Ingénierie des Systèmes d'Information
Vol. 25, No. 2, April, 2020, pp. 139-152
Journal homepage: http://iieta.org/journals/isi

A Secure Cloud Password and Secure Authentication Protocol for Electronic NFC Payment Between ATM and Smartphone

Samir Chabbi[1*], Rachid Boudour[1], Fouzi Semchedine[2]

[1] Dept. of Computer Science, Badji Mokhtar University, Annaba 23000, Algeria
[2] Institute of Optics and precision Mechanic, University of Setif 1, Setif 19000, Algeria

Corresponding Author Email: s.chabi@univ-soukahras.dz

https://doi.org/10.18280/isi.250201

Received: 9 December 2019
Accepted: 26 February 2020

Keywords:
authentication; confidentiality, hash function, NFC, automated teller machine, smartphone payment, secure element

ABSTRACT

NFC (Near Field Communication) is a radio frequency wireless communication technology for less distance (less than 10cm). It operates at a frequency of 13.56 MHz recently, it has been used for electronic payment between an Automated Teller Machine (ATM) and a Smartphone. It can be menaced by attacks which stole personal data like the user password, the user bank account number ant its amount. So, it must be protected and secured. In this paper, we present a cloud secured password, a simple secured authentication protocol, a simple proposed hash function and a simple test of intrusion to secure the NFC payment between an ATM and a Smartphone against eleven attacks. The analysis of our solution proves that it defends against eleven attacks; it is cost-effectiveness in terms of hardware, cost of calculation, storage space and cost of communication. The proposed technique of password and the protocol use simple cryptography operations, a simple hash function and simple operators.

1. INTRODUCTION

The NFC (Near Field Communication) technology is a radio frequency wireless communication technology. In Near-Field Communication (NFC), the two communicating devices use an electromagnetic induction field to transfer information at a short distance [1]. The NFC is now used in many products such as Smartphone, Point of sale and Automated Teller Machine. Recently, more consumers are using their NFC Smartphone for electronic NFC payments [2]. In the electronic NFC payment with a Smartphone, the safety is to verify the security properties that are: mutual authentication of the two NFC devices (Smartphone and ATM), the confidentiality and the integrity of the exchanged data [3].

In our study, the authentication is to verify the identity of the NFC device (Smartphone and ATM) before a payment operation. The confidentiality of the user private data is to don't give the permission to use the NFC payment only to a proclaimed identity (authorized user). It requires the use of authentication methods like password or biometric technology [4]. The integrity of the user private data is to protect the confidential data (password, secure element identifier) to not be modified during the transfer between the Smartphone and the ATM [5]. The secure element (SE) is a circuit implemented in the Smartphone which integrates NFC applications, protocols and cryptography routines, secret keys and the private data of the user [6]. Several solutions are proposed to secure the electronic payment using NFCATM. They are based on password, biometric technology or authentication protocol. However, each solution has its limits and drawbacks.

In this paper, we propose a cloud secured password, an authentication protocol based on the use of nonce, a secure element identifier and a simple hash function that we propose and a simple test of intrusion to secure the electronic NFC payment between a Smartphone and an Automated Teller Machine. The protocol tries to authenticate the user with the password, using a proposed hash function. It is embedded in the secure element of the Smartphone. On the other hand, the server lunches an intrusion test when attempting to access the user account by an attacker. Further, the solution allows the user to select a credit card or a bank card from a list of cards embedded in the Smartphone. The security properties (Mutual authentication, confidentiality, data integrity and privacy) are verified by analysis and by the AVISPA tool. The proposed solution well verifies the recalled security properties and guarantees the protection of the user privacy. It is secured against eleven attacks and presents an authentication time of 2,17sec. It uses few numbers of xor operators, concatenation operators, and hash function operations. The proposed solution uses the technique of cache memory in the server and less degree of difficulties concerning the older user. A comparison with some existing solutions is investigated to evaluate the performance and the effectiveness of our solution.

The rest of the paper is organized as follows: Section 2 explores the related works for securing the NFC communication. We find in this section the existing security solutions used in the NFC payments. The Section 3presents our secure system architecture and describes its main components. It presents the proposed cloud password, the protocol, the hash function and the test of intrusion. The Section 4 illustrates how to check the security of the proposed solution against multiple attacks where verification by analysis and an automatic verification are presented. In Section 5, we perform a comparison between our solution and some well known methods used for authentication, and we present a performance evaluation. Finally, conclusions with some future works are presented in Section 6.

2. RELATED WORKS

Several works have been established as a part of securing RFID or NFC communications. We present in this section each solution and its limitation.

2.1 Active jamming

This technique is to use a device that sends a radio frequency field strong enough to prevent a reader from accessing the card or understanding its response. This solution can create a denial of service and may destroy the contactless systems nearby [7].

2.2 Distance bounding

It consists in setting an expulsion boundary between a transmitter and a receiver to detect relay attacks. This solution is difficult to achieve because it is difficult to isolate the small propagation time compared to the processing time which is not constant [7].

2.3 Google Wallet application security

On a Smartphone, the access to the Google Wallet application is protected using a personal code selected by the user. This code is encoded in binary form with other information and its hash function (SHA-256). If the PIN is recovered (by theft or loan), it is possible to make payments with a prerecorded credit card.

The GoogleWallet (GoogW) application does not fight against the attack of theft or loan of the credit card or Smartphone where the attacker can use the prepaid card [6]. The password capture is simple and direct. So, it suffers from the attacks of shoulder-surfing, simple record, multiple record, screen record, registration camera, auxiliary canal and spyware.

2.4 French payment cards

In April 2012, a security flow on an implementation of contactless payment on recent French cards was discovered where the disclosed information is sensitive (name, card number, transaction histories) [8].

2.5 EMV chip cards

The EMV (Euro Master Visa) chip cards (EmvCC) were targeted by a Man-In-The-Middle attack during the PIN code verification. This attack is called the Cambridge attack [6]. The EMV smart cards are vulnerable against the Man-In-The-Middle (Cambridge) attack [6]. It uses a simple and a direct password capture. So, it suffers also from the attacks of shoulder-surfing, simple record, multiple record, screen record, registration camera, auxiliary canal and spyware.

2.6 BioRFID protocol (Brfid)

The system is proposed as an authentication system based on the combination of two subsystems: a RFID system and a biometric system [9]. It is used for authenticating the user and the two communicating devices in a RFID communication. This system can be used in an electronic NFC payment between an NFC card and a reader. The BIORFID protocol is resistant to the Man in the Middle, Replay and Monitoring trace attack [9]. It is vulnerable to the theft attack of card and user fingerprint. It is expensive compared to our protocol because it uses a sensor for the user fingerprint.

2.7 FakePIN technique

In this system, the password consists of a sequence of alphanumeric characters and a direction. At each authentication session, the order of appearance of the characters or the numbers is changed on a displayed keyboard. To deceive the observer, the user must type a character provided by the combination of this character with the direction of the password to give the exact character of the password. This must be done for all characters of the password [10]. The FakePIN technique used to ensure user authentication by password is vulnerable against recording attack [11].

2.8 PassWindow method

This method is based on the use of a grid of icons. One icon must be preselected and considered as a password. The position of this icon must be memorized by the user. To ensure the authentication operation, a virtual keyboard with a grid without icons is displayed on the screen. To enter the PINs, the user must enter each digit in the location of the password icon while moving the device and must hide the input by isolating the camera back [12]. The PassWindow technique used to authenticate the user with a password is weak against the intersection of multiple records [11].

2.9 Cappcha (Capp) method

This technique provides authentication by entering a PIN code with the operation of tilting the device to a degree displayed on the screen for one second. This degree of inclination can be generated by an accelerometer integrated into the secure element of the Smartphone [13]. The technique suffers from the Shoulder-surfing attack and a concealed camera that can be hidden on an ATM to record the typed PIN or password. In the case of Smartphone theft, the attacker who steals the PIN code can carry out the payment operation because the Cappcha technique integrated in the secure element will display a random degree of inclination of the Smartphone that it will be simply followed and hence, the restored PIN code will be entered.

2.10 BrightPass technique

In this technique, the SE generates a sequence of 0 and 1 called Lie Overhead. From this sequence, a series of circles of different brightness will be displayed on the Smartphone screen in the area reserved for entering the PIN code. A low-brightness circle (corresponding to the value 0) tells the user to type a random digit that is not part of the PIN code, while the high-brightness circle (corresponding to the value 1) indicates to the user to type a real number that is part of the PIN code. This technique is used for fighting the spyware attack that tries to find the PIN code typed by the techniques of screen capture or recording [14]. The BrightPass technique also suffers from the Shoulder-surfing attack and the concealed camera attack. The illustration of a BrightPass authentication session [11] is a great proof of the possibility of

returning the PIN code by an attacker using one of the two attacks.

