SECURED SMART CARD SIMULATION

Zahraa Ali Jwad, Sarkout N. Abdulaah, and Oday A.L.A Ridha

ABSTRACT:

Different methods of encryption that are widely used in smart card have been presented. Because of the usefulness and widespread application of Food Ration Card throughout our country, three models of designing and simulations are developed. A comparison between the different models is done. The first model is the Food Ration Card without using any security method. The second model is the Food Ration Card with using an AES algorithm as a security method. The third model is the Food Ration Card with using RSA method. All models are implemented and simulated using
BasicCard Development kit Environment. For the first model, a Compact BasicCard version ZC1.1 is used. While for the second and third models, a Professional BasicCard version ZC4.5 is used. From obtained results, we noticed that AES-method is faster than RSA and takes less memory size, but RSA is more secure than AES, since it uses two different keys instead of one as in AES.

الخلاصة:

دراسة البطاقة الذكية واهم طرق التشفير المستخدمة فيها. في هذه المقالة تم تقديم ثلاث نماذج للبطاقة التموينية باستخدام البطاقة الذكية لما تمتلكه الاولى من اهمية كبيرة في قطرنا. حيث انها تعتمد كوثيقة رسمية في الكثير من الدوائر الحكومية. النموذج الاول هو تنفيذ البطاقة التموينية بدون استخدام اي طريقة من الطرق الامينة. اما النموذج الثاني فهو تنفيذ البطاقة التموينية باستخدام طريقة معيار التشفير المتقدم كطريقة امينة. اما ما يخص النموذج الثالث فتم استخدام طريقة RSA كطريقة امينة لتنفيذ البطاقة التمونية. في النموذج الاول تم استخدام بطاقة BasicCard المضغوطة باصدار1.1، بينما النموذج الثاني والثالث تم استخدام بطاقة BasicCard المتخصصة باصدار4.5. بعد تنفيذ النماذج الثلاث تم عمل مقارنة بينها. من خلال النتائج التي حصلنا عليها من تنفيذ البرامج لاحظنا بان طريقة معيار التشفير المتقدم اسرع من RSA وتاخذ حجم ذاكرة اقل. في حين استخدام RSA يعطي البطاقة اكثر امنية من طريقة معيار التشفير المتقدم كونها تعتمد على مفتاحين مختلفين بدل المفتاح الواحد.

Keywords: Smart card, AES, RSA, Encryption, Foot Ration Card.

INTRODUCTION

The smart cards are the youngest and cleverest member of the family of identification cards in the ID-1 format. Its characteristic feature is an integrated circuit embedded in the card, which has components for transmitting, storing and processing data. The data can be transmitted using either contacts on the surface of the card or electromagnetic fields, without any contacts.
Smartcards offer several advantages compared with magnetic-stripe cards. For instance, the maximum storage capacity of a smart card is many times greater than that of a magnetic-stripe card. However, one of the most important advantages of smart cards is that their stored data can be protected against unauthorized access and manipulation [Wenzhou L., 2010]. Since the data can only be accessed via a serial interface that is controlled by an operating system and security logic, confidential data can be written to the card and stored in a manner that prevents them from ever being read from outside the card. Such confidential data can be processed only internally by the chip's processing unit. In principle, both hardware and software mechanisms can be used to restrict the use of the storage functions of writing, erasing and reading data and tie them to specific conditions. This makes it possible to construct a variety of security mechanisms, which can also be tailored to the specific requirements of a particular application[Stefan M., 2010]. In combination with the ability to compute cryptographic algorithms, this allows smart cards to be used to implement convenient security modules that can be carried by users at all times, for example in a purse or wallet. Some additional advantages of smartcards are their high level of reliability and long life compared with magnetic-stripe cards, whose useful life is generally limited to one or two years [Rankl W. and Effing W., 2003].

SECURITY OF SMART CARD

The encryption algorithm (also called a cipher) is used to transform plaintext (the original information to be hidden) into ciphertext (the hidden information) or ciphertext into plaintext (decryption).

An encryption algorithm has known as an "asymmetric algorithm" if the encryption and decryption keys that it uses are different, includes RSA and elliptic curve; otherwise, it is a "symmetric algorithm", examples include the Data Encryption Standard (DES), the International Encryption Standard (IDEA) and Advanced Encryption Standard (AES) [Schneier B., 1996]. In general, security provides the following objectives or services:

- Confidentiality is a service used to keep the content of information accessible to only those authorized to have it.
- Integrity is a service that requires that computer system assets and transmitted information be capable of modification only by authorized users. Modification includes writing, changing, changing the status, deleting, creating, and the delaying or replaying of transmitted messages.
- Authentication is a service that is concerned with assuring that the origin of a message has correctly identified. That is, information delivered over a channel should be authenticated as to the origin, date of origin, data content, time sent, etc.
- Non-repudiation is a service, which prevents both the sender and the receiver of a transmission from denying previous commitments or actions [Everett D., 1993].

The advantage of asymmetric algorithm is increased security. Since the receiver decrypts the message by using his/her private key, there is no need to expose the secret key to the insecure transmission medium [Schneier B., 1996].

Among the standard methods that have been used in the smart card are DES, AES, RSA and elliptic curve cryptography (ECC) [Everett D., 1993].

