Building an Application that reads Secure Information Stored on the Chip of the Citizen Identity Card in Vietnam

Van-Hoan Le
Weapon Institute, Vietnam
hoanle.aiot@gmail.com

Nhu-Quynh Luc
Academy of Cryptography Techniques, Vietnam
quynhln@actvn.edu.vn
(corresponding author)

Toan Thanh Dao
University of Transport and Communications, Vietnam
daotoan@utc.edu.vn

Quang-Trung Do
Academy of Cryptography Techniques, Vietnam
trungdq1980@actvn.edu.vn

Received: 29 November 2022 | Revised: 24 December 2022 | Accepted: 31 December 2022

ABSTRACT

Reading the information on the CIC/passport is very meaningful in serving the life activities of citizens in Vietnam and of foreign citizens visiting Vietnam. This research is based on the operating modes, such as BAC, FACE, and EAC to read the data contained in the chip put on the Citizen Identity Card (CIC) out securely. Specifically, the authors used the BAC mode to perform safe data reading from the CIC's chip. BAC mode uses 3DES and SHA1 algorithms to encrypt data to ensure security, so when the data are transmitted from the chip they are encrypted and decryption is performed by the application. In this paper, a complete application has been built for reading personal information stored securely on CIC. This application is built based on the BAC reading mode corresponding to CIC in Vietnam and meets the requirements of the ICAO 9303 standard, so it is able to read passports or CICs of other countries that use this standard. The performance of this application when reading data stored on fields DG1, DG2, and DG13 is improved with a speed of about 1.2s - 1.3s for all 3 data fields.

Keywords-BAC; EAC; PACE; 3DES; MRZ; SHA; digital signature

I. INTRODUCTION

Smart card technology was first proposed in 1977 [1]. During the '90s, there was an explosion in smart card technology in Europe with the appearance of SIMs (in GSM devices) and the international payment systems MasterCard, Visa, and Euro-pay [2-4]. In 1994, the EMV system (Euro-pay, MasterCard, Visa) was widely used in many countries [5, 6]. Currently, smart cards are applied in many fields [7]. Physically, the smart card is compact in size and embedded with an integrated circuit to control access to resources [4, 8]. Communication between the smart card and the card reader is direct physical contact (Contact Smartcard) or through short-range wireless connection such as radio frequency identification (Contactless Smartcard) [8-10]. Operating systems designed for smart cards can provide personal identification, authentication, data storage, and application processing [9, 11, 12]. Smart cards using chips have very high reliability and safety for practical applications because some security algorithms and biometric features are installed [13-15]. Smart card technology with security is applied to digital ID cards (e-passport) as legal proof for different services in many countries [13, 16-18].

In Vietnam, the Government promulgated the Decree 577 on ID cards and regulations on issuance of ID cards in 1957, to

help citizens facilitate daily transactions [19]. After a while, this ID card showed some disadvantages: it can be faked, many people share the same number of ID cards, and the information on the ID card is lost after a period of use. Currently, with the trend of global digital transformation, the Government of Vietnam has launched a chip-based CIC to store and secure user information [20], in which, if the data retrieved in the chip are exactly the same as the data recorded on the card, the user information is valid [21]. Moreover, CIC with chips have many other advantages such as making international, online, and digital transactions [22].

CICs with chip in Vietnam are manufactured based on smart card technology in compliance with ICAO 9303 [22, 23]. In particular, the chip used in CIC has built-in cryptographic solutions to secure personal data, ensuring that information is safe and reliably authenticated. According to ICAO 9303, the access process to read security information stored on the CIC's chip operates in three modes: BAC (Basic Access Control), PACE (Password Authenticated Connection Establishment), and EAC (Extended Access Control) [17, 22-23]. The command architecture used is the APDU (Application Protocol Data Unit) protocol [7, 21, 24]. Data are stored and organized on the chip according to ICAO 9303 [22, 25], ISO/IEC 14443-4 [26], and ISO/IEC 7816-4:2013 [27] standards.

In Vietnam, a new chip-based CIC is going to be deployed in the community in 2022. There are almost no studies related to the reading of the information contained in the CIC/passport chip. Almost every citizen's transactions related to finance (such as withdrawing money from an ATM, buying and selling online, inter-bank financial transactions, etc.) and activities in life are related to the CIC/passport. Therefore, reading the information on the CIC/passport will make the transaction process between citizens, government agencies, and service providers much more convenient. In this article, an application that can read security information stored on the DG1, DG2, and DG13 partitions of the CIC's chip has been built and is discussed.

II. RELATED WORK

A. The Physical Surface of the CIC

Figure 1 shows the physical surface (front and back) design of the CIC. The information recorded on CIC includes full name, date of birth, expiration date, gender, hometown, portrait, and country. The back side has the information on the relevant characteristics of the citizen. The MRZ information area recorded on the back of the CIC is an area used to be read by a machine, facilitating inspection and reducing execution time in administrative procedures [25]. In addition, MRZ also provides the ability to verify information in the VIZ. MRZ is formatted according to ICAO 9303 standard to ensure machine readability in different countries. Therefore, OCR-B font is used to store data in MRZ [25].

