphering used for the recording of certificate segments before the digital signature application. The 8 bites ASCII is used (value '2'), it means that the certificate is in this ciphering when the Certificate Authority creates its signature. The element is filled in at the certification.

0543 - Character selection
This element defines the selection of the character set for the certificate segments. Here, it is specified that the characters are used according to the UN/EDIFACT syntactic level A (value '1'). Since syntactic level A is usually used for EDIFACT application, this element will not be used and the value '1' will be considered as default. Its use is assumed in later versions.

0546 - Level of rights
This element defines the status word which specifies the key use included in the certificate, the rights of the key owner etc. The value of the rights is alphanumeric, the first character from the left is mandatory, the others are optional; the format of the rights is as follows:
'AB-CCCC-DDDD..' where
'A' - is a character defining the use of the key and has the following values:
'A' - the key is intended for digital signature and ciphering of symmetric keys
'B' - the key is intended only for digital signature
'C' - the key is intended only for ciphering of symmetric keys
'Z' - the key has an other use
'B' - is a character defining the level of user's rights and has values in the range of '0' - '9' where:
'0' - is the lowest level of rights
'1' - '8' - are graduated levels of rights
'9' - is the highest level of rights
'-' - is a separator character, it occurs only when further characters follow it
'C' - are four characters reserved for future need
'-' - is a separator character, it occurs only when further characters follow it
'D' - are characters delivered by the certificate owner for his needs, there can be placed up to 27 characters.
This element is not used in this application and the default value is 'A0'. Later, the use is assumed in new EDISEC2 versions.

S505 - Separators
These elements are not used. Standard separators (which also represent default) are used. Later, the use for specific applications is assumed.

S501 - Date and time (first appearance)
This element defines the date and time of the certificate creation (qualifier 0517 = '2') or the date and time of certificate cancellation (qualifier 0517 = '6'). The data are complemented by CA at the creation/cancellation of the certificate.

S501:0517 - Date and time qualifier
see the table

S501:0338 - Date
The date 1 value must have the specified format YYYYMMDD (e.g. 19950403).

S501:0314 - Time
The time1 value must have the specified format HHMMSS (e.g. 182033). The time value represents the common time.

S501:0336 - UTC offset
This element is used for distinguishing between standard and daylight saving time. The offset 1 value specifies the local time difference from the Greenwich Mean Time, + 1 hour (value'0100') in the winter and +2 hours (value'0200') in the summer.

S501 - Date and time (second appearance)
This element defines the date and time of the certificate validity start. The data are complemented by CA at the creation of the certificate. See above for a description of the simple elements.

S501 - Date and time (third appearance)
This element defines the date and time of the certificate validity start. The data are complemented by CA at the creation of the certificate. See above for a description of the simple elements.

0567 - Security status
This element defines the certificate status; if the certificate is valid this element is not stated its default values '1' (see below). The status value may be as follows:
'1' - certificate is valid
'2' - certificate is canceled (for security reasons)
'3' - the certificate status is unknown
'4' - certificate is terminated (for formal reasons)
'5' - certificate is suspicious
'6' - certificate is expired

GROUP 2 (C, 1) SEGMENT USA (C, 1)

Comp. Elem.	Element	M.	Format	Meaning	Content	Comments
S502		*M*		*Security algorithm*		*Algorithm for digital signature of the certificate owner*
S502	0523	M	an..3	Algorithm use - code	'6' or '7'	Algorithm is used for message digital signature or for symmetric key ciphering
S502	0525	C	an..3	Operating mode - code	'0'	No meaning for the given algorithm
S502	0533	O	an..3	List of operating modes	'1'	List is defined by UN/EDIFACT SJWG
S502	0527	C	an..3	Algorithm - code	'10'	Algorithms RSA(Rivest, Shamir, Adleman,1978)
S502	0529	O	an..3	List of algorithms	'1'	List is defined by UN/EDIFACT SJWG
S503		*C*		*Algorithm parameters*		*Parameters for RSA*
S503	0532	C	an..512	Parameter value	length	length= length of the module,decimal
S503	0531	C	an..3	Parameter qualifier - code	'14'	length of the module
S503		*C*		*Algorithm parameters*		*Parameters for RSA*
S503	0532	C	an..512	Parameter value	exp	exp= exponent for RSA algorithm, filtered
S503	0531	C	an..3	Parameter qualifier - code	'13'	exponent of the public key
S503		*C*		*Algorithm parameters*		*Parameters for RSA*
S503	0532	C	an..512	Parameter value	mod	mod= module for RSA algorithm, filtered
S503	0531	C	an..3	Parameter qualifier - code	'12'	module of public key
S503		*O*		*Algorithm parameters*		*Omitted - no meaning for given algorithm*
S503		*O*		*Algorithm parameters*		*Omitted - no meaning for given algorithm*

Table 5 (Structure of the security segments.)

This segment contains the public key of the certificate owner and algorithm parameters for which the key is determined.

The element description:

S502 - Security algorithm
This element describes the asymmetric ciphering algorithm (RSA) used for digital signature of messages sent by user (code 0523 = '6') or used for the digital signature and at the same time for symmetric keys ciphering (code 0523 = '7').

S502:0523 - Algorithms use
see table

S502:0525 - Operating mode
see table

S502:0533 - List of operating modes
This element defines the list of operating modes. In this case the list defined in the 1994 UN/TRADE/WP.4/R.1026 (value '1') is used.

S502:0527 - Algorithm
This element defines the used algorithm. The detailed specification of the algorithm and its parameters is specified in the chapter Parameters of the used ciphering algorithms.

S502:0529 - List of algorithms
This element defines the used list of algorithms. In this case the list defined in the 1994 UN/TRADE/WP.4/R.1026 (value '1') is used.

S503 - Algorithm parameters (first appearance)
This element defines the length of the module for the RSA algorithm.

S503:0532 - Parameter value
The length value defines the module length in bits The value is stated in standard decimal notation

S503:0531 - Parameter qualifier
see table

S503 - Algorithm parameters (second appearance)
This element defines the exponent of the certificate owner public key for RSA algorithm.

S503:0532 - Parameter value
The exp value represents the public key exponent. The value is stated as filtered one (see the description of the element 0505 in the segment USC). For the used fix exponent (Fermat's number F4) is the hexadecimal exp value '10001'.

S503:0531 - Parameter qualifier
see table

S503 - Algorithm parameters (third appearance)
This element defines the exponent of the certificate owner public key for RSA algorithm.

S503:0532 - Parameter value
The mod value represents the public key module. The value is stated as filtered one (see the description of the element 0505 in the segment USC).

S503:0531 - Parameter qualifier
see table

S503 - Algorithm parameters (further appearances)
These elements are not used; for the RSA algorithm no further parameters are needed.

GROUP 2 (C, 1) SEGMENT USA (O, 2)

Comp. Elem.	Element	M.	Format	Meaning	Content	Comments
S502		*M*		*Security algorithm*		*Algorithm for certificate hash (made by CA)*
S502	0523	M	an..3	Algorithm use - code	'4'	Algorithm for certificate hashing
S502	0525	C	an..3	Operating mode - code	'0'	No meaning for given algorithm
S502	0533	O	an..3	List of operating modes	'1'	List defined by UN/EDIFACT SJWG
S502	0527	C	an..3	Algorithm - code	'6'	Algorithm MD5 (Rivest, Dusse - RSA Security Inc., 1991)
S502	0529	O	an..3	List of algorithms	'1'	List defined by UN/EDIFACT SJWG
S503		*O*		*Algorithm parameters*		*Omitted - no meaning for given algorithm*
S503		*O*		*Algorithm parameters*		*Omitted - no meaning for given algorithm*
S503		*O*		*Algorithm parameters*		*Omitted - no meaning for given algorithm*
S503		*O*		*Algorithm parameters*		*Omitted - no meaning for given algorithm*
S503		*O*		*Algorithm parameters*		*Omitted - no meaning for given algorithm*

Table 6 (Structure of the security segments.)
This segment contains a description of the algorithm used by Certification Authority for the certificate hash for its digital signature. This segment is not used in the EDISEC2 application, because a standard use of the agreed algorithm is assumed.

GROUP 2 (C, 1) SEGMENT USA (O, 3)

Comp. Elem.	Element	M.	Format	Meaning	Content	Comments
S502		*M*		*Security algorithm*		*Certificate signature algorithm (made by CA)*
S502	0523	M	an..3	Algorithm use - code	'3'	Algorithm is a certificate digital signature
S502	0525	C	an..3	Operating mode - code	'0'	No meaning for given algorithm
S502	0533	O	an..3	List of operating modes	'1'	List defined by UN/EDIFACT SJWG
S502	0527	C	an..3	Algorithm - code	'10'	RSA algorithm (Rivest, Shamir, Adleman,1978)
S502	0529	O	an..3	List of algorithms	'1'	List defined by UN/EDIFACT SJWG
S503		*C*		*Algorithm parameters*		*Parameters for RSA*
S503	0532	C	an..512	Parameter value	length	length= module length, decimal
S503	0531	C	an..3	Parameter qualifier - code	'14'	module length
S503		*C*		*Algorithm parameters*		*Parameters for RSA*
S503	0532	C	an..512	Parameter value	exp	exp= exponent for RSA algorithm, filtered
S503	0531	C	an..3	Parameter qualifier -	'13'	exponent of the public key

				code		
S503		*C*		*Algorithm parameters*		*Parameters for RSA*
S503	0532	C	an..512	Parameter value	mod	mod= module for RSA algorithm, filtered
S503	0531	C	an..3	Parameter qualifier - code	'12'	module of the public key
S503		*O*		*Algorithm parameters*		*Omitted - no meaning for given algorithm*
S503		*O*		*Algorithm parameters*		*Omitted - no meaning for given algorithm*

Table 7 (Structure of the security segments.)
This segment contains the description of the asymmetric ciphering (RSA) algorithm used by the Certification Authority for certificate digital signature creation and also the CA public key, which is used for the certificate signature. This segment is exceptionally used in the certificate of the CA public key. The segment replaces the USA segment with the public key of the subject (however practically it means only the change of the qualifier S502:0523 in the USA segment included in certificate). The certificate of the CA public key is distributed by a method described in the Key management chapter.