2.11 Secure credit card protocol

Jensen et al. [15] proposed a secured credit card protocol (Secure CCP). It is an authenticated proxy's credit card protocol based on pre-computed hashes, indexing, and xor operation. We find in this solution that the protocol is vulnerable against the theft or the lost of the credit card because the owner of the credit card is not authenticated with a Personal Identifier Number (PIN). In case where a password is used with their protocol, the solution can be vulnerable against registration camera, Spyware, shoulder-surfing and force brute attacks.

2.12 Jie et al.'s protocol

Jie et al. [16] proposed a protocol based on NFC to guarantee the privacy protection security. They use the technique of chebyshev-map and certificateless public key cryptography. For registration process and verification of the true identity of the user, they use TSM as a trusted third party. When we analyze the protocol, we find that it is complicated and it uses mores messages and computations compared to our protocol.

2.13 Nana et al.'s solution

Nana et al. [17] proposed an authentication scheme based on an embedded fingerprint biometric for Automated Teller Machine. We find this solution vulnerable against the attack that uses a default reader to acquire the fingerprint of the user. Even if the solution also uses a password or a PIN, the attacker can first steal it by Shoulder-Surfing attack or registration camera attack, then he can steal the fingerprint by a default reader and in this way, he can affect an electronic payment.

2.14 Symmetric random function generator (SRFG)

Saha and Geetha [18] proposed a function generator that creates an output string symmetric balanced in the number of 1s and 0s and irrespective of the input string. This function considers the hamming distance of output string and it outputs a combined function consisted of basic Boolean functions. This function can be used in hash algorithms, stream ciphers and round function module of block ciphers to be more robust. The result of this function is obtained as following: A random selection of two input variables is established from 'N' variables. Then we apply a random logic gate from four: AND, OR, NOT and XOR on the two variables. The result is the term1. Random selections on a list of 'N' variables generate result variables which are combined with a random of four logic gates: AND, OR, NOT and XOR for creating the first term. For generating the second term, a variable is obtained from the 'N' variables and combined with the first term using a random of four logic gates. This operation is repeating until generating the term number 'L'. The result of the last term is the result of the function generator. The authors prove that the function generator can be used for any cryptography algorithm and cannot be traced back due its randomness. Their proposed work can be used in MD5 or SHA series, block cipher round functions, stream ciphers and cryptography function modules. They prove by their experimentation that the functions generated by the proposed generator provide a good non-linearity, resiliency and balanced effect.

2.15 Healthcare framework dealing with electronic medical records (EMRs)

Saha et al. [19] proposed a system model that is used to process online electronic medical records while preserving their confidentiality. The system collects patient data (DME) across peripheries, terminals, and devices and submits them to the Identity Manager (IM) by performing a secure cryptography exchange. The IM then maps the details of the patient with the pseudo identity and stores them in a Cloud Server. Authors propose Data Aggregators which receives the DME in pseudo-identity through the IM, the information of the Key Center and the information from the Query Handler. After acquiring this information, the Data Aggregators applies a dotted check before sending the data to the public Cloud.

3. SECURE PAYMENT SYSTEM

Our solution is composed of a cloud password, a protocol, a hash function and a test intrusion. They are used in the following hardware system. The following notations are used in our solution:
Natm: Random challenge generated by the ATM.
ATM: Automated Teller Machine
BcN: Bank card Number.
Icvv: unpredictable value freshly generated for every solicitation response, and is subsequently used by the Server bank to validate the transaction.
Id: Identifier of the secure element
CCode: Password
Amount: amount withdrawer.
H: Hash function
||: Concatenation
⊕: xor (exclusive or)

3.1 The hardware system

The hardware system is composed of: A Smartphone (SP), an Automated Teller Machine (ATM) and a Server (S) (see Figure 1).

Figure 1. NFC payment system

3.1.1 Smartphone

The proposed Smartphone has a non-centric SIM architecture. The secure element is implemented with a form of a movable secure support that has a type of memory card (Secure Memory Card: SMC) [20] (Figure 2). The Smartphone is equipped with NFC and is characterized by a movable Secure Element that stores its identifier (Id), the protocols and

the NFC applications (e.g. Payment application), etc. The architecture chosen for the secure element give an advantage to the user which can change the secure element in case of change of country (travel for example) and another improvement in case of stealing of the Smartphone, the operator can disable it.

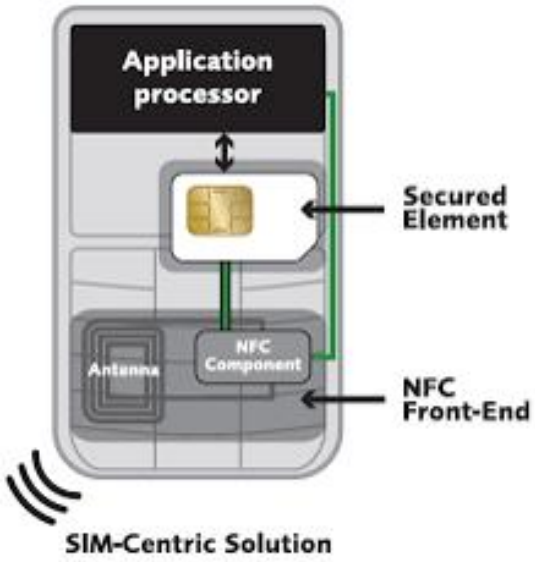

Figure 2. NFC mobile with SMC architecture

3.1.2 Automated Teller Machine (ATM)

It has an ability to read a bank card embedded in a Smartphone. It is capable of generating a nonce. The ATM generates a random challenge and communicates it to the Smartphone. It communicates with the NFC Smartphone and the server. In our system, the ATM has the advantage to generate nonce which are used in encryption to protect against the replay attack.

3.1.3 Server

The Server is characterized by:

- A database containing a set of records. Each record contains the identifier (Id) of the secure element integrated on the Smartphone, the password, the result of application of hash function to the concatenation between the Id and the password (represents the index of research), the access mode state, and the reference to the database containing the list of credit cards or bank cards information. Each credit card or bank card is specified by its number, the bank name, the user account number, its amount, etc. The information is saved during the registration phase by the transmitter (the operator that transmits the details of the secure element). It is updated in the case when the owner changes his Smartphone.
- The server activates the access mode state in the corresponding record of the user when the user sends to the server, its password concatenated with the secure element identifier and signed with the hash function. The server disables the access mode state just after the end of the payment operation.
- The cache memory that represents an advantage to accelerate the access to the record of user who want effect an ATM operation.
- An application that permits the user to send his password in cloud away from ATM which can be targeted by attacks.

The Figure 3 presents the architecture of our system resumed in three phases.

(1) User in his car (for example) sends the encrypted password (message S) to the server