The hardware design of the chip ID card is organized into two main parts: Power Reserve and Main Card. In particular, the Power Reserve helps CIC collect and reserve energy to ensure that the chip on CIC works when performing transactions related to the CIC. The Main Card part consists of the RFID chip, the CIC chip, antenna, and integrated circuit. The RFID chip is in charge of receiving power and making the connection with the antenna. The data will be transmitted through the antenna and if encryption or decryption is needed, the RFID chip will make the connection and perform encryption and decryption with the CIC chip. The CIC chip performs the main job of executing cryptographic algorithms and communicating with the RFID chip to transmit information securely and store encrypted personal information. The CIC chip is connected to the power source to ensure the energy used to store the password key and ID of the card when the power is not directly supplied.

Fig. 1. (a) Front side and (b) back side (b) of the CIC.

B. File Organization on the Operating System in the Chip of the CIC

According to the ICAO standard [19, 20], the data in the chip of CIC are in a hierarchical form. First, there is the root directory (Master File -MF) and then the dedicated directory (Dedicated File - DF), and finally the Elementary Files (EF). The root directory of the CIC is divided into 2 DFs and 3 EFs, in which Future Application is a dedicated directory for future development, EF.CardAccess is the elementary file used for PACE mode access, EF.CardSecurity is the elementary file used for PACE mode access. EMRTD is a dedicated directory that stores the basic data files of each citizen. This is the most important folder of the CIC because all user information will be located in there. The EMRTD stores basic files, also known as data partitions and is described according to the international standards ICAO 9303 for CIC/e-passport. Citizens' data are stored in basic files in the EMRTD Application directory including EF.COM, EF.SOD, and DG1-16. EF.COM stores Unicode version information and a list of data groups available in the CIC, DG1 stores individual basic details (MRZ), DG2 stores citizens' facial images, DG3 stores the fingerprint, DG4 stores an iris picture, DG5 stores features of the face, DG6 is expanded to store information that will be added later, DG7 stores citizen's signature, DG8, DG9, DG10, DG11 are currently not used, DG12 stores supporting information, DG13, DG16 store citizen-related information, DG14 is used for the EAC access mechanism, DG15 stores the public key, and EF.SOD stores the hash value of each field.

C. Communication between CIC and the Computer

The computer is connected to the card reader via a serial port, USB, or Bluetooth to read the card data through either contact or contactless communication. After the CIC is inserted into the slot of the card reader, the computer connected to the card reader will detect the device through the ATR message sent from the CIC. After receiving the card, the computer and the card establish a connection and select the protocol for data transmission (TAPDU). When the data transfer protocol is

selected, the process of sending and receiving data between the computer and the card is done through APDU commands. When the computer sends the APDU command to the card, the card will receive the command and send a response back to the computer. Finally, after the successful data exchange, the CIC is removed from the slot. The computer will detect the discarded device through the card reader.

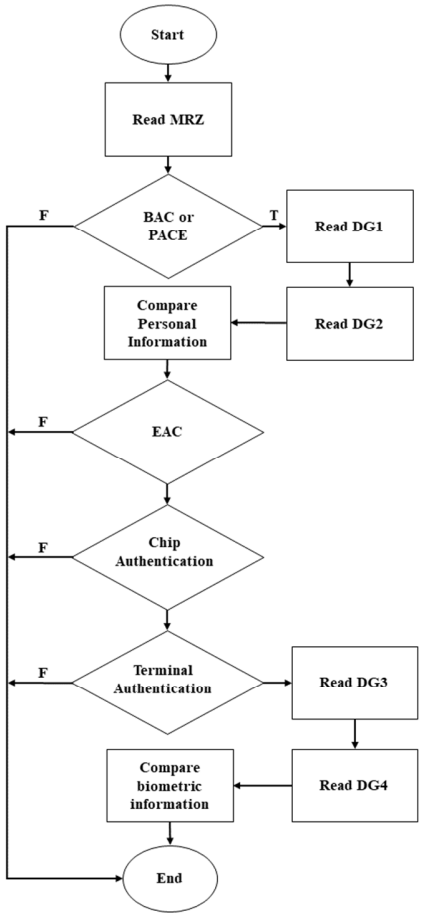


Fig. 2. Reading CIC data with BAC, FACE, and EAC modes.

D. Data Reading Modes

The read access process of the CIC chip includes three secure access modes, each of them determining the data partitions allowed to be accessed. The CIC access modes are BAC, PACE, and EAC [22, 24, 25]. The BAC mode ensures that only authorized parties have access to read the data. The PACE mode implements password-based access and Diffie-Hellman key agreement. The password used for PACE depends on the digital certificate of the CIC, using AES or 3DES algorithms to encrypt and authenticate the data [28, 29]. This mode allows reading, similar to BAC mode, the user's information data including name, date of birth, card number, portrait, relative information, and other personal characteristics. EAC mode is an access mode that combines advanced security features between the terminal and the CIC to protect and restrict access to sensitive personal data contained on the chip. It enables mutual authentication and establishes a secure communication channel between the CIC and a terminal. In contrast to ordinary personal data that can be protected by basic mechanisms, more sensitive data (such as fingerprints or iris images) must have enhanced protection to prevent unauthorized access. To access this mode, users need to establish a secure connection based on either BAC or PACE mode. EAC access mode is implemented through two parts: Chip Authentication that uses the Diffie-Hellman key exchange algorithm to exchange new session keys and Terminal Authentication that uses RSA digital signature algorithm or digital signature on an elliptic curve. Figure 2 describes in detail the process of reading data when combining the three modes. To retrieve the data in CIC, we first need to get the MRZ information contained on the CIC that can be entered or scanned. Then, we proceed to access with two modes, BAC or PACE. When either of these two modes is successfully set up, the individual can access the partitions DG1 and DG2 to get information, portraits and some information in the related data partition (if permitted). That information is then compared with the personal information contained on the CIC to check whether the user information is valid. After successful data retrieval, EAC advanced access mode can be selected to read data in partitions DG3 and DG4 to get information about fingerprint and iris images and compare them with other personal information. In this study, we aim to use BAC mode to read information stored on the CIC chip. The data read include: information of citizens themselves stored on the DG1, portrait information stored on the DG2, and citizen-related information such as parents, special identities, etc. that are stored on DG13.

