Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Challenge-response mutual authentication protocol for EMV contactless cards

*Ossama Al-Maliki, Hisham Al-Assam**

School of Computing, The University of Buckingham, UK

ARTICLE INFO

Article history:
Received 25 March 2020
Revised 9 November 2020
Accepted 5 January 2021
Available online 8 January 2021

Keywords:
EMV security
Point of sales
Contactless cards
Payment protocol
Mutual authentication

ABSTRACT

Europay MasterCard and Visa (EMV) is the most popular payment protocol with almost 7.1 billion EMV based credit and debit cards around the world. This payment protocol supports different kinds of payment transactions such as Chip & PIN, Chip & signature, contactless card, and mobile payment transactions. This paper focuses on the EMV contactless card transactions and highlights one of such transactions' vulnerabilities that allows attackers to gain access to most of the EMV card sensitive information using off-the-shelf hardware and software. In the EMV card payment protocol, the EMV card must authenticate itself as a genuine card to the point of Sale (POS) in each transaction while the reverse is not happening. An attacker can take an advantage of such vulnerabilities in the EMV specifications especially in contactless cards due to the wireless connectivity between the cards and POSs. In this paper, we propose a cost-effective mutual-authentication solution that relies on two-way challenge-response between EMV contactless cards and POSs in order to prevent sniffing attacks launched by NFC enabled readers or smartphones. To demonstrate the viability of the proposed authentication protocol, we present a Java framework to illustrate the practicality of the proposed solution. The paper argues that the proposed protocol can be easily integrated into the EMV infrastructure with minor changes at the personalization and transaction phases.

1. Introduction

Nowadays, Near Field Communication (NFC) and Radio Frequency Identification (RFID) technologies are being integrated into different hardware such as smartphones and smartcards. One of the most biggest deployments of these technologies is the contactless smart card whether for identification, accessing a building or contactless payment (Lacmanović et al., 2010). The Europay MasterCard and Visa (EMV) is an international chip-based standard payment protocol owns by the EMVCO and was introduced to replace the magnetic stripe payment protocol (Murdoch et al., 2010). There are almost 7.1 billion EMV based cards around the world as reported by the EMVCO in 2019 (EMVCO, 2018a). The EMV payment protocol in general supports five different payment methods, namely Chip & PIN, Chip & Signature Contactless card, Mobile and magnetic stripe payments. The EMVCO reported in the last quarter of 2018 that 85.5% of the cards in Europe and the UK were EMV based cards and 97.3% of the transactions were EMV transactions (EMVCO, 2018b).

The EMV contactless cards allow the users to make a quick and easy payment for goods or services with an amount less than £30 in the UK without the need to enter the Personal Identification Number (PIN). Contactless card payments depend on the NFC technology and ISO 14443 where the cardholders should place their contactless card within 10cm of the Point of Sale (POS) in order to make a contactless card

* Corresponding author.
E-mail address: hisham.al-assam@buckingham.ac.uk (H. Al-Assam).
https://doi.org/10.1016/j.cose.2021.102186

transactions (ISO/IEC, 2001). The advantage of using a contactless card to end-users is the ability to do faster transactions within around 500 milliseconds while merchants spend less time processing each transaction compared to Chip & PIN or cash transactions. The Smart Payment Association reports that customers who are using contactless cards transaction are likely to spend 30% more than when they are using Chip &PIN or cash (The Uk Cards Association, 2017). However, there are several possible attacks that can be quite easy to launch against the EMV contactless cards such as skimming, eavesdropping, replay and relay attacks (Diakos *et al.*, 2013) (Hutter et al., 2008) (Francis et al., 2012). Such attacks are possible due to the vulnerabilities in the EMV contactless card protocol, especially the wireless connectivity between POSs and EMV contactless cards. Moreover, the skimming attack is possible due to the one-sided authentication in the EMV specification as the EMV card must authenticate itself to the POS while the opposite is not happening.

The EMV payment specifications in general support three methods of card authentication namely Static Data Authentication (SDA), Dynamic Data Authentication (DDA) and Combine Data Authentication (CDA) (EMVCo, 2011a). The main aim of these card authentication methods is to authenticate the EMV card to the POS or Automated Teller Machine (ATM). The authentication process is different for each method of these three EMV card authentication methods. However, the main concept behind it is that the EMV card generates a digital signature to authenticate itself as a genuine EMV card to the POS/ATM. This could be done in three different ways. In the case of the SDA, the digital signature consists of the Static Information (SI) of the EMV card such as the Primary Account Number (PAN), cardholder name and expiry date. All these data are singed by the Issuer Bank RSA Private key (IB_{sk}) and sent by the EMV card to the POS in order to be verified. While in the case of DDA, there are two digital signatures consist of the same first digital signature in the SDA along with another one that consists of the transaction data such as the transaction amount, date, time and currency which is signed by the card RSA private key (C_{sk}). Both these digital signatures are sent by the EMV card to the POS/ATM in order to be verified. Moreover, in the case of the CDA, the EMV card includes the Application Cryptogram (AC) to the second digital signature that is used in the DDA and then the two signatures are sent to the POS/ATM to verify the EMV card as a genuine one. Fig. 1 shows the three EMV card authentication methods.

The one-sided authentication highlighted earlier allows attackers to use off-the-shelf hardware and software to obtain the sensitive information of the EMV contactless cards and use this information to launch different kinds of attacks such as Card Not Present (CNP) attack (Bond *et al.*, 2014). Therefore, the mutual authentication protocol is essential to implement to protect the EMV contactless cards from such an attack.

The UK Finance reported in December 2018 that the total number of EMV contactless cards was 123.5 million cards in the UK. Furthermore, the same source showed that the number of contactless card transactions in the UK was increased dramatically between June 2016 and December 2018 where almost 700 million contactless card transactions were made in just December 2018. Moreover, Due to the increase in the use of contactless card transactions by the cardholders in the UK, the value of these transactions was almost 6.7 billion GBP for the same month while the value was around 0.5 billion GBP in June 2015(UK Finance, 2019b).

On the other hand, the UK Finance reported in their newest report that the total losses amount of remote purchase frauds (CNP) on the EMV cards in the UK 2018 has increased by 24%. The total CNP fraud losses were £506.4 million GBP (UK Finance, 2019a). It can be argued that the fraud increase could be due to several vulnerabilities in the EMV protocol that has made different attacks possible. Furthermore, the report states that 76% of the losses on the card payment was due to the remote purchases or CNP attacks. The fraudsters gain an advantage by the ease of launching skimming attacks on contactless cards to sniff sensitive information of the EMV contactless cards and performing CNP attacks. Therefore, it is vital to provide practical solutions that can stop EMV contactless card from being easily read by unauthorized NFC enabled readers/smartphones to stop such attacks.

