Ingénierie des Systèmes d'Information
Vol. 27, No. 1, February, 2022, pp. 41-47
Journal homepage: http://iieta.org/journals/isi

Vulnerability of the Dynamic Array PIN Protocol

Samir Chabbi*, Djalel Chefrour

Dept. of Mathematics and Informatics, University of Souk-Ahras, BP 1553 Souk-Ahras 41000, Algeria

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

https://doi.org/10.18280/isi.270105

Received: 6 November 2021
Accepted: 12 January 2022

Keywords:
PIN authentication, automated teller machine, NFC smartphone, dynamic array PIN protocol, vulnerability

ABSTRACT

We recently proposed the Dynamic Array PIN protocol (DAP), which is a novel approach for user authentication on Automated Teller Machines. DAP replaces bank cards with smartphones that support Near Field Communication (NFC) and allows a user to enter his PIN code in a secure way. We showed that DAP is resistant to 13 different attacks and is therefore better and more cost effective than several other solutions from the literature. However, after carrying a deeper analysis we found that DAP is vulnerable to a complex attack that might lead to unauthorized transactions on ATMs if the user smartphone and his PIN code are both stolen. In this paper we expose how the user PIN code can be discretely discovered using multiple eavesdropping videos or camera records. We also propose several fixes for this vulnerability.

1. INTRODUCTION

Near Field Communication (NFC) is a communication technology that allows contactless information transfer between a reader like an Automated Teller Machine (ATM) and a bank card or a smartphone at a short distance [1]. It uses an electromagnetic field to transfer messages within 10 centimeters at a frequency of 13.56 MHz [2]. NFC became very popular and is now used in many products to perform proximity payments [3]. For instance, customers can perform a secure payment by bringing an NFC enabled smartphone close to an ATM [4]. This technology simplified ATM transactions by removing the need for bank cards and is therefore beneficial for both the banks and their users [5]. Unfortunately, it is possible for an attacker to steal the banking information that is stored in an NFC bank card [6]. The user password or his Personal Identification Number (PIN) is among the sensitive information that can be stolen by various attacks such as: shoulder-surfing [7], brute force [8], side channel [9], video recording [10], replay [11], spyware [12], camera recording [13], smudge [14] and multiple registrations [15].

When an NFC enabled bank card or smartphone is used for payment with ATM, securing the PIN code becomes extremely mandatory [16] because the PIN is subject to several attacks.

Authentication is an important security feature that protects an entity access to a valuable resource [17]. In the literature, several techniques are proposed for a contactless and secure user authentication at an ATM. Some solutions rely on a PIN or a password. Others exploit a biometric technology [18], such as a fingerprint readers or camera based human face recognition. Each solution has its limits and inconvenient.

Nevertheless, the cost of biometric methods outweighs their benefits in most use cases, when compared to the password or PIN based authentication [19]. Additionally, if a biometric characteristic is stolen, the user cannot change it as he can do with a password or a PIN. In an NFC enabled smartphone, the user password or PIN can be stored in a hardware circuit called Secure Element (SE). The SE ensures the security of the data it stores and the execution of sensitive programs like electronic payment applications [20].

We have recently proposed a novel PIN based method for the user authentication at an ATM using an NFC enabled smartphone. This method, called Dynamic Array PIN protocol (DAP), is more cost effective and offers a better security than many literature techniques [21]. When a customer carrying his smartphone approaches and ATM, the screen of the latter executes the DAP protocol to authenticate him.

In this paper, we present our discovery of a vulnerability in the DAP protocol that can allow an attacker to steel the user PIN using the intersection of multiple recordings. Before detailing this vulnerability, we discuss related works in Section 2. Next, we review how the DAP protocol works in Section 3. Then, we detail the vulnerability in question in Section 4. Finally, we present our conclusion in Section 6.

2. RELATED WORKS

Several works have been proposed to secure the user password or PIN code used for authentication in NFC enabled payment. We present here after the most important and recent ones; and analyze their strengths and limitations.

2.1 Cappcha

The Completely Automated Public Physical test to tell Computer and Humans Apart (Cappcha) method provides authentication by displaying a PIN pad for the user to enter his code only if he tilts the device to a degree displayed on the screen and holds it there for one second. This degree is generated randomly for each authentication, while the smartphone inclination carried by the user is measured by an accelerometer integrated into its secure element [22].