III. DESIGNING AND BUILDING AN APPLICATION TO READ INFORMATION SAFELY FROM CIC WITH CHIP IN VIETNAM

A. Setting Up the BAC Mode to Read Data

Figure 3 shows the details of the four stages needed to be performed when the application connects to the CIC with chip, including: First, get information from the MRZ on the CIC as a key seed. Second, establish session key pair (KENC, KMAC) from the key seed. Third, authenticate and exchange new session keys with CIC. Finally, proceed to read the data on the chip using the new session key pair to encrypt and authenticate each other. BAC mode is the access mode to the CIC using 3DES symmetric encryption in CBC mode to encrypt as well as generate the token authentication code. Figure 3 presents details of the test execution procedure for the CIC with chip to connect to the card reader with BAC mode.

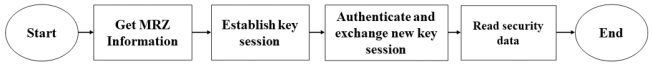


Fig. 3. Software read stages with the BAC mode.

First, the PC will get the MRZ information contained on the CIC either manually or by scanning. The PC receiving the MRZ will generate a key encryption code and a key message authentication code (KENC, KMAC). Next, the PC requests a notification from the CIC via the APDU command sent to the CIC. CIC generates a response message M sending a response

back to the PC. The PC receives the message M from the CIC and then generates a message M' and the derived key Kifd. It calculates S = M ‖ M' ‖ Kifd, encrypts S using a Kenc key, and generates message authentication code with ciphertext using the Kmac key. Next, the PC constructs the APDU instruction with the data E ‖ MAC sent to the CIC. CIC's chip will receive decryption and authentication data with Kenc and Kmac keys and check if the message M sent before is correct. If it is correct, the CIC will generate the derived key Kic and a new key seed by xor-ing Kic with the key derived from the incoming PC code (Kifd). Finally, the CIC generates a new session key pair, i.e. Kenc(new) and Kmac(new), generates a new message M2 = M ‖ M' ‖ Kik, encrypts message M2 with the Kenc key, generates the token authentication code, constructs the APDU instruction with the data E ‖ MAC, and sends a response back to the PC. At this point, the PC receives, decrypts, and authenticates the data to check if it is the message M' sent before. If it is correct, the PC will generate the key seed by xor-ing Kifd and the incoming CIC derived key (Kic), and will generate a new session key pair Kenc(new), Kmac(new).

When both the chip and the card reader complete the connection tests successfully, the two sides will communicate using the new session key pair to encrypt and authenticate data, with the following process: the PC sends the APDU command read data to CIC, CIC encrypts data and sends the APDU commands back to the PC. Then, the PC receives the decoded data, validates the data, and checks the received information. This is demonstrated in the application by the following steps: First, the application will proceed to get the MRZ information data as the key seed, then will calculate and set up a session key pair including KENC and KMAC. The application then sends a connection to the CIC and the CIC authenticates and initiates a new session key exchange. When the new session key exchange is successful, the application will proceed to read the specified data partitions using the session key pair to encrypt and authenticate the data.

1) Session Key Establishment

Session Key Establishment is the process by which a user generates a session key pair based on the MRZ information provided by the CIC. The MRZ information serves as the key seed to generate the session key pair [25]. According to the ICAO 9303 standard, the process of establishing session key [22, 25] consists of four main steps:

- **Step 1**: Build MRZ information (organized with 24 numbers according to the following structure: *Document Number* - Consists of 9 digits of CIC from the end, *Birth* is 6 digits on the CIC, *Expiry* is 6 digits of the expiration date on the CIC, *Check* is 1 byte calculated by the formula Check = (Document Number + 7 + 3 +1 + 7 + 3 +1 + 7 + 3 +1) mod 10).
- **Step 2**: Generate the key (here using SHA1 hash).
- **Step 3**: Generate KENC encryption key (using true random source in chip).
- **Step 4**: Generate key message authentication key (using SHA1 and KMAC). This process is detailed in the ICAO standard [19].

After successful session key setup, a key pair consisting of an encryption key (KENC) and a key message authentication key (KMAC) will be generated for secure exchange with the CIC.

