A Novel Approach For Generating One-Time Password With Secure Distribution

Mohamed.H.S.AbouSteit
Computer engineering
Arab Academy for Science,
Technology & Maritime Transport
Cairo, Egypt
m.hazem.shaker@gmail.com

Dr. Ashraf Farouk Tammam
Computer engineering
Arab Academy for Science,
Technology & Maritime Transport
Cairo, Egypt
ashraf.tammam@aast.edu

Dr. AbdelMoneim Wahdan
Computer engineering
Arab Academy for Science,
Technology & Maritime Transport
Cairo, Egypt
Abdelmoniem.wahdan@aast.edu

***Abstract*— Authentication nowadays is an important issue that cannot depend only on static passwords because it is vulnerable and subjected to many types of attacks. Many researchers addressed this problem by introducing a dynamic password framework called a one-time password (OTP). However, this framework still has many limitations. One of these limitations is sending OTP that is done without any encryption. In this paper, we will introduce a new OTP framework that achieves the confidentiality of exchanging the OTP by using a combination of AES-256 bit, RSA, SHA-512. OTP should be delivered to the client in a secure way, as an example is using network traffic to avoid the delay in SMS. In the proposed model the server encrypts OTP using the RSA public-key cryptosystem with the client's public key, in order not to send OTP as a plain text.**

Keywords*— *One-time password, AES, RSA, SHA, SMS based OTP

I. INTRODUCTION

The authentication technique has great importance in the security control process. The most common and traditional technique for authentication is the static passwords, which is one of the simplest and commonly used for authentication phase; however, it is not secure enough because the password always still the same. Static password authentication cannot resist some well-known attacks for example (replay attacks, password leakage, guessing attacks, or exhaustive attacks, eavesdropping, phishing attack, spoofing, and man in the middle attack, Denial-of-service and malware attacks). To overcome the defects of the static password for authentication, OTP was firstly introduced (Lamport, 1981) [1].

The aim of using OTP could be divided into:

1- The generation of OTP should be hard for shoulder surfing attacks and Brute force attacks. So, OTP is recommended to be alphanumeric, case-sensitive, and contains special characters.

2- The process of generation should be resilient to reverse engineering or predicting the next generated OTP.

3- The delivery of the generated OTP should be without delay especially in critical decision systems as online money transfer, and stock markets, ... etc., as the delay even for second cost a lot of money and may lead to a great loss.

4- The generated OTP should be delivered securely without any modification or subjected to any type of attack.

A. One Time Password:

OTP is a password that is effective only for a short period in each session or transaction. OTP overcome the weakness of the traditional password (static password), OTP also is considered as a two-factor authentication that requires something that the user has (cell phone, e-mail, token, etc.).

The most substantial preferred feature for using OTPs rather than static passwords is that they are not powerless against replay attacks and most of the common attacks on traditional passwords for example Brute force attacks, eavesdropping, man in the middle attacks, …etc. Even these types of attacks could not be effective because the main feature of OTP is its short lifetime. That means the password lifetime in some systems is at most 5 minutes but, in many systems, the lifetime is only 1 minute or less than that. This implies if it is possible for an attacker who could figure out how to record an OTP that was used to log in to a system or money transaction won't be able to reuse it again, as it will never be effective again. So, the main target for the attacker is to try to predict what would be the next OTP. So, the main issue for generating OTP is to be random and could not be predicted.

B. OTPs generation and delivery:

OTPs have different methods for a generation:

1- Time based OTP that depends on the time synchronization between the server and the client.

978-1-7281-6823-4/20/$31.00 ©2020 IEEE

2- Mathematical algorithm that generates the new password depending on the last generated password.

3- Mathematical algorithm that generates the new password depending on a challenge predefined between the client and server (Image, secret number ...etc.).

OTPs could be delivered by different methods:

1- Phone (SMS, OTP application)

2- Tokens (RSA token)

3- E-mail

4- Hard copy (are used in online banking where the bank gives to the client a list of printed OTPs (hard copy).