To address the above limitations, this paper proposes a cost-effective mutual-authentication solution that relies on a two-way challenge-response between EMV contactless cards and POSs to withstand the sniffing attacks explained earlier. The paper also presents a Java framework to illustrate the practicality of the proposed solution as explained in Section 4. We also argue that the proposed protocol can be easily integrated into the EMV infrastructure with minor changes at the personalization and transaction phases (see the discussion in Section 5).

The rest of the paper is organised as follows. Section 2 presents the literature review and some of the countermeasures to prevent reading the EMV contactless cards by unauthorized NFC enabled readers/smartphones. While Section 3 presents our mutual authentication proposed scheme for the EMV contactless cards. Next, Section 4 details our implementations and results while Section 5 presents several discussion points about the proposed scheme leaving Section 6 to conclude the research presented in the paper.

2. Literature Review

Several countermeasures have been proposed in the literature to prevent reading the EMV contactless cards by unauthorized NFC enabled readers/smartphones to prevent skimming and relay attacks. One of the earliest proposals was relying on an RFID blocking wallet that designed to stop any RFID and NFC signals to communicate with EMV contactless cards while it is inside the cardholder's RFID blocking wallet (Hong et al., 2012).

Furthermore, two solutions introduced the idea of turning the EMV contactless cards into an active card with a built-in battery instead of the original passive cards (battery less). The first method was based on using cards with an activation button where the cardholder needs to push the activation button to activate the card and allow the contactless transaction. The cardholder then needs to deactivate the card after the transaction is performed (Emms and van Moorsel, 2011). The second mothed was suggested the use of a built-in light sensor on the EMV contactless cards. Once the card is exposed to the light (outside the cardholder's wallet), the light's sensor activates the card while if the card inside the cardholder's wallet, the

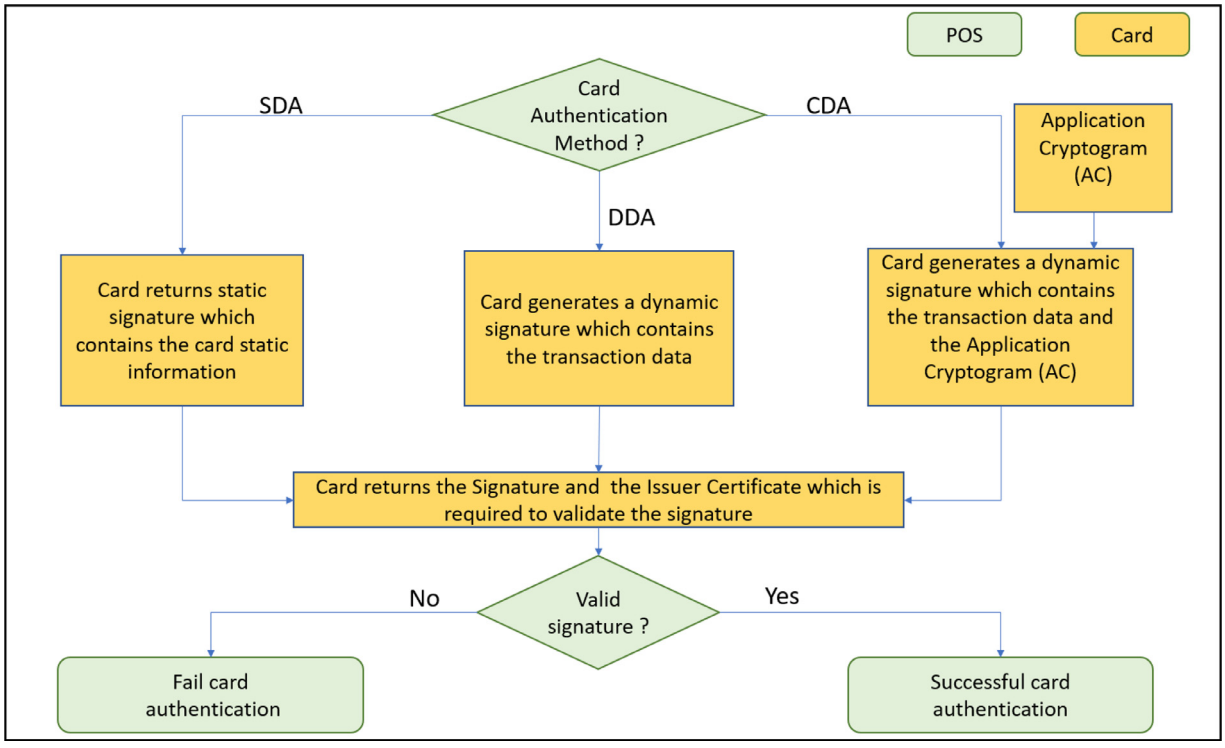

Fig. 1 – EMV Card Authentication Methods

card is always deactivated. When a genuine cardholder needs to do a transaction, the sensor activates the card and allows it to communicate with the POS (Madhoun et al., 2018). In both methods, reading the EMV contactless cards by unauthorized NFC readers/smartphones is not easily possible.

Another approach was presented by MasterCard to use a fingerprint built-in sensor to prevent unauthorized NFC enabled readers/smartphones from obtaining the EMV contactless cards sensitive information. In this method, the cardholders should place their fingerprint on the built-in sensor and place the EMV card next to the POS. The POS then powers up the EMV card and a fresh fingerprint data is compared with the stored template (stored at the personalization phase). If both fingerprints are matched, the EMV contactless card is activated (Vats et al., 2016).

More solutions based on mutual authentication between an NFC mobile payment and POSs for EMV mobile payment were proposed to ensure the confidentiality of the sensitive information of the EMV mobile payment. The first solution proposed the use of a cloud-based secure authentication protocol for the EMV mobile payment (Madhoun et al., 2015). The protocol uses asymmetric cryptography to ensure mutual authentication and encryption of the payment information during the transaction. At the time of the transaction, the NFC mobile requests from a trusted cloud infrastructure to verify the POS before sending any sensitive payment information to the POS. If the POS was authenticated successfully, the cloud generates a session key to be used by NFC mobile and POS to encrypt the transaction details to ensure the confidentiality of the transaction data. Other existing proposal was based on Needham-Schroeder protocol to ensure mutual authentication and confidentiality between the NFC mobile payment and POS (Ceipidor *et al.*, 2012). The protocol uses an authentication server to verify both, the NFC mobile and the POS and provide them with a session key to be used for encrypting transaction data.

Another mutual authentication solution was proposed for the EMV contactless cards in which the POS has its own pair of RSA public/private key along with the POS public key certificate (Martin et al., 2013). The POS needs to sign the EMV contactless card nonce by its own RSA private key and sends it back along with the POS's public key certificate to the EMV card. Then, the card retrieves the POS public key to verify its own nonce and authenticate the POS. However, the main limitation of such proposals is the size of the "APDU commands" where the maximum size of a single APDU command is 256 bytes.