Data Encryption standard (DES)

DES is block cipher; it encrypts data in 64-bit block. The key length is 56-bits (the key is usually expressed as 64-bit number, but every 8^{th} bit is used for parity checking and is ignored). DES is fast but is relatively insecure. Generally, it is avoided when possible [Schneier B., 1996].

Advanced encryption standard (AES)

NIST (National Institute of Standards and Technology) had selected Rijndael algorithm to be the proposed AES due to its high security strength, computational and memory efficiency, high configurability and simplicity. It can be implemented in wide ranges of devices from low memory devices like smart card to high-end workstations. It is a stronger symmetric encryption algorithm that was approved by NIST to replace the Data Encryption Standard (DES) and Triple DES encryption algorithm. For the AES algorithm, the length of the input block, the output block and the State is 128 bits**.** For the AES algorithm**,** the length of the Cipher Key, K, is 128, 192, or 256 bits [Daemen J. and Rijmen V., 199].

RSA (Rivest Shamir Adelman) public-key encryption

The RSA algorithm belongs to the public-key systems, namely architectures where the key is published to all parties**.**

The keys used in RSA are quadruples of (p, q, e, d), The mathematical definition of this scheme is $EK(P) = P^e \bmod pq$, $DK(C) = C^d \bmod pq$, with EK being the encrypting function, and DK being the decrypting function. P is the plaintext and C is the cipher text. The security of the RSA depends on the problem of factoring large numbers. That is meaning the difficulty of recovering p and q if only their product n has known, so cannot calculate the private exponent [Schneier B., 1996].

Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) is public key cryptography. In public key cryptography each user or the device taking part in the communication generally have a pair of keys, public key and a private key, and set of operations associated with the keys to do the cryptographic operations [Schneier B., 1996].

The mathematics of cryptographic systems based on elliptic curves is relatively difficult. One factor limiting the use of elliptic curves for asymmetric cryptographic algorithms is that they are regarded as a relatively new discovery in the cryptographic world, even though they have been known for a long time. It will no doubt take some time until the use of ECC systems becomes commonplace in the cautious world of cryptographers and smart card application designers [Rankl W. and Effing W., 2003].

BASICCARD

Since 1996, a smart card operating system with an interpreter for the Basic programming language has been available from the German company Zeitcontrol. This operating system has called Basic Card, and it is available in various versions with different features and for hardware platforms with various memory sizes.

The procedure for generating downloadable programs for Basic Card has based on traditional Basic interpreters. A compiler translates the source code into P-code, which is transferred to a memory region of the smart card microcontroller reserved for this purpose. The transformation is done using a special loader program. After that, the program code stored in this region can be processed by the interpreter as necessary [Rankl W. and Effing W., 2003].

The ZeitControl's BasicCard family includes Compact BasicCard, the Enhanced BasicCard, Professional BasicCard, and the MultiApplication BasicCard. A BasicCard contains 256-1768 bytes of RAM, and 1-31 kilobytes of user-programmable EEPROM.

BasicCard has the following features:

- Inexpensive, where the development software is free of charge and most versions of the BasicCard cost less than half as much as other currently available programmable processor cards.
- Easy to program, where it uses a very Basic like programming language.
- Secure, where state-of-the-art cryptographic algorithms are available for all BasicCard types [Guilfoyle T., 2005].

FOOD RATION CARD APPLICATION WITHOUT SECURITY

The application of Food Ration Card is chosen as an example of the smart card applications, because it is useful and widespread application throughout our country. It is an official and important document.

The card issuer for the Food Ration Card will be Ministry of Trade. The following information will be written on the card: name of family head, the card serial number, the address, number of family members, number of infants, last date updated and the PIN (personal identification number) code (which will be known only by the owner), so as to prevent other people from making use of the card in case of theft or loss.

The name of process in which the information is written on the card is "personalization". After that, the cards will be distributed to the citizens by agents. Therefore, when a citizen wants to receives his/her share of rational food, he/she has to use his/her card. The last date updated refers to the date of issuing the card by Ministry of Trade.

At the beginning, the program is written without adding any security algorithms to check the size of the program, the speed of its implementation, and the required memory to implement the program.

The BasicCard Development Kit is selected as the development environment for this project, due its widespread, low cost and technical support available via internet. The application is written in ZC-Basic language, which is supported by the development kit.

The program includes three basic parts:

1. Terminal-Write program: this program is used to issue the Food Ration Card.
2. Terminal-Read program: this program is used to read the information written on the Food Ration Card.
3. Food Ration Card program: this represents the program of the smart card which saves the following information: name of family head, the card serial number, the address, number of family members, number of infants, last date updated and the PIN code.

We will conduct a simulation to the three programs mentioned above using a software development package, so that we can check the physical requirement of the BasicCard.

Terminal- Write program of Food Ration Card without security

This program is used to issue the Food Ration Card to users. The first step of the program is to check the status of the card whether it is inserted in to card reader or not, see fig.1. If not, the program will remain in the pool state, waiting for the card to be inserted. After the card is inserted, the communication between terminal and BasicCard will begin by sending Reset card command from the terminal to card. The card will respond by an Answer to Reset to the terminal program. The process will happen according to communication protocol, which is defined in ISO/IEC 7816-3.

