[1]Amanapu Yaswanth

[2]K. Thammi Reddy

Vulnerabilities in One Time Passwords and Protection Method Using Secret Key and Biometric Hash Code

Journal of Electrical Systems

Abstract: **-** In the present era, conducting financial transactions online is both simple and widely embraced for its convenience. However, engaging in internet-based financial transactions exposes individuals to potential threats like password attacks, malware, phishing, and other illicit activities. Recognizing these risks, many banks have enhanced security measures by incorporating One-Time Password (OTP) authentication alongside traditional login credentials. Despite these improvements, security issues persist in online transactions. To mitigate such concerns, OTPs can be delivered via SMS to the account owner's mobile number. Even with heightened security, internet banking remains susceptible to various attacks, including online phishing, Man-in-the-Middle attacks, SQL injection, and brute force attacks. To address these challenges, this proposed model introduces a Biometric mobile OTP-based transaction authentication method.

Keywords: OTP, biometric, hash code, password assaults, malware, phishing.

I. INTRODUCTION

E-banking, or online banking, facilitates financial transactions over the internet [5]. Major banks currently offer diverse online banking services to their clients, reducing the necessity for in-person visits, increasing cost-efficiency, and enhancing convenience [17]. Online banking enables users to check balances, make deposits and withdrawals, and pay bills from anywhere, as highlighted by Bhatt [5]. In our research, the primary focus is on the first two factors: what the user knows and possesses. Specifically, we concentrate on single-factor authentication, involving a username and password, as opposed to biometric-based techniques. Various combinations of something the user possesses (e.g., a card) and something they know (e.g., a PIN or password) are referred to as two-factor or multifactor authentication, which is known for being more secure and challenging to breach than single-factor authentication [7]. Banks are actively exploring new measures to enhance security and deter attacks, especially given the evolving landscape of online transactions [6]. Additionally, SMS OTP is susceptible to phishing scams online. Attackers may compel victims to log in to a fraudulent website that mimics their bank's site. This malicious site captures the victim's login credentials and requests the OTP. Meanwhile, the attacker gains access to the victim's bank account, conducts transactions, and awaits the OTP sent to the victim's phone. The victim unknowingly shares the OTP with the attacker, who then uses it to verify the transaction with the bank, making the attack successful.

II. RELATED WORK

Enhancing the security and standards of online banking is essential to protect user information from unauthorized access and promote customer retention. The initial and crucial measure in safeguarding data is authentication. While passwords and usernames represent the most commonly employed method of authentication [3], relying solely on single-factor authentication, which solely necessitates a password and login, is deemed inadequate. To strengthen password authentication, a partial password approach can be employed. Instead of the conventional password, this approach involves the system requesting random characters from the password [25].

Several authentication systems have incorporated a more recent form of verification using images, either concealed or identifiable, in lieu of traditional passwords [11]. Passwords and PINs are susceptible to various advanced threats, including brute force, dictionary attacks, and password guessing attempts. Attackers attempt to gain unauthorized access by attempting to discover the login credentials. Nevertheless, the presence of key loggers poses a continuous risk as they can record every keystroke a user makes on their keyboard [10]. It's important to note that advanced

[1,2]Department of Computer Science and Engineering, GITAM School of Technology

GITAM (Deemed to be University), Andhra Pradesh, Visakhapatnam, India,

E-mail: 1yamanapu@gitam.in.

key loggers can even capture the victim's screen [24]. However, not all types of attacks require highly specialized skills; some, such as social engineering and phishing, rely more on the attacker's manipulation abilities.

In the realm of authentication, the literature generally acknowledges and employs three widely accepted techniques. As mentioned earlier, biometrics is employed in various aspects of an individual's life, encompassing their password, ATM card, and physical characteristics. Brainard et al. (2006) [17] describe a modern and secure approach that heavily relies on third-party verification. An effective method to enhance security is the adoption of two-factor authentication. This can be accomplished by combining elements mentioned earlier, such as pairing a password with biometrics or a One-Time Password (OTP). Two-factor authentication, such as the combination of an ATM card and a PIN, is commonly utilized in ATMs. Passwords alone are recognized as one of the most vulnerable targets for hackers.

Biometrics are highly secure but have limited use in online transactions and ATMs due to the high equipment costs. Instead, banks and corporations often opt for tokens as a form of two-factor authentication. Users receive services through a security token, typically a physical item also known as a cryptographic token used for authentication. There are two primary types of tokens: software and hardware tokens. Hardware tokens are compact, portable devices that can be easily carried. Some tokens display a constantly changing PIN, while others incorporate cryptographic keys, hashes, or biometric data. When a client or user needs to authenticate themselves at a specific moment, they use the token's PIN rather than their daily account credentials.

tokens generated by software programs are dynamic and create One-Time Passwords (OTPs). These OTPs are crucial for securing critical systems because unauthorized users or attackers cannot predict or deduce the next password or OTP in the sequence. The series of OTPs must be as random, unpredictable, and unrepeatable as possible. Elements like names, seeds, timestamps, random numbers, and more can be employed to generate these OTPs.

In a 2009 US patent by Bemmel and Mian [18], a system is detailed that employs biometric identification to authorize payments made by consumers using their mobile phones at point-of-sale counters. According to Aloul, F. et al. (2009) [19], two-factor authentication, which includes a biometric identity element, provides enhanced security levels for mobile-based financial transactions when compared to traditional username and password login methods. The system generates One-Time Passwords (OTPs) that can be utilized for various services, including ATM transactions, online banking, and mobile banking. Their approach, based on evidence, demonstrated the ability to accurately recognize and identify users. De Marsico et al. (2014) [24] introduced an innovative method for biometrics in mobile interactions, utilizing face and iris recognition. They developed a multimodal biometric technology called "FIRME," specifically tailored for Android-powered mobile devices. Face and iris recognition are separate modules operating independently until they are combined. They claim that this multimodal authentication method effectively confirms the user's identity.