2) Authenticating and Exchanging the New Session Key

This process is done after the session key pair has been previously established. The application will request a message from the CIC, then proceed to use the previously set session key pair to encrypt and generate an authentication code for that message, and then send it to the CIC. The CIC decodes and authenticates the incoming data. If successful, the card will proceed to generate a new session key pair consisting of KENC(new) and KMAC(new) and then send it back to the application. The two sides of the application and the CIC will be accessed through this new session key pair [25]. First, during the process of authentication and exchange, a new session key is performed, an RND.IC message (8 bytes) is randomly requested from the CIC. The application performs the following operations: Generate random RND.IFD (8 bytes), generate a random K.IFD (16 bytes) key, Join S = RND.IFD ‖ RND.IC ‖ K.IFD (RND.IC sends request to CIC), encode EIFD = E (KEnc, S), calculate the message authentication code MIFD = MAC(KMAC, EIFD), build command APDU(EIFD ‖ MIFD), and send the APDU command to the CIC. Then, CIC receives the data sent to the APDU(EIFD ‖ MIFD). EIFD conducts decryption and authentication with application via MAC, extracts RND.IC from S and checks if it returns the correct value, generates K.IC key and generates Kseed(new) = K.IC XOR K.IFD, generates a new session key pair KEnc(new), KMac(new), joins R = RND.IC ‖ RND.IFD ‖ K.IC, encrypts EIC = E(KEnc, R) and computes the message authentication code MIC = MAC(KMAC, EIC), constructs the PDU(EIC ‖ MIC) command, and sends the APDU command back to the application. The application receives the response data from the APDU tag(EIC ‖ MIC), conducts EIC decryption and authentication with the CIC via MAC, extracts the RND.IFD from R and checks if the tag returns the correct value, generates Kseed(new) = K.IC XOR K.IFD, and generates a new session key pair (KEnc(new), KMac(new)). Finally, the application and the CIC will generate a new session key pair that is used to encrypt, decrypt, and authenticate the data when reading the data file used for secure data reading.

3) The Process of Reading Secure Data

It is done after the software and the CIC have authenticated and set up the session key. During the reading process, the application and the CIC encrypt, decrypt, and authenticate the data based on a new session key pair to ensure that they are confidential and authenticated. Card reading is done through properly constructed APDU commands sent to the chip [DO'85' or DO'87'] [DO'97'] DO'8E'. The response to the APDU command is [DO'85' or DO'87'] [DO'99'] DO'8E'. In this module, the data on each partition are different, so the structure of the APDU command sent to the card is also built differently and must follow the rules of sending access commands to that partition. After sending a command to that partition, there will be response data with status bytes. If the returned data are correct and the status bytes return "9000", then the command sent is correct. In the CICs in Vietnam, the data areas are: DG1

(MRZ information), DG2 (face image encoding), DG3 (fingerprint encryption), DG4…DG12 (backup), DG13 (additional personal information), DG14 (read condition of EAC mode), DG15 (passive authentication), DG16 (backup).

B. Analysis, Evaluation, and Design of the Application

Figure 4 presents the details of the system and the operation of the reading data application. The application is built with the following devices: Scanner (gets MRZ information from the card), card reader (communication device that transmits APDU commands from the computer to the CIC and vice versa), and software (retrieve and process data in CIC). The user will provide a CIC chip and the scanner connected to the computer will take the MRZ information from the card as input. At the same time, a card reader is connected to the computer through the built-in data reader software. After obtaining the MRZ information as input to the data reader software, CIC will communicate with the software through the previously connected card reader. The application then processes and retrieves data from the card.

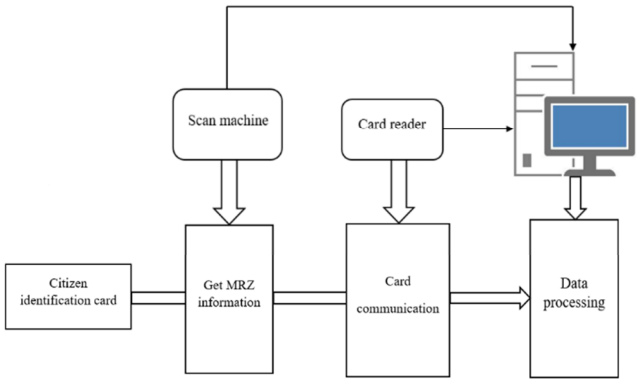


Fig. 4. Built-in system and program that reads the security information of the CIC.

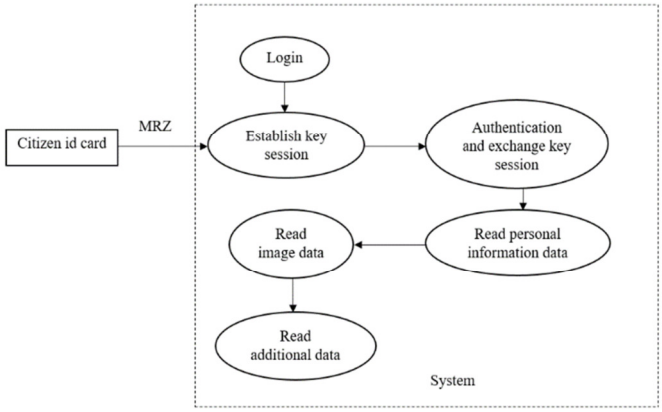


Fig. 5. ID card security information reader program flow.