GROUP 2 (C, 1) SEGMENT USR (C, 1)

Comp. Elem.	Element	M.	Format	Meaning	Content	Comments
S508		*M*		*Result of the sec. function*		*Certificate digital signature*
S508	0560	M	an..256	Resulting value	sig_val	sig_val = result of the digital signature, filtered
S508	0560	O	an..256	Resulting value		Omitted

Table 8 (Structure of the security segments.)

The element description:

S508 - Result of the security function
This contains the result of the certificate digital signature.

S508:0560 - Resulting value (initial appearance)
The sig_val value contains the result of the digital signature in the filtered (text) form (see the description of the element 0505 in the segment USC).

S508:0560 - Resulting value (second appearance)
This element is not used.

GROUP n (C, 1 - 9) SEGMENT UST (M, 1)

Comp. Elem.	Element	M.	Format	Meaning	Content	Comments
	0534	M	an..14	Check reference	link	link='01' for one signature

Table 9 (Structure of the security segments.)

The element description:

0534 - Checking reference
This element is used as an unambiguous key for connecting group 1 (Security Header) and group n (Security Trailer) - i.e. parameters defined in group 1 are related to group n, which has the same element value 0534. For this application the use of one digital signature is intended - the link is then '01'. For further group repetition (more signatures) the link is incremented.

GROUP n (C, 1 - 9) SEGMENT USR (C, 1)

Comp. element	Element	M.	Format	Meaning	Content	Comments
S508		*M*		*Security function result*		*Message digital signature*
S508	0560	M	an..256	Resulting value	sig_val	sig_val = digital signature result, filtered
S508	0560	O	an..256	Resulting value		Omitted

Table 10 (Structure of the security segments.)

The signature result contained in this segment is bound with parameters in the segment USH and with the certificate (group 2) by the segment UST (see above).

The element description:

S508 - Security function result
This contains the result of the message digital signature.

S508:0560 - Resulting value (initial appearance)
The sig_val value contains the digital signature result in a filtered (text) form (see the description of the element 0505 in the segment USH).

S508:0560 - Resulting value (second appearance)
This element is not used.

Example of a signed message

Segments	Comments
UNB+UNOD:2+BANK+CNBASUD+960521:2002+000010033'	Header of the interchange file, it is the interchange file from BANK application to the CNBASUD application.
UNH+236+GESMES:D:95A:UN'	Header of the message, it is a GESMES message, the reference number is 236.
USH+94W+1+01+1+2+2+2+1+++236+1:19960521:200246:0200' USA+1:0:1:6:1' USC+CATEST000000021'	Initial security segments. USH segment defines the function type (digital signature), the signature reference (01), acknowledgment requirement (code 2), the used filter function (hexadecimal), the secure copy of the message ref. number, timestamp of the signature creation. The USA segment defines the used hash function (MD5). The USC segment contains a reference to the certificate, which must be used by signature checking.
BGM+:::Issue of the statistic occurence+ZKUS_96.01+X05' DTM+137:960521:102' DSI+ROSIFE20.04.00.197' STS+X09+X01' ARR+X00+6700' ARR+X00+19960331' ARR+X00+1' ARR+X00+30000000.00' ARR++2' ARR++40000000.00' ARR++3' ARR++30000000.00'	The actual message body, it is made by the user's segments which were in the message before securing.
UST+01' USR+64E13B655B71EEEA17353D99A443B9BF015A6C2BF856A3B38DAB0502D0F00AB7C44EA791A39F4295A17B3D2130FD8E273BD93444C03847C7C8A7CE8DB17EA8D6786D94209C6654CE947BDC7FDA3B0ED331E2520B8C76AA262E60F8B66FC5A85520F368A617875750805E00E6482FFFE7615C73561C815B271AAAF07E3E24F0B1'	Trail security segments. The UST segment binds the trail security segments with the header by using a signature reference (01). The USR segment contains the digital signature of the message in a filtered form.
UNT+19+236'	The message footer, the message contains 19 segments (including the service one).
UNZ+1+000010033'	The interchange file footer, the interchange file contains 1 message.

Tab. 11 (Example of the interchange file with signed message (segments are here separated by CRLF for clear arrangement, in the actual interchange file are following directly one after the other)

A certificate example, which is referred to in the message (i.e. must be used for message check):

Segments	Comments
USC+CATEST000000021+3:BANK_KEY1:BANK:::::OWNER+4:CA_KEY0:CA+94W+2+2+4++++++2:19951215:093243:0200+3:19960101:000000:0100+4:19960701:000000:02000'	The USC segment contains the certificate number (CATEST000000021), the organization identification (BANK) and the certificate owner (OWNER), owner's keys (BANK_KEY1), CA identification (CA), the CA key used for certificate signature (CA_KEY0), the filter type for the public key and the signature and further date and time of the certificate creation, start of the certificate validity and end of the certificate validity.
USA+7:0:1:10:1+1024:14+010001:13+C1164701726B49F75B9CAB59E0BF9F28657D78ADCA738EC70EF256F9657272602ABD32E4F4AF731F0BC4515D9EE1B07638E1CBDB94D791A7463DE2E2AC62B009040B9F4E7D47B5EF9594E91E4D3136421D876BC5552C4A87BD4DB14C6A271C257C71A6D44F3E3427D03CE9D36B0904D50834C5B99E31A7815E11CE17ED8AD2CD:12'	The USA defines user signature algorithm (RSA) and contains a public key; values are filtered: key length 1024 bits, exponent (010001H) and modulus (C1164...H) of the key.
USR+61098948944B7A62BA67389DEB73273D4DDA56BCCE593F4E2CA32870E09ECB3833ADCDA276B92D329C6E051BBDEF6B90529EAEF16D096356292EB6CC14EFFCA912006EE6536BB5593E8C6BC8EC156A64A88C518394A1DE60109FEDA9C4A36B6EA37D9F1CDD5A21698A4176C767E3D5F5C6CF9ED765DF5AFBC81DFCC098F3F6F7'	USR segment contains a description of the certificate, which is filtered.

Table 12 (A certificate example, which is referred to in the message.)

The AUTACK message

The principle of using the AUTACK message

AUTACK is an acknowledgment message which makes possible the safe acknowledgment of receipt of a certain message or providing information on a security error which occurred while checking the message. AUTACK contains references to messages received and a hash code of messages received (so-called fingerprint). The hash code is used in order not to reject the content of the message. The AUTACK message is provided with the signature of the original message's recipient. The signature enables the functioning of receipt non-rejection. AUTACK makes possible the authentication of even more messages.

The following figure shows the creation of an authentication message AUTACK for messages received within one interchange file:

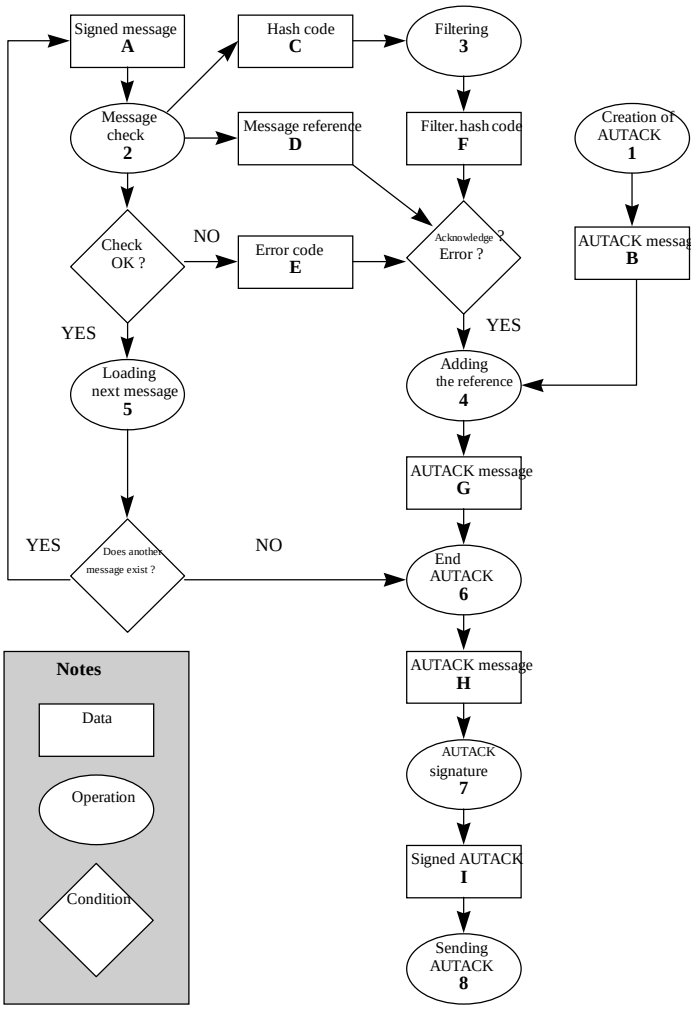

Fig. 5 (The scheme of creating the AUTACK message.)