C. SMS based OTP:

SMS based OTP means is the delivery of an OTP via SMS which is most commonly used for the delivery of an OTP because it is an easy way for delivery as it is cheap and handy as there is no need for token only what the client needs to receive it is a cell phone. On the other hand, it is important to deliver OTP securely in order not to be subjected to any type of attack such as stealing or by modification. Many pieces of research concerned with the SMS based OTP and its security. SMS if it is encrypted it would be encrypted using A5/1 which is a very weak encrypting algorithm and easy to be compromised, or sometimes it is sent as a plain text so it could be easily captured by the attacker. So, the security of SMS is important especially in our case of delivery of OTP. Some proposed models encrypt OTP before sending it via SMS, But the main weak point is if they used a good encrypting algorithm, on the other hand, they used a fixed key for encryption or using also a weak encrypting algorithm for the key and send it via SMS. So, if the key is compromised because it is fixed or weakness of the encryption so the OTP would be easy to be captured.

The proposed model aimed to solve the problem of secure generation of OTP, by using random text from a list RSS feeder of CNN and BBC that are updated hourly. This random text is encrypted using AES -256 bits and the key is generated during the generation time of OTP. The output of the encryption process is hashed using SHA 512. By XORing the hashed text would be our OTP (8 bytes).

The problem of secure delivery of the generated OTP to the client without delay especially in critical decision systems was being solved by not sending the generated OTP as a plain text, it would be encrypted using RSA with the client's public key and send it as network traffic, at the client-side the application would decrypt the message by using the client's private key and display it directly to the application. The encrypted OTP would not be sent via SMS because of the delay sometimes in the messaging center or the network coverage. So, the encrypted OTP was sent to the client using network traffic to avoid any delay in the basic SMS system.

II. LITERATURE SURVEY

The related works would be divided into two sections, the first one for the generation of the OTP, and the second section would be for sending the generated OTP via SMS to the client's mobile device.

A. Generation of OTP:

Chang-Seop Park overcame the drawbacks of the finite Lamport OTP that uses the hash chain function and it this technique is finite because it generates a finite number of hash chain function and after that, it needs a re-registration again to continue, but he generated an infinite number of OTP without re-registration.[3]

Dongdong Zhao, Wenjian Luo they generated OTP by using Negative database (NDB) by generating a random seed, this random seed is concatenated with the user password, after that the result of the concatenation is hashed this would be DB and the resulted DB would be an input to function using the same generated random seed to get the NDB.[1]

Yuji Suga proposed a new model for the generation of OTP by using a combination of the hash chain using Merkle tree and L-divided tree and called this model Sausage-style one-time password, but there is no implementation yet for the proposed model.[4] He also proposed the same model but in a different name and still without implementation for the proposed model, [5] he also made more clarification to his proposed model by giving more examples to clarify his idea, but still no implementation yet for his proposed model for testing it.[6]

S Prayla Shyry, M. Mahithaasree, M. Saranya proposed a new model for generating OTP by using a 3*3 Vedic multiplier, firstly the get the client login details from the Certificate Authority and is converted to 8 bits, after that the length of the message is also converted to 8 bits too. These 16bits (2*8 bits) is converted to a decimal using CB2D, the output of the conversion is two 3 decimal digits are multiplied together using 3*3 Vedic multiplier and the result of the multiplication would be OTP, But the generation of the OTP is based on knowing the user credentials (username, password) which is not secure, as the attacker succeeded to get these credential he would be able to create OTP. [7].