Hence, Cappcha prevents malware from carrying an

unwanted authentication even if it steels the PIN code, as it cannot tilt the device without a human intervention. However, Cappcha suffers from the Shoulder-surfing attack as well as the concealed camera recording attack. In both cases an attacker can see the PIN code and if he manages to steal the user phone, he can simply carry out a fraudulent payment operation.

2.2 BrightPass

In this technique, the SE generates a sequence of 0 and 1, called Lie Overhead, from which a series of circles of different brightness will be displayed in the Smartphone PIN pad. A low brightness circle (corresponding to the value 0) tells the user to type a random digit that is not a part of the PIN code, while a high-brightness circle (corresponding to the value 1) indicates to the user to type a real number that is a part of the PIN code. This technique is used for fighting the spyware attack that tries to find the typed PIN by capturing the phone screen [23]. Nonetheless, the BrightPass technique suffers from the Shoulder-surfing attack and the concealed camera attack like the previous one.

2.3 CWPIN

In the Color Wheel PIN (CWPIN) method, the bank server shares with the user a secret composed of his PIN code and a unique table of ten colors (indexed from 0 to 9, where for instance 0 corresponds to red, 1 to orange, etc.). This table is randomly shuffled by the server for each new authentication, then the shuffled index is sent to the ATM, which also relays it to the user smartphone via NFC or via a QR code. Next, the ATM displays on its screen a random wheel of 10 colors and a seek-bar to spin it. This wheel is surrounded by a fixed index from 0 to 9. On the other hand, the smartphone displays a colored array built from the shuffled index and the color table stored in its secure element. This color array is also topped with an index that goes from 0 to 9.

To authenticate, the user picks the color corresponding to his PIN first digit from the phone array and spins the ATM wheel to match this color with his PIN second digit. After which the wheel rotates randomly. Then, the user repeats the previous step to match array color corresponding to the PIN third digit with its fourth digit in the ATM screen [24]. In this way, CWPIN protects the PIN against smudge and malware attacks. However, it is not applicable if the PIN code has an odd number of digits.

3. THE DAP PROTOCOL

The DAP protocol uses only the PIN code as a shared secret between the user and the bank server. In addition, it relies on NFC communication between a smartphone equipped with a secure element and an ATM furnished with a small touch pad covered with a shell to mask the user's finger movements. When the smartphone is brought near the NFC reader of the ATM, the latter displays on tis screen two vertically aligned arrays containing each 10 random digits. While the content of the upper array is fixed on the screen for the authentication session, the lower one can slide horizontally following the movement of the user finger on the ATM touch pad. We note that the content of the lower array rotates horizontally in a circular way, so that the cells that disappear from one side will appear immediately on the opposite side in the same order.

To authenticate, the user following the steps of the next protocol:

(1) Pick the digit in the lower array that is below to the PIN's first digit in the upper array. Then look for this picked digit in the upper array and consider its position there as a reference which will be used in the remaining steps of the session. For instance, as shown in Figure 1, to introduce the PIN code '8642', the user looks for 8 in the upper array and picks the digit below it in the lower array, which is 4 in this case. Then, he looks for 4 in the upper array and considers its cell (which is the third one from the left) as the reference for this authentication session.

(2) Then, using the touch pad the user slides the lower array to the left or to the right and only releases his finger when the first digit of the PIN code in the lower array aligns vertically with the reference position in the upper array. The ATM graphical user interface (GUI) uses the technique of "snapping", which is well known in drawing programs, to ensure that the cells of two arrays are perfectly vertically aligned when the user releases his finger. In Figure 1, the horizontal scroll bar is used to simulate the ATM touch pad.

(3) The previous step is repeated for every digit remaining in the PIN code. To end of this operation, the user presses the 'Validate' button. More details about the internal working of the DAP protocol can be found in the study [21].

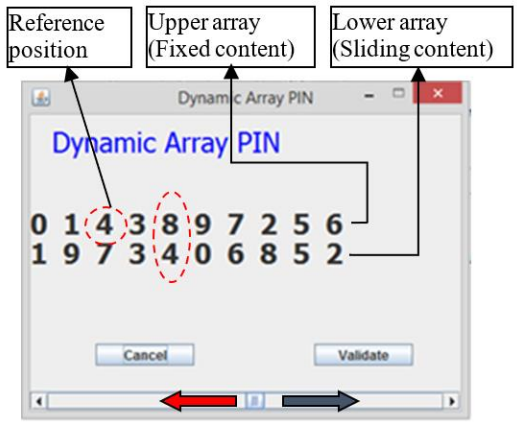

Figure 1. The DAP protocol interface

4. VULNERABILITY OF THE DAP PROTOCOL

In this section, we demonstrate by example that the Dynamic Array PIN protocol is vulnerable to the multiple record intersection attack.