Other interesting mutual authentication protocols were proposed for various network applications. Although such solutions may not be directly applicable to EMV contactless cards, the underlying techniques used for mutual authentication are application independent. For examples, a privacy-preserving protocol was used to authenticate machine-type communications by providing anonymity and preventing traceability in which the mutual authentication component relies on the complexity of the elliptic curve algorithm to act as a filter of unauthorised devices (Fu et al., 2016). A similar mutual authentication solution was proposed for as a privacy-preserving handover authentication scheme based on pseudonyms to mutually authenticate mobile station and the

target base station with minimal communication overhead (Fu et al., 2012). Another solution based on a secure distributed attestation was proposed to reduce the possibility of single point of failure verifier in which a distributed attestation was utilised to verify the integrity of IOT nodes (Kuang et al.,2019).

Most of the above reviewed solutions can arguably achieve the same objective of the proposal presented in the paper i.e. preventing unauthorised reading of contactless card's information. However, one can argue that the RFID blocking wallet (Hong et al., 2012), the activation buttons (Emms and Moorsel, 2011), the built-in light sensors on the EMV contactless cards (Madhoun et al., 2018) and the use a fingerprint built-in sensor (Vats et al., 2016) all require major changes the EMV contactless cards hardware. Furthermore, attaching a battery to the cards creates additional problems related to the life span of the battery i.e. when the card's battery is dead, the battery needs to be replaced or recharged.

Although the solution presented in (Martin et al., 2013) does not require changes to the hardware of contactless cards, the size of the two certificates required to be sent to the EMV contactless card by the POS exceeds the 256-byte maximum size of APDU commands. On the other hand, the solutions (Madhoun et al., 2015) and (Ceipidor *et al.*, 2012) for EMV mobile payments depend on the capability of the NFC mobile phone to connect to the cloud or the authentication server to authenticate the POS. Obviously, such capabilities are not available in EMV contactless cards. As a result, a reasonable mutual authentication protocol for the EMV contactless cards should depend on the EMV contactless cards themselves without the need for external servers. Other considerations when designing a mutual authentication for EMV contactless card is the size of the APDU commands and responses in addition to the contactless card transaction time as the EMV specifications allow up to 500 milliseconds to process an overall contactless card transaction (Van Den Breekel et al., 2016). Further discussion on how the proposed solution addresses the above limitation are presented in Section 5.

3. The Proposed Mutual Authentication Scheme

This section describes a cost-effective mutual authentication proposal that relies on two-way challenge-response between EMV contactless cards and POSs. This section also shows that the proposal does not require any change in the existing EMV payment protocol infrastructures as all required changes are done at the personalization and transaction phases. The overall stages of the proposal are summarised in Fig. 2 where the key distribution protocol and the personalisation phases are explained in Section 3.1. The figure shows the mutual authentication scheme is basically dependent on generating one-time random challenges generated for each transaction and exchanged based on a shared secret key of a symmetric cipher such as the Advance Encrypt Standard (AES) between both EMV card and the POS. This secret key is distributed among all the parties at the card's personalization phase as details in the next subsection. At the transaction stage, the key is used to encrypt and decrypt challenges between both POSs and EMV cards as explained in Section 3.2.

3.1. *Keys Distribution & Personalisation Phases*

The proposed solution requires that all EMV contactless cards and the POSs must have a shared secret key in order to process a mutual authentication between them and prevent reading the EMV contactless cards by unauthorized NFC enabled readers/smartphones. To minimise the changes in the EMV infrastructures, we suggest the use of existing methods that the EMV payment protocol is currently using to distribute the keys between different EMV components. As shown in Fig. 3, the original processing of the EMV keys distribution is shown with the sold-bordered boxes while our additional steps are shown with the dash-bordered boxes.

In the original EMV keys distribution process, the Certificate Authority (CA) (such as VISA and MasterCard) sends its own public key (CA_{pk}) to both Issuer Bank (IB) and Acquirer Bank (AB) so that CA_{pk} can be uploaded into the POSs. However, in the mutual authentication solution, we propose that:

- The CA distributes another key to both IB and AB in the same way that the CA_{pk} is being distributed. We refer to this key by CA Shared Secret (CA_{ss}).
- No changes are proposed to the IB public key certificate (IB_{pk}) as shown in Fig. 3 where the IB sends IB_{pk} to the CA.
- The CA then signs the IB_{pk} with its own private key (CA_{sk}) to generate the IB public key certificate.
- The AB uploads the CA_{sk} in all the POSs.

In the original EMV card personalization phase, the IB uploads the EMV card with a different type of data based on the type of the underlying EMV authentication methods (SDA, DDA or CDA). The Static Information (SI), such as the PAN, expiry date and cardholder name, is uploaded by the IB into the EMV card. The signed version of the SI is also uploaded into the card along with the IB public key certificate.

We propose one minor change to the original EMV personalization phase, that is uploading the CA_{ss} into the EMV contactless card. As a result, both the POS and EMV cards are uploaded with the same CA_{ss} at the end of both keys distribution and personalization phases.

3.2. *Transaction Phase*

The proposed mutual authentication relies on generating one-time random challenges for each transaction and employing the shared secret key, distributed to both EMV cards and the POSs as explained earlier. To stop skimming attacks, we propose that the EMV contactless card must authenticate the POS before the Get processing Options (GPO) response. The reason behind that is to prevent the EMV contactless card from revealing any sensitive data such as PAN, cardholder name and expiry date before the authentication process is completed. Please note that such sensitive data is revealed in the original EMV transaction in both responses of "GPO" and "Read Record" Commands. Fig. 4 shows the main POS's commands and EMV contactless card's responses where the original EMV transaction steps are shown in black and the additional steps for the