Now, the program allows the issuer to enter the required information, which is as follow: name of family head, the card serial number, the address, number of family members, number of infants, last date updated and the PIN code. The PIN code aims at verifying the bearer of the card. The PIN code is four-character string.

The program will check the length of the entered PIN code, which has to be four-characters. If the length is not four-character, then the issuer will be allowed to enter another PIN. Then the COMD1 is called by terminal-write, to transfer information from terminal-write temporary buffer to EEPROM of BasicCard. Afterwards, SW1SW2 will be checked for each command.

The SW1SW2=9000 is defined in ISO/IEC 7816-4 (Interindustry commands for interchange) as a status word for successful command. The value that will back from the command has to be checked always so as to remove the card form the reader in case there are no error conditions. Finally, a message will appear on the screen, which says: "personalization successful".

Terminal-Read program of Food Ration Card without security

The second program in this application is the terminal read, fig.2. This program is used to read the information from BasicCard. The first step of the program is to check the status of the card whether it is inserted in to card reader or not. If not, the program will remain in the pool state, waiting for the card to be inserted.

After the card is inserted, the communication between terminal and BasicCard will begin by sending Reset card command from the terminal to card. The card will respond by an answer to Reset to the terminal program. Then the user is asked to enter the name of the application, which is saved in the smart card. If the name is wrong, then the user will be out of the program. But if the name is right, then a message will be displayed on the screen which says: "Right application name".

Then, the program will call "COMD2" to get the saved information on the card. The information will appear on the screen. Then the program asks the user whether he/she needs an update. This phase will be used when the citizen wants to receive his share from the agent. So if the citizen enters any number that means he/she does not want to receive his/her share. And he/she used the card in another place. If this happened, the user will go out of the program. But if the user entered "1", that means he/she wants to receive his/her share. If this happened, the program will ask the citizen to enter PIN code. Then the program will call the COMD3 command by which we can make sure whether the PIN is right or wrong.

Then the citizen has asked to enter a new date. That is the date in which he/she received his/her share. The reason we use this date is to make sure whether the citizen has received his/her share or not.

Food Ration Card program without security

The program of the Food Ration Card that stored in smart card contains series of commands that will be called from the terminal write/Read programs. All the details of the commands are explained below:

COMD1 command

This command is called by terminal-write program. This command makes the personalization process. The command reads entered information from the keyboard then stores them on internal EEPROM of the BasicCard.

COMD2 command

This command is called by terminal-Read program. It checks whether the card is personalized or not. If not, the command will terminate terminal-Read program; but if yes, the command reads the information (name of family head, number of serial card, the address, number of family members, number of infants, last date updated and the PIN code) from EEPROM of the card and send them to terminal-Read program.

COMD3 command

This command is called by terminal-Read program. It first checks whether the card personalized or not. If not, the command will terminate Read program; but if yes, it reads entered PIN code and compares it with actual PIN code. If they equal, the program will continue to update date. Otherwise the command permits the user to re-enter PIN code. The re-entering is permitted for 4 tries as maximum.

FOOD RATION CARD APPLICATION WITH SECURITY

In this section, the program is implemented with a security to verify whether terminal is a genuine one or not. Two methods of encryption will be used to implement the security. These methods are AES and RSA with a key length of 128 bits. To make a comparison between the two methods, the size, speed, and required memory are considered.

The programs that are changed are the terminal-Read program and Food Ration Card, where there is no need to alter the terminal-write program because it only issues the card and the issuer of the Food Ration Card is the ministry of Trade.

Terminal-Read program of Food Ration Card with security

The security level is added to manage the genuineness of the terminal. This security level is described as terminal generates a message. This message is sent to BasicCard to encrypt it and

returns an encrypted value to the terminal. The terminal decrypts the encrypted value received from BasicCard. The value obtained from terminal decryption process must equal to the original message, which generates before. If the value obtained from terminal decryption process is not equal to the original message the communication between the terminal and BasicCard stops (this mean that Terminal-Read program cannot read the information stored on the card), so the information stored on the card is protected from counterfeit terminal.
While if the two values are equal the communication between the terminal and BasicCard continues because the card ensures that, its communication partner is genuine terminal.

Terminal-Read program of Food Ration Card using AES-method

The first step of the program is to check the status of the card whether it is inserted in to card reader or not. If not, the program will remain in the pool state, waiting for the card to be inserted.
After the card is inserted, the communication between terminal and BasicCard will begin by sending Reset card command from the terminal to card. The card will respond by an answer to Reset to the terminal program.
AES method is applied to the terminal-Read program as a secretive method. Then, the program will generate message and sends it to Food Ration Card program to encrypt it by calling Encrypt command. See fig.3.
This command will encrypt the generated message. Then, by calling the COMD4 the terminal-Read program will get the encrypted value, after that the program decrypts the encrypted value by AES-function, by using the key stored in the program which the same the key stored in BasicCard program. The terminal-Read program compare between the decrypted value and the original message generated from this program, if not equal, the program will exit and print on the screen “The Terminal is counterfeit”. While, if the two values are equal the program will print on the screen “The Terminal is genuine” and complete the other steps that are declared in the “Terminal-Read program of Food Ration Card without security”.