4.1 Principle of the multiple record intersection attack

This attack is typically applied to discover the PIN entered by the user when using an intelligent technique (the digits of the PIN are not entered directly). It is usually carried out by a concealed camera that records the steps of PIN based authentication protocol.

4.2 Steps of the demonstration

To apply the intersection of multiple records attack successfully, the attacker must target a victim, record at least two of his ACM authentication sessions. For the purpose of

demonstrating how this attack works we assume that the PIN code of the victim is 3582 and record his steps during two authentication sessions.

4.2.1 First DAP session

We suppose that the window of Figure 2 is shown to the user when he starts his first authentication session. Hence, according to the DAP protocol detailed previously, the user will pick the first left cell of the upper array as a reference position. This is because it holds the digit 0 which exists (in the lower array) below the first digit of the PIN, that is 3. The reference position is for now unknown to the attacker.

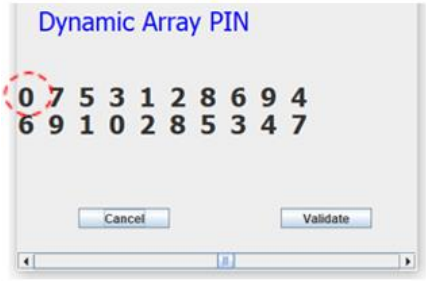

Figure 2. Start of the first DAP session

Next, the user slides the lower array to the left (i.e., with 7 positions without the need to count them) until the PIN's first digit (i.e., 3) aligns with the reference position, at which time the user lifts his finger and the movement of the lower array stops, as shown in Figure 3.

Figure 3. Input of the PIN first digit '3'

By repeating the same process for the remaining digits of the PIN code, the user and also the attacker will notice the following screens, depicted by Figures 4 to 7, each time the sliding movement of the lower array stops. This process is carried on until the 'validate' button is pressed.

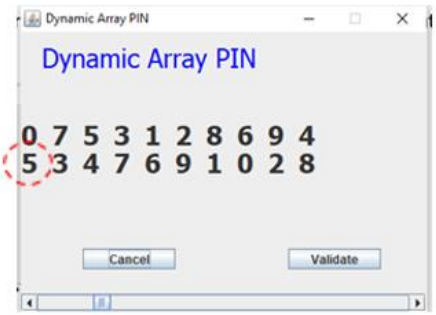

Figure 4. Input of the PIN second digit '5'

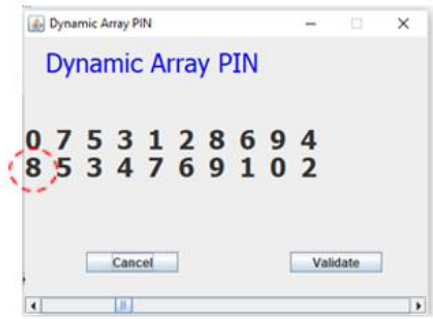

Figure 5. Enter of the third PIN digit '8'

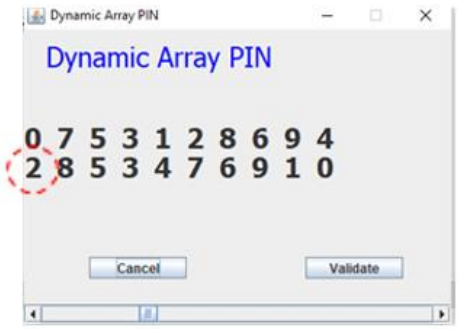

Figure 6. Input of the PIN fourth digit '2'

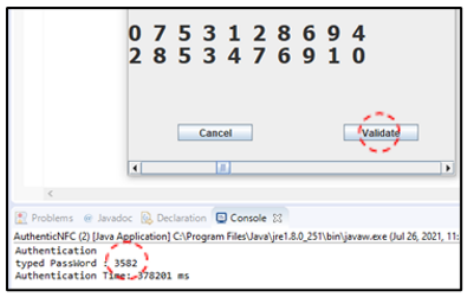

Figure 7. End of the first DAP session

Once the PIN is validated, the attacker can regroup the states of the lower array from the previous screens to obtain the results shown in Table 1. Each row corresponds to the content of the lower array when its sliding movement stopped, that is when a digit from the PIN code was input.