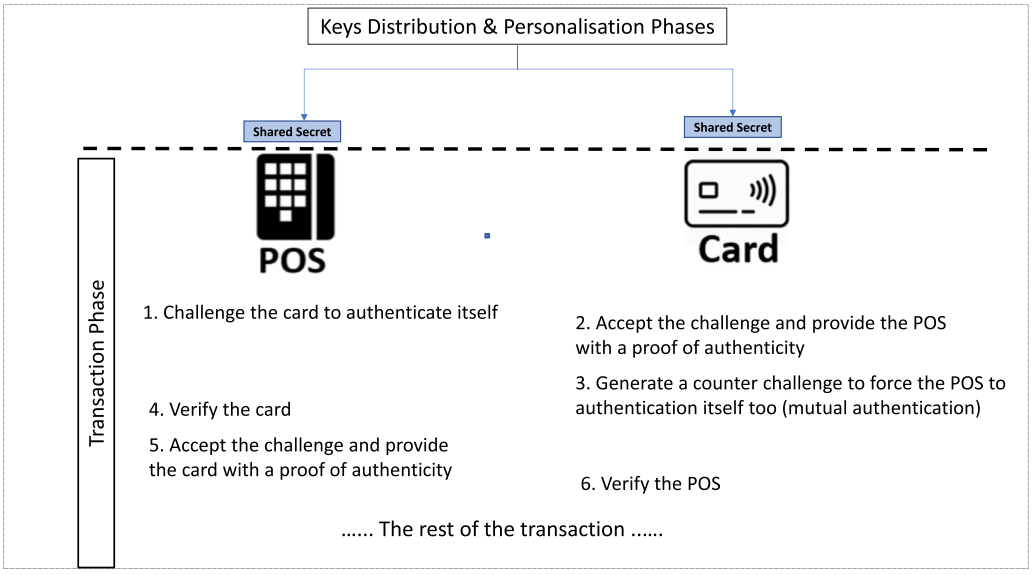

Fig. 2 – The overall phases of the proposal

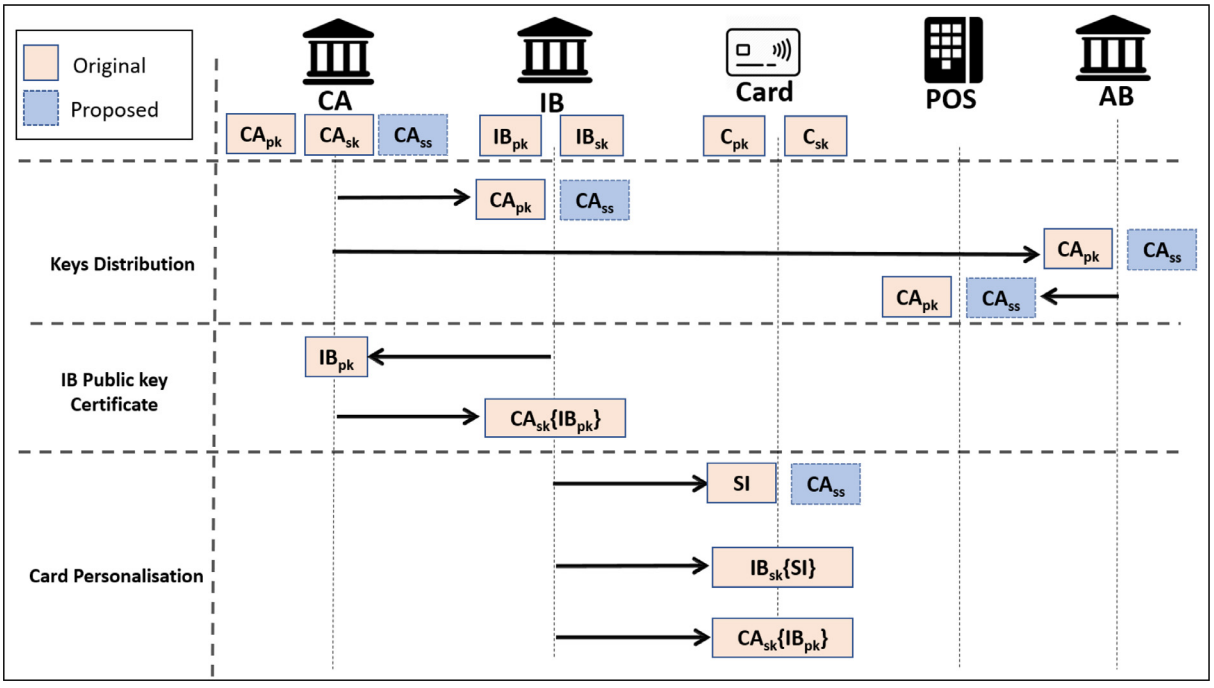

Fig. 3 – The Keys Distribution in the Proposed Scheme

proposed mutual authentication are shown in red and can be summarised as follows.

The first proposed change is shown in Fig. 4 in step 5.3 in which the POS generates its own 8 bytes of Random Challenge (POS_RC) and sends it to the card along with the Application Identification (AID) as illustrated in Fig. 4 in both steps 5.3 and 6.

- Once the card receives the POS_RC, it generates its own 8 bytes random challenge (Card_RC) as shown in step 7.1 in the same Figure.
- The EMV contactless card calculates A as shown in Fig. 4 in step 7.2 as following

$$A = (POS\_RC)XOR(Card\_RC)$$

- The card then uses the symmetric key CA_{ss} (uploaded into the card at the personalization phase by the IB as explained in the previous section) to encrypt A and send it back to the POS along with the "Select AID" response as shown in Fig. 4 in both steps 7.3 and 8.