Figure 5 shows that the software is built into many different modules, each with different functions. The software is built based on the BAC mode. The software is divided into the following modules: Login module, module for setting session key, authentication and session key exchange module, personal data reading module, image data reading module, and supplementary information reading module. First, users need a password account to log in to the main module of the software. The MRZ of CIC is then put into the session key setup module to generate a session key pair for the next process. The session key exchange and authentication module receives the session key pair, establishes a secure connection to the CIC, and exchanges a new session key pair. The personal information data read module retrieves the data from the DG1 partition. The image data reading module will get the data from the DG2 partition. The additional information reading module retrieves data from partition DG13.

IV. RESULTS AND DISCUSSION

A. Building and Designing Modules in CIC Reader Software

Figure 6(a) describes in detail the operation flow of the login module. At first, on the login interface of the software, the user needs to enter the account/password. Next, when the software's module works, if the information is valid, it will take the user to the main interface. In case of incorrect input, the software will not allow login. Figure 6(b) details the operation flow of the authentication module and session key establishment. This module will process information according to the following steps:

- Step 1: The user enters the MRZ on the CIC into the software.
- Step 2: The entered MRZ information is processed in accordance with the provisions of ICAO 9303 standard to get the necessary information.
- Step 3: After obtaining the necessary information from the MRZ, the module will proceed to generate the key.
- Step 4: After having the key seed, the processing software generates a pair of session keys KENC (encryption) and KMAC (key message authentication code).

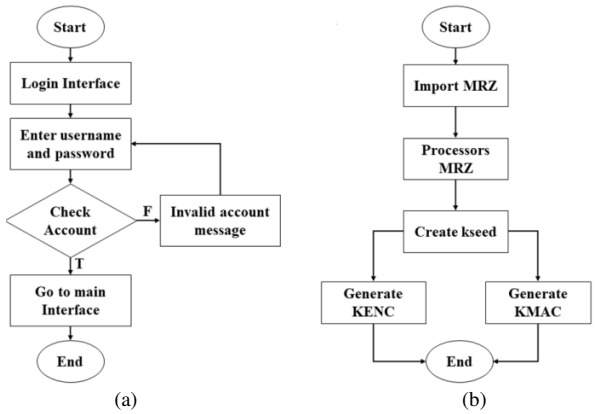


Fig. 6. (a) Login module activity flow, (b) session key exchange and authentication module activity flow.

The new session key exchange and authentication module is operational:

- Step 1: User generates a random plaintext block.

- Step 2: Encrypt and generate token authentication code from plaintext block from KENC AND KMAC set up before.
- Step 3: Connect the cipher-text and the generated message authentication code.
- Step 4: Send S to the card for personal authentication.
- Step 5: If it is legitimate, the card will respond and receive a resend notice.
- Step 6: Decrypt the message and authenticate with the card.
- Step 7: Get the key seed from the card's response message.
- Step 8: Set up a new session key for encryption and token authentication. The details of the operation flow of this module are shown in Figure 7(a).

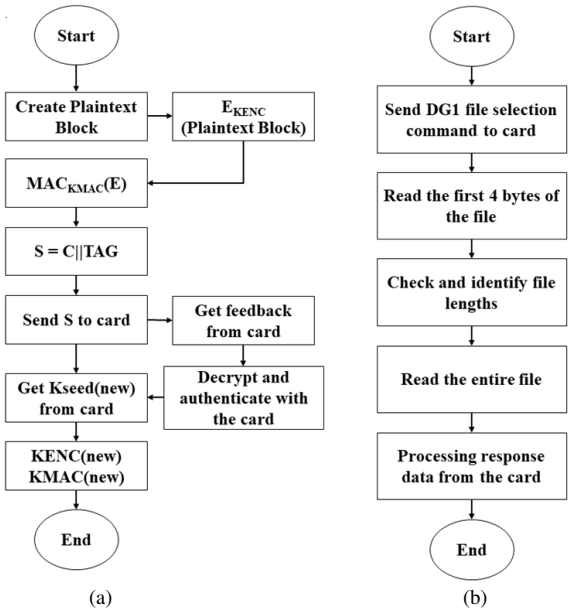


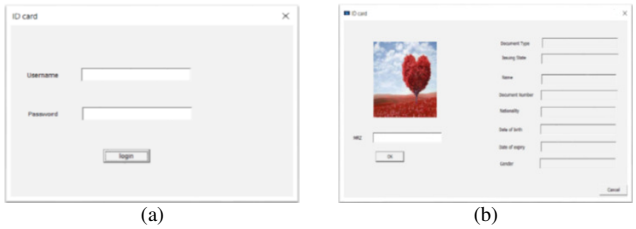
Fig. 7. (a) New session key exchange and authentication module activity flow, (b) operation flow of the data reading module (personal information).

Figure 7(b) shows the details of the operation flow of the personal data reading module. When this module works, the information stored on the CIC read out is processed through the following steps:

- Step 1: Send the file selection command to DG1 to read the card's personal information.
- Step 2: Send the command to read the first 4 bytes of the DG1 file.
- Step 3: Get a response from a valid check card and determine the length of data contained in the file.
- Step 4: Proceed to read the entire file.
- Step 5: Get response from the data processing card from decoding, validating, and extracting valid information. In addition, there is a module to read image data and additional information. The module that reads image data and reads additional information is basically the same as the module that reads personal data, except for the method of file selection and the processing of the received data.