(2) Activation of the corresponding record in the database of the server

(3) Authentication between the Smartphone and the ATM using the protocol

(4) Encrypted data transmission between ATM and the server using the protocol

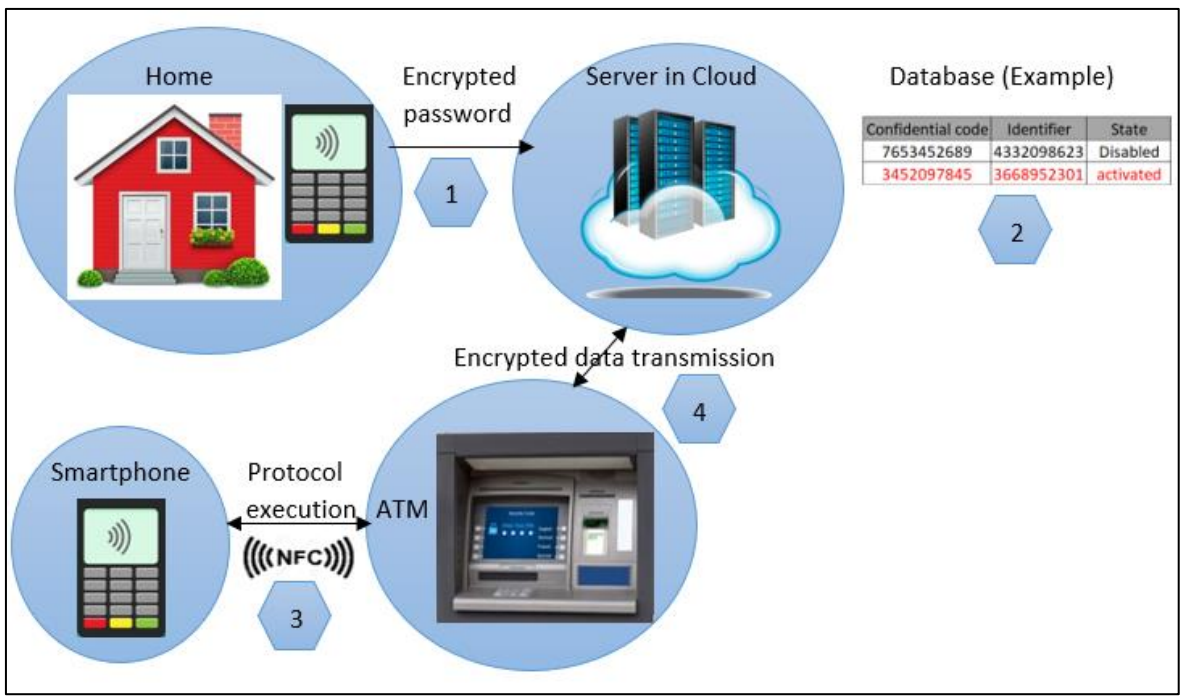

Figure 3. System architecture

3.2 Objects

The ATM communicates with the Smartphone by a radio frequency communication and by an online communication with the Server. The last communication is considered secure by TLS protocol (Transport Layer Security). Our system offers the acquisition of a password with the NFC Smartphone by an operator application to authenticate the owner of the Smartphone. It concatenates the secure element identifier with the password, encrypts the result with a proposed hash function and sends the encrypted message on the cloud. This message is considered secure by using the TLS protocol because our principal goal is to secure the NFC payment between the ATM and the Smartphone. The operator server that has a database indexed on the hash function value of the password concatenated with the secure element identifier looks for the sent value. If it is found, it activates the access mode state in the founded record (the record corresponds to the secure element identifier and its owner). Only when this state is activated, the user is authorized to effect the NFC payment with the ATM. The proposed solution secures the communication between the NFC Smartphone and the ATM by a simple secured proposed protocol which secures messages transmitted during the NFC payment. We believe that our solution is efficient in computation cost, communication cost, storage space and it is safe by checking the following safety features:

- Confidentiality: the secure element identifier (its MAC address) and the user password are not sent in clear over the radio frequency interface or on the cloud but encrypted in a hash function.
- Authentication of the user: the user is authenticated by the secure element identifier encrypted with the password.
- Authentication of the Smartphone: in our solution, the Server is able to authenticate the true Smartphone. This is proved when we present the protocol.
- Authentication of the ATM: The Smartphone must ensure that it communicates with the real ATM. This is proved when we present the protocol.
- Data Integrity: in our solution, the security of the secure element identifier and the password are not infected by the loss of messages, the lack of energy and the failure of connection.

After checking these properties, the protocol provides the payment transactions. To ensure the payment processing system, the protocol prohibits the payment execution when a security property is not verified (the case of attack). In the opposite case, the payment processing system is authorized.

3.3 The protocol

The proposed protocol is called Cpass (Cloud Pass). It is inspired from the protocol named secure CC (Current Card) protocol used for the point of sale [15]. Our protocol is adapted to the electronic payment with ATM using Smartphone. It is characterized by authenticating the user with the password, by its hash function and by protecting the NFC payment against eleven attacks especially the theft of the Smartphone attack. On the other hand, the server runs an intrusion test and posts the amount on the screen of the Smartphone and visualizes an SMS confirmation on the Smartphone in the case of changing the amount by a new further attack (the worst case). Our protocol uses the secure element identifier integrated in the Smartphone. It uses for cipher: a hash function, a challenge number, the concatenation and xor (exclusive or) operators.

The protocol must be embedded in the secure element of the Smartphone. An advantage of our solution is that allows the user selecting a credit card or a bank card from a list of cards embedded in the Smartphone. The phases of the solution can be described as follows:

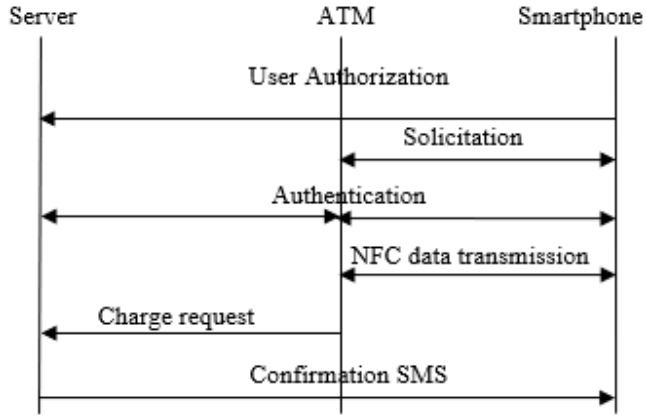

Figure 4. Secured Smartphone protocol for NFC operation with ATM

3.3.1 Registration phase

For the NFC Smartphone, we propose the use of the non centric SIM architecture where the secure element is a secure memory card (SMC) and it is characterized by:

- It offers a high level of security;
- It is conforming to EMV, GlobalPlatform, ISO/IEC 7816, javacard;
- It has an important capacity of memory;
- It is movable since it can be placed with its NFC applications and its secret keys in a new Smartphone.

At the local operator, the following information is saved in the database of the Server:

- The password;
- The Id: the UUID (Universally Unique IDentifier) used to identify the secure element;
- The phone number;
- The access mode state (disabled);
- The Bank card number: it references the name bank, the account number, the amount, the transactions' histories, etc.

The messages exchanged during the protocol process are shown in Figure 4.

3.3.2 Authentication and confidentiality phase

This part includes the following steps:

Step 0: User authorization

This step is important before each operation at the ATM. It informs the server that the owner of a managed account will perform a transaction at an ATM in the coming hours.

- Before each operation, the user situated in a secure place far from the ATM (for example in his car, few minutes before acceding to the ATM), uses a Smartphone application provided by the operator to enter his confidential secret password.
- This application creates a value S = H (Id ‖ CCode), which represents the index of the user in the operator server database.
- The value S is sent to the server which looks for it in its database. If it is found, the server activates the state of the

access mode field in the corresponding record to authorize the operation. Otherwise, no operation is allowed at the ATM.

Initially, the state of the access mode is disabled, and it will also be automatically disabled by the server immediately after the end of the operation. In order to protect against the spyware that may exist in the Smartphone and that can steal the password, the BrightPass technique can be used by the secure element of the Smartphone.

Step 1: Solicitation

- The ATM solicits the Smartphone for its information. It sends a message to the Smartphone identifying the bank card type.
- The Smartphone sends a message to the ATM identifying the bank card type embedded in the Smartphone and selected by the user.

Step 2: Authentication

- The ATM generates a random challenge Natm and then, sends it with a request to the Smartphone.
- The Smartphone calculates A= H(Id ⊕ Icvv ‖ Natm) and sends to the ATM the message A, the Icvv and the bank name. The Icvv and the bank name are saved on the bank card embedded on the Smartphone and selected by the user.
- The ATM returns the received message A, the Icvv and the Natm to the Server.
- The Server looks in the set of records, which have the activated access mode state and situated in a cache memory, for a record with an identifier secure element Idi such as: Ai=H(Idi⊕Icvv ‖ Natm) is equal to the message A. To win the time of research, the server uses a cache memory. The cache memory contains only the records of the users who want to effect NFC payment in the current day. If Idi exists, the Smartphone is considered legitimate and it is authenticated. Otherwise, it is illegitimate and the protocol is interrupted.
- In the first case (the Smartphone is legitimate), the Server calculates the message Bi and sends it to the ATM. Bi= H(Idi ‖ Icvv ‖ Natm)
- The ATM sends the message Bi to the Smartphone.
- The Smartphone calculates B=H(Id ‖ Icvv ‖ Natm) and compares it with the message Bi. If they are equal, the ATM is considered authenticated. Otherwise, the protocol is interrupted.

Step 3: NFC data transmission

- The Smartphone responds to the solicitation by sending back the following information to the ATM:
- The bank card number (The card selected by the user on the Smartphone).
- The bank card's expiration date.

Step 4: Charge request

- The ATM issues a charge request to the Server. This request is composed of:
- The bank card number.
- The bank card's expiration date.
- The amount.