H.S. Elganzoury, A.A. Abdelhafez, and A.A. Hegazy proposed a model for generating banking OTP by using three different inputs known by the client in addition to an IV, this IV is assessed using NIST test randomness suite. The client gets an SD card containing IV from the bank and encrypted by AES-256-XTS, this IV consists of 256 bits and is divided to two parts 128 bits ant the client-side and the other 128 is at the server side to add another level of authentication, the factors XOR'ed and after that concatenated together and then hashed using SHA-256 and this will be the OTP, this OTP is encrypted by AES-256-CBC mode to be sent to the client through an HTTPs secure channel to the client's mobile device.[8]

B. Delivering the generated OTP to mobile device:

D. Mahto and D.K. Yadav in their work they focused on the encryption of the generated OTP in order not to send it as a plain text, so the used Elliptic curve cryptography (ECC) for encrypting OTP by using a public key (based on iris code). But in this model, they used a fixed key for encryption that can be attacked.[9]

Ananthi Sheshasaayee and D. Sumathy Proposed a model for the secure transmission of SMS based OTP by using Feistel cipher for encrypting the OTP and used steganography technique for hiding this encrypted OTP in a normal SMS, But the main defect in this technique is using the user's Personal Identification Number (PIN) code and user's Date of Birth (DOB) as a key for encrypting and decrypting the OTP and also a fixed key (stego-key) is used to reveal the OTP from the steganographic SMS, which is subjected to many types of attacks for getting the key[10].

Ramkrishna Das, Sarbajit Manna and Saurabh Dutta in their model they depend on dividing the authentication into two parts, first is the encrypted image using the biometric image of the user using BitWise Masking Alternate Sequence (BWMAS), the original image is selected randomly from image database on the server. In the second part, they generated OTP on many steps, and send the intermediate OTP to the user via E-mail, the user has to decrypt the encrypted image using the biometric image, and the second part he has to generate the final OTP from the sent OTP by using the predefined scheme. The decrypted image and final OTP are used for authentication.[11]

V. Kaveri Reddy and S.J.Saritha proposed a new model for a secure SMS by encrypting the SMS by using symmetric key algorithm AES and the key will be encrypted using Caesar cipher and sent to the user's mobile phone, the main problem is using Caesar cipher for encrypting the key which is easy to be compromised, so the key could be decrypted.[12]

III. Motivation

The One-time password was thought to be secure an invulnerable because of its short lifetime but lately, NIST published research that presented the insecurity of using OTP[13], on the other hand, another research of Positive Technologies Company[14] that also showed that how their employees succeeded to hack Facebook's OTP. Facebook users now are about 2.41 billion monthly active users on Facebook as of June 30, 2019[15].

Sending OTP via e-mail could not be considered a secure level of authentication because of the e-mail attacks as phishing attacks that target the data of the victim such as passwords [16], using e-mail and its password nowadays is not a sufficient proof that the user who received OTP via e-mail is the intended person who should receive it. As the client could leave his e-mail opened on his PC / laptop and during his absence, anybody could access his device to acquire his OTP. So that we exclude e-mail solution from our proposed model.

The SMSs are encrypted during transmission with the A5/1 algorithm and stored as plain text at network operators. Recently discovered that this algorithm is not secure anymore if the network operator is compromised so the attackers could gain access to SMS. [17]

All of these attacks because of the weakness of Signaling System No.7 (SS7) networks that there is a weak encryption A5/1[18].

SMS based OTP is very critical to be sent as clear text or even encrypted by A5/1, which could be subjected to many types of attacks as a man-in-the-middle attack (MITM), Over the air (OTA) modification and many other types of attacks.

Most banking systems use OTP as numbers that are easy for shoulder surfing attacks, But generating a long OTP with special characters and case sensitive makes it hard for these types of attacks.

Encryption of the SMS based OTP using symmetric key encryption because of the limited size of the SMS, and the distribution of the key. That should be dynamic and not sent every time with the SMS.

The delay in SMS due to high traffics which is not suitable for OTP delivery especially in critical decisions like online money transfer and deals in stock markets

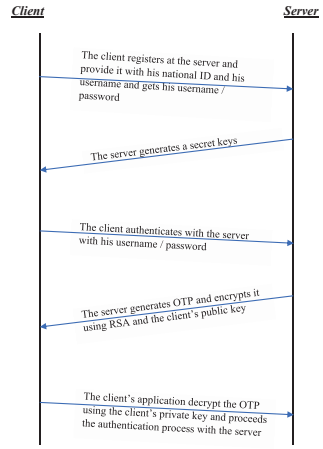

Fig.1 Workflow for the proposed model

IV. Proposed System

In the proposed system we divided the model into phases as follow:

Secret key = *f* (client's National ID + username + time stamp)

A. Server OTP generation algorithm:

Step 1: Getting a random input text that is randomly selected from RSS feeder from daily news sites CNN and BBC.
Step 2: This random text would be encrypted using AES-256bits using the secret key.