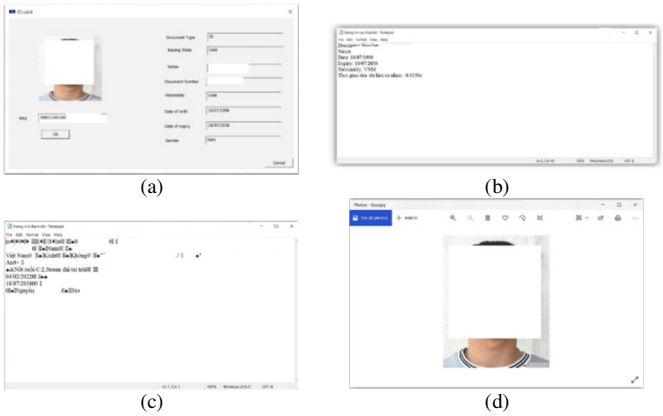
B. Building and Perfecting the Functions of the CIC Reader Software

In this study, the results of the readings from the CIC will be obscured, due to concerns regarding confidentiality and privacy. To build an application that reads security information on a CIC, we used a number of devices and supporting tools: a Duali DE-620 smart card reader, a PC with Visual Studio 2019 installed, programming language: C/C++, OpenSSL library. Figure 8(a) shows the designed login module interface. On this interface, citizens only need to enter the correct account and password and then they can start working with the CIC reader software with the main interface of Figure 8(b). On this main interface, if a citizen wants to read the information stored on the CIC, it is only needed to enter the MRZ information string (written on the back of the CIC).



(a) (b)

Fig. 8. (a) Login interface, (b) main interface.



(a) (b) (c) (d)

Fig. 9. (a) Results of reading personal information on the CCCD, (b) results of reading data on DG1, (c) data reading results on DG13, (d) data reading results on DG2.

Figure 9 shows the results by the software when reading security information stored on the CIC's chip. Figure 9(a) is the citizen's output displayed on the CIC reader software. Figures 9(b),(c) show the detailed results of personal information read on DG1 and DG13. The read out results are saved on 2 separate corresponding txt files. Figure 9(d) shows the result of a detailed portrait of an individual read on the DG2 partition that is saved on a txt file. The results obtained when reading individual data on the chip CIC include: Document Type, Issuing State, Full Name, CIC Number, Nationality, Date of

Birth, Expiration Date, Gender, Portrait Image, and other personal data.

TABLE I. READING TIME RESULT OF SECURED DATA ON THE CIC CHIP

No	End-to-end authentication time (s)	Reading time of DG1 (s)	Reading time of DG2 (s)	Reading time of DG13 (s)	Sum (s)
1	0.1950	0.0260	0.9630	0.0360	1.22
2	0.1960	0.0260	0.9640	0.0380	1.224
3	0.1960	0.0260	0.9570	0.0370	1.216
4	0.1950	0.0260	0.9430	0.0380	1.202
5	0.1950	0.0250	0.9640	0.0390	1.223
Average	0.1954	0.0258	0.9582	0.0376	1.217

Finally, in order to evaluate the performance of the CIC reader software, the time to read citizen information stored on each partition DG1, DG2, DG13 and end-to-end authentication of chip CIC was determined. Table I gives the results of the software's execution speed performance when reading data. It is seen that the total time to read the entire data (end-to-end authentication, DG1, DG2 and DG13) is approximately 1.217s. So, the software reading data on the CIC has an average time of about 1.2 to 1.3s, which is a relatively fast read speed, which is suitable for real time applications.

C. Analyzing, Evaluating, and Testing the Source Code of the CIC Reader Software

The Fortify Static Code Analyzer toolkit (Version 22.1.0.0166) was used to analyze and evaluate the CIC reader software source code. Table II gives the detailed results. There are error issues related to Buffer Overflow, Variable Never Used, Integer to Character, Signed to Unsigned, Unchecked Return Value, but these problems are not of a serious nature and don't affect the security of the software. Specifically, these issues are analyzed below.

- There is a problem related to buffer overflow (Buffer Overflow: 1 issue), which is warned in the `memcpy` function (`void *memcpy(void *str1, const void *str2, size_t n)` of the reader, occurring when the size of the copied array is smaller than the data array size, which does not guarantee program optimization according to [30-33]. The Fortify Static Toolkit Code Analyzer showed how to fix the source code (check the correct copy array length and number of bytes to copy) according to [34].

TABLE II. RESULTS OF SOURCE CODE OF CIC SOFTWARE TESTING WITH THE FORTIFY STATIC CODE ANALYZER

Category	Fortify Priority (audited/total)				Total Issues
	Critical	High	Medium	Low	
Buffer Overflow	0	0 / 1	0	0	0 / 1
Poor Style: Variable Never Used	0	0	0	0 / 2	0 / 2
Type Mismatch: Integer to Character	0	0 / 1	0	0	0 / 1
Type Mismatch: Signed to Unsigned	0	0 / 1	0	0	0 / 1
Unchecked Return Value	0	0	0	0 / 1	0 / 1
Weak Cryptographic Hash	0	0	0	0 / 5	0 / 5

- There are two problems related to unused variables (Variable Never Used: 2 issues). The error occurs because two variables, `num` and `lRetvalue` have been declared but not used. The fix is to delete these two variables.
- There is an issue related to integer to character conversion (Integer to Character: 1 issues). The error is that the function returns an unsigned char to an int type, but the return value is assigned to a char type. The fix is to resynchronize the return value with the same type.
- There is an issue with the signed to unsigned conversion (Signed to Unsigned: 1 issues). The error occurs because the arrays `ch[] (type char)` and `mrz_input[] (type unsigned char)` are not of the same data type. The fix is to re-declare one of the two arrays with the same data type char or unsigned char.