The creation of AUTACK requires the following steps:

1. If some received messages are to be acknowledged, an interchange file is created which is intended for the sender of the interchange file being processed. This contains an AUTACK [B] message into which the required references will be entered (see below).
2. One signed message [A] is obtained from the received interchange file and checked. The result of the check (see chapter Principle of Creating and Checking the Digital Signature) is the message hash code [C] and the message reference [D]. If an error is detected while checking the message, an error code [E] is created.

3. The hash code [C] is filtered into the text form so that it can be transmitted in the UN/EDIFACT message.
4. If the checked message is to be acknowledged (this is determined in the opening security segments), or an error has been detected while checking the message, the reference of a given message [D] is added together with the filtered hash code [C] (if it has been generated) and the error code [E] (if an error occurred) into the body of the AUTACK message [B].
5. After the message is checked, another signed message from the interchange file is loaded, the message is checked again and acknowledged as described above - items 2-4. If there is no other signed message in the interchange file, item 6 shall be the next step.
6. The AUTACK message [G] is ended.
7. The AUTACK message [H] is provided with the digital signature of the recipient using standard techniques (see chapter Security Header and Trailer for the Digital Signature).
8. The signed AUTACK message [I] is sent to the sender of the checked interchange file.

The sender of the original message can check from the references and from the fingerprint of the original message whether the message has been received correctly. Checking the AUTACK signature he finds out whether the message has been transmitted to the party entitled to receive it. AUTACK is proof of receipt. Or, if the message has not been further processed, the sender can identify the reason for this in the error code.

Syntactic rules and formal rules for the AUTACK message

AUTACK is implemented according to the recommendations UN/EDIFACT UN/TRADE/WP.4/R.1026/Add.3 and Add.4 and according to ISO/CD 9735-6.

This message informs the sender of any message on the results of the acknowledgment of security functions carried out by the recipient. The AUTACK message responds to the transmitted message for which the acknowledgment requirement has been set (element 0503 in the USH segment - i.e. that the message received must be signed - see chapter Security Header and Trailer for the Digital Signature), thus providing the function of non-rejection of receipt. The sender is therefore informed that his/her message has been received and the signature acknowledged. AUTACK does not check the factual correctness of the transmitted message - this is carried out by a special protocol at the application level. AUTACK can also be used as a report on an error in message securing check or an error in message deciphering. If this is the case, the error code in S508:571 in the USY segment in group 3 relating to the referred message must be filled in appropriately.

If used for acknowledgment of the receipt, the AUTACK message must include the hash code of the message received (element S560:571 of the USY segment in group 3 relating to the referred message). This is created while checking the message signature using the routine described in the chapter Principle of Creating and Checking the Digital Signature. If the AUTACK message gives information on a securing error, it can contain a hash code only in case the hash code could have been created. The hash code included in the AUTACK message is filtered by the same algorithm that has been used for filtering the signature of the given message.

An AUTACK message must always be provided with the digital signature of the original message recipient (i.e. the AUTACK sender). AUTACK is secured via the opening and end security segments as with any other message (see chapter Security Header and Trailer for the Digital Signature), i.e. opening and end segments are added to the message to be used for the signature.

The AUTACK message can respond to more transmitted messages. These messages, however, must be a part of one interchange file.

AUTACK is not acknowledged by the AUTACK message any more (cycling could occur).

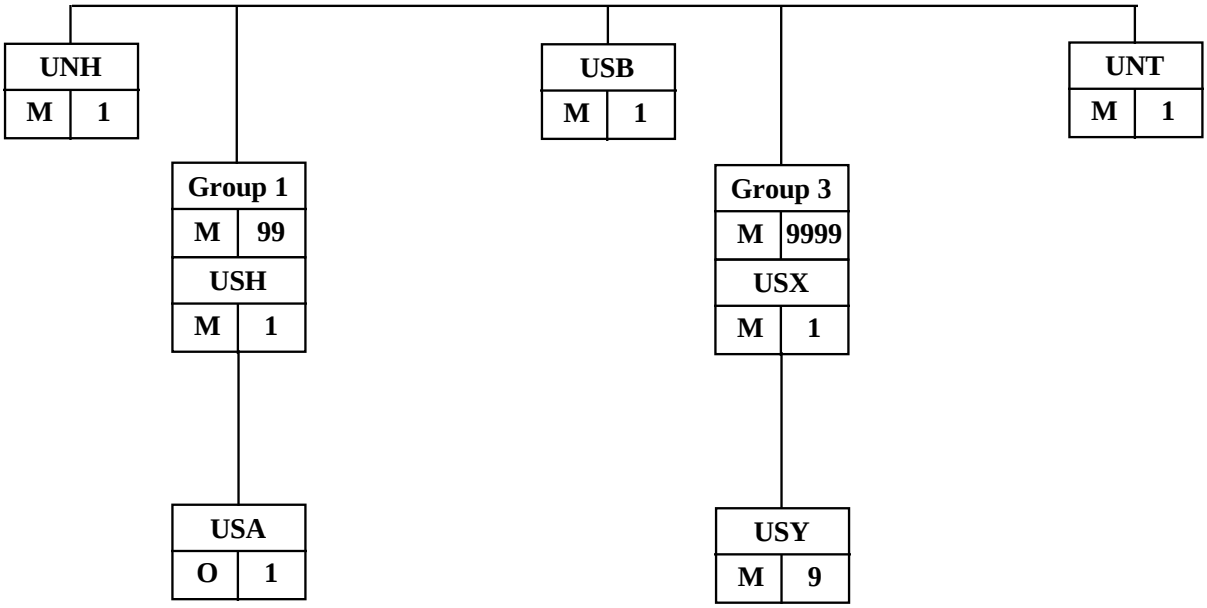

Fig. 6 (The structure of the AUTACK message.)

M/C - mandatory (M), use non-mandatory (C), not commonly used (O) segment (group)
Op. - number of recurrences, the maximum number allowed by the standard is indicated in the brackets.

GROUP SEGMENT	M/C	Op.	DESCRIPTION
1	M	1(99)	This group of segments is used to identify the security mechanism used for a response to the received message.
USH	M	1	Defines security services used for the response to the message received.
USA	O	1	Algorithm used for hashing the received message.
USB	M	1	Provides the identification of involved parties and contains a time mark.
3	M	1(9999)	Defines the messages to which AUTACK responds as well as the result of security functions applied on receipt.
USX	M	1	Contains references to the message received.
USY	M	1(9)	Contains the results of message processing

Table 13 (Detailed description of individual segments.)

S.Prv. - The number of the composite element in the UN/EDIFACT Standard Directory
Element - The number of the element in the UN/EDIFACT Standard Directory
P. - Mandatory (M), use non-mandatory (C), not commonly used (O) segment, element
Format - Specification of the format according to UN/EDIFACT conventions
Content - constants are specified in ' ', text identifiers refer to variable values supplied by the security application

SEGMENT UNH (M, 1)

Comp. Elem.	Element	P.	Format	Meaning	Content	Commentary
	0062	M	an..14	Message ref. number	ref_no	ref_no = ref. number, unambiguous
S009		*M*		*Message identifier*		*Identifies the type of the UN/EDIFACT message*
S009	0065	M	an..6	Message type	'AUTACK'	
S009	0052	M	an..3	Message version	'1'	
S009	0054	M	an..3	Version number	'4'	
S009	0051	M	an..2	Responsible agency	'UN'	
S009	0057	O	an..6	Special code		Skipped
	0068	O	an..35	Joint reference		Skipped
S010		*O*		*Transmission status*		*Skipped*

Table 14 (The structure of AUTACK message segments.)

UNH is the standard service segment which is used by all UN/EDIFACT messages. Its usage is specified by standard UN/EDIFACT rules.

GROUP 1 (M,1) SEGMENT USH (M, 1)

Comp. Elem.	Element	P.	Format	Meaning	Content	Commentary
	0552	M	an..3	Version of segment structure	'94W'	The 1994 version
	0501	M	an..3	Security function - code	'5'	Non-rejection of receipt
	0534	M	an..14	Control reference	'00'	Number 00 (to avoid collision with the USH of the message received)
	0541	O	an..3	Security extent - code	'1'	Opening security segments + body of message 1)
	0503	O	an..3	Type of response - code		Skipped
	0505	O	an..3	Filter (function) - code	filter	Filter for binary data 1)
	0507	O	an..3	Coding of characters - code	'2'	ASCII 8-bit 1)
	0509	O	an..3	Role of the party - code	'1'	Sender of the document
S500		*O*		*Party identification*		*Skipped*
S500		*O*		*Party identification*		*Skipped*
	0516	O	an..35	Reference number		Skipped
S501		*O*		*Date and time*		*Skipped*

Table 15 (The structure of AUTACK message segments.)

1) These elements copy the elements from the Security Header of the message received. It is therefore not necessary to indicate them; the message sender knows them implicitly.

Group 1 contains data on the function of the AUTACK message and the parameters for creating acknowledgment references. This group appears only once in the implementation.