Table 1. Lower array states extracted from the first DAP session

3	4	7	6	9	1	0	2	8	5
5	3	4	7	6	9	1	0	2	8
8	5	3	4	7	6	9	1	0	2
2	8	5	3	4	7	6	9	1	0

4.2.2 Second DAP session

As for the first DAP authentication session above, we assume for this second session that both the user and the eavesdropper will see the beginning of the authentication protocol illustrated by Figure 8. This time the reference position is the fourth cell from the left in the upper array. Both arrays contain initially random numbers and the reference position is still unknown to the attacker. Furthermore, both the user and the attacker will observe the next screens as the user input goes on until the session finishes successfully.

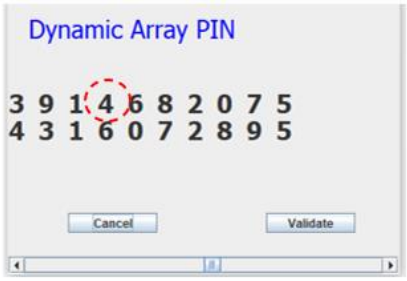

Figure 8. Start of the second DAP session

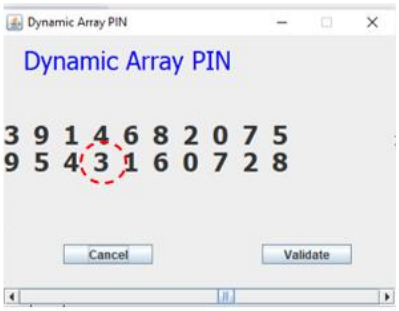

Figure 9. Input of the PIN first digit '3'

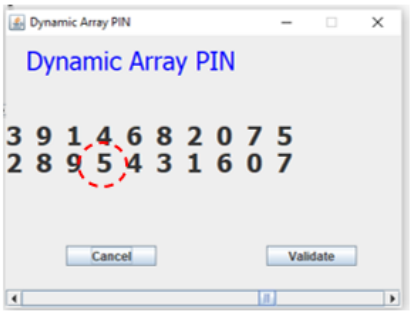

Figure 10. Input of the PIN second digit '5'

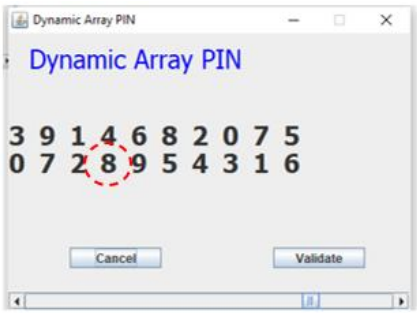

Figure 11. Input of the PIN third digit '8'

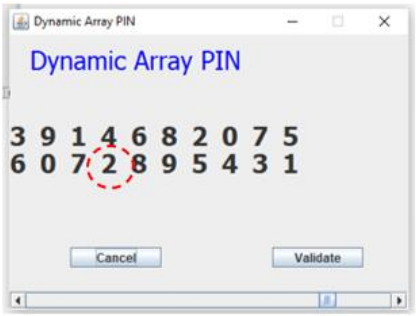

Figure 12. Input of the PIN fourth digit '2'

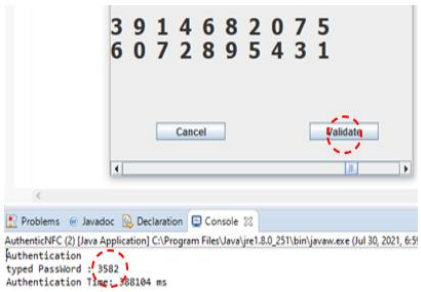

Figure 13. End of the second DAP session

Figures 9 to 12 show how the user enters the digits of his PIN code. Figure 13 shows that after clicking on the validate button, the application displays the PIN code typed and the authentication result (authentication in our case).

Once more, the attacker will group the states of the lower array from the previous screens and will have the results shown in Table 2.