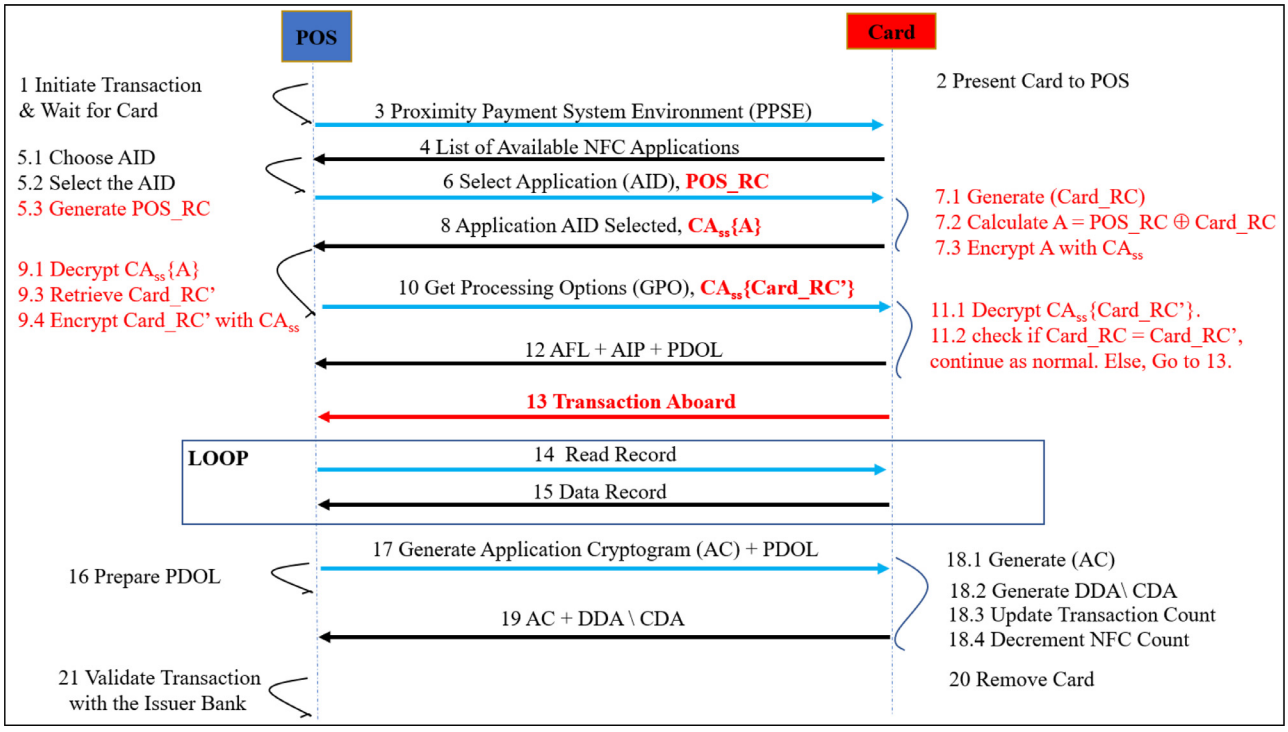

Fig. 4 – Transaction Phase in the proposed Scheme

- When the POS receives the encrypted version of A, it uses the CA_{ss} (uploaded at the key distribution phase by the AB) to decrypt and obtain A as shown in step 9.1.
- Then the POS uses its own POS_RC (generated at the previous command) to obtain the Card challenge by doing the following

$$Card\_RC' = A \text{ XOR } POS\_RC$$

- The POS encrypts the Card_RC' by its own copy of CA_{ss} and sends it to the EMV card at the "GPO" command as shown in Fig. 4 in both steps 9.4 and 10.
- When the EMV contactless card receives the encrypted version of Card_RC', it uses its own copy of CA_{ss} and decrypts it to obtain the Card_RC' as shown in step 11.1.
- Finally, the EMV card compares Card_RC and Card_RC'. If they are identical, the card authenticates the POS as a genuine one and it continues with the transaction as normal. Otherwise, the transaction should be aboard by the card as the POS fails to authenticate itself to the card as shown in steps 11.2.

The above steps show how the EMV contactless card authenticates the POS before releasing any information. The POS, on the other hand, authenticates and verifies the card based on the original EMV card authentication methods supported by the EMV payment protocol (Liu *et al.*, 2007). As a result, incorporating the proposed solution into any of the EMV card authentication methods (SDA, DDA and CDA) leads to establishing a mutual authentication between both of the POSs and EMV cards.

4. Implementation & Results

This section demonstrates the practicability of the proposed mutual authentication solution to withstand sniffing attacks on EMV contactless cards launched by unauthorized NFC enabled readers/smartphones. The section explains the hardware and software used to implement the proposed solution. Then, it presents implementation and simulation results of the proposed scheme.

4.1. *Hardware & Software*

The NFC ACR 122U reader was used in our simulations to represent the POS. The reader sends and receives all the required APDU commands (send by a standard POS to EMV cards) and responses (send back by the cards to the POS) according to the EMV specifications. We also used two Java contactless cards, one to represent an original EMV contactless card and the other Java contactless card to represent the proposed mutual authentication protocol.

Moreover, a genuine EMV contactless card (belongs to the first author) was used to duplicate its APDU responses to both Java contactless cards. Also, this EMV contactless card was used to calculate the times of the APDU commands and responses in order to compare each time with the time of the proposed solution as explain next. Fig. 5 shows all the hardware used in our implementation.

Furthermore, Java Card Integrated Development Environment (JCIDE) was used as a software to develop two applets to represent the original EMV contactless card and our proposed protocol(Attali *et al.*, 2001). In addition, PyApduTool software

Fig. 5 – Hardware used in our Implementations & Simulations

was used to send and receive the APDU commands and responses between both EMV contactless cards or Java contactless cards and the NFC ACR 122U reader (JavaCard OS, no date). The same software was used to download and install the two applets to the Java contactless cards.

4.2. *Implementations*

As explained in the previous section, all the required changes for the mutual authentication proposed solution were done at both the "Select AID" and "GPO" APDU commands and responses. Since the EMV sensitive information is revealed by the EMV contactless cards at the response of "GPO" command, the proposal forces the POS to authenticate itself to the card before the card's response to the "GPO" command in order to withstand sniffing attacks. However, the proposed solution could modify easily in order to be implemented in other EMV contactless cards kernels such as the Fast DDA (FDDA).

To demonstrate and evaluate the proposal, we used the JCIDE to develop two Java applets to represent both an original EMV contactless card and the proposed solution. Both applets were developed to respond to the first three APDU commands to serve the purpose of the implementation.

The first Java applet duplicates the original EMV contactless card shown in Fig. 5 A while the second applet was designed to simulate the proposed solution. The PyApduTool was used to connect both of the NFC ACR 122U reader and the three cards that are used in the implementation. Also, the PyApduTool was used to send the first three APDU commands namely Proximity Payment System Environment "PPSE", "Select AID" and "GPO" to the three cards according to the EMV specifications (EMVCo, 2011b). All the three APDU responses are static i.e. the responses are the same for each transaction as the three commands and responses serve the purpose of handshaking between the POS and the EMV contactless card. Table 1 confirms the correctness of implementing the first applet as it precisely duplicates the same APDU responses of the original card.

For the second applet, the applet loads the Java contactless card with the CA_{ss} key at the personalization phase. We developed the applet in a way that gives the researchers the ability to upload the Java card with AES keys of three different key sizes (128,192 and 256 bits) in order to simulate the solution. As stated earlier, no changes are required at the "PPSE" APDU command and response. Therefore, the second applet returns a similar response to the original one as shown in Table 1.

Table 1 shows all the three APDU commands ("PPSE", "Select AID" and "GPO") and their responses. The table shows the "APDU" commands of the original EMV and those of the proposed mutual authentication protocol. Two differences between the two APDU commands can be seen in the table. 1) at the "Select AID" command, the POS sends its own 8 bytes of random number(POS_RC), and 2), at the "GPO" command, the POS sends 16 bytes of the encrypted version of Card_RC.