Terminal-Read program of Food Ration Card using RSA-method

The first step of the program is to check the status of the card whether it has inserted in to card reader or not as described in previous program.
RSA-method is applied at the beginning of the terminal-Read program. First, the program will call COMD5 command to calculate the public key modulus (n), and then this program generate message and sends it to Food Ration Card program to encrypt it by calling Encrypt command. This command will encrypt the generated message. Then, the program decrypts the encrypted message by using RSA-function with key length 128-bits. The terminal-Read program compares between the decrypted value and the original message generated from this program, if the two values are different, the program will exit and print on the screen “The Terminal is counterfeit". While, if the two values are equal the program will print on the screen “The Terminal is genuine" and complete the other steps that are declared in the “Terminal-Read program of Food Ration Card without security”.

Food Ration Card program with Security

This program uses the same the commands that are used in the previous section but with adding some new commands. These new commands are:

Encrypt command

First, the command will check if the card is personalized or not. If not, then the command will exit. Else, the command will encrypt the received data (by using AES-function in the Food Ration Card project with AES-128 and RSA-function in the Food Ration Card project with RSA-128).

COMD4 command

The function of this command is to send the encrypted data N1 (that is obtained from Encrypt command) to terminal-Read program to decrypt it.

COMD5 command

First, the command will check the key length if less than 128-bit, if yes the command will exit from the program, while if more than 128-bits this command calculates the public key modulus (n) from p, q.

THE OPTION WINDOWS

To simulate this project, we must compile it and before we compile it must make setting to simulation options for both terminal program (write/Read) and BasicCard program.

The option windows of Food Ration Card application without security

In terminal program (Write/Read) option window the output file type must select one of 5 files (.DEG, .EXE, .IMG, .LST&.MAP). In our case .DBG files extension has chosen for debugging the software. **Fig. 4** shows the project option windows for terminal programs.

The initial value of communication port (COM port) must be selected. For simulation purposes, virtual card reader was chose in port 201. **Fig. 5** shows the setting virtual port for terminal program.

In BasicCard, option window COM 201 has been selected as virtual COM port and the Compact BasicCard version ZC1.1 has been selected as card type (EEPROM size 1K , RAM 256 bytes, Protocol Support: T=1). The option window for card program has shown in **fig. 6**.

The option windows of Food Ration Card with security

In BasicCard, professional BasicCard version ZC4.5 has been selected as card type (EEPROM size 30K, RAM 1K, Protocol Support: T=0, T=1) because it used AES & RSA encryption methods and its memory is enough for the application requirements.

The options of this project chosen just like what has done in the Food Ration Card without security; but setting of the card type to professional BasicCard ZC4.5 and virtual port to 201.

THE SIMULATION RESULTS

In order to discuss the results from each simulation of three models, we will discuss each one individually.

Simulation Results of the Food Ration Card without security

After writing (Terminal-Write, Terminal-Read &BasicCard) programs, the first step of the simulation was to compile the project files from source code form to p-code debugging form. The compilation results have shown in the **table 1**.

After, the compilation the Terminal-Write program will be opened in parallel with Food Ration Card to issue the initial information to card as shown in **fig. 7.** The follow of the Command-Response is done manually by press “step to card” and” step to terminal” bottoms in debugger window. After this all the information is written in the Food Ration Card and read from the card when Terminal-Read program opened in parallel with the Food Ration Card program.

Because, this project in simulation phase, each time the Food Ration Card program is closed, it must be reinitialized when we opened again, because no EEPROM is valid in the virtual card.

Simulation Results of the Food Ration Card with security

The compilation results of Food Ration Card with AES-128, RSA-128 have shown in table 2, 3 respectively.

After that, the simulation of the Terminal-Read program and BasicCard has done to read all the information has written in the Food Ration Card. The **fig. 8** shows the simulation process of Food Ration Card with AES-128.

The execution time of three models, (Food Ration Card without security and Food Ration Card with security by using AES-128 &RSA-128) has shown in **table 4**.

CONCLUSIONS

The following conclusions have been drawn from the simulation results of three models of the Food Ration Card (the first is Food Ration Card without using any security technique, the second simulates the same application but with security technique by using AES-method and the third by using RSA-method.).

- When we added the security to Terminal-Read program, we found that the code size and the data size of it is increased because of adding the code of security as well as the function of it.

Where, the difference in code size is (159 bytes) in AES-128 and (157 bytes) in RSA-128. While difference in data size is (84bytes) in AES-128 and (199 bytes) in RSA-128.

- When we added the security to Food Ration Card program, we found that difference in EEPROM of BasicCard is (387 bytes) in AES-128 and (675 bytes) in RSA-128.
- Another important point in this comparison is the execution time, where the execution time of Food Ration Card without security is (25 seconds), while the execution time of Food Ration Card with AES-128 is (29 seconds) & RSA-128 is (30 seconds).

REFRENCES

Wenzhou L., ChangQing C., and Zhuo Z., 2010, "Fingerprint ID Cards and the Recognition System", The 5th International Conference on Computer Science & Education.

Stefan M., Markus D., and Christoph G., 2010,"A Smartcard based approach for a secure energy
management node architecture", IEEE.