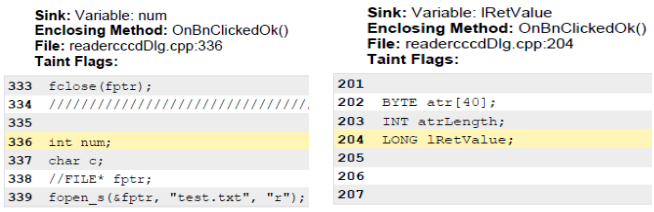


Fig. 10. Unused variables.

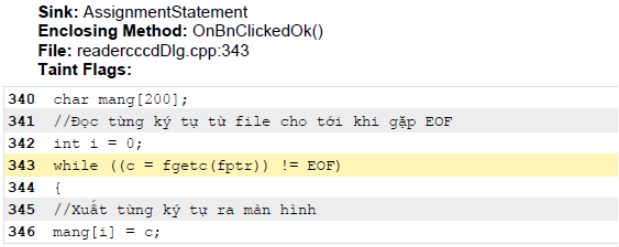


Fig. 11. Integer to character conversion issue.

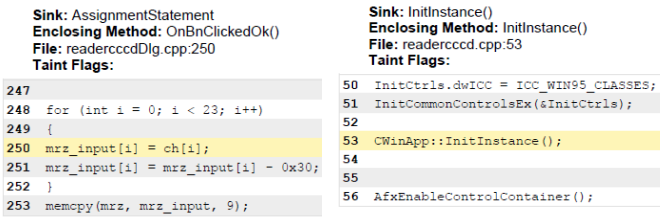


Fig. 12. Unchecked return value issue.

- There is an issue with the unchecked return value (Unchecked Return Value: 1 issue). This error occurs when the `InitInstance()` function returns no value. Since `InitInstance()` is an interface function of the MFC available when creating the interface, it cannot be dropped.

Analyzing, evaluating, and testing the problems in CIC reader software with Fortify Static Code Analyzer toolkit (Version 22.1.0.0166), shows that the developed and built software is guaranteed source code-safe. This is enough to confirm that the CIC reader software will be secure against

some types of attacks when deployed in real-world commercial applications.

V. CONCLUSION

Reading citizen information stored on a CIC with chip is practical for daily services, such as electronic payments, online money transfers, online public services, etc. In this paper, we used the BAC mode to set up a secure channel in the process of reading citizen information stored on CIC. We used cryptographic algorithms such as 3DES, SHA1, and MAC, to ensure that during the authentication process between the chip and the terminal the communication in the process of reading citizen information from the chip on CIC is secure. As a result, the software reads the information stored on DG1, DG2, and DG3 fields with safety and security. The performance of the software has a reading and data processing speed of approximately 1.2 s to 1.3 s. The CIC reader software source code has been evaluated and tested with the Fortify Static Code Analyzer tool (Version 22.1.0.0166) to confirm the software its robustness and safety.

ACKNOWLEDGMENT

The authors thank the Academy of Cryptography Techniques and the Minister of Education and Training (MOET) for supporting this work under grant number B2022-GHA-10.

REFERENCES

[1] L. C. Guillou and M. Ugon, "Smart Card a Highly Reliable and Portable Security Device," in *Lecture Notes in Computer Science*, New York, NY, USA: Springer, 1987, pp. 464–479.

[2] K. Vedder, "GSM: Security, Services, and the SIM," in *Lecture Notes in Computer Science*, Berlin, Heidelberg: Springer, 1998, pp. 224–240.

[3] V. Guyot, "Smart card, the stealth leaker," *Journal in Computer Virology*, vol. 8, no. 1, pp. 29–36, May 2012, https://doi.org/10.1007/s11416-012-0159-y.

[4] W. Rankl and W. Effing, *Smart Card Handbook*, Fourth Edition. New York, NY, USA: John Wiley & Sons, 2010.

[5] D. Basin, R. Sasse, and J. Toro-Pozo, "The EMV Standard: Break, Fix, Verify," in *IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, Dec. 2021, pp. 1766–1781, https://doi.org/10.1109/SP40001.2021.00037.

[6] M. E. Haykin and R. B. J. Warnar, *Smart Card Technology: New Methods for Computer Access Control*. Gaithersburg, MD, USA: National Institute of Standards and Technology, 1988.

[7] K. Markantonakis, "Multi Application Smart Card Platforms and Operating Systems," in *Smart Cards, Tokens, Security and Applications*, K. E. Mayes and K. Markantonakis, Eds. Boston, MA, USA: Springer, 2008, pp. 51–83.

[8] W. Rankl and W. Effing, *Smart Card Handbook*, 1st Edition. Chichester, WS, England: Wiley, 2010.