Moreover, Table 1 shows that there are three kinds of "APDU" responses. The first APDU responses belong to the original EMV contactless card while the second APDU responses belong to the first applet that duplicates the original EMV contactless card and, hence, both APDU responses are identical. The third APDU responses, on the other hand, belong to the second applet that represents the proposed protocol where one difference compared to the original APDU re-

Table 1 – APDU commands & responses for all the three tested contactless cards

Name of APDU Command	POS Original APDU Commands	POS Proposed APDU Commands	Genuine EMV Card (A) APDU Responses	Java Card Duplicates the Genuine EMV Card (B) APDU Responses	Java Card implements the Proposed Protocol (C) APDU Responses
PPSE	00 A4 04 00 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 00	00 A4 04 00 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 00	6F 3A 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 28 BF 0C 25 61 23 4F 07 A0 00 00 00 03 10 10 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 0A 08 00 01 05 01 00 00 00 00 90 00	6F 3A 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 28 BF 0C 25 61 23 4F 07 A0 00 00 00 03 10 10 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 0A 08 00 01 05 01 00 00 00 00 90 00	6F 3A 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 28 BF 0C 25 61 23 4F 07 A0 00 00 00 03 10 10 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 0A 08 00 01 05 01 00 00 00 00 90 00
Select AID	00 A4 04 00 07 A0 00 00 00 03 10 10 00	00 A1 04 00 0F A0 00 00 00 03 10 10 **11 22 33 44 55 66 77 88** 00	6F 52 84 07 A0 00 00 00 03 10 10 A5 47 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 38 18 9F 66 04 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 BF 0C 1A 9F 5A 05 31 08 26 08 26 9F 0A 08 00 01 05 01 00 00 00 00 BF 63 04 DF 20 01 80 90 00	6F 52 84 07 A0 00 00 00 03 10 10 A5 47 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 38 18 9F 66 04 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 BF 0C 1A 9F 5A 05 31 08 26 08 26 9F 0A 08 00 01 05 01 00 00 00 00 BF 63 04 DF 20 01 80 90 00	6F 52 84 07 A0 00 00 00 03 10 10 A5 47 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 38 18 9F 66 04 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 BF 0C 1A 9F 5A 05 31 08 26 08 26 9F 0A 08 00 01 05 01 00 00 00 00 BF 63 04 DF 20 01 80 **06 7E 25 89 0C 24 83 BD 3E C0 F6 5D C4 9B EF EF** 90 00
GPO	80 A8 00 00 23 83 21 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	80 A8 00 00 33 83 21 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 **CA 97 8C 5D A9 95 0E 0E 05 B7 44 9E 56 83 01 D2** 00	77 4D 82 02 20 00 94 04 10 02 05 00 57 13 47 51 39 05 86 29 91 07 D2 21 02 21 99 20 00 00 00 00 0F 5F 34 01 00 9F 10 07 06 01 0A 03 A0 00 00 9F 26 08 92 FB E4 3F 5B D5 3D B6 9F 27 01 80 9F 36 02 00 1B 9F 6C 02 2E 00 9F 6E 04 20 70 00 00 90 00	77 4D 82 02 20 00 94 04 10 02 05 00 57 13 47 51 39 05 86 29 91 07 D2 21 02 21 99 20 00 00 00 00 0F 5F 34 01 00 9F 10 07 06 01 0A 03 A0 00 00 9F 26 08 92 FB E4 3F 5B D5 3D B6 9F 27 01 80 9F 36 02 00 1B 9F 6C 02 2E 00 9F 6E 04 20 70 00 00 90 00	77 4D 82 02 20 00 94 04 10 02 05 00 57 13 47 51 39 05 86 29 91 07 D2 21 02 21 99 20 00 00 00 00 0F 5F 34 01 00 9F 10 07 06 01 0A 03 A0 00 00 9F 26 08 9B 47 D9 91 EC 49 A7 12 9F 27 01 80 9F 36 02 00 1C 9F 6C 02 2E 00 9F 6E 04 20 70 00 00 90 00

sponse. This difference can be seen when the card sends 16 bytes of the encrypted version of XORing POS_RC and Card_RC at the response of “Select AID” command.

5. Discussions

This section presents three key discussion points that address the following questions: 1) is the proposed solution efficient enough to meet the 500ms threshold?, 2) does the proposal satisfy the current EMV requirements related to the APDU size of commands and responses between the card and the POS? and 3) why the proposal is secure against sniffing attacks and what are the limitations?

5.1. *Time & Computational Overhead*

As mentioned earlier the transaction time is a vital point when assessing the proposed mutual authentication protocol for the EMV contactless card transactions. Such transactions should be processed very quickly. Therefore, any proposal that takes a long time to be processed must be rejected automatically.

We used PyApduTool to calculate the overall time to perform all the three APDU commands and responses. We repeated these three commands 10 times and reported the average time on the three different cards that are used in the implementation individually. Fig. 6 shows the times for the genuine EMV contactless (Card A) and the first Java contactless card that was designed to duplicate the genuine card (Card B)

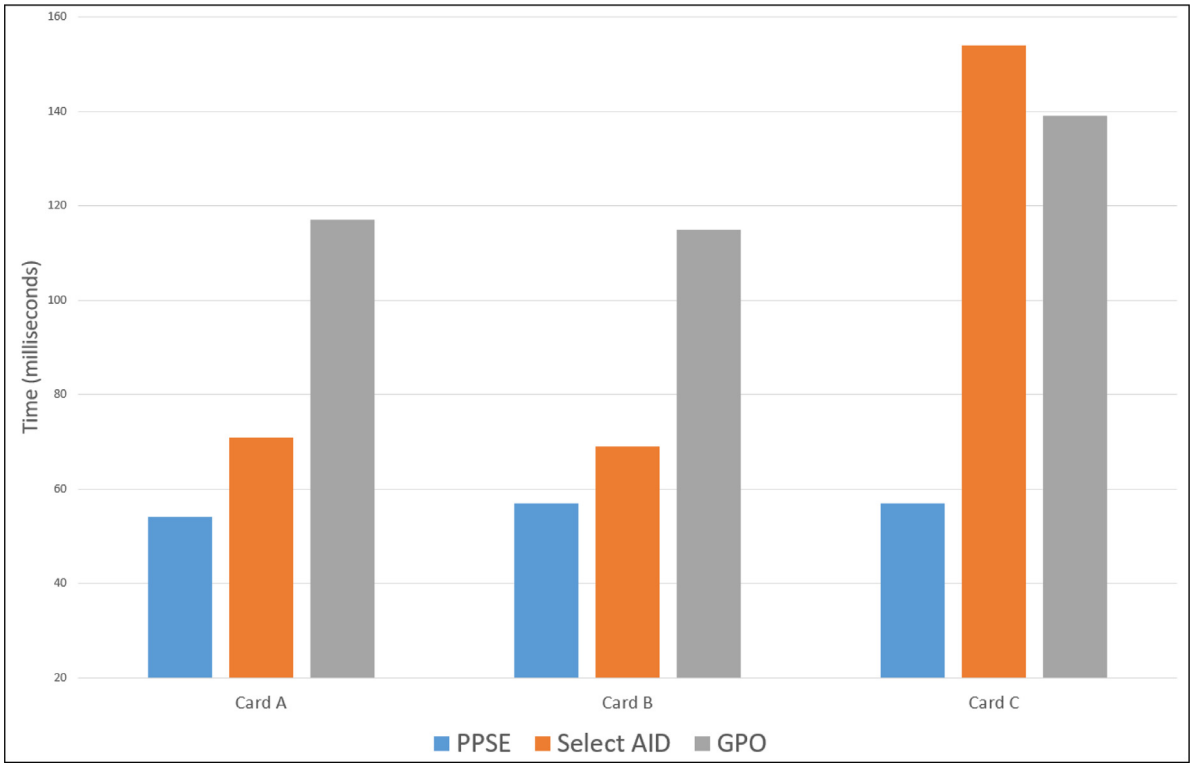

Fig. 6 – Timing Results for the proposed scheme

while the second Java contactless card (card C) is designed to implement the proposed scheme.