Step 5: SMS Confirmation

- The transaction is executed. The server sends an SMS confirmation (considered secure by TLS) indicating the details of the transaction (the user account number, the removed amount, the sold, the date and the time of the transaction, etc.) to the Smartphone. The access mode state is disabled.

3.3.3 Metrics

The transfer rate between a Smartphone and an ATM in an NFC payment communication is: Bit Rate 1 = 424Kbit / s [21].

Since the ATM used in the NFC payment must be connected to a modem with a dedicated 4G chip, the transfer rate between a server and an ATM is of the order of: Bit Rate 2 = 100 Mbps downlink and Bit Rate 3 = 50 Mbps in uplink [22].

The size of the random number Natm can be of 128 bits [18]: the 128 bits concatenated with the 32 bits of Icvv gives 160 bits that can be linked with the Id which is 160 bits using the XOR operator to form the messages A or Ai.

The size of the variable Icvv is: 32 bits [15].

The size of the secure element identifier (Id) is: 20 characters = 20 bytes = 160 bits [23].

The size of the bank key is: 20 characters = 160 bits.

The size of the bank name is: 30 characters = 240 bits. The biggest message to encrypt with our hash function is: Id || Icvv || Natm whith a size equal to: 160 + 32 + 128 = 320 bits. The size of the different messages and variables are presented in Table 1. The notations and measures used in this table will be used in the calculation of the response time including the execution time of the hash function.

Table 1. Variables and messages size

Message or variable	Specification	Size (bits)
Natm	Random number	128
Message1	Natm	128
Id	Secure element Identifier	160
Bankey	Bank key	160
Bankname	Bank name	240
Icvv	Unpredictable value freshly generated for every solicitation response	32
A ou Ai	Id⊕Icvv‖Natm	160
Message2	A+Icvv+Bankname	432
B ou Bi	Id‖Icvv‖Natm	320
Message3	Bi	320

3.3.4 Proposed Hash function

Our proposed hash function is used by our protocol to generate the encrypted messages A and B in the Smartphone and to generate the encrypted messages Ai and Bi in the server (see Figure 5).

To present our hash function, we consider the following steps:

- We define our hash function H.
- We identify four properties P1, P2, P3 and P4 that the proposed hash function must satisfy.
- We prove that our hash function satisfies the properties P1, P2, P3 and P4.

Definition of the hash function. Our hash function is used to encrypt a message containing the secret information (identifier of the secure element or password) that is linked to other information (Icvv, Natm) using the Xor or the concatenation operator. This message, which is of certain size generally greater than 160 bits, is encrypted by this hash function using the bank key that is of 160 bits of size, as follows:

The function breaks the message into parts of size equal to the size of the bank key.

The function crypts each part. The encryption result of each part is a message of 160 bits generated as follows:

If the bit of the part to be encrypted is equal to 1, the bit generated is the result of the xor between this bit and the corresponding bit (of the same index) in the bank key. In the opposite case (the bit to be encrypted is equal to 0), the generated bit is the result of the xor between this bit and the negation of the corresponding bit in the bank key. The encryption result of each part will be a message of 160 bits.

Finally, the function performs the binary sum of the resulting parts to generate a final encrypted message on the maximum of 161 bits (the worse case) knowing that the max number of parts in our protocol is two because the biggest message to encrypt is: ID || Icvv || Natm that is of 320 bits = 160 * 2 bits.

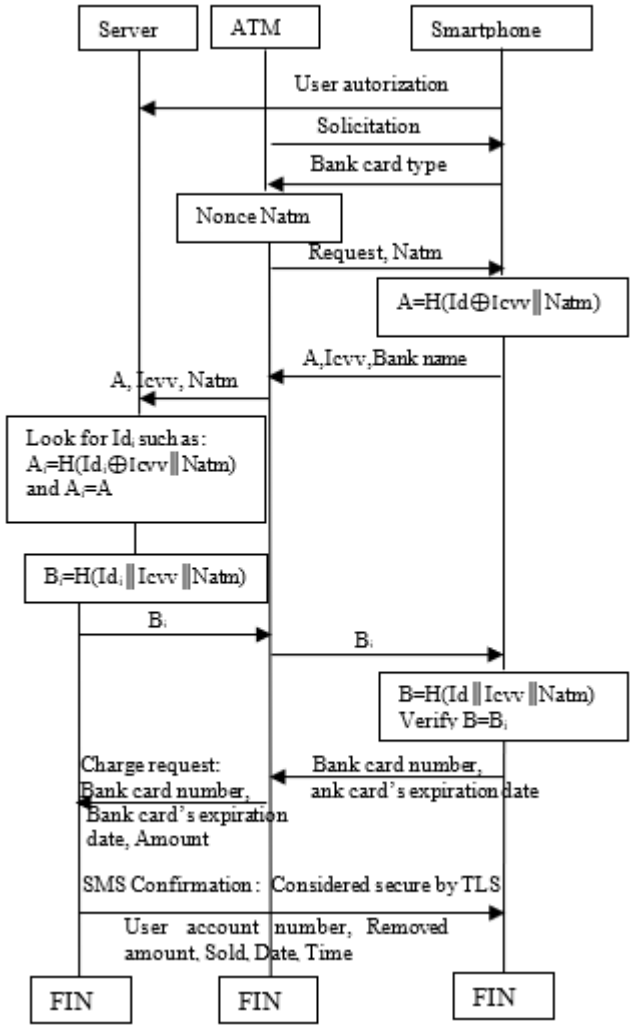

Figure 5. The proposed protocol

The implementation of our proposed hash function is described as follows:

```
H (Message, Bankey, Result)
      ⊳ Message, Bankey, Result and ht are bit sequences
     ⊳ ht is an intermediate variable
     Init (result) ⊳empty initialization
     j ← 1
     while j <= Message.size() do
              i ← 1
             Init (ht)
             while i <= bankey.size() and  j <=
message.size()
                       if (message[j] = 1)
                                ht[i] ← message[j] xor
bankey[i]
                       else
                                ht[i] ← messag[j]  xor  not
(bankey[i])

             i ← i + 1
             j ← j + 1

          Result.insert (ht)
```

Properties of the hash function H. To ensure a secure encryption, the H function must satisfy the following properties:

P1: H is irreversible. Given 'mh' a message encrypted by the function 'H'. It is difficult to find the message 'm' such that: mh = H (m) [24].

P2: Given x. It is impossible to find a pair x, y such that H (x) = H (y) [25]

P3: If Icvv is random, then H (m) is random (m: the message to be encrypted by H) [15].

P4: If we know H (m) for m = Id || Icvv || Natm, and we only know Natm and Natm0 such as Natm <> Natm0, it is impossible to deduce H (m ') for m' = Id || Icvv || Natm0 without knowing Id and Icvv [15].

Properties verification. The value of Icvv is chosen pseudo-random because the generation method not publicly known and depends on the bank that can use its own arbitrary generating function [15].

P1: Given 'mh' a message encrypted by the function 'H'. The question is: can we deduce the original message 'm'? The answer is no for the following reasons:

When an attacker processes a bit of the message H (m) generated after an addition of two binary numbers, and each bit of a number is the result of the Xor between a bit of the message and either the corresponding bit of the bank key, or the negation of this last, it cannot have any information on the bit of the message m. So, it is impossible for this attacker to reconstruct the message m.

P2: For this property, we demonstrate that it is impossible to find a pair x, y such that H (x) = H (y).

Let Table 2 represents the truth table of the Xor.

Table 2. Xor results

Bit (message)	0	1	0	1
Bit (Bankkey)	0	0	1	1
Xor	0	1	1	0

Bit (message)	0	1	0	1
Bit Negation (Bankkey)	1	1	0	0
Xor	1	0	0	1

From this table, we notice that each modification of the message bit modifies the xor result. This implies obtaining a new binary sequence and therefore a new number. It means a new footprint and therefore, a new encrypted message.

P3: If Icvv is random, and we already have Natm a random number, then the message m = Id || Icvv || Natm or the message m = Id⊕Icvv || Natm is random and therefore H (m) will be random too.

P4: If we know H (m) for m = Id || Icvv || Natm, and we know Natm, we cannot deduce the message m according to the proof presented in P1. In this way, the message m remains

unknown. Hence, it is impossible to know the information Id and Icvv. This implies that we cannot calculate m'= Id || Icvv || Natm0 even with the knowledge of Natm0 and therefore, impossible to know H (m').

As the properties p1, p2, p3 and p4 are verified, we can consider that our function with its simple code is a secure hash function.