Daemen J. and Rijmen V., 1999, "AES Proposal:Rijndeal-Second Edition "

Everett D., 1993, "Smart Card Tutorial "

Guilfoyle T., March 2005, "The ZeitControl BasicCard Family", ZeitControl CardSystem GmbH, Germany

Rankl W. and Effing W., 2003, "Smart Card Handbook-Third Edition", Jhon Wiley& Sons, Ltd., Germany

Schneier B., 1996,"Applied Cryptography Algorithms and protocols- Second Edition", Jhon Wiley& Sons

LIST OF ABBREVIATIONS

AES**:** Advanced Encryption Standard
DES**:** Data encryption standard
ECC**:** Elliptic Curve Cryptography
EEPROM**:** Electrical Erasable Programmable
ISO**:** International Standard Organization
NIST: National Institute of Standards and Technology
PIN**:** Personal Identification Number
RAM**:** Random Access Memory
RSA**:** Rivest-Shamir-Adelman
SW1SW2**:** Status Signal.

Table 1: The compilation results of Food Ration Card

Program name	Size (bytes)	Description	Usage percentage
Termina 1-Write	3020	Code size	5%
	2623	Data size	4%
Termina 1-Read	3495	Code size	5%
	2847	Data size	4%
Food Ration Card	406	EEPROM size	41%

Table 2: The compilation results of Food Ration Card with AES-128

Program name	Size (bytes)	Description	Usage percentage
Terminal-Write	3020	Code size	5%
	2623	Data size	4%
Terminal-Read	3654	Code size	6%
	2931	Data size	4%
Food Ration Card	793	EEPROM size	3%

Table 3: The compilation results of Food Ration Card with RSA-128

Program name	Size (bytes)	Description	Usage percentage
Terminal-Write	3020	Code size	5%
	2623	Data size	4%
Terminal-Read	3652	Code size	6%
	3046	Data size	5%
Food Ration Card	1081	EEPROM size	4%

Table 4: The execution time of Food Ration Card

Program name	The execution time (sec)
Food Ration Card without security	25
Food Ration Card with AES-128	29
Food Ration Card with RSA-128	30