Step 3: Applying SHA-512 for the encrypted text in step 2, we will get 512 bits as an output.
Step 4: Divide the previous 512 bits into 8 blocks such that the size of each block is 64 bits.
Step 5: XORing the previous 8 blocks we get 64 bits which are 8 bytes. That represents 8-character OTP.

B. *Authentication process:*

Step 1: The client connects to the server home page to make a transaction using his username/password.
Step 2: The server generates a secret key.
Step 3: The server generates an OTP using the server OTP generation algorithm mentioned above.
Step 4: The output of step 3 would be encrypted using RSA and sent to the client via network traffic using the client's public key.
Step 5: The client's application decrypts the obtained message using his private.
Step 6: The client's application gets OTP in step 5 and displays it in the text box in the bank application.

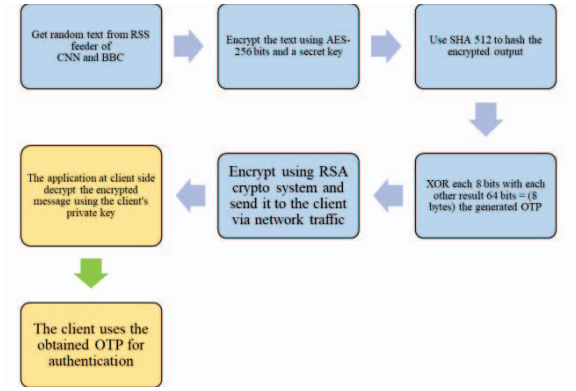

Fig.2 Workflow for the OTP generation and sending

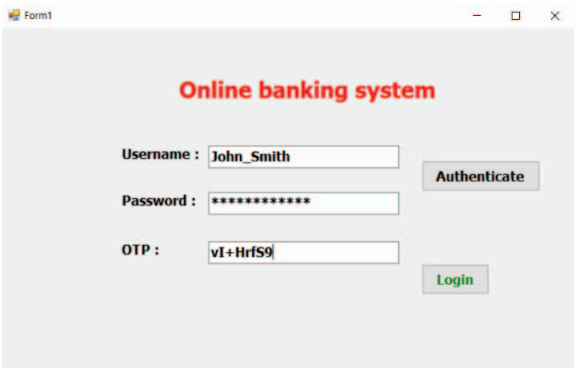

Fig.3 Client's application for authentication with the bank

C. *Abbreviations and Acronyms*

OTP – One-Time Password
SMS – Short Message Service
AES – Advanced Encryption Standard
SHA – Secure Hash Algorithm
RSA - Rivest-Shamir-Adleman cryptosystem
TPM – Trusted Platform Module
OTA – Over the Air
MITM – Man in the Middle
SS7 - Signaling System No.7
IMSI - International Mobile Subscriber Identity

V. PERFORMANCE ANALYSIS

We implemented the proposed model as shown in Table I. As we encrypted the obtained random plain text from the RSS feeder (CNN-BBC) using AES-256 bits. To compare the results of the proposed model with the results of [8], so will use the output of hashing function that will be 512 bits = 64 bytes which are alphanumeric and case sensitive and contains special characters. The proposed model was tested against offline/online guessing attack, impersonation attack, brute force attack, dictionary attack, rainbow attack, a man in the middle attack, birthday attack, key-recovery attack, and collision attack using different online and offline tools. "Online Domain Tools" [19] which concludes the stats shown in Table II regarding the required time to break one OTP. Gibson Research Corporation password checker [20] has been used for checking the OTP strength as well, and the results are shown in Table III, Kaspersky Lab password checker [21], The password meter that assesses the strength of password strings which give the instantaneous visual feedback provides the user a means to improve the strength of their passwords [22] and the results are shown in Table IV.