4. SECURITY ANALYSIS

Two types of verification are established:

4.1 Security control by analysis

The analysis of the proposed solution shows that it is impassable by the following attacks: Brute force, Card cloning, Registration camera, Spyware, Shoulder-surfing, Eavesdropping, Skimming, denial of service, relay, replay, Man-In-The-Middle and theft of Smartphone attack.

4.1.1 Brute force attack

This attack involves installing malware at the ATM or on the Smartphone that tries all possible combinations of characters or numbers to find the password received by the ATM or the Smartphone. In the ATM, our solution fights this attack because there is no password transfer to the ATM through the radio frequency interface. In the Smartphone, our method is protected against this attack by BrightPass technique and after a certain number of failed attempts (two or three times), the cloud server blocks the operator application used to enter password. Although the communication between the Smartphone and the server is considered safe, our object is to secure the communication between the ATM and the Smartphone and we have proved that it is secured.

4.1.2 Card cloning attack

This attack involves the use of cloning devices at the ATM which are undetectable by the user to secretly record the bank card data embedded in the Smartphone to create a cloned copy for later use. In our system, using a copy of the bank card or even a copy of the Smartphone is not sufficient to validate a transaction with ATM because this requires the activation of the access mode state in the server database by entering correctly the password, and since the attacker does not have the password and only has a cloned copy of the bank card or the Smartphone, he cannot achieve a transaction with ATM. On the other hand, for the attacker who has succeeded in restoring the information related to the bank card, he can't decrypt the secure element identifier since it is protected and it is necessary for the execution of the protocol.

4.1.3 Registration camera attack

In such attack, the attacker uses a camera near the ATM to register the password entered by the user using the Smartphone or the ATM. Since the solution is based on entering a password as far from the ATM, and this password is sent to the cloud server and not to the ATM, this attack will have no effect.

4.1.4 Spyware attack

In this case, the attacker uses malware installed in the ATM as the Keylogger to steal the user's credentials and specifically, the password. For the proposed solution, this attack is inefficient because on one side, the attacker can't do anything with a secure element identifier that is encrypted with a random number (i.e. Natm) when he retrieves it using a spyware. On the other hand, he can't steal the password that does not pass through the ATM.

The password entered by the Smartphone is protected against a Spyware installed in a Smartphone using the BrightPass technique.

4.1.5 Shoulder-surfing attack

By this attack, the attacker tries to get ready for the ATM and waits for the arrival of a user to see the entered password and memorize it. As the solution uses a password entered in a secure location away from the ATM, this attack will have no effect.

4.1.6 Auxiliary channel attack

Auxiliary channel attack is used by a spyware to capture the user keystrokes. Since the secure element uses protection ways to secure sensitive data inputted by the user, the spyware uses the shared resources between the mobile operating system and the secure element to make this type of attack succeed [11]. The use of the BrightPass technique for entering the password by Smartphone combats this type of attack which consequently protects the solution. If the spyware is installed on the ATM, it has no effect because there is no password to enter by the ATM.

4.1.7 Eavesdropping attack

The secure element identifier Id transmitted between the Smartphone and the ATM is encrypted in a hash function that uses a random value (Natm) to guarantee that no sensitive information is leaked.

The eavesdropper can't receive in clear the Id or the password for replaying them in a next operation with ATM especially when random numbers are used in encryption.

4.1.8 Skimming attack

The skimmer finds the message using the Id encrypted with a random challenge Natm. When the skimmer tries to perform a purchase using this message, he finds another random challenge generated by the ATM. On the other hand, the skimmer can't receive the password that doesn't pass by the NFC communication between the ATM and the Smartphone.

4.1.9 Denial of Service (DoS) attack

The solution doesn't require synchronization. The loss of messages, the lack of energy and the failure of connection does not affect the safety of the secure element identifier or the password. This means that the solution can withstand the denial of service attack. On the other hand, with a short distance between the Smartphone and the ATM, it is difficult to realize the network flooding or the connection disruption.

4.1.10 Relay attack

It is impossible for the mole attacker to activate the Smartphone of the victim without his consent because the Smartphone must be open and the NFC option must be activated by the user. We suppose that the mole attacker has received the bank card information and the Id from the Smartphone and has transmitted them to the proxy. The proxy that is near to the ATM, can't effect an operation with the ATM because it requires the user password that it didn't recovered by the proxy and only the user is its owner. On the other hand, in a relay attack, the mole sends a random

challenge Natm1 to the Smartphone. The mole receives A1=H(Id⊕Icvv‖ Natm1) from the Smartphone and sends the message A1 to the proxy. The proxy tries to perform a transaction with the ATM that generates a random challenge Natm2 and sends it to the proxy. The proxy can't construct the message A2= H(Id⊕Icvv‖Natm2) and can't use the message A1 to communicate with the ATM.

4.1.11 Theft of Smartphone attack

After stealing the Smartphone of the victim, the attacker can't access to the ATM because he doesn't have the password.

- If the stealer directly approached the Smartphone near the ATM, the protocol will be interrupted in step 2 (authentication) when the Server don't find the appropriate record in the set of records which have an activate state of access mode. So, the access mode state in database stays disabled and the NFC transaction can't be established.
- If the stealer uses the operator application on the cloud and tries to enter a password (CCode0) that is very possible different from the CCode, the message M0=H(Id‖CCode0) will be transmitted to the server. The server after looking in the database, it doesn't activate the access mode state since there is no value equal to M0. In this case, the protocol will be interrupted by the server in step 2 (authentication). On the other hand, when the stealer enters a password CCode0 that is different from the CCode, the SE detects the difference and don't send the message that encrypts the Id with the password to the server. So, the access mode state stays disabled and it is impossible to effect the NFC transaction.

After this analysis based on logic, we have proven the security of our method against the 11 attacks mentioned.

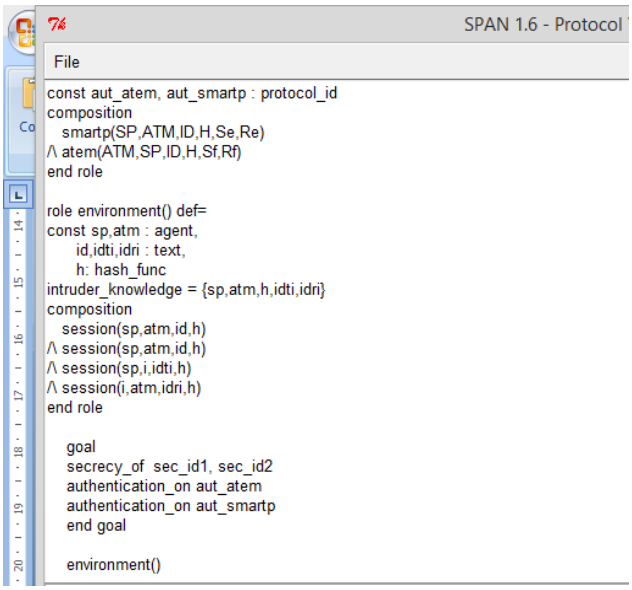

Figure 6. Protocol specification

4.2 Automatic security control

This section presents the validation results of the solution using the AVISPA tool that enables automatic verification of the protocol security. As shown in Figure 6, the protocol specification could detect Replay and Man-In-The-Middle attacks if they exist.

After the verification of the protocol by the OFMC tool of AVISPA (Figure 6), the result is shown in Figure 7.

After the verification of the protocol by the ATSE tool of AVISPA (Figure 6), the result is shown in Figure 8.

The result obtained in Figure 7 and Figure 8 means that there is no replay or Man-In-The- Middle attacks.