Elements description:

0552 - Version of segments structure
Value '94W' states that service security segments described in the UN/TRADE/WP.4/R.1026 and ISO/CD 9735 - 5 documents are used.

0501 - Security function
The AUTACK message carries the non-rejection of receipt function (value '5').

0534 - Control reference
This element serves as a non-ambiguous key for joining groups 1 (Security Header) and 3 (USY segment). The index is however not being used because the USY segment is referred to the Security Header of the message received. The link value is '00' to avoid collisions with the indexes from the message received.

0541 - Security extent
Value '1' states that the hash of the message received is calculated from the text of opening security segments (groups 1 and 2) - from the first letter of the USH segment (i.e. 'U') up to (and including) the separator ending these segments, and from the text of the message body which is directly added - from the first character following after the separator concluding the opening security segments (i.e. 'B' from the BGM segment) up to (and including) the separator placed prior to the end security segments. If there is only one signature, the hash function is applied to a continuous text from the 'U' segment USH up to ' ' ' placed before the UST segment.
This element copies the value of the received message. It is therefore not necessary to indicate it.

0503 - Type of response
The element is skipped (the AUTACK messages must not be acknowledged by the AUTACK message any longer - a cycle could occur).

0505 - Filter (function)
States the type of function used for filtering binary data resulting from hashing the message received before the data is entered into the AUTACK message (into the S508:0560 element in the USY segment, group 3).
Filtering can be done either using a hexadecimal filter or a filter defined in ISO 9735-5 (also in R.1026), so-called UNO-A filter, both of them being fully applicable for the UN/EDIFACT syntactic level A (multi-purpose filters). Chosen filter is then applied to all binary data in the AUTACK message.
The hexadecimal filter represents one byte by a pair of characters ('0' - '9', 'A' -'F'), the first one representing the upper 4 bites and the second representing the lower. Characters on the left side of the hexadecimal record represent more important bytes. Non-relevant zeroes on the left can be skipped.
The filter code has the following values:
'2' - hexadecimal filter
'5' - UNO-A filter
This element copies the value from the received message and therefore need not be indicated.

0507 - Coding of characters
Specifies coding the characters of the received EDIFACT message before using the hash function. 8-bit ASCII is applied here (value '2').
This element copies the value from the received message. It is therefore not necessary to indicate it.

0509 - Role of the signing party
The party is the sender of the AUTACK message (value '1')

S500 - Party identification (first recurrence)
S500 - Party identification (second recurrence)
0516 - Reference number
S501 - Date and time
These elements are skipped.

GROUP 1 (M, 1)SEGMENT USA (O, 1) 1)

Comp. Elemm	Element	P.	Format	Meaning	Content	Commentary
S502		*M*		*Security algorithm*		*Algorithm for the hash of the message received*
S502	0523	M	an..3	Application of the algorithm - code	'1'	The algorithm is used for hashing the message
S502	0525	C	an..3	Operation mode - code	'0'	No relevance for given algorithm
S502	0533	O	an..3	List of operation modes	'1'	List defined by UN/EDIFACT SJWG
S502	0527	C	an..3	Algorithm - code	'6'	Algorithm MD5 (Rivest, Dusse - RSA Security Inc., 1991)
S502	0529	O	an..3	List of algorithms	'1'	List defined by UN/EDIFACT SJWG
S503		*O*		*Algorithm parameters*		*Skipped - no relevance for given algorithm*
S503		*O*		*Algorithm parameters*		*Skipped - no relevance for given algorithm*
S503		*O*		*Algorithm parameters*		*Skipped - no relevance for given algorithm*
S503		*O*		*Algorithm parameters*		*Skipped - no relevance for given algorithm*
S503		*O*		*Algorithm parameters*		*Skipped - no relevance for given algorithm*

Table 16 (Security segments structure.)

This segment is copied from the Security Header of the received message and it is therefore not necessary to indicate it. The sender knows its elements implicitly.

Elements description:

S502 - Security algorithm
The element describes the algorithm of the user applied for hashing the received message to create the fingerprint which is then indicated in the USY segment.

S502:0523 - Application of the algorithm
see table

S502:0525 - Operation mode
see table

S502:0533 - List of operation modes
The element defines the list of operation modes used. In this case the list defined in the material UN/TRADE/WP.4/R.1026 from 1994 is used (value '1').

S502:0527 - Algorithm
The element defines the algorithm used. Detailed specification of the algorithm and its parameters is contained in the chapter Parameters of Applied Cryptographic Algorithms.

S502:0529 - List of algorithms
The element defines the list of algorithms used. In this case the list defined in the material UN/TRADE/WP.4/R.1026 from 1994 is used (value '1').

S503 - Algorithm parameters

These elements are not used. The MD5 algorithm does not need any input parameters.

SEGMENT USB (M, 1)

Comp. Elem.	Element	P.	Format	Meaning	Content	Commentary
	0503	M	an..3	Type of response - code	'1'	The message shall not be acknowledged by the AUTACK message
S501		*C*		*Date and time*		*Date and time of creating the AUTACK message*
S501	0517	M	an..3	Date and time qualifier	'1'	Time mark
S501	0338	C	n..8	Date	date	date= date, YYYYMMDD format
S501	0314	C	n..15	Time	time	time= time, HHMMSS format
S501	0336	C	n4	UTC offset (time offset)	offset	offset = '0100' - offset from UTC is + 1 hour (winter time) offset = '0200' - offset from UTC is + 2 hours (summer time)
S002		*C*		*Interchange file sender*		*Identification of the AUTACK message sender*
S002	0004	M	an..35	Identification of the sender	send	send = sender (according to the identification element of the UNB segment)
S002	0007	O	an..4	Identification qualifier	qual_s	qual_s = a copy from the UNB segment (if used)
S002	0008	O	an..35	Identification of the sender - 2. level		Skipped
S002	0040	O	an..35	Identification of the sender - 3. level		Skipped
S003		*C*		*Interchange file recipient*		*Identification of the AUTACK message recipient*
S003	0010	M	an..35	Recipient identification	rec	rec = identification (according to the identification element of the UNB segment)
S003	0007	O	an..4	Identification qualifier	qual_r	qual_r = copy from the UNB segment (if used)
S003	0014	O	an..35	Recipient identification - 2. level		Skipped
S003	0044	O	an..35	Recipient identification - 3. level		Skipped

Table 17 (Security segments structure.)

Elements description:

0503 - Type of response
This element states that the AUTACK message receipt must not be acknowledged by the AUTACK message (value '1') any longer - a cycle could occur.

S501 - Date and time
This element contains the date and time of the AUTACK message creation.

S501:0517 - Date and time qualifier
see table

S501:0338 - Date
The date value must have the specified YYYYMMDD format (e.g. 19950403).

S501:0314 - Time
The time value must have the specified HHMMSS format (e.g. 182033). The time value represents the time used in the Czech Republic.

S501:0336 - UTC offset
This element is used for distinguishing between summer time and winter time. The offset value states the offset of local time from the standard world time, i.e. +1 hour for the winter time (value '0100') and + 2 hours for the summer time (value '0200').

S002 - Interchange file sender
This element contains the identification of the AUTACK message sender. The data is taken from the heading of the interchange file in which AUTACK is being sent (or from the interchange file in which the message is referred in case the message recipient is the AUTACK sender).

S002:0004 - Identification of the sender
The element contains the application ID of the message sender. The send value is copied from the S002:0004 element from the UNB sender of the interchange file sent (or the S003:0010 UNB element of the file received).

S002:0007 - Identification qualifier
If the qual_s value indicated is copied from the S002:0007 element of the UNB segment of the interchange file being sent (or the S003: 0007 element of the received UNB file).

S002:0008 - Identification of the sender - 2nd level
S002:0040 - Identification of the sender - 3rd level
These elements are skipped.

S003 - Interchange file recipient
This element contains the application ID of the AUTACK message recipient. The data is taken from the heading of the interchange file in which AUTACK is being sent (or from the interchange file in which the message is referred in case the message sender is the AUTACK recipient).

S003:0010 - Recipient identification
The element contains the message recipient ID. The rec value is copied from the S003:0010 element of the UNB segment of the interchange file sent (or the S002:0004 UNB element of the file received).

S003:0007 - Identification qualifier
If the qual_r value indicated is copied from the S003:0007 element of the UNB segment of the interchange file being sent (or the S002:0007 UNB element of the file received).

S003:0014 - Recipient identification - 2nd level
S003:0044 - Recipient identification - 3rd level
These elements are skipped.

GROUP 3 (M, 1 - 9999) SEGMENT USX (M, 1)

Comp. Elem.	Element	P.	Format	Meaning	Content	Commentary
	0020	M	an..14	Interchange file ref. number	itc_ref	itc_ref = reference of the interchange received (value from UNB)
S002		*O*		*Interchange file sender*		*Skipped*
	0048	O	an..14	Group ref. number		Skipped
S006		*O*		*Application identification of the sender*		*Skipped*
S007		*O*		*Application identification of the recipient*		*Skipped*
	0062	C	an..14	Message ref. number	msg_ref	msg_ref = reference of the message received (value from UNH)
	0800	O	an..14	Package reference number		Skipped
S501		*C*		*Date and time*		*Date and time of creation of the referred messages*
S501	0517	M	an..3	Date and time qualifier	'5'	Date and time of creation of the referred message
S501	0338	C	n..8	Date	date	date= date, format YYYYMMDD
S501	0314	C	n..15	Time	time	time= time, format HHMMSS
S501	0336	C	n4	UTC offset (time offset)	offset	offset = '0100' - offset from UTC is + 1 hour (winter time) offset = '0200' - offset from UTC is + 2 hod (summer time)
S509		*O*		*Security reference*		*Skipped*

Table 18 (Security segments structure.)