By comparing the proposed system for secure delivery of the SMS based OTP by providing end-to-end encryption with the other existing models the result is shown in Table V.

The proposed model worked to solve the weak points of the models [10] and [11], as in [9] they used PIN code combined with DOB to be a fixed key, meanwhile [12] used AES for encryption but the key is sent as an SMS but encrypted by weak encrypting algorithm Caesar cipher that could be easily compromised.

TABLE I. IMPLEMENTATION PLATFORM SPECIFICATIONS

Implementation	
Core processor	Intel Core i5 @2.4 GHz
Operating system	Windows 10 pro 64 bit
RAM	8 GB

TABLE II. PASSWORD PROPERTIES (BY ONLINE DOMAIN TOOL)

	Proposed model		[8]	
Property	***Value***	***Comment***	***Value***	***Comment***
Password length:	64	OK	64	OK
Numbers:	9	USED	44	USED
Letters:	52	USED	20	USED
Uppercase Letters:	24	USED	0	NOT USED
Lowercase Letters:	28	USED	20	USED
Symbols	3	USED	0	NOT USED
Charset size	95	HIGH (A-Z, a-z, 0-9, symbols) 26+26+10+33 = 95	36	MEDIUM (0-9, a-z) 10+26 = 36
TOP 10000 password	NO	Password is NOT one of the most frequently used passwords.	NO	Password is NOT one of the most frequently used passwords.
Exact Search Space Size (Count): count of all possible passwords with this alphabet size and up to this password's length)	3,792,333,218,144,715,689,214 ,286,733,276,153,973,356,796, 205,513,695,361,373,369,020,8 59,784,852,595,066,624,036,31 6,469,616,424,830,959,697,422 ,432,772,656,704,517,120	Exact Search Space Size (Count): (count of all possible passwords with this alphabet size and up to this password's length)	65,150,009,139 ,058,228,497,7 66,923,716,720 ,797,655,893,5 03,594,612	Exact Search Space Size (Count): (count of all possible passwords with this alphabet size and up to this password's length)
Search Space Size (as a power of 10)	3.79 x 10126		6.52 x 1049	

TABLE III. TIME REQUIRED TO EXHAUSTIVE SEARCH THIS PASSWORD'S SPACE (GIBSON RESEARCH CORPORATION)

Online Attack Scenario: (Assuming one thousand guesses per second)	1.21 million trillion trillion trillion trillion trillion trillion trillion trillion trillion centuries
Offline Fast Attack Scenario: (Assuming one hundred billion guesses per second)	12.06 billion trillion trillion trillion trillion trillion trillion trillion trillion centuries
Massive Cracking Array Scenario: (Assuming one hundred trillion guesses per second)	12.06 million trillion trillion trillion trillion trillion trillion trillion trillion centuries

TABLE IV. PASSWORD PROPERTIES (BY PASSWORD METER)