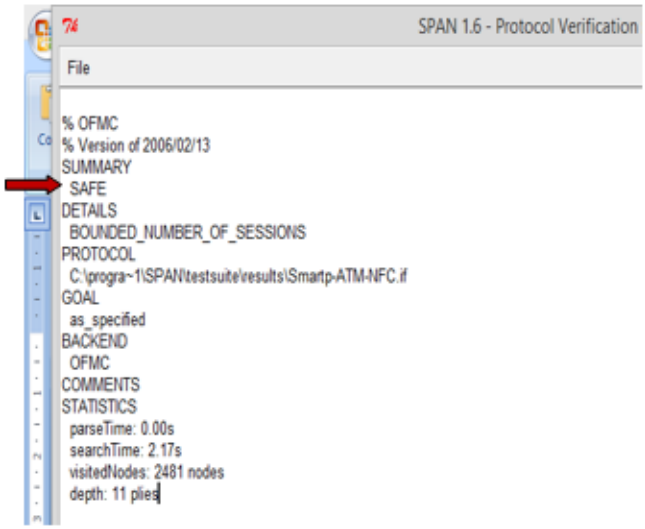

Figure 7. Protocol Result generated with OFMC tool

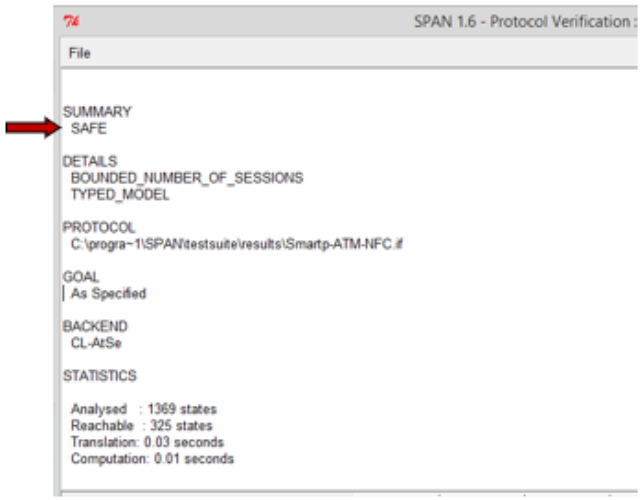

Figure 8. Protocol Result generated with ATSE tool

5. PERFORMANCE COMPARISON

In this section, we compare our solution with some well known existing solutions.

5.1 Security analysis

The aim of this analysis is to check the attacks which could cross the methods or the protocols. Some results are obtained by the AVISPA tool and the others are obtained by the analysis of the attack towards the method or the protocol.

The FakePIN and PassWindow methods use simple passwords, despite their resistance against the Shoulder-Surfing attack and the concealed camera attack. To protect these methods against the brute force attack, they must either use long or complicated passwords that require a large number of possible combinations, or they must use in addition to the password, values of motion secured and stored in the SE.

The BrightPass and Cappcha methods are vulnerable to the

Shoulder-Surfing or the concealed camera attack because an attacker or a camera can use a direct observation by watching or filming the direct entry of the password. In the BrightPass technique, an attacker who knows the operating principle, records in his memory the numbers entered in the circles having a high luminosity. If thereafter the attacker can steal or borrow the Smartphone, it can carry out a payment transaction with the ATM following the indications of the secure element.

In the proposed solution, the Shoulder-Surfing or the concealed camera attack is impossible because the user can enter his password on the cloud and in a secure place, for example his home, which means that the theft of Smartphone will have no effect on the transaction payment.

The messages used in our protocol are difficult to be decrypted because they are composed of nonce, Xor operator, concatenation operator and they are signed with a hash function. So, it is difficult to discover the password and the secure element identifier. The security control by analysis and automatic security control in the above section prove that it presents a high level of security.

We present in the following table, a comparison between our solution and other methods used for securing the NFC payment.

The Table 3 presents the security limits of well-known existing methods and protocols. We note by 'V' that the method or the protocol is vulnerable to the attack and by 'R' that it is resistant to the attack.

This table presents the security limits of well-known existing methods and protocols. We note by 'V' that the method or the protocol is vulnerable to the attack and by 'R' that it is resistant to the attack. Table 3 shows the vulnerability and the resistance to the mentioned attacks concerning the following methods: GoogW and EmvCC methods [6], FakeP, PassW, Capp, and Bpass [11]. The rest of information in this table is obtained by our analysis. According to this table, we show that the different methods are vulnerable against the attacks except the proposed solution (Cpass) that is the safer and prevents all the attacks listed.

We can present now the strong points of security of the proposed solution as follows:

- The use of a password that is typed in a secure side by the user, encrypted with a secure element identifier and sent on the cloud. This means that the user types his password far from the attackers.
- The secure element identifier and the password are protected by encryption using the xor and the concatenation operators and our proposed hash function. This means that the confidentiality is verified.
- The user, the Smartphone and the ATM are authenticated.
- The data integrity is verified.
- The type of the used secure element offers a high level of security.
- In the worst case, if a new further attack is discovered and breaks our protocol, a confirmation SMS is sent to the Smartphone of the owner and that represents an alarm message, indicates that his sold is changed. This means that the protocol uses a technique of intrusion test.
- The protocol does not use any biometric modality. Thus, the user is not exposed to the related attacks.

Table 3. Security limits of methods

Attack / Method	GoogW	EmvCC	Brfid	FakeP	PassW	Capp	Bpass	Cpass
Theft of card or Smartphone	V	/	/	R	R	R	R	R
Man-In-The-Middle	/	V	R	/	/	/	/	R
Replay	/	/	R	/	/	/	/	R
Monitoring trace	/	/	R	/	/	/	/	R
Theft of user fingerprint and card	/	/	V	/	/	/	/	R
Shoulder-surfing	V	V	/	R	R	/	/	R
Multiple record	V	V	/	V	V	R	R	R
Auxiliary canal	V	V	/	R	R	R	R	R
Simple record	V	V	/	R	R	R	R	R
Force brut	/	/	/	V	V	R	R	R
Spyware	V	V	/	R	R	R	R	R
Screen record	V	V	/	R	R	R	R	R
Card cloning	/	/	/	R	R	R	R	R
Registration camera	V	V	/	R	R	/	/	R
Eavesdropping	/	/	/	/	/	/	/	R
Skimming	/	/	/	/	/	/	/	R
Relay	/	/	/	/	/	/	/	R
Compromised ATM	/	/	/	/	/	/	/	R
Denial of Serves	/	/	/	/	/	/	/	R
Registration camera and theft of Smartphone	V	V	/	/	/	V	V	R
Shoulder-surfing and theft of Smartphone	V	V	/	/	/	V	V	R

Table 4. Runtime of the hash function

Message	Runtime (ms)										
	Trial1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trial 8	Trial 9	Trial 10	Average
A	1	2	1	1	1	2	2	1	1	2	1.4
Ai	Same as message A since it has the same size										1.4
Bi	Same as message B since it has the same size										1.9
B	2	1	3	3	2	1	2	2	2	1	1.9

Table 5. Transfer time of messages

Message	Size (bits)	Communicating parts		Transfert time (ms)
		Smartphone – ATM Bit rate 1 = 424kbit/s	Server – ATM Bit Rate2 = 100Mbps (downlink) Bit Rate3 = 50Mbps (uplink)	
Message1	128	*		Tm1=0.30
Message2	432	*		Tm2=1.02
Message2	432		*	Tm3=0.001
Message3	320		*	Tm4=0.0004
Message3	320	*		Tm5=0.75
Total				2.07

5.2 Authentication time

5.2.1 Usability tests of the hash function:

We present a usability test of a java code that represents our hash function. We have implemented the hash function in Java. Table 4 shows the runtime of the hash function on messages A (160 bits) and B (320 bits):

The runtime of the hash function during authentication is: 1.9 * 2 + 1.4 * 2 = 6.6 ms.

Calculation of message transfer time. By analyzing the protocol diagram, it is possible to estimate the authentication time 'At' by the addition of the following times:

Tm1: transmission time of Message 1 from ATM to Smartphone. Tm1 = size of message 1 / bit rate 1 = 128/424 ms = 0.30 ms.

Th1: runtime of the hash function to build the message 'A' in the Smartphone. Th1 = 1.4 ms (Table 4).

Tm2: transmission time of Message 2 from Smartphone to ATM. Tm2 = size of message 2 / bit rate 2 =432/424 ms = 1.02 ms.

Tm3: transmission time of Message 2 from the ATM to the server. Tm3 = size of message 2 / bit rate 3 = 432/50 * 1024*8 ms = 0.001 ms.

Th2: runtime of the hash function to build the message 'Ai' in the server. Th2 = 1.4 ms (Table 4).

Th3: runtime of the hash function to build the message 'Bi' in the server. Th3 = 1.9 ms (Table 4).

Tm4: transmission time of Message 3 from the server to the ATM. Tm4 = size of message 3 / bit rate 2 = 320/100 * 1024*8 ms = 0.0004 ms.

Tm5: transmission time of Message 3 from ATM to Smartphone. Tm5 = size of message 3 / bit rate 1 = 320 / 424ms = 0.75 ms.

Th4: runtime of the hash function to build the message 'B' in the Smartphone. Th4 = 1.9 ms (Table 4).

At = Tm1 + Th1 + Tm2 + Tm3 + Th2 + Th3 + Tm4 + Tm5 + Th4 = 0.30 + 1.4 + 1.02 + 0.001 + 1.4 + 1.9 + 0.0004 + 0.75 + 1.9 = 8.67 ms.