The group of segments No. 3 (USX, USY) contains references of the message received and the results of the message receipt. This group appears repeatedly for each referred message within AUTACK.

Elements description:

0020 - Interchange file reference number
The itc_ref value contains the value of the 0020 element from the UNB segment of the interchange file in which the message referred has been contained.

S002 - Interchange file sender
0048 - Group reference number
S006 - Application identification of the sender
S007 - Application identification of the recipient
These elements are skipped.

0062 - Message reference number
The msg_ref value contains the value of the 0062 element from the UNH segment of the referred message.

0800 - Package reference number
This element is skipped.

S501 - Date and time
This element contains the date and time of creation of the referred message creation. The time of creation of the digital signature is considered to be the time of message creation. The data for this element is taken from the S501 element in the USH segment of the referred message.

S501:0517 - Date and time qualifier
see table

S501:0338 - Date
The date value must have the specified YYYYMMDD format (e.g. 19950403).

S501:0314 - Time
The time value must have the specified HHMMSS format (e.g. 182033). The time value represents the time used in the Czech Republic.

S501:0336 - UTC offset
This element enables distinguishing between summer time and winter time. The offset value states the offset of local time from the standard world time, i.e. + 1 hour for the winter time (value '0100') and + 2 hours for the summer time (value '0200').

S509 - Security reference
This element is skipped.

GROUP 3 (M, 1-9999) , SEGMENT USY (M, 1-9)

Comp. Elem.	Element	P.	Format	Meaning	Content	Commentary
	0534	M	an..14	Control reference	link	link = index from USH/UST of the received message
S508		*C*		*Result of the security function*		*The result of the message receipt*
S508	0560	M	an..256	Resulting value	chck_val	chck_val = hash of the message received
S508	0560	O	an..256	Resulting value		Skipped
	0571	C	an..3	Securing error - code	err_code	err_code = code of the error detected on the receipt

Table 19 (Security segments structure.)

Elements description:

0534 - Control reference
The element is referred to the USH and UST segments from the referred message which is being acknowledged. The link value is then taken from the 0534 element in the given USH segment of the referred message.

S508 - Result of the security function
Contains the fingerprint of the referred message.

S508:0560 - Resulting value (the first appearance)
The chck_val value is the fingerprint of the referred message in filtered form (see the 0505 element) which is being calculated using the same algorithm that has been used for checking the referred message.

S508:0560 - Resulting value (second appearance)
This element is not used.

0571 - Securing error
This element contains the error code of an error detected while checking the referred message.
If the element is not indicated, acknowledgment of the message receipt is assumed.
The err_code values are as follows:
'1' - message authentication check error
'2' - certificate authentication error (i.e. the certificate needed for checking the signature)
'3' - CA certificate is not available for this certificate (incomplete certification path)
'4' - asymmetric ciphering algorithm is not supported
'5' - hash function is not supported
'6' - certificate validity has expired
'7' - the certificate is not yet valid
'8' - the certificate has been canceled
'9' - unknown certificate (only the reference number has been sent, the certificate is not placed in the local database)
'10' - wrong time mark (exceeds current time)
'11' - wrong security segment syntax
'12' - wrong fingerprint
'13' - message securing does not correspond to given requirements
'14' - message deciphering error
'999' - non-specified error

UNT SEGMENT (M, 1)

Comp. Elem.	Element	P.	Format	Meaning	Content	Commentary
	0074	M	n..6	Number of segments	seg_no	seg_no = number of segments in the message
	0062	M	an..14	Message ref. number	ref_no	ref_no = ref. number, unambiguous

Table 20 (Security segments structure.)

UNT is a standard service segment used by all UN/EDIFACT messages. It is used in accordance with standard UN/EDIFACT rules.

Example of an AUTACK message

Example of an interchange file with an AUTACK message which acknowledges the message receipt from the example contained in chapter Example of signed message:

Segments	Commentary
UNB+UNOA:1+CNBASUD+BANK+960521:2007+0000523048'	Interchange file heading - interchange file from the CNBASUD application for the BANK application.
UNH+120000000637+AUTACK:1:4:UN'	Message heading - AUTACK message with reference number 120000000637.
USH+94W+1+01+1+1+2+2+1+++120000000637+1:19960521:200742:0200' USA+1:0:1:6:1' USC+CATEST000000022'	Opening security segments which define the digital signature of the AUTACK message carried out in a standard way. The USH segment defines in particular the type of the function (digital signature), the reference of the signature (01), the message shall not be acknowledged (code 1), the filtering function used (hexadecimal), the safe copy of the message reference number, the timestamp of the signature creation. The USA segment defines the hash function used (MD5). The USC segment contains a reference to the certificate which must be used when checking the signature.
USH+94W+5+00'	USH segment defines the AUTACK message function, i.e. the non-rejection of receipt (code 5).
USB+1+1:19960521:200740:0200+CNBASUD+BANK'	The USB segment contains the date and time of creation of the AUTACK message. Furthermore, it includes the sender (CNBASUD) and the recipient (BANK) of the AUTACK message.
USX+000010033+++++236++5:19960521:200246:0200'	The USX segment contains a reference to the message being

	acknowledged: the interchange file number (000010033), the message number (236) and the date and time of creation of the signature.
USY+01+A4D37E33EB795A9B115BA36646C01F3A'	The USY segment contains a reference of the received message signature (01) and a filtered hash code of the received message.
UST+01' USR+58AFDF12B0EC9CC398C57C67F4268C49FBC0CD7F539766D0DE020A6808A9CEAA5B8806A26B8DC68B4DADBA46F87D977EAD07C8FF349843F849B4D3D7E0096209C316ABD943315C4436889A9F0100D2814F6AEC185BA1C7F6589CD5B77B3A58840BAB458FEC74A2E31AF86B00DA12D3CAC506B5A2D9E90BDB17833CC9FE2EF42F'	End security segments containing the digital signature of the AUTACK message which is carried out in a standard way. The UST segment ties the end security segments to the opening segments using the signature reference (01). The USR segment contains the message digital signature itself in filtered form.
UNT+11+120000000637'	The bottom of the message. The message contains 11 segments (including the service segments).
UNZ+1+0000523048'	The bottom of the interchange file. The file contains 1 message.

Table 21 (Example of an interchange file with an AUTACK message.)

CIPHER message

The principle of using the CIPHER message

The CIPHER message is used for ciphering transmitted UN/EDIFACT messages. A CIPHER message can contain ciphered text of any EDIFACT message. It is placed into the interchange file instead of the original message. The original message is then on receipt again deciphered from the CIPHER message and replaces the CIPHER message in the interchange file.

The figure below shows the scheme of ciphering a message:

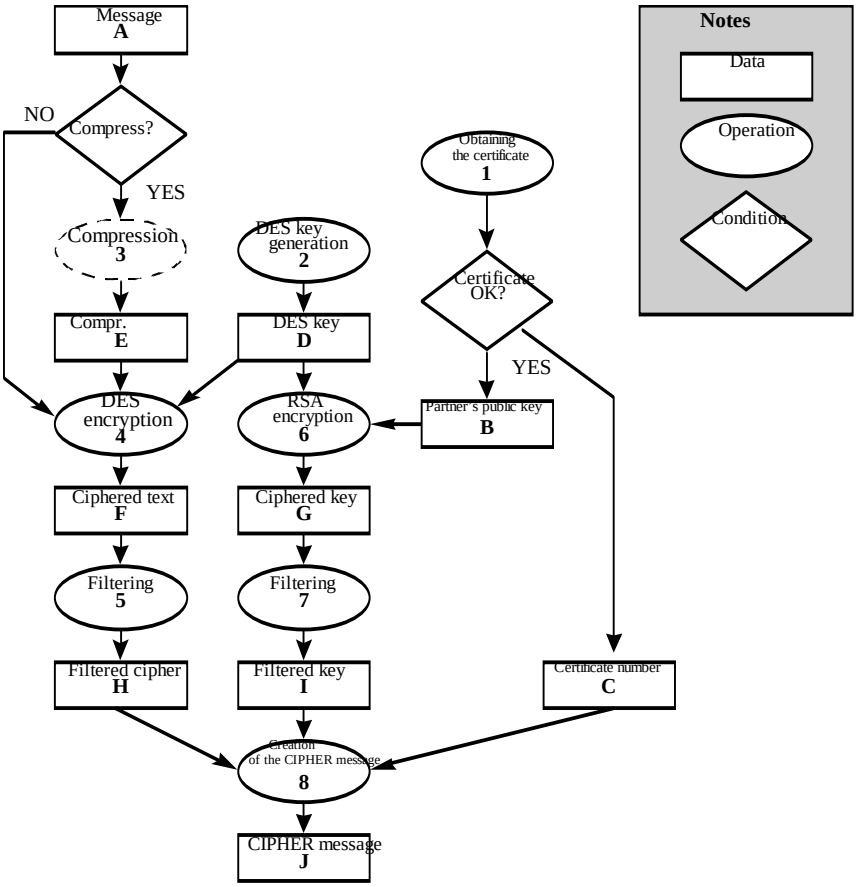

Fig. 7 (The scheme of creating the CIPHER message.)