As shown in Fig. 6, the three cards (A, B and C) are spending almost the same time in order to process the first APDU command "PPSE" within almost 58 ms. This is an important observation not only to illustrate that the proposal does not change this specific APDU command and response but also to confirms that the reported execution times in this paper are comparable to those in real-life i.e. the running of the same commands on Java and cards and an original bank card are comparable.

The figure also shows that both the original EMV contactless card (Card A) and the first Java contactless card (Card B) spend in average around 70 ms to process the second APDU response "Select AID". However, the proposed scheme (Card C) spends around 155 ms to process the same command i.e. an extra 85 ms as an overhead time of processing all the required steps in the proposed mutual authentication scheme.

Moreover, in the third APDU response of "GPO command", the proposed solution takes around 25ms more than Card A and B. This extra time is required to process the steps of the proposed scheme. That makes the total extra time to process the proposed mutual authentication scheme is around 110 ms. It can be argued that the extra 110 ms is a reasonable overhead to be added to the EMV contactless card transactions as some of the genuine EMV contactless cards can take longer than 500 ms (van den Breekel, 2015). In fact, it has been shown that the POS usually tolerate contactless transactions even if they take more than 650 ms and the POS process and accept the contactless cards transactions within maximum time of 2500 ms (Chothia *et al.*, 2015).

5.2. *APDU Size & Required Storage Size*

The second consideration that the proposed mutual authentication protocol must consider to be integrated into the existing EMV infrastructure is the APDU size. The previous section showed that the whole process of the proposal requires just 40 bytes to be added to the APDU commands and responses of the original EMV contactless transactions. As shown in Table 1, there are 8 bytes added to the "Select AID" APDU command that represents the POS random number. Furthermore, there are 16 bytes added to the response of the same command "Select AID". The last 16 bytes are added on the "GPO" command. Therefore, these 40 bytes consider as a light added bytes to the original EMV APDU size.

Furthermore, the proposed mutual authentication scheme requires to store the CA_{ss} into the second Java contactless card (Card C) at the personalization phase. This additional AES and key storage is about 3 KB. This added storage size is acceptable as the Java card used in the implementation had 72.8 KB of storage size. The way that this extra storage size is measured in the implementation is by comparing the size of the Java applet of Card B and Card C. As the applet Java size for Card B is 12 KB while it is 15 KB for Card C.

5.3. *Security Analysis*

This section discusses the security the shared secret (CA_{ss}) and the robustness against the man-in-the-middle attack and presents a comparison summary with other schemes reported in the literature.

Table 2 – Comparing existing scheme with the proposal

	Robust against Sniffing	Robust against MITM	Source of security	Applied on contactless cards	No hardware changes	Within APDU maximum size
(Hong et al., 2012) (Emms et al. 2011) (Vats et al., 2016)	✓	✓	Extra Hardware on the card	✓	×	N A
(Madhoun et al., 2015) (Ceipidor et al., 2012)	✓	✓	Smartphones	×	N A	NA
(Kuang et al.,2019) (Fu et al., 2016)	✓	✓	Mutual authentication + elliptic curve	×	N A	NA
(Martin et al., 2013)	✓	✓	Mutual authentication +RSA	✓	✓	×
The Proposal	✓	✓	Mutual authentication +AES 256 bits	✓	✓	✓

5.3.1. *Robustness against the sniffing attack*

The main objective of the proposal is to stop unauthorised readers from sniffing the card's details without the cardholder's knowledge. The whole security of the mutual authentication scheme rests on the security of the shared secret CA_{ss} uploaded into the EMV card's chip and the Secure Access Module (SAM) of the POS.

The EMV chip itself considers as a secure environment to store sensitive information such as the card RSA private key and the Shared key of IB and the EMV chip. Also, the SAM is currently used to store all the CA certificates and several EMV applications. To the best of the authors' knowledge, no successful skimming attacks on the EMV card's chip or the SAM have been reported in the literature.

If an attacker eavesdrops the communication between a POS and an EMV contactless card during the time of the contactless card transaction (practically hard but possible), can the attacker obtain the CA_{ss}?. In the proposed scheme, the CA_{ss} is not being sent by either POSs or the EMV contactless cards during the transaction. Therefore, even if the attacker eavesdrops to the communication, it is currently computationally hard to brute-force the CA_{ss} if it is a 256-bit key.

5.3.2. *The robustness against the man-in-the-middle attack*

Attackers could also launch a Man-In-The-Middle (MITM) attack during the EMV contactless card transactions and modify the APDU commands and responses between the POSs and the EMV contactless cards. The POS_RC could be modified by an attacker which results to failure of the proposed scheme. However, this attack is possible in theory while in the real-life scenario, the attack is very difficult to achieve due to the limited distance (maximin of 10 cm) between both POSs and EMV contactless cards during the contactless card transactions.

5.3.3. *A comparison with other schemes reported in the literature*

To summarise the differences between the proposed mutual authentication scheme and other existing scheme, Table 2 below presents compact comparisons in terms of security, application to contactless cards and the size of APDU commands.

We appreciate that proposing a new solution to the existing EMV payment protocol affects almost 7.1 billion EMV based cards around the world (EMVCO, 2018b). Therefore, we suggest that the proposal can be deployed in newly issued EMV based cards while the existing EMV cards continue to work as usual using the current EMV payment protocol with the one-sided authentication protocol. To achieve that, we propose the use of one bit from the two bytes of the Application Interchange Profile (AIP)(EMVCo, 2011b). Using the bit as a flag to inform the POS of whether the EMV contactless card supports the proposal or not so the POS can respond accordingly.

6. Conclusion

One of the biggest threats on the EMV contactless cards is the skimming attacks that might lead to other relevant attacks such as CNP, replay and cloning attacks. The paper argued that launching skimming attacks on EMV contactless cards is quite easy to perform by attackers using off-the-shelf hardware and software. The paper showed that the main vulnerability in the EMV payment specifications that led to skimming attacks is the single-sided authentication protocol. The EMV specifications force the EMV card to authentication itself to the POS while the reverse is not happening. Therefore, Any NFC enabled readers/smartphones can obtain most of the sensitive information of the EMV contactless card without the knowledge of cardholders. The paper presented a mutual authentication protocol between the EMV contactless cards and the POSs. The proposed protocol is cost-effective and easy to integrate into in the exiting EMV infrastructures. The paper then identified and explained the minor changes that need to be made into the EMV payment protocol to adopt the proposed solution. To demonstrate the effectiveness of the proposal, we developed two Java applets on two Java contactless cards to simulate the original EMV contactless card and the

proposed mutual authentication protocol. The implementation and simulation results showed how minimal the changes to the EMV specifications and infrastructures could be. Initial results show that the proposal is efficient enough to be meet the time constrain associated with contactless card transactions and the additional data required by the proposal do not exceed the APDU maximum size.