The transfer time of messages during authentication is: 2.07 ms (Table 5).

The authentication time is: 6.6 + 2.07 = 8.67 ms.

Calculating the complexity. In the hash function, we have an extern loop that starts from the first bit of the message and ends at its last bit. In this loop we have another loop and the complexity could be equal to O (n).

5.3 Performance analysis

The object of this analysis is to show that the proposed solution is economic and prove that it has the best authenticate time and it is efficient.

We present in Table 6 and Figure 9 the authentication times in second of some well-known methods compared to that of the proposal. Table 6 shows the authentication times of the methods FakeP, PassW, Capp and Bpass [11].

The Table 6 and Figure 9 show that the estimated authentication time for our protocol is the best and the smallest. In fact, the protocol does not require any password entry in the front of the ATM which saves time.

Table 6. Authentication time comparison

Method	Authentication time (s)
FakeP	10.90
PassW	18.12
Capp	4.12
Bpass	8.20
Cpass	0.00867

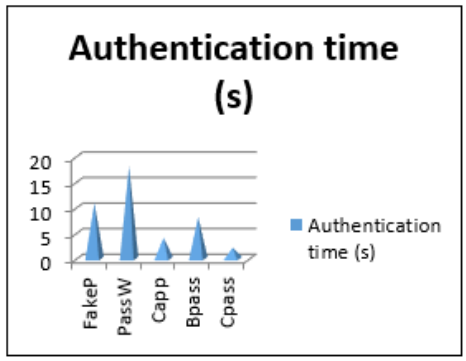

Figure 9. Authentication time comparison

We calculate now, for our protocol, the number of hash function operations carried out in the Smartphone to evaluate the calculation cost, the number of variables stored in the Smartphone to evaluate the storage space and the number of transmitted variables to assess the communication cost.

5.3.1 The calculation cost

The calculation cost: The proposed protocol requires 03 operations to calculate the hash function in the Smartphone. This means that it isn't expensive in computation time compared to the Smartphone capacity and the memory capacity of the secure element. The calculation cost: The proposed protocol requires 03 operations to calculate the hash function in the Smartphone. This means that it isn't expensive in computation time compared to the Smartphone capacity and the memory capacity of the secure element.

5.3.2 The storage space

The protocol requires 04 L (L: bit size for each variable) of memory on the secure element to store the secure element identifier (Id), the Icvv, the bank card number and the bank

card's expiration date. Also, there is no storage of biometric modalities which means a low storage space on the secure memory.

5.3.3 The communication cost

The proposed protocol is efficient in communication cost because there are few exchanged messages between the Smartphone and the ATM during the electronic payment and, the user authentication is separated from the NFC communication.

The protocol is cost effective considering the following points:

- The protocol is secure according to the presented verification.
- At the Smartphone level, there is little calculations for the hash function (only two) to calculate the message 'A' and the message 'B' with a time equal to: Th1 + Th4 = 1.4 + 1.9 = 3.3 ms. It is therefore economical in calculation cost.
- The number of messages transmitted between the Smartphone and the ATM is three: message1, message2 and message3, with a transfer time equal to: Tm1 + Tm2 + Tm5 = 0.30 + 1.02 + 0.75 = 2.07 ms. It is therefore economical in communication cost.

The protocol is compared to the Brfid protocol by considering the number of xor, concatenations and the used hash function. Table 7 and Figure 10 present the obtained results. We show that Cpass is the best compared to Brfid because it presents the less number of ⊕, ‖ operators and hash functions.

Table 7. Cpass vs Brfid

	Brfid	Cpass
Number of ⊕	5	2
Number of ‖	5	4
Number of hash functions	5	4

The protocol is compared to the Brfid protocol by considering the number of xor, concatenations and the used hash function. Table 7 and Figure 10 present the obtained results. We show that Cpass is the best compared to Brfid because it presents the less number of ⊕, ‖ operators and hash functions.

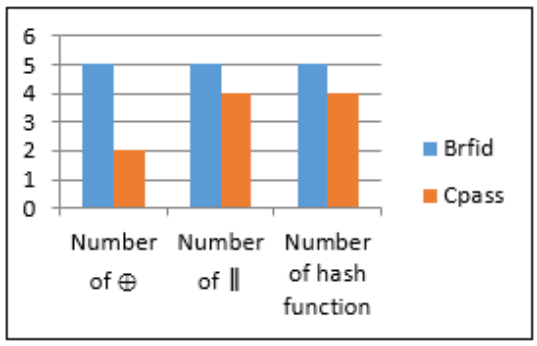

Figure 10. Cpass vs Brfid

On the other hand, by comparing Cpass with the secure credit card protocol (secure CCP), we can present some characteristics of the proposal (see Table 8).

When we compare our protocol with the Jie et al.'s protocol, we find that the last uses more messages and more computation operations. The comparison is shown in Table 9.

Table 8. Cpass vs secure CCP

Secure CCP	Cpass
Uses only credit card	Uses Smartphone. User can select a card from a list
Vulnerable to the theft of the credit card	Resistant to the theft of the Smartphone
No password	Password
No confirmation message after payment	Confirmation message after payment indicates transaction details and can be used as an intrusion test
No secure element	Secure element

Table 9. Cpass vs Jie et al.'s protocol

Parameter / Protocol	Jie et al.'s protocol	Cpass
Number of sent messages	04	01
Number of received messages	03	02
Number of computation operations	07	02
Number of verification operations	02	01
Number of generated random values	04	00

In Jie et al.'s protocol [16], each communicating device sends 04 messages: For example, the device 'A' sends the following messages:

- The message containing the password and the identity of the device: $\{q_A, ID_A\}$
- The message m1 containing the concatenation of 'h' result, 'A' identity and 'B' identity:
- $m_1 = h(ID_A \| ID_B \| S_A) \| ID_A \| ID_B$
- The message $\{Q'_A \| N_A \| Q''_A\}$
- The message $m_3=\{MacTag_A\}$

The received messages for the device 'A' are as follow:

- R_B
- The message $m_2=Enc(Q_A,R_B) \| ID_{TSM} \| S_{TSM}$
- The message $m_4 =\{MacTag_B\}$

The device 'A' effects the following computations:

- $m_1 = h(ID_A \| ID_B \| S_A) \| ID_A \| ID_B$
- $Enc (Q_A,R_B)$
- $Q'_A=r_AQ_A$, $Q''_A=r_Ad_AQ_S+Q_A$
- $P_A = r_Ad_AQ''_BTS_A(R_B)$
- 'Z' value from 'P'
- $SSK=KDF(N_A,N_B,ID_A,ID_B,Z)$
- $MagTag_A = f(SSK,ID_A,ID_B,Q_A,Q_B)$

The verifications established by the device 'A' are as follow:

- Verify TSM signature (S_{TSM})
- Verify the message m_4 validation

At the end, the random values generated by the device 'A' are:

- The password q_A
- The identity ID_A
- The nonce N_A
- The random number r_A

However, for the proposed protocol, the sent messages for the Smartphone are as follow:

- A, Icvv, Bank name

The received messages are as follow:

- Request, Natm
- Bi = H (Idi ‖ Icvv ‖ Natm)

The computation operations realized by the Smartphone are as follow:

- A = H(Id⊕Icvv ‖ Natm)
- B = H(Id ‖ Icvv ‖ Natm)

The verification operations are as follow:

- Verify B = Bi

We compare now between our solution and the Nana et al.'s solution. The comparison shown in Table 10 indicates that our solution is safer and economic.

Table 10. Cpass vs Nana et al's solution

Attack / Solution	Nana et al.'s solution	Cpass
Default fingerprint reader	V	R
Registration camera (if password is used)	V	R
Shoulder-Surfing (if password is used)	V	R

Table 11. Difficulties comparison

Method	Difficulties
FakeP	-Memorize two passwords: one is alphanumeric and the other is a direction. -Virtual keyboard that can be modified with each authentication attempt. -The letter to press is the combination of the letter of the password and the direction.
PassW	-Memorize two passwords: PIN code and a preselected icon. -Memorize the location of the preselected icon in an editable grid for each authentication operation and containing other random icons. -Displaying a virtual keyboard and a grid without icons. - Incline the phone to move the grid on the virtual keyboard to enter the PIN digit in the location of the preselected icon. -Hide the lens of the camera. -Store the PIN code.
Capp	-Incline the mobile phone to a specific degree displayed on the screen and hold it in such position for one second to have access to the PIN code. -Store the PIN code.
Bpass	-Display a set of small circles that take the positions of the password numbers. In a circle of low light, the user enters a false number. In high brightness, it enters a real number.
Cpass	-The user only brings his Smartphone near to the ATM because he has taken his time to introduce safely his password so far from the ATM.