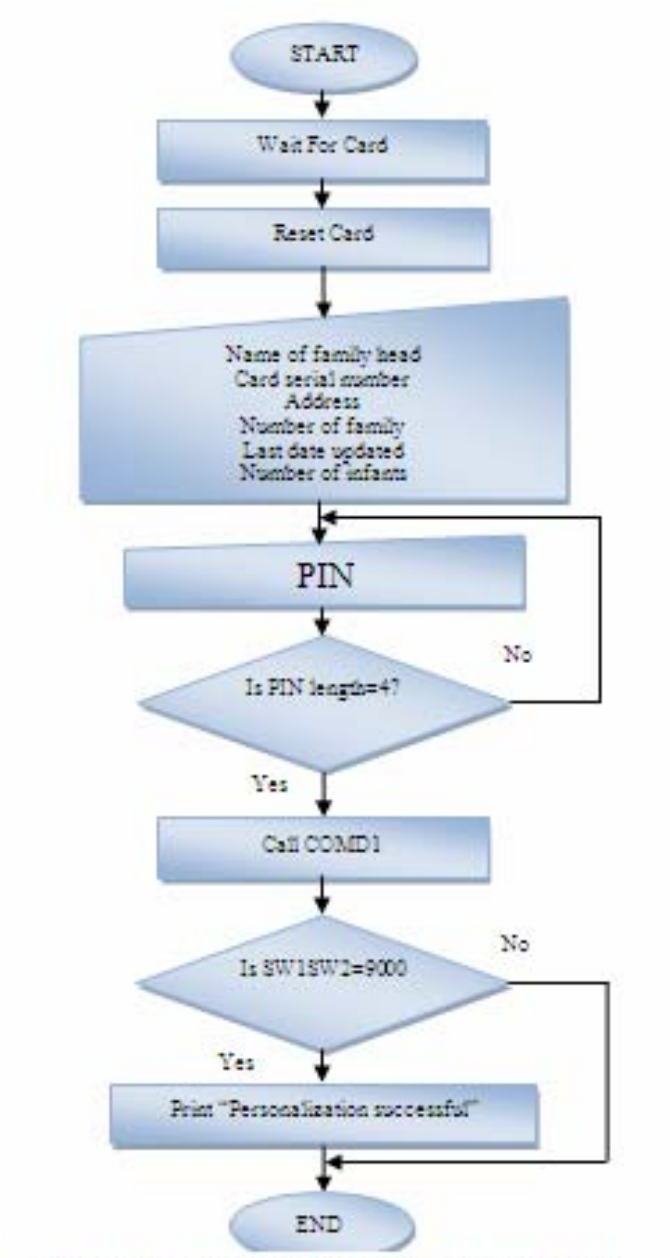

Fig. 1: Terminal- Write program flowchart

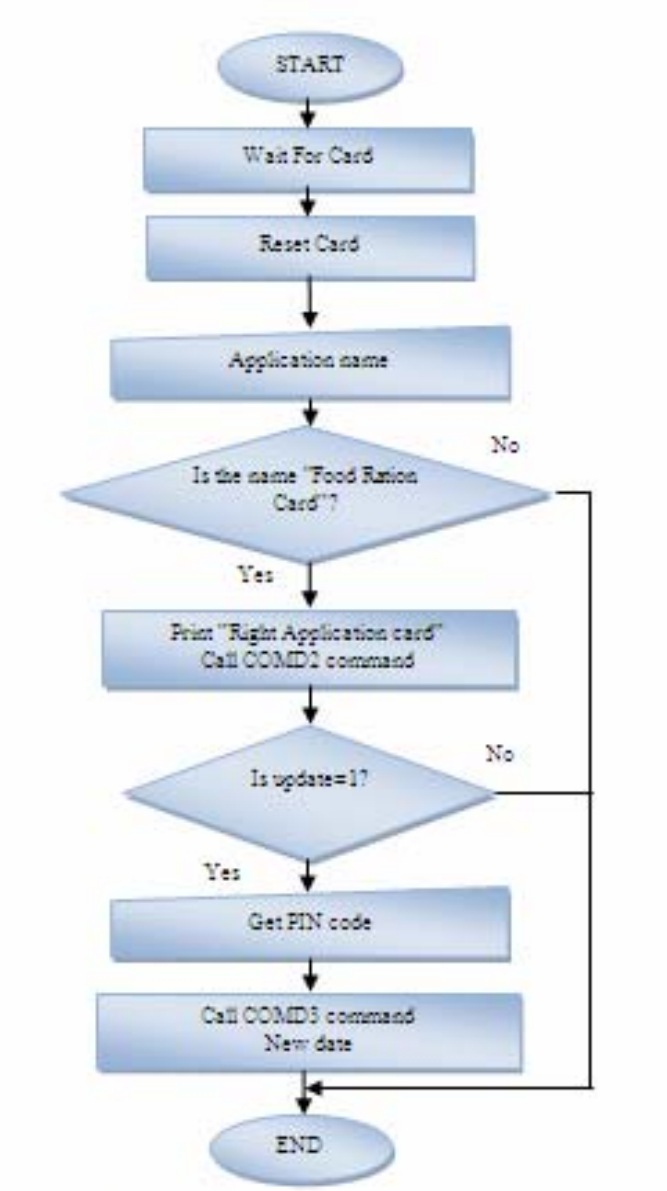

Fig. 2: Terminal- Read program flowchart

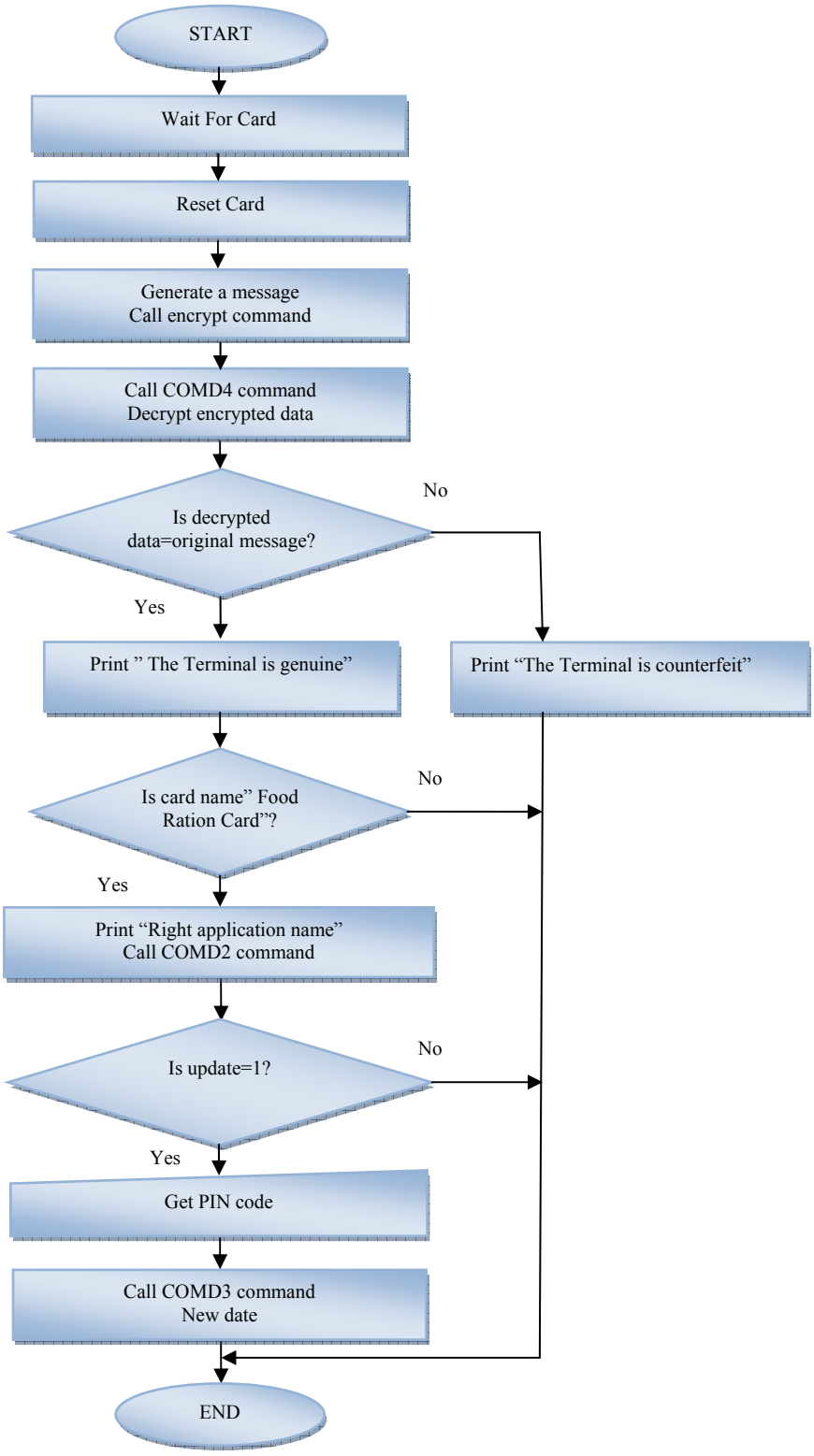