Declaration of Competing Interest

The authors have no conflict of interest.

CRediT authorship contribution statement

Ossama Al-Maliki: Software, Writing - original draft, Visualization, Investigation. **Hisham Al-Assam:** Methodology, Validation, Writing - review & editing, Visualization.

REFERENCES

Attali I, et al. An integrated development environment for Java Card. Computer Networks 2001. doi:10.1016/S1389-1286(01)00162-1.

Bond, M. et al. (2014) 'Chip and skim: Cloning EMV cards with the pre-play attack', in *Proceedings - IEEE Symposium on Security and Privacy*. doi:10.1109/SP.2014.11.

Van den Breekel J. In: BlackHat Asia, Singapore. Relaying EMV Contactless Transactions using Off-The-Shelf Android Devices; 2015.

Van den Breekel J, et al. EMV in a nutshell. In: Technical Report; 2016. p. 1–37.

Ceipidor, U. B. et al. (2012) 'A protocol for mutual authentication between NFC phones and POS terminals for secure', pp. 115–120.

Chothia T, et al. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Relay cost bounding for contactless EMV payments; 2015. doi:10.1007/978-3-662-47854-7_11.

Diakos TP, et al. Eavesdropping near-field contactless payments: a quantitative analysis. The Journal of Engineering 2013;2013(10):48–54. doi:10.1049/joe.2013.0087.

Emms M, van Moorsel A. Practical Attack on Contactless Payment Cards. HCI2011 Workshop - Heath, Wealth and Identity Theft 2011.

EMVCo (2011a) *Book 2: Security and Key Management, EMV Integrated Circuit Card Specifications for Payment Systems*. doi:10.7591/9780801469121-014.

EMVCo (2011b) *Book 3: Application Specification, Integrated Circuit Card Specifications for Payment Systems*. doi:10.7591/9780801469121-015.

EMVCO (2018a) '2018: a Year in Review', 392(10165), pp. 2669–2670. doi:10.1016/s0140-6736(18)33244-6.

EMVCO (2018b) *Contact EMV Global Adoption, EMVCO*. Available at: https://www.emvco.com/about/deployment-statistics/ (Accessed: 14 February 2020).

Francis L, et al. Practical relay attack on contactless transactions by using NFC mobile phones. Cryptology and Information Security Series 2012;8:21–32. doi:10.3233/978-1-61499-143-4-21.

Fu A, Song J, Li S, Zhang G, Zhang Y. In: A privacy-preserving group authentication protocol for machine-type communication in LTE/LTE-A networks, 9. Security and Communication Networks; 2016. p. 2002–14.

Fu A, Zhang Y, Zhu Z, Jing Q, Feng J. An efficient handover authentication scheme with privacy preservation for IEEE 802.16 m network. Computers & Security 2012;31(6):741–9.

Hong L, Yong HC, Zhang QH. In: 2012 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2012. The survey of RFID attacks and defenses; 2012. doi:10.1109/WiCOM.2012.6478720.

Hutter M, Schmidt JM, Plos T. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). RFID and its vulnerability to faults; 2008. doi:10.1007/978-3-540-85053-3_23.

ISO/IEC (2001) *ISO/IEC 14443-4 - Identification cards — Contactless integrated circuit(s) cards — Proximity cards — Part 4: Transmission protocol*. doi:10.1021/nl8019328.

JavaCard OS (no date) *Java Card Development Kit*. Available at: https://www.javacardos.com/tools (Accessed: 5 March 2020).

Kuang B, Fu A, Yu S, Yang G, Su M, Zhang Y. ESDRA: An efficient and secure distributed remote attestation scheme for IoT swarms. IEEE Internet of Things Journal 2019;6(5):8372–83.

Lacmanović I, Radulović B, Lacmanović D. Contactless payment systems based on RFID technology. In: MIPRO 2010 - 33rd International Convention on Information and Communication Technology, Electronics and Microelectronics, Proceedings; 2010. p. 1114–19.

Liu MH, et al. Security mechanism research of EMV2000. Proceedings - 2007 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT Workshops 2007, 2007.

Madhoun NEl, Bertin E, Pujolle G. An overview of the EMV protocol and its security vulnerabilities. In: 2018 4th International Conference on Mobile and Secure Services, MOBISECSERV 2018; 2018. p. 1–5. doi:10.1109/MOBISECSERV.2018.8311444.

Madhoun NEl, Guenane F, Pujolle G. A cloud-based secure authentication protocol for contactless-NFC payment. In: 2015 IEEE 4th International Conference on Cloud Networking, CloudNet 2015; 2015. p. 328–30. doi:10.1109/CloudNet.2015.7335332.

Martin E, Budi A, J H, A van M. POS Terminal Authentication Protocol to Protect EMV Contactless. In: *TECHNICAL REPORT SERIES: Newcastle University.*, No. CS-TR-(2); 2013. p. 2–9. doi:10.1145/335527.335528.

Murdoch SJ, et al. Chip and PIN is broken. Proceedings - IEEE Symposium on Security and Privacy, 2010.

The Uk Cards Association (2017) 'Guide for retailers: Accepting contactless and higher value contactless payments'.

UK Finance (2019a) *2019: Half Year Fraud Report*. Available at: https://www.ukfinance.org.uk/system/files/2018-half-year-fraud-update-FINAL.pdf.

UK Finance (2019b) 'Contactless Transit: Implementation in the UK', 2, pp. 1–42.

Vats H, Ruhl R, Aghili S. In: 2015 10th International Conference for Internet Technology and Secured Transactions, ICITST 2015. Fingerprint security for protecting EMV payment cards; 2016. doi:10.1109/ICITST.2015.7412065.

Hisham Al-Assam holds BEng in Software Engineering and Information Systems, and a PhD in Computer Science. He is a senior lecturer in Computer Science at the School of Computing, University of Buckingham. Hisham's research into cyber security include deep learning for biometric recognition, dynamic signa-

tures, privacy-aware biometric template security and multi-factor remote authentication, identity based encryption and EMV contactless cards security.

Ossama Al-Malki is a PhD student at the School of Computing, University of Buckingham. His research project focuses on analysing and enhancing the security of the EMV contactless cards. Ossama is working on a number of security-improved solutions that can withstand different attacks on contactless bank cards such as sniffing, relay and card-not-present attacks.