[9] H. Guo, *Smart Cards and their Operating Systems*. Helsinki, Finland: Helsinki University of Technology, 2002.

[10] K. Mekki, O. Necibi, C. Boussetta, and A. Gharsallah, "Miniaturization of Circularly Polarized Patch Antenna for RFID Reader Applications," *Engineering, Technology & Applied Science Research*, vol. 10, no. 3, pp. 5655–5659, Jun. 2020, https://doi.org/10.48084/etasr.3445.

[11] K. R. Wilcox, "Multi-application smart cards: Card operating systems and application security," presented at the 21st Computer Science Seminar, 2003.

[12] E. Barker and N. Mouha, *Recommendation for Triple Data Encryption Algorithm (TDEA) Block Cipher*. Gaithersburg, MD, USA: National Institute of Standards and Technology, 2017.

[13] B. Robisson *et al.*, "Smart security management in secure devices," *Journal of Cryptographic Engineering*, vol. 7, no. 1, pp. 47–61, Apr. 2017, https://doi.org/10.1007/s13389-016-0143-4.

[14] T. Rosteck, *Connected Secure Systems (CSS) Division Call hosted by Deutsche Bank*. infineon, 2021.

[15] A. H. Al-Omari, "Lightweight Dynamic Crypto Algorithm for Next Internet Generation," *Engineering, Technology & Applied Science Research*, vol. 9, no. 3, pp. 4203–4208, Jun. 2019, https://doi.org/10.48084/etasr.2743.

[16] P. A. Karger, D. C. Toll, E. R. Palmer, S. K. McIntosh, S. Weber, and J. W. Edwards, "Implementing a High-Assurance Smart-Card OS," in *Financial Cryptography and Data Security*, Tenerife, Canary Islands, Jan. 2010, pp. 51–65, https://doi.org/10.1007/978-3-642-14577-3_7.

[17] O. Dagdelen, "The Cryptographic Security of the German Electronic Identity Card," Ph.D. dissertation, Technical University of Berlin, Berlin, Germany, 2013.

[18] U. Iftikhar, K. Asrar, M. Waqas, and S. A. Ali, "Evaluating the Performance Parameters of Cryptographic Algorithms for IOT-based Devices," *Engineering, Technology & Applied Science Research*, vol. 11, no. 6, pp. 7867–7874, Dec. 2021, https://doi.org/10.48084/etasr.4263.

[19] *Order of Identification and Regulations for Issue Issuance*. 1957.

[20] *Circular No. 59/2021/TT-BCA detailing the implementation of the Law on Citizen Identification*. 2021.

[21] *ISO/IEC 18013-3:2017, Information technology — Personal identification — ISO-compliant driving licence — Part 3: Access control, authentication and integrity validation*. ISO, 2017.

[22] *Doc 9303: Machine Readable Travel Documents: Part 3: Specifications Common to all MRTDs*, 8th ed. ICAO, 2021.

[23] "Radio Frequency Protocol and Application Test Standard for eMRTD – Part 3," ICAO, Technical Report, Mar. 2018.

[24] D. Cooper, H. Ferraiolo, K. Mehta, S. Francomacaro, R. Chandramouli, and J. Mohler, *NIST Special Publication 800-73-4: Interfaces for Personal Identity Verification – Part 1: PIV Card Application Namespace, Data Model and Representation*. Gaithersburg, MD, USA: NIST, US Department of Commerce, 2015.

[25] *Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS Token – Part 1*. Germany: Federal Office for Information Security, 2015.

[26] *ISO/IEC 14443-4:2008, Identification cards — Contactless integrated circuit cards — Proximity cards — Part 4: Transmission protocol*. ISO, 2008.

[27] *ISO/IEC 7816-4:2020(en), Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange*. ISO, 2020.

[28] H. Mestiri, I. Barraj, and M. Machhout, "AES High-Level SystemC Modeling using Aspect Oriented Programming Approach," *Engineering, Technology & Applied Science Research*, vol. 11, no. 1, pp. 6719–6723, Feb. 2021, https://doi.org/10.48084/etasr.3971.

[29] A. S. Alshammari, "Comparison of a Chaotic Cryptosystem with Other Cryptography Systems," *Engineering, Technology & Applied Science Research*, vol. 10, no. 5, pp. 6187–6190, Oct. 2020, https://doi.org/10.48084/etasr.3745.

[30] G. McGraw, "Software Security: Building Security In," in *17th International Symposium on Software Reliability Engineering*, Raleigh, NC, USA, Nov. 2006, https://doi.org/10.1109/ISSRE.2006.43.

[31] A. Apvrille and M. Pourzandi, "Secure software development by example," *IEEE Security & Privacy*, vol. 3, no. 4, pp. 10–17, Jul. 2005, https://doi.org/10.1109/MSP.2005.103.

[32] J. Koziol *et al.*, *The Shellcoder's Handbook: Discovering and Exploiting Security Holes*. New York, NY, USA: Wiley, 2004.

[33] M. Howard and D. LeBlanc, *Writing Secure Code, Second Edition*, 2nd ed. Redmond, WA, USA: Microsoft Press, 2003.

[34] "About Strsafe.h - Win32 apps," *Microsoft*. https://learn.microsoft.com/en-us/windows/win32/menurc/strsafe-ovw.