Table 2. Lower array states extracted from the second DAP session

9	5	4	3	1	6	0	7	2	8
2	8	9	5	4	3	1	6	0	7
0	7	2	8	9	5	4	3	1	6
6	0	7	2	8	9	5	4	3	1

At this stage, the attacker can reveal the user PIN code by comparing the columns of both tables 1 and 2. The column that is present in both of them, in other words their intersection set, will contain the digits of the PIN code ordered from top to down. The top cell of this common or intersection column corresponds to the left most PIN digit. Table 3 below regroups the two previous ones and shows the column in question.

Table 3. PIN revealed by the column common to the two extracted tables

Table 1	3	4	7	6	9	1	0	2	8	5
	5	3	4	7	6	9	1	0	2	8
	8	5	3	4	7	6	9	1	0	2
	2	8	5	3	4	7	6	9	1	0
Table 2	9	5	4	3	1	6	0	7	2	8
	2	8	9	5	4	3	1	6	0	7
	0	7	2	8	9	5	4	3	1	6
	6	0	7	2	8	9	5	4	3	1

The fact that the DAP arrays are generated randomly at the start of each authentication session guaranties that no other column will be present in intersection of the two tables. Only the one containing the PIN will be present in these tables.

This multiple records attack is made possible because the position of the reference cell is fixed once the DAP authentication session starts. Although, this simplifies the user task, it makes it easy for an attacker to guess the PIN code even with one video recording. Indeed, a fixed reference position means that the digits that compose the PIN code will always be in the same column of the table extracted from the lower array. So there is one chance out of ten to figure out the right PIN. Fortunately, most ATMs will lock the user account and require a PIN change after three failing authentication attempts. Hence, to succeed from the first trial the attacker needs the records of at least two normal user DAP sessions.

5. PROPOSED SOLUTIONS

The first obvious solution to the DAP vulnerability explained above is to randomize the reference position after the input of each digit of the PIN. This idea is similar to what happens in the CWPIN method described in the related works section. In CWPIN, once the user spins the wheel to enter the two digits of his PIN, the color wheel is rotated on the ATM a screen with a random angle. Likewise, to improve security of the DAP protocol we propose to randomize the content of the upper array before the input of each digit of the PIN code, so the reference position is not fixed during the session. The user will have to memorize the digit in the reference cell when the session starts. Then look for it in the new randomized array and consider its position the new reference for the next digit of the PIN code.

In the following Table 4, we show step by step how this idea improves the security of the DAP protocol. It is applied on a new example that uses the same PIN code 3582 as above. We use the green color to indicate the reference cell in the upper array and the red color to indicate the entered digit in the lower array.

Table 4. The steps of the improved DAP protocol

Steps	Content of the tow DAP arrays									
Session start	0	7	5	3	1	2	8	6	9	4
	6	9	1	0	2	8	5	3	4	7
1st digit input	0	7	5	3	1	2	8	6	9	4
	3	4	7	6	9	1	0	2	8	5
upper array randomized	4	1	5	0	3	6	2	8	9	7
	3	4	7	6	9	1	0	2	8	5
2nd digit input	4	1	5	0	3	6	2	8	9	7
	0	2	8	5	3	4	7	6	9	1
upper array randomized	3	7	4	9	6	1	5	8	0	2
	0	2	8	5	3	4	7	6	9	1
3rd digit input	3	7	4	9	6	1	5	8	0	2
	3	4	7	6	9	1	0	2	8	5
upper array randomized	5	8	4	7	9	3	0	2	6	1
	3	4	7	6	9	1	0	2	8	5
4th digit input & validation	5	8	4	7	9	3	0	2	6	1
	4	7	6	9	1	0	2	8	5	3

The two DAP arrays are initially randomized and displayed to the user when the authentication starts. The reference position is initially the leftmost cell of the upper array as in the previous vulnerable DAP version and contains the digit 0. Once the user releases his finger to input the PIN first digit, the content of the upper array is randomized again. Therefore, the user has to look again for the new position of digit 0 and consider it the reference to enter the next digit. In this case, it is now the fourth cell from the left in the upper array. The same process continues for the remaining digits of the PIN till validation.

When the attacker extracts the four states of the DAP lower array, he will not be able to guess the PIN code as its digits are now scattered randomly in both ways (horizontally and vertically) for every new DAP session, as shown in Table 5. Moreover, comparing the tables extracted from multiple records will not yield any useful information for the same reason.