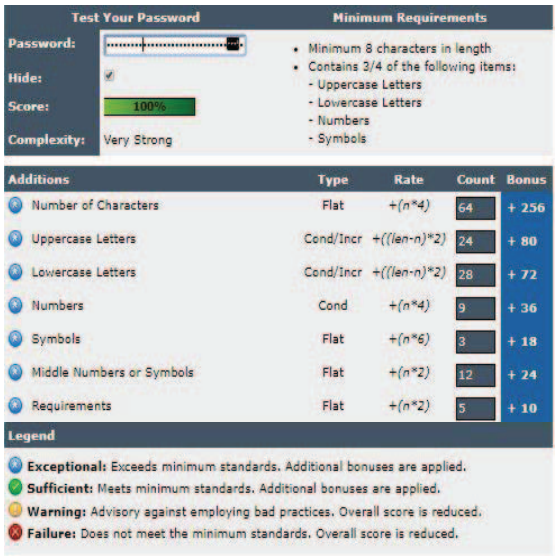

TABLE V. COMPARISON OF EXISTING ALGORITHMS FOR SECURE SMS DELIVERY WITH THE PROPOSED MODEL

Item	**Traditional OTP**	**[10]**	**[9]**	**[11]**	**Proposed method**
End-to-End encryption	No	Yes	Yes	No	Yes
OTP transfer via Internet	Plain text	cipher-stego text	cipher-text	Encrypted image + Intermediate OTP	Encrypted
Cryptography	No	Yes. Light weight Feistal cipher using fixed key	Yes. ECC (Elliptic Curve Cryptography)	Yes. (BWMAS) Bit Wise Masking Alternate Sequence	Yes. AES using dynamic key
Steganography	No	Yes. Text Steganography fixed stego key	No.	No	No
H/W circuitry	Basic h/w	No extra h/w	No extra h/w	No extra h/w	No extra h/w
S/W requirements	Basic s/w	Mobile app to extract and decipher OTP	s/w for OTP retrieval from image	s/w for decrypting image and generate the final OTP	Mobile app to extrac and decipher OTP
Sending media	SMS	SMS	Email	Email	SMS
Levels of authentication	Two	Three	Two	Two	Three

VI. FUTURE WORKS

OTP nowadays become an effective way for fast and easy authentication. The phase of generation of OTP should be fast and at the same time should be hard to be subjected to any attacks, on the other hand, the phase of sending OTP to the client should not be sent as a plain text in order not to be captured easily by any attacker.
Speed up the generation process of OTP because it takes much time compared with the existing systems, but it is not recognized as it will run on servers with high specifications so generation speed would be about the same with the existing generators.

VII. CONCLUSIONS

The main problem of using a static password that is not sufficient for providing authentication to critical systems like online banking, so depending on the (OTP) becomes very important for providing a second level of authentication. So, the main problem is divided into two important issues, the first is the generation of OTP, the second issue is the distribution of the generated OTP. The generated OTP is alphanumeric, case-sensitive, and contains special characters. The proposed model is resilient to different types of attacks for reverse engineering or predicting the next generated OTP because of using random input text encrypted using AES-256 bits and hashed using SHA 256, that is resilient to brute force attacks for cracking the password because of the short lifetime of the password, resilient to dictionary attacks and MITM attacks. The proposed model solved the problem of OTP distribution by using a list of secret keys. This list would be sent to the client using public-key cryptography RSA, the list is updated periodically by the server and send the updated list to the client. Encrypt the generated OTP using AES-256 bits. By using this technique, the MITM attacks also over the air (OTA) modification would be useless.

The proposed system solved the problem of key distribution as the system used symmetric key encryption and the key should be transmitted and should be dynamic. Also, the proposed model solved the problem of security of SMS and it avoided the delay in basic SMS systems by encrypting OTP using RSA before sending it via network traffic, and the decryption process would take place at the client-side.

REFERENCES

[1] D. Zhao and W. Luo, "One-time password authentication scheme based on the negative database," *Eng. Appl. Artif. Intell.*, vol. 62, no. December 2016, pp. 396–404, 2017, DOI: 10.1016/j.engappai.2016.11.009.

[2] E. Conrad, S. Misenar, and J. Feldman, "Domain 3: Security Engineering (Engineering and Management of Security)," *CISSP Study Guide.*, pp. 103–217, 2016, DOI: 10.1016/b978-0-12-802437-9.00004-7.

[3] C. S. Park, "One-time password based on hash chain without shared secret and re-registration," *Comput. Secur.*, vol. 75, pp. 138–146, 2018, DOI: 10.1016/j.cose.2018.02.010.

[4] Y. S. B, "Advances on Broad-Band Wireless Computing, Communication and Applications," vol. 12, pp. 2–10, 2018, DOI: 10.1007/978-3-319-69811-3.

[5] Y. Suga, "An Extended Lamport-Like One-Time Password Scheme and its Applications," *2018 IEEE Int. Conf. Consum. Electron. - Asia, ICCE-Asia 2018*, pp. 206–212, 2018, DOI: 10.1109/ICCE-ASIA.2018.8552134.