Fig. 3: The flowchart of the terminal-Read program with security

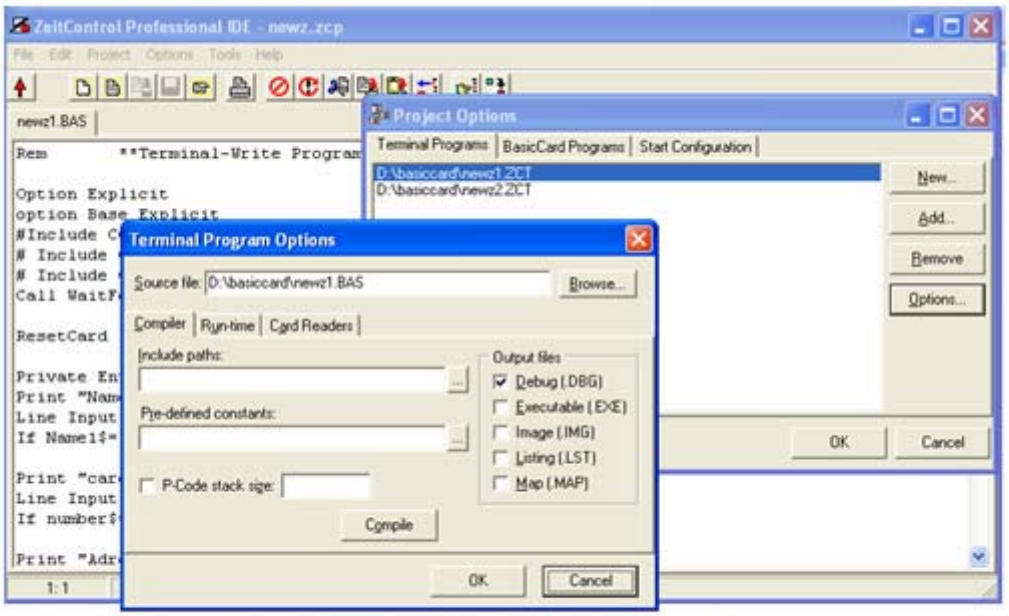

Fig. 4: Project option window for Terminal program

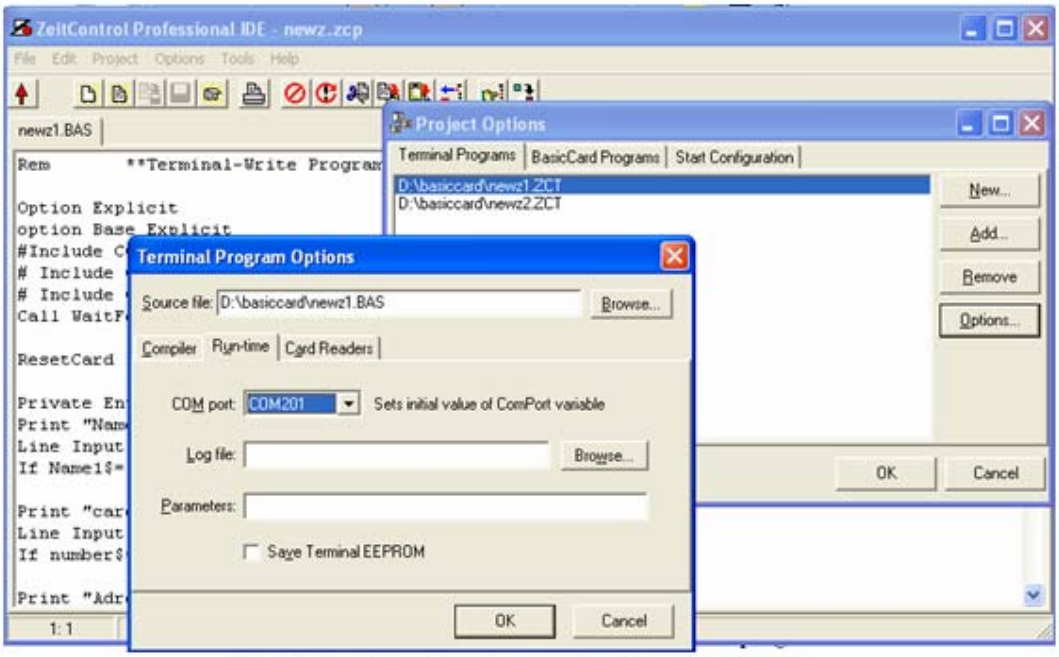

Fig. 5: Setting virtual port for Terminal program