Table 5. The states of the lower array during a session with the improved DAP version

3	4	7	6	9	1	0	2	8	5
0	2	8	5	3	4	7	6	9	1
3	4	7	6	9	1	0	2	8	5
4	7	6	9	1	0	2	8	5	3

The second solution we propose to solve the DAP vulnerability exposed earlier is to confuse the possible attacker with fake PIN digits that are entered in the ATM in addition to the correct ones. This is inspired by the BrightPass technique reviewed in the related works section. Nevertheless, our new proposal will resist shoulder-surf, camera recording and the intersection of multiple records attacks unlike the BrightPass method.

Our idea is to exploit the multi-touch feature that is nowadays present in most touch pads. Multi-touching means that the user can perform certain gestures using two fingers. For instance, he can move his fingers together or apart (i.e. pinch them) to zoom in or out in a photo or a map. Also, on recent laptops, the user can scroll his screen by moving two fingers in the same direction on the touch-pad. We can easily use this second example with DAP to distinguish between fake digits and real ones.

In more details, we keep the original DAP protocol as is and require that the ATM touch pad supports multi-touching. We also ask the user to put two fingers under the shell of the touch pad up from the session start and keep moving them both for every gesture. Then, to confuse the attacker the user can enter additional random fake digits in his PIN code by touching the pad, hence sliding the lower array randomly, with two fingers instead of one. He can do that without picking a random digit value, but only performing a multi-touch gesture. On the other hand, the correct PIN digits are entered in the normal way by touching the pad with one finger only even if both fingers are moved at the same time. The attacker cannot guess if a gesture was a single touch or a multi-touch one as the finger tips are covered by a shell.

To improve the resilience of this solution we can let the user choose how many fake digits to add to the correct ones, for instance, as many; and when to enter a fake digit and when to enter a correct one. We also recommend to change these two parameters (i.e., the number of fake digits and their order in the whole sequence) in every new authentication session. In this way, an attacker that captures the state of the lower array from one or multiple records cannot guess the PIN code. He will might very get a different number of rows in the extracted tables for every session. Furthermore, even if the correct PIN digits exist in the same column of the extracted table, they will be scattered among other random fake digits in a random order.

Compared to the first solution, the second one will not increase the hardware cost of the ATM as most recent touch pads support multi-touching. It is also cognitively better for the user in the sense that he does not need to look for a new reference position for every digit of the PIN. Furthermore, chasing the reference position will in total take a slightly longer time than a sequence of additional random gestures (assuming that the number of fake digits is lower than or equal to the number of correct ones). So the duration of an authentication session with the second solution is shorter that

the first solution, even if both are longer than the original DAP protocol.

6. CONCLUSION

We started this work by a deeper security analysis of the DAP protocol and found one vulnerability that could disclose the PIN code by using multiple records attack. After a review of the related works and revisit of the DAP protocol, we demonstrated its vulnerability by example. We then proposed two solutions that can make DAP resistant to the intersection of multiple records attack. A brief comparison of the two solutions showed that the second one is better in many aspects.

As a future work, we intend to experiment with the retained solution to strike a good balance between its augmented security and its increased authentication time. A possible improvement would be to alleviate the user from the burden of choosing himself the number and the order of the fake digits added to his PIN. We think this part could be automated without re-exposing the improved version of DAP to the multiple records attack.

REFERENCES

[1] Badra, M., Badra, R.B. (2016). A lightweight security protocol for NFC-based mobile payments. Procedia Computer Science, 83: 705-711. https://doi.org/10.1016/j.procs.2016.04.156

[2] Giese, D., Liu, K., Sun, M., Syed, T., Zhang, L. (2019). Security Analysis of Near-Field Communication (NFC) Payments. ArXiv1 - 04.10623.

[3] 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): 102-111.