Message encryption and creation of the CIPHER message require the following approach:

1. A partner certificate containing the partner's public key [B] is obtained. This key is intended for ciphering messages. Also, its reference number [C] is gained from the certificate. If the certificate is not valid or cannot be obtained, the message cannot be ciphered.
2. A DES key [D] is generated. This must be a random key which can only be used for ciphering this particular message.
3. The text of the original message [A] can be compressed using a selected algorithm. This will not be done if a SUD application is used.
4. The complete original message [A] or a compressed message [E] is ciphered using the DES algorithm and a generated DES key [D].
5. The resulting ciphered text [F] is filtered into text form so that it can be transmitted in a UN/EDIFACT message.

6. The DES key is ciphered by a RSA algorithm using the partner's public key [B].
7. The ciphered DES key [G] is filtered into text form so that it can be transmitted in a UN/EDIFACT message.
8. A CIPHER message is created which contains a filtered and ciphered message [H], a filtered and ciphered DES key [I], the number of the certificate which has been used for ciphering the DES key [C], and other standard data.
9. Syntactic and formal rules applying to the CIPHER message.

The following figure shows the scheme of CIPHER message deciphering and obtaining the original message:

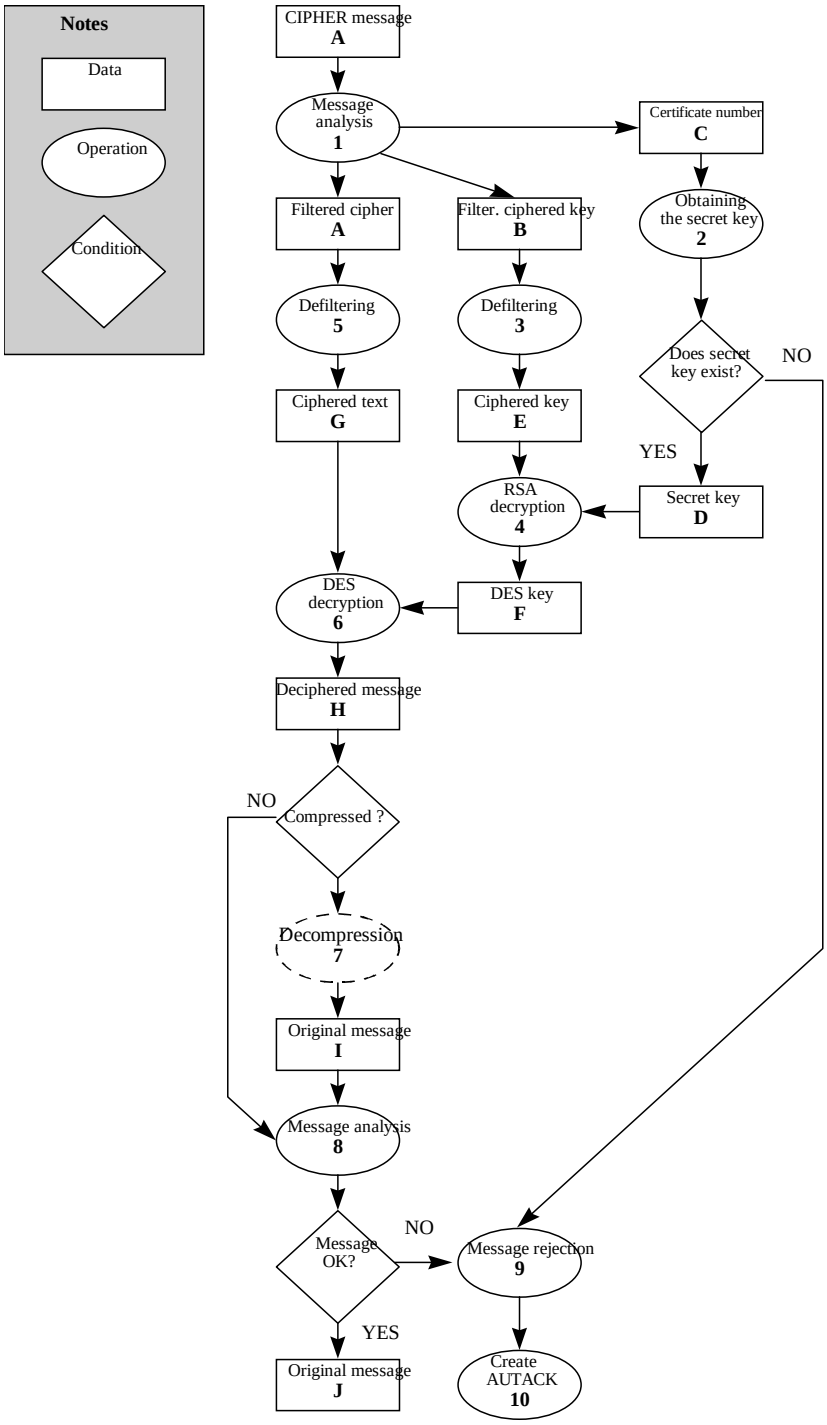

Fig. 8 (The scheme of deciphering the CIPHER message and obtaining the original message.)

On receipt of the CIPHER message the original message is obtained as follows:

1. The CIPHER message [A] is analyzed and required data obtained, in particular the filtered ciphered text of the original message [B], the filtered ciphered DES key [C] and the number of the certificate used for ciphering the DES key [D].
2. The required secret key of the user [E] is obtained using the certificate number [D]. The secret key enables the ciphering of the DES key (i.e. the key making up a pair together with the public key from the certificate).
3. The filtered DES key [C] is de-filtered into binary form [F].
4. The ciphered DES key is deciphered by the RSA algorithm using the secret key of the user [E].
5. The filtered text of the original message [B] is de-filtered into binary form [H].
6. The ciphered text of the original message [H] is deciphered by the DES algorithm using the DES key [G] obtained from the message.
7. If the text of the original message [I] has been compressed before ciphering, it is now decompressed.
8. The original message [J] is checked for the correctness of deciphering (must correspond to the UN/EDIFACT syntax).
9. If the message has not been deciphered correctly or it is impossible to obtain a secret key for its deciphering, the CIPHER message must be rejected. It cannot be further processed.
10. If the message has not been deciphered correctly or it is impossible to obtain a secret key for its deciphering, the recipient processes an error message AUTACK which is then sent to the CIPHER message sender. (see chapter The AUTACK).

Syntactic and formal rules for the CIPHER message

The CIPHER message is implemented according to the document "Cipher Text Message - EDIFACT Message Implementation Guidelines", UN/ECE/SJWG and to documents ISO/CD 9735-5,6.

The CIPHER message contains ciphered text of the entire UN/EDIFACT message, i.e. this text (as a sequence of bytes) represents the input into the ciphering algorithm from the first character of the UNH segment (i.e. ´U´) of the original message up to (and including) the end separator of the UNT segment (i.e. ' ' ') of the original message. If compression is used, this text first enters the compression algorithm and only the result of the compression represents the input into the ciphering algorithm.

If the text shall be compressed before ciphering, the compression function in the 0519 element of the USH segment is specified (not implemented in current version).

Both the ciphered text and the ciphered DES keys are filtered by the same algorithm which is specified in the 0505 element of the USH segment.

The resulting ciphered text is first filtered into text form and then divided into blocks of 512 bytes each (the last block can be smaller, depending on the number of remaining data) and each block is then entered into the USD segment in the CIPHER message body (advancing from the left to the right, i.e. the first 512-byte block in the left = first USD recurrence etc.). The ciphered text is similarly assembled back from USD segments (i.e. advancing from the left to the right).

If the message is secured using the digital signature and ciphered using the CIPHER message at the same time, the digital signature must be done first and then the message can be ciphered.

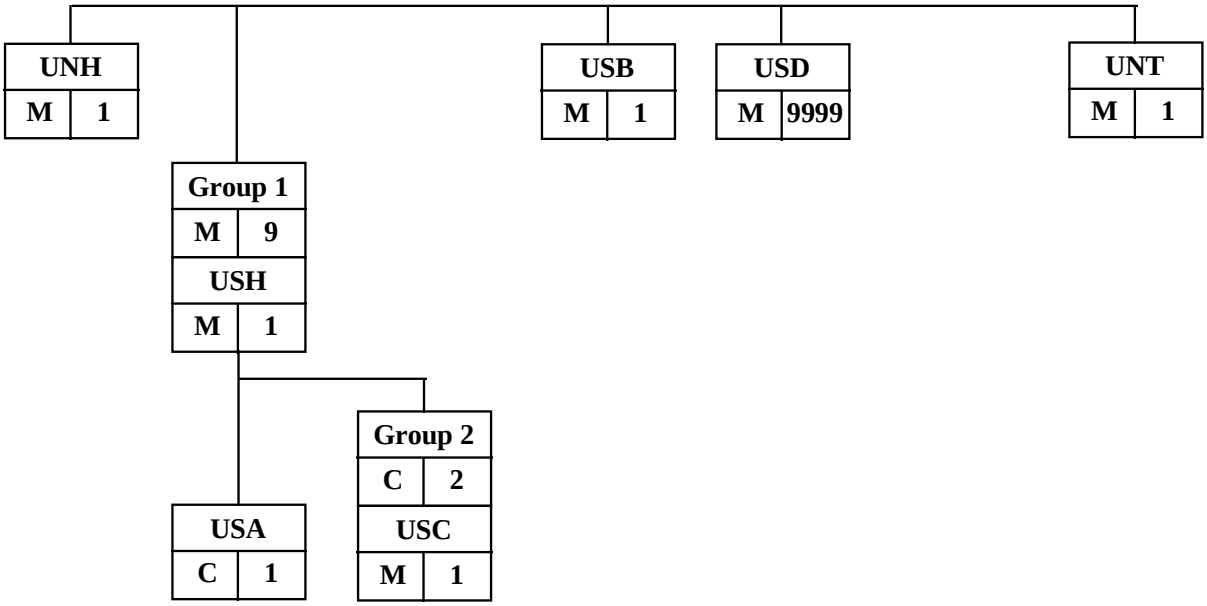

Fig. 9 (Structure of the CIPHER message.)