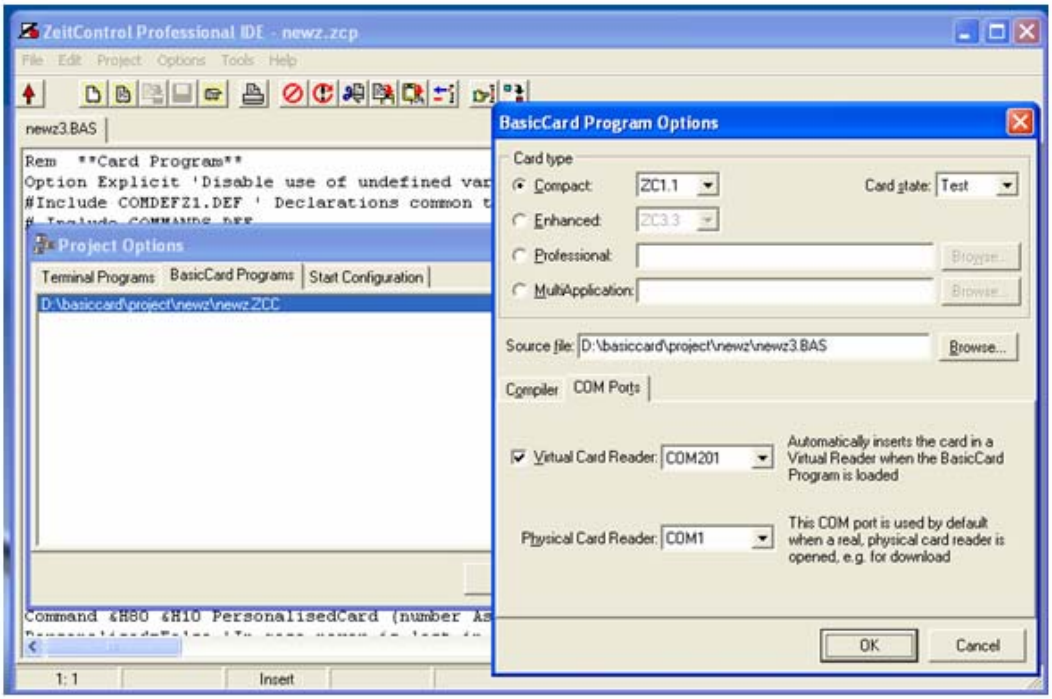

Fig. 6: Option windows for setting Card program

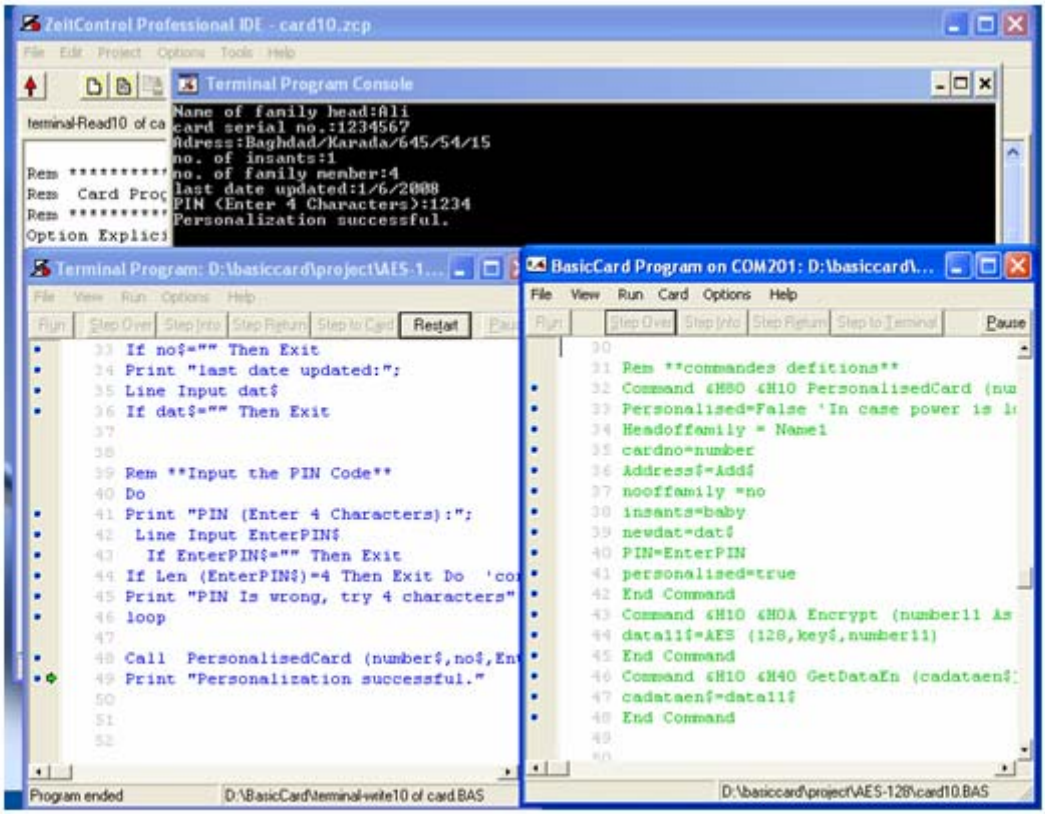

Fig. 7: The debugger windows for Terminal-write and Food Ration Card programs