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

[5] Wadii, E.L., Boutahar, J., Ghazi, S.E. (2017). NFC technology for contactless payment echosystems. International Journal of Advanced Computer Science and Applications (IJACSA), 8(5): 391-397. https://dx.doi.org/10.14569/IJACSA.2017.080548

[6] El Madhoun, N., Pujolle, G. (2016). Security enhancements in EMV protocol for NFC mobile payment. IEEE Trustcom-16 Conference, Tianjin, China, pp. 1889-1895. https://doi.org/10.1109/TrustCom.2016.0289

[7] Alsuhibany, S.A. (2021). A camouflage text-based password approach for mobile devices against shoulder-surfing attack. Security and Communication Networks, 11 pages. https://doi.org/10.1155/2021/6653076

[8] Kurnaz, S., Mohammed, A.H. (2020). Secure pin authentication in java smart card using honey encryption. IEEE International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), pp. 1-4. http://dx.doi.org/10.1109/HORA49412.2020.9152936

[9] Chen, D., Zhao, Z., Qin, X., Luo, Y., Cao, M., Xu, H., Liu, A. (2020). MAGLeak: A learning-based side-channel attack for password recognition with multiple sensors in IIoT environment. IEEE Transactions on Industrial Informatics, 18(1): 467-476. https://doi.org/10.1109/TII.2020.3045161

[10] Guerar, M., Migliardi, M., Palmieri, F., Verderame, L., Merlo, A. (2020). Securing PIN-based authentication in smartwatches with just two gestures. Concurrency and Computation: Practice and Experience, 32(18): e5549. https://doi.org/10.1002/cpe.5549

[11] Shang, J., Wu, J. (2020). LightDefender: Protecting PIN Input using Ambient Light Sensor. IEEE International Conference on Pervasive Computing and Communications (PerCom), Austin, TX, USA, pp. 1-10. https://doi.org/10.1109/PerCom45495.2020.9127361

[12] Shammee, T.I., Akter, T., Mou, M., Chowdhury, F., Ferdous, M.S. (2020). A systematic literature review of graphical password schemes. Journal of Computing Science and Engineering, 14(4): 163-185. http://dx.doi.org/10.5626/JCSE.2020.14.4.163

[13] Jain, S. (2017). ATM frauds: Detection & prevention. International Journal of Advances in Electronics and Computer Science, 4(10): 82-89.

[14] Shin, H., Sim, S., Kwon, H., Hwang, S., Lee, Y. (2021). A new smart smudge attack using CNN. International Journal of Information Security, 21: 25-36. http://doi.org/10.1007/s10207-021-00540-z

[15] Kobayashi, K., Oguni, T., Nakagawa, M. (2020). A series of PIN/password input methods resilient to shoulder hacking based on cognitive difficulty of tracing multiple key movements. IEICE Transactions on Information and Systems, 103(7): 1623-1632. https://doi.org/10.1587/transinf.2019EDP7181

[16] Yadav, K., Mattas, S., Saini, L., Jindal, P. (2020). Secure card-less ATM transactions. IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA), Kurukshetra, India, pp. 1-4. https://doi.org/10.1109/ICMICA48462.2020.9242713

[17] Chanal, P.M., Kakkasageri, M.S. (2020). Security and privacy in IOT: A survey. Journal of Wireless Personal Communications, 115(3): 1667-1693. https://doi.org/10.1016/B978-0-12-822844-9.00009-8

[18] Smart Payment Association. (2018). Biometrics in Payment: Breaking down barriers with high value payments. https://smartpaymentassociation.com/index.php/news-smart-payment-association/news-smart-payment-association/entry/biometrics-in-payment, accessed on Dec. 7, 2021.

[19] Promontory an IBM Company. (2017). Biometric authentication in payments: Considerations for Policymakers.

[20] GSMA. (2018). NFC Functions and Security Certification overview. https://www.gsma.com/newsroom/resources/14517/attachment/nfc-functions-and-security-certification-overview_v1-0/, accessed on Dec. 7, 2021.

[21] Chabbi, S., Boudour, R., Semchedine, F., Chefrour, D. (2020). Dynamic array PIN: A novel approach to secure NFC electronic payment between ATM and smartphone. Information Security Journal: A Global Perspective, 29(6): 327-340. https://doi.org/10.1080/19393555.2020.1773583

[22] 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. IEEE International Conference on High Performance Computing & Simulation (HPCS), pp. 203-210. https://doi.org/10.1109/HPCSim.2015.7237041

[23] 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

[24] Guerar, M., Benmohammed, M., Alimi, V. (2016). Color wheel pin: Usable and resilient ATM authentication. Journal of High Speed Networks, 22(3): 231-240. http://dx.doi.org/10.3233/JHS-160545