M/C - mandatory (M), use non-mandatory (C), normally not used (O) segment (group)
Op. - number of recurrences, maximum number allowed by the standard is indicated in the brackets. Recurrence that will not be used is indicated in (), recurrences that can be used in other implementations are indicated in [].

GROUP SEGMENT	M/C	Op.	DESCRIPTION
1	M	1(9)	This group of segments is used to identify the applied security functions.
USH	M	1	Defines parameters for the function of the message confidentiality.
USA	C	1	Contains the specification of the ciphering algorithm and a ciphered symmetric key.
2	C	1(2)	Group of segments no. 2 identifies the applied certificate of the message recipient.
USC	M	1	Contains the number of the certificate which has been used to cipher the symmetric key or the Identification of the certificate owner..
USB	M	1	Opening segment of the body, contains a time mark.
USD	M	9999	Contains a ciphered message.

Table 22 (The structure of security segments.)

Composite Element - Number of the composite element in the UN/EDIFACT Standard Directory
Element - Number of the element in the UN/EDIFACT Standard Directory
P. - mandatory (M), use non-mandatory (C), normally used (O) segment, element
Format - format specification according to the UN/EDIFACT conventions
Content - constants are indicated in ' ', the text identifiers refer to variable values supplied by the security application

SEGMENT UNH (M, 1)

Comp. Elem.	Element	P.	Format	Meaning	Content	Commentary
	0062	M	an..14	Message reference number	ref_no	ref_no = ref. number, identical with the number used for the original message
S009		*M*		*Message identifier*		*Identifies the type of the UN/EDIFACT message*
S009	0065	M	an..6	Message type	'CIPHER'	
S009	0052	M	an..3	Message version	'2'	
S009	0054	M	an..3	Version number	'951'	
S009	0051	M	an..2	Responsible agency	'UN'	
S009	0057	O	an..6	Special code		Skipped
	0068	O	an..35	Joint reference		Skipped
S010		*O*		*Transmission status*		*Skipped*

Table 23 (The structure of security segments.)

UNH is a standard service segment which is used by all UN/EDIFACT messages. Its usage is given by standard UN/EDIFACT rules. Special approach is required only for the element 0062.

0062 - Message reference number
The message reference number (ref_no) is identical with the number used for the original message

GROUP 1 (M,1) SEGMENT USH (M, 1)

Comp. Elem.	Element	P.	Format	Meaning	Content	Commentary
	0552	M	an..3	Segments structure version	'94W'	Version from 1994
	0501	M	an..3	Security function - code	'4'	Data confidentiality
	0534	M	an..14	Control reference	'00'	Number 00 - to avoid collision with the digital signature
	0541	O	an..3	Securing extent - code		Skipped
	0503	O	an..3	Type of response - code		Skipped
	0505	C	an..3	Filter (function) - code	filter	Filter for binary data
	0507	C	an..3	Coding of characters - code	'2'	ASCII 8 bites 1)
	0509	C	an..3	Role of the party - code	'1'	Document sender
S500		*O*		*Party identification*		*Skipped*
S500		*O*		*Party identification*		*Skipped*
	0516	O	an..35	Reference number	ref_no	Skipped
S501		*O*		*Date and time*		*Skipped*
	0519	O	an..3	Compression function - code	comp	comp = code of the compression function applied

Table 24 (Security segments structure.)

Group 1 contains data on the CIPHER message function and the parameters of the ciphering algorithm including the ciphered DES key. This group appears only once in the implementation.

Description of elements:

0552 - Segments structure version
The value '94W' defines that service security segments are used which are described in the document UN/TRADE/WP.4/R.1026 and in ISO/CD 9735 - 5.

0501 - Security function
The CIPHER message carries the function of data confidentiality (code '4').

0534 -Control reference
link is '00' to make clear that it is not tied to any Security Trailer.

0541 - Securing extent
This element is skipped.

0503 - Type of response
This element is skipped.

0505 - Filter (function)
Identifies the type of the function used for filtering binary data resulting from ciphering the message before the data is loaded into the CIPHER message (USD segment).
Filtering can be done either using a hexadecimal filter or a filter defined in ISO 9735-5 (also in R.1026), a so-called UNO-A filter, both of them being fully applicable for the UN/EDIFACT syntactic level A (multi-purpose filters). The chosen filter is then applied to all binary data in the CIPHER message.
The hexadecimal filter represents one byte by a pair of characters ('0' - '9', 'A' -'F'), the first one representing the upper 4 bytes and the second representing the lower. Characters on the left side of the hexadecimal record represent the more important bytes. Non-relevant zeroes in the left can be skipped.
The filter code has following values:
'2' - hexadecimal filter
'5' - UNO-A filter

0507 - Coding of characters
Specifies coding the characters of the received EDIFACT message before using the hash function. 8-bit ASCII is applied here (value '2').

0509 - Role of the signing party
The party is the sender of the CIPHER message (value '1')

S500 - Party identification (first recurrence)
S500 - Party identification (second recurrence)
0516 - Reference number
S501 - Date and time
These elements are skipped.

0519 - Compression function - code
This element states whether the text has been compressed before ciphering. If this is the case, the element specifies the compression function applied. The compression function will not be implemented for current application but can be applied in the future.
If the element is not stated, the compression has not been applied.
The comp values have not been defined so far.

GROUP 1 (M, 1) SEGMENT USA (C, 1)

Comp. Elem.	Element	P.	Format	Meaning	Content	Commentary
S502		*M*		*Security algorithm*		*Algorithm for the hash of the message received*
S502	0523	M	an..3	Application of the algorithm - code	'2'	Symmetric algorithm for message ciphering
S502	0525	C	an..3	Operation mode - code	'2'	DES mode CBC - ISO 8372
S502	0533	O	an..3	List of operation modes	'1'	List defined by UN/EDIFACT SJWG
S502	0527	C	an..3	Algorithm - code	'1'	Algorithm DES - FIPS Pubs 46
S502	0529	O	an..3	List of algorithms	'1'	List defined by UN/EDIFACT SJWG
S503		*C*		*Algorithm parameters*		*Ciphered DES key*
S503	0532	C	an..512	Parameter value	DES_key	DES_key = ciphered DES key
S503	0531	C	an..3	Param. qualifier - code	'6'	Symmetric key ciphered by a public key
S503		*O*		*Algorithm parameters*		*Skipped - no relevance for given algorithm*
S503		*O*		*Algorithm parameters*		*Skipped - no relevance for given algorithm*
S503		*O*		*Algorithm parameters*		*Skipped - no relevance for given algorithm*
S503		*O*		*Algorithm parameters*		*Skipped - no relevance for given algorithm*

Table 25 (Security segments structure.)

Elements description:

S502 - Security algorithm
The element describes the symmetric ciphering algorithm (DES) of the user used for ciphering messages sent by the user (code 0523 = '2').

S502:0523 - Application of the algorithm
see table

S502:0525 - Operation mode
see table

S502:0533 - List of operation modes
The element defines the list of operation modes applied. In this case the list defined in the material UN/TRADE/WP.4/R.1026 from 1994 (value '1') is used.

S502:0527 - Algorithm
The element defines the algorithm applied. Detailed specification of the algorithm and its parameters is defined in chapter Parameters of Applied Cryptographic Algorithms.

S502:0529 - List of algorithms
The element defines the list of algorithms applied. In this case the list defined in the material UN/TRADE/WP.4/R.1026 from 1994 (value '1') is used.

S503 - Algorithm parameters (first appearance)
This element contains a DES key ciphered by the public key of the message recipient.

S503:0532 - Parameter value
The DES key value is a DES key ciphered by the public key of the message recipient. The value indicated is filtered (see the description of element 0505 in the USH segment).

S503:0531 - Parameter qualifier
see table

S503 - Algorithm parameters (further appearances)
These elements are not being used. No other parameters are needed for the scheme used.