Table 12. Difficulties degree comparison

Method	Degree of difficulties
Fakep	4
PassW	6
Capp	2
Bpass	2
Cpass	0.5

Now, we study the difficulty that the older users can find during the authentication operation with an ATM using one of the last four methods (Fake PIN, Passwindow, Cappcha and BrightPass) and we compare it with the proposed solution. Results are shown in Table 11 and 12. By assumption, we can evaluate each difficulty by one point. For the proposed method, we can evaluate the difficulty by 0.5 point because the older agent has only to bring his Smartphone in the front of the ATM and don't enter any password.

The results shown in Table 11 and 12 indicate that the proposed method Cpass presents fewer difficulties for older users during their payments using the ATM.

6. CONCLUSIONS

In this paper, we have proposed a technique of cloud pass that uses a password concatenated with the secure element identifier and signed with a proposed hash function. The result message is sent on the cloud to the server to activate the access mode state of the record corresponding to the owner of the secure element. Also, we have proposed an authentication protocol for a system consisting of: a server, an ATM and an NFC Smartphone. The protocol enables or disables the NFC payment. We have verified the proposed protocol by analysis and by the AVISPA tools and we have proved that it is efficient. We have proposed a hash function in order to sign the messages and a test of intrusion to indicate to the user that his account sold is modified. Further, we have compared our solution with some well-known existing security protocols and methods. The comparisons shown that it is the best one. Also, we have proved that it is not expensive because it didn't require additional hardware devices. We have shown that it provided three security features: authentication, data integrity and confidentiality to secure the NFC payment between an ATM and a Smartphone and, it resists against several violations and attacks.

Further, the proposed solution presents some interesting features:

- The use of Smartphone that replaces the bank card or the credit card which can be selected by the user.
- The Server accesses rapidly to the database record of the user wanting to authenticate with the ATM for the NFC payment. In fact, during the authentication attempt, the server localizes only the records having the activate state of access mode (like a cache memory).
- The type of the used secure element is conforming to EMV, Globalplatfrom and Javacard. It has an important capacity of memory and it is movable. So, it could be placed with its NFC applications and its secret keys in a new Smartphone.
- The proposal does not use any sensor which means that it is economic in terms of hardware resources.

As future works, we would consider securing the information flows between the host controller, the NFC controller and the secure element by creating security tunnels. We can propose also:

- The use of sensors capable of acquiring the user's iris in addition to physiological characteristics in real time to prevent attack using iris images.
- The use of sophisticated cameras associated with ATM capable of taking the 3D image of the user face in real time and disabling all the input channels of the image except the channel of the secure element and this during the payment operation.

To implement these solutions without any security problems, we propose:

- The user must maintain the confidentiality of his password.
- The user must not borrow his Smartphone and must quickly declare the loss of his phone.
- These solutions must be implemented by specialized technicians who are able to install the necessary hardware and software and, to control and verify it at specific times.

REFERENCES

[1] Giese, D., Liu, K., Sun, M., Syed, T., Zhang, L. (2019). Security Analysis of Near-Field Communication (NFC) Payments. arXiv preprint arXiv:1904.10623.

[2] Merkus, J. (2018). 'Security evaluation of the NFC contactless payment protocol using Model Based testing' (Master's thesis, Open University Nederland).

[3] Desta, G. (2012). Security for mobile payment transaction. Master of Science Thesis, Stockholm, Sweden. https://pdfs.semanticscholar.org/21c5/2539247ef56a71ec8bb1c9370cdda2b2bf6e.pdf.

[4] Promontory an IBM Company. (2017). Biometric authentication in payments. Considerations for Policymakers. https://app.ranenetwork.com/article/8001000008492/

[5] Pourghomi, P. (2014). Managing near field communication (NFC) payment applications through cloud computing (Doctoral dissertation, Brunel University, School of Information Systems, Computing and Mathematics).

[6] Vincent, A.M. (2012). Contribution to the deployment of mobile services and to the analysis of transaction security (PhD thesis, University of Caen Basse- Normandie).

[7] Thevenon, P.H. (2011). Securing the physical layer of contactless communications of type RFID and NFC (PhD thesis, University of Grenoble).

[8] Lifchitz, R. (2012). Hacking the NFC credit cards for fun and debit. Hackito Ergo Sum Conference.

[9] Chikouche, N., Cherif, F., Benmohammed, M. (2012). An authentication protocol based on combined RFID-biometric system RFID-biometric system. arXiv preprint arXiv:1207.5627.

[10] Kim, S., Yi, H., Yi, J.H. (2014). FakePIN: Dummy key based mobile user authentication scheme. In Ubiquitous Information Technologies and Applications, Springer, Berlin, Heidelberg, 157-164. https://doi.org/10.1007/978-3-642-41671-2_21

[11] Guerar, M., Migliardi, M., Merlo, A., Benmohammed, M., Messabih, B. (2015). A completely automatic public physical test to tell computers and humans apart: A way to enhance authentication schemes in mobile devices. In 2015 International Conference on High Performance Computing & Simulation (HPCS), Amsterdam, Netherlands, pp. 203-210. https://doi.org/10.1109/HPCSim.2015.7237041

[12] Yi, H., Piao, Y., Yi, J.H. (2014). Touch logger resistant mobile authentication scheme using multimodal sensors. In Advances in Computer Science and its Applications, Springer, Berlin, Heidelberg, pp. 19-26. https://doi.org/10.1007/978-3-642-41674-3_4

[13] Guerar, M., Migliardi, M., Merlo, A., Benmohammed, M., Palmieri, F., Castiglione, A. (2016). Using screen brightness to improve security in mobile social network access. IEEE Transactions on Dependable and Secure Computing, 15(4): 621-632. https://doi.org/10.1109/TDSC.2016.2601603

[14] Guerar, M. (2017). Security problems in embedded systems. PhD Thesis, University of Oran-Algeria.

[15] Jensen, O., Gouda, M., Qiu, L. (2016, January). A secure credit card protocol over NFC. In Proceedings of the 17th International Conference on Distributed Computing and Networking, 32: 1-9. https://doi.org/10.1145/2833312.2833319

[16] Ling, J., Wang, Y., Chen, W. (2017). An improved privacy protection security protocol based on NFC. IJ Network Security, 19(1): 39-46. https://doi.org/10.6633/IJNS.201701.19(1).05

[17] Gyamfi, N.K., Mohammed, M.A., Nuamah-Gyambra, K., Katsriku, F., Abdulah, J.D. (2016). Enhancing the security features of automated teller machines (ATMs): A Ghanaian perspective. International Journal of Applied Science and Technology, 6(1).

[18] Otterbein, F., Ohlendorf, T., Margraf, M. (2017). The german eID as an authentication token on android devices. International Journal of Computer Science and Information Security, 14(12). arXiv:1701.04013.

[19] Saha, R., Kumar, G., Rai, M.K., Thomas, R., Lim, S.J. (2019). Privacy ensured *e*-healthcare for fog-enhanced IoT based applications. IEEE Access, 7: 44536-44543. https://doi.org/10.1109/ACCESS.2019.2908664

[20] Otterbein, F., Ohlendorf, T., Margraf, M. (2017). The German eID as an authentication token on android devices. Available: https://arxiv.org/ftp/arxiv/papers/1701/1701.04013.pdf.

[21] Bouazzouni, M.A. (2017). Processus sécurisés de dématérialisation de cartes sans contact. PhD, Réseaux, Télécommunications, Systèmes et Architecture, Institut National Polytechnique de Toulouse.

[22] Patel, S., Shah, V., Kansara, M. (2018). Comparative Study of 2G, 3G and 4G. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 3: 2456-3307.

[23] Jdaida, S. (2016). Analyse de sécurité des applications d'authentification par NFC. Mémoire de maîtrise, École Polytechnique de Montréal. Tiré de. https://publications.polymtl.ca/2149/.

[24] Aste, T., Tasca, P., Di Matteo, T. (2017). Blockchain technologies: The foreseeable impact on society and industry. Computer, 50(9): 18-28. https://doi.org/10.1109/MC.2017.3571064

[25] Zheng, Z., Xie, S., Dai, H.N., Chen, X., Wang, H. (2018). Blockchain challenges and opportunities: A survey. International Journal of Web and Grid Services, 14(4): 352-375.