[6] Y. S. B, *Advances in Internet, Data & Web Technologies*, vol. 17. Springer International Publishing, 2018.

[7] S. P. Shyry, M. Mahithaasree, and M. Saranya, "Implementation of One Time Password by 3∗3 Vedic Multiplier," in *2nd International Conference on Computer, Communication, and Signal Processing: Special Focus on Technology and Innovation for Smart Environment, ICCCSP 2018*, 2018, no. Icccsp, pp. 1–5, DOI: 10.1109/ICCCSP.2018.8452861.

[8] H. S. Elganzoury, A. A. Abdelhafez, and A. A. Hegazy, "2018 , 35 th NATIONAL RADIO SCIENCE CONFERENCE A New Secure One-Time Password Algorithm for Mobile Applications 2018, 35 th NATIONAL RADIO SCIENCE CONFERENCE," no. Nrsc, pp. 249–257, 2018.

[9] D. Mahto and D. K. Yadav, "Security Improvement of One-Time Password Using Crypto-Biometric Model," in *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics*, 2016, pp. 347–353.

[10] A. Sheshasaayee and D. Sumathy, "A framework to enhance security for OTP SMS in E-banking environment using cryptography and text steganography," in *Advances in Intelligent Systems and Computing*, 2017, vol. 469, pp. 709–717, DOI: 10.1007/978-981-10-1678-3_68.

[11] R. Das, S. Manna, and S. Dutta, "Secure user authentication system using image-based OTP and randomize numeric OTP based on user unique biometric image and digit repositioning scheme," in *Lecture Notes in Electrical Engineering*, 2017, vol. 470, pp. 83–93, DOI: 10.1007/978-981-10-8585-7_8.

[12] V. K. Reddy and S. J. Saritha, "An end to end protocal transmission for secure Ciphertext," in *2017 International Conference on Energy, Communication, Data Analytics, and Soft Computing, ICECDS 2017*, Aug. 2018, pp. 3629–3634, DOI: 10.1109/ICECDS.2017.8390140.

[13] P. A. Grassi, M. E. Garcia, and J. L. Fenton, "NIST 800-63-3: Digital Identity Guidelines," *NIST Spec. Publ.*, p. 68, 2017, DOI: 10.6028/NIST.SP.800-63-3.

[14] P. T. Researchers, "One Time Passcodes Sent via SMS Intercepted and Used to Hack Accounts," *Positive Technologies*, 2016. https://www.ptsecurity.com/ww-en/about/news/one-time-passcodes-sent-via-sms-intercepted-and-used-to-hack-accounts/.

[15] Facebook, "Facebook Q2 2019 Results," 2019, [Online]. Available: https://s21.q4cdn.com/399680738/files/doc_financials/2019/Q2/Q2-2019-Earnings-Presentation-07.24.2019.pdf.

[16] M. Jakobsson, "Two-factor inauthentication – the rise in SMS phishing attacks," Comput. Fraud Secur., vol. 2018, no. 6, pp. 6–8, 2018, DOI: https://doi.org/10.1016/S1361-3723(18)30052-6.

[17] M. Noman Riaz and A. Ikram, "Development of a Secure SMS Application using Advanced Encryption Standard (AES) on Android Platform," *Int. J. Math. Sci. Comput.*, vol. 4, no. 2, pp. 34–48, 2018, DOI: 10.5815/ijmsc.2018.02.04.

[18] Z. Li, "Optimization of Rainbow Tables for Practically Cracking GSM A5/1 Based on Validated Success Rate Modeling," in *Topics in Cryptology - CT-RSA 2016*, 2016, pp. 359–377.

[19] Online-Domain-Tools.com,"Online-Domain-Tools.com." http://password-checker.online-domain-tools.com/.

[20] G. R. Corporation, "Gibson Research Corporation." https://www.grc.com/haystack.htm.

[21] Kaspersky, "Kaspersky: Secure Password Check." https://password.kaspersky.com/.

[22] T. P. Meter, "The Password Meter.".