GROUP 2 (C, 1) SEGMENT USC (M, 1)

Comp. Elem.	Element	P.	Format	Meaning	Content	Commentary
	0536	C	an..35	Certificate ref. number	ref_num	ref_num= certificate ref. number - unique
S500		*O*		*Party identification*		*Identification of the certificate owner*
S500	0577	M	an..3	Party qualifier	'3'	Certificate owner
S500	0538	C	an..35	Key name	key1	key1= number (name) of the certified key
S500	0511	C	an..17	Party ID	EDI_ID	EDI_ID= EDI identification of the key owner's organization
S500	0513	O	an..3	List of parties applied	'1'	List of banks - code (bank EDI applications)
S500	0515	O	an..3	Agency maintaining the list	'CNB'	Czech National Bank
S500	0586	O	an..35	Name of the party	org_name1	org_name1= name of the organization
S500	0586	O	an..35	Name of the party	org_dep1	org_dep1= department (branch) of the organizat.
S500	0586	O	an..35	Name of the party	org_pers1	org_pers1= responsible personnel
S500		*O*		*Party identification*		*Skipped*
	0544	O	an..3	Version of certificate format		Skipped
	0505	O	an..3	Filter (function) - code		Skipped
	0507	O	an..3	Coding of characters - code		Skipped
	0543	O	an..3	Selection of characters - code		Skipped
	0546	O	an..35	Rights level		Skipped
S505		*O*		*Separators*		*Skipped*
S505		*O*		*Separators*		*Skipped*
S505		*O*		*Separators*		*Skipped*
S505		*O*		*Separators*		*Skipped*
S501		*O*		*Date and times*		*Skipped*
S501		*O*		*Date and time*		*Skipped*
S501		*O*		*Date and times*		*Skipped*
	0567	O	an..3	Security status, code		Skipped

Table 26 (Security segments structure.)

The USC segments identify the certificate (or the public key contained in it) which has been applied for ciphering the DES key contained in the USA segment. It is therefore a certificate of the key belonging to the CIPHER message recipient. The certificate is uniquely identified using a corresponding reference number.

Elements description:

0536 - Certificate reference number
This element contains the certificate reference number.

S500 - Party identification (first recurrence)
This element can serve for the additional specification of the recipient. Its values must correspond to values indicated in the certificate. Since the certificate reference number uniquely identifies both the certificate and the recipient, it shall not be normally used.

S500:0577 - Party qualifier
see table

S500:0538 - Key name
Contains the identification of the user's public key contained in the certificate.

S500:0511 - Party ID
Contains the identification of the organization for EDI. The EDI_ID value is assigned by CA or RA.

S500:0513 - List of parties applied

S500:0515 - Agency maintaining the list
Only one list shall be used for the current application. The values will therefore not be indicated; the values indicated in the table are considered to be default values. They are supposed to be applied in the future versions of EDISEC2 product.

S500:0586 Name of the party
Intended for a more detailed specification of the party.

Other elements in the USC segment are skipped.

SEGMENT USB (M, 1)

Comp. Elem.	Element	P.	Format	Meaning	Content	Commentary
	0503	M	an..3	Type of response - code	'1'	The message shall not be acknowledged by the AUTACK message
S501		*C*		*Date and time*		*Date and time of ciphering the message*
S501	0517	M	an..3	Date and time qualifier	'1'	Time mark
S501	0338	C	n..8	Date	date	date= date, format YYYYMMDD
S501	0314	C	n..15	Time	time	time= time, format HHMMSS
S501	0336	C	n4	UTC offset (time offset)	offset	offset = '0100' - offset from UTC is + 1 hour (winter time) offset = '0200' - offset from UTC is + 2 hours (summer time)
S002		*O*		*Interch. file sender*		*Skipped*
S003		*O*		*Interch. file recipient*		*Skipped*

Table 27 (Security segments structure.)

Elements description:

0503 - Type of response
This element specifies that the CIPHER message receipt shall not be acknowledged by the AUTACK message (value '1') because possible message acknowledgment is done only after the CIPHER message is deciphered. The AUTACK message is created for the original message provided with a digital signature according to the requirement indicated in the element 0503 of the USH segment (see also chapter Security Header and Trailer for the Digital Signature).

S501 - Date and time
This element contains date and time of the CIPHER message creation.

S501:0517 - Date and time qualifier
see table

S501:0338 - Date
The date value must have the specified YYYYMMDD format (e.g. 19950403).

S501:0314 - Time
The time value must have the specified HHMMSS format (e.g. 182033). The time value represents time used in the Czech Republic.

S501:0336 - UTC offset
This element is used for distinguishing between summer time and winter time. The offset value states the offset of local time from the standard world time, i.e. +1 hour for winter time (value '0100') and + 2 hours for summer time (value '0200').

S002 - Interchange file sender
S003 - Interchange file recipient
These elements are skipped.

SEGMENT USD (M, 9999)

Comp. Elem.	Element	P.	Format	Meaning	Content	Commentary
	0522	M	an..512	Ciphered text	ciph_txt	ciph_txt = ciphered text divided into blocks, filtered

Table 28 (Security segments structure.)

The USD segment contains a filtered and ciphered text. The number of segments recurrences is selected to cover the entire text. Dividing the text into 512 blocks and saving into the USD segment is done from the left to the right.

Elements description:

0522 - Ciphered text
Contains 512 bytes (except the last recurrence) of the filtered (by the function specified in USH 0505) and ciphered text of the original message.

SEGMENT UNT (M, 1)

Comp. Elem.	Element	P.	Format	Meaning	Content	Commentary
	0074	M	n..6	Number of segments	seg_no	seg_no = number of segments in the message
	0062	M	an..14	Message ref. number	ref_no	ref_no = ref. number, unambiguous

Table 29 (Security segments structure.)

UNT is a standard service segment used by all UN/EDIFACT messages. Its application is given by standard UN/EDIFACT rules.

Example of CIPHER message

Segments	Commentary
UNB+UNOD:2+BANK+CNBASUD+960521:2002+000010033'	Interchange file heading - interchange file from the BANK application for the CNBASUD application.
UNH+236+CIPHER:2:951:UN'	Message heading - CIPHER message, ref. no. 236.
USH+94W+4+00+++2+2+1	The USH segment defines the function of the CIPHER message (data confidentiality), the filter applied for the ciphered data and the ciphered DES key (hexadecimal).
USA+2:2:1:1:1+801D992110315F5A796AB70F1E8D1A56EA4E9867EDDA55291D898E8062825C1A173A7B0DA11F99D98D838E2C2D69FB3B6C8A21F2AA6875290D2F89EE7B61BEA9F808517EE2B0B9BB73E8478DD0DD285673480DE9E2D0352BE0B16FFE57CDBD6029097F69B85E7F67126D2B5A87B819A048A4E0139EFE8A08E4915B63EC7BE30B:6'	The USA segment defines the parameters of the ciphering algorithm (DES mode CBC) and contains a filtered DES key.
USC+CATEST000000022'	The USC segment contains a reference to the certificate used for ciphering the DES key.
USB+1+1:19960521:200249:0200	The USB segment contains the date and time of creation of the CIPHER message.
USD+940CB4938F319A5E37E1F882E3CF5C7A938B4C7269A5EE082B0D2F17BE4779E904E28E6F617F2F75E755E560A7D9DB2E1F208EE41B1BB97F8504B80B5F779B80542DF98C46F39561FE4F375C635D010DE828F3B492615229E20212A91545DABA60BF2E15FE1B89931C860E0C1DA5AC54EF5959F3CBF051436B5254503C84163CD61D9CF7CE778B8D816E61152879B007710718F30221D2EBCC6E9BEEAB4EA77A872DB2BBA37A9798189CB8A9F0B71B27551F119F56390752C6D44874AF982827576FB2733C6C30329BA3C6BAA0D0951E831F334E2605E5B66875038406B581C090FDE4FB95A20D1DAC1A7F80CD8EB8E013B6C1E178F933F16C3448D653E54ECF' USD+FED688E43D7EEDA8FCCA04564B81B04BB041FEEA6CE64D711F6D4061A7806F0C6C8D55CF39624D71FC6A6D6A91073BB25D38E878E9B10F92BDF7971580368EFBAC28091EC25D09BEF2A04FF5E363163D5035A9E1A66B2631924DCF24BC5D9E186068501E964327030341E061EBEDC878B3C827417C4DEDE958034DFBB0C15C1BCD5D5382ACD998922E9EEA60F8B55E6EE8F55DA5429164783BA6BEC7E2FD09AC1ED9ACA9D303958C2FA522F4C778CAA2068E3E64335DC9C41A86C47E3D9A478DC076057C10EEAD80CE6C0309C8C1BA740A3680F10373F1047A1517DF1C9F8862B9D60C099B12EDC202FED4593979E56C41A122BEBBD6FE938E02E5383EAB8BAC' USD+98862F5F151638D2CF2BBB8490F91468C0B5B20445786741077F3B1525101F0189B3D47C3073AB4D3909944BD8B443934AFF330C434CA46DD5862615F2CA585080DAF0F5607999BF1549E436CCF385A8E5E588B8FF64EC71'	The USD segments contain the original message - filtered and ciphered.
UNT+9+236'	Bottom of the message. The message contains 9 segments (including the service segments).
UNZ+1+000010033'	The interchange file bottom. The interchange file contains 1 message.

Table 30 (Example of an interchange file with a ciphered message. (CIPHER))

APPENDIX B - LIST OF ERROR CODES IN THE AUTACK MESSAGE

Code	Description
1	message authentication check error
2	certificate authentication check error (i.e. the certificate needed for checking the signature)
3	CA certificate is not for this certificate (incomplete certification path)
4	asymmetric ciphering algorithm is not supported
5	hash function is not supported
6	the certificate validity has expired
7	the certificate is not yet valid
8	the certificate has been canceled
9	unknown certificate (only the reference number has been sent, the certificate is not placed in the local database)
10	wrong time mark (exceeds current time)
11	wrong security segments syntax
12	wrong fingerprint
13	message securing does not correspond to given requirements
14	message deciphering error
500	Multiple message acknowledgment (in the database only)

Table 31 (Appendix B - LIST OF ERROR CODES IN THE AUTACK MESSAGE)