Kumar, D., and Ryu, Y. (2009) [25] conducted research indicating that biometric payment systems do not require memorization, in contrast to usernames and passwords. They argue that small enterprises can benefit from biometric readers' costs as long as there is a maximum number of customers utilizing the biometric model [26–27]. Yoo, J. H. et al. (December 2007) [28] introduced a novel embedded biometric model that combines face, fingerprint, or iris fingerprint for individual authentication, distinguishing it from existing embedded systems of that time. This new system addressed issues related to low memory and processing power in existing embedded systems. Their implementation showed that face, iris, and fingerprint authentication had similar execution times and error rates of 1.68%, 1.50%, and 4.53%, respectively. In June 2009, Xi, K., and Hu, J. introduced a novel biometric fingerprint model [29] that relies on multiple or composite characteristics, offering effectiveness, reliability, distortion tolerance, and no registration requirements. They validated their findings using a public database, demonstrating that this new structure could significantly enhance verification performance.

III. Main Vulnerabilities With SMS OTP

An attacker’s sole objective is aim is to acquire One-Time Passwords (OTP), and there are several methods through which this objective can be achieved, including wireless interception, mobile phone Trojans, and SIM card Swap Attacks. Here is a concise breakdown of these attacks:

A. Wireless interception

A wireless attack can be accomplished by connecting the unauthorized device to the wireless network and bypassing security measures. People who want to improve their cellular signal can purchase femtocells from their carriers [5]. It is possible to exploit femtocells for the purpose of recording various user activities, including SMS. This could result in the recording of all voice calls, interception of incoming SMS and MMS, and researchers being able to eavesdrop on the users [6]. The use of weak encryption techniques and absence of mutual authentication make GSM technology for sending SMS messages to the intended receiver results in security vulnerabilities. Only the air portion of communication is private when using an A5/1 encryption technique. In 1998, the encryption on the broadcast portion was compromised [4]. An additional study demonstrates that the connectivity between base stations and mobile phones can eavesdrop on and decrypted using protocol weakness [2].

B. Mobile Phone Trojans

The development of mobile phone Trojans specifically designed to intercept SMS messages is becoming an increasing threat. The first known piece of malware designed exclusively for intercepting mTANs is the ZITMO (Zeus in The Mobile) Trojan for Symbian OS [2]. ZITMO is a Trojan virus designed to intercept one-time passwords (OTP) that banks send via SMS. This virus has a specific target and its main purpose is to allow malicious users, or a server in the case of ZITMO for Android, to obtain incoming text messages that contain mTAN codes. These codes can be used to perform financial transactions using bank accounts that have been compromised [7]. ZeuS for Windows Mobile was discovered in February 2011 and named Trojan-Spy. WinCE.Zbot.a [2] is a. The main distinguishing aspect of ZITMO is its “partnership” with the well-known PC-based ZeuS Trojan [7]. Without the latter, ZITMO is just text message forwarding spyware. Cybercriminals can get beyond the mTAN security mechanisms employed in online banking because of the two components’ “teamwork” [7]. Users are tricked into installing the malicious program by ZeuS-infected PCs by telling them that activating their phone is necessary for additional security precautions. When the victim enters his phone number, a text message with a link to the infected application is sent to the phone [8]. Users install malicious software known as SMS OTP Trojans [2]. This software employs social engineering to trick the user into installing the malware rather than taking advantage of the afflicted platform’s security flaw.

C. SIM card Swap Attacks

SIM card Swapping is one of the most recent frauds in the second stage of a phishing scam. Through the use of phishing, a criminal can obtain a victim's fundamental personal information, which can then be used to intercept private communications such as conversations and texts. A SIM switch attack is a spear phishing attack [10] in which a criminal utilizes social engineering tactics are employed to persuade the owner of the mobile phone to transfer the victim's mobile number to a SIM card controlled by the criminal. When the victim’s phone is supplied with one-time banking passwords, the thief begins to receive incoming calls and texts. The offender can then carry out transactions using the personal information they have obtained via keylogging software or phishing. The fraudster receives an OTP sent by the bank through SMS and completes the transaction’s authorization. SIM Swap fraud has been documented in South Africa, where SMS-delivered TAN numbers are typical [9]. To prevent phishing assaults and Trojans like key loggers, banks have put safeguards like one-time passwords sent to the intended user through SMS. Since the operator wants to provide prompt and efficient customer service (and maintain revenue streams), SIM swap fraud is a relatively simple fraud channel for the determined fraudster [10]. The cost of SIM Swap fraud is increasing for the banks. Customers’ confidence in mobile devices as general banking and payment tools and a method for acquiring relatively basic security codes could be damaged [10].

D. Other challenges in SMS OTP

One of the primary drawbacks of the conventional approach is the SMS transmission delay [11]. When the roaming service is off, the bank cannot deliver the SMS-OTP, which stops the user from continuing any additional activities. When compared to bank transaction statistics, SMS costs are higher. Customers cannot finish a transaction that has been authorized due to network coverage issues.”.

E. Maintaining the Integrity of the Specifications

To ensure secure authentication, it is essential. This can be accomplished by preventing replay and tampering attacks by combining timestamping, hashing, and encryption. By verifying and certifying OTPs, the use of digital signatures, secure transmission channels, and message authentication codes (MAC) reinforces security even more.

IV. PROCESS MODEL

The proposed system utilizes a random selection process among a provided set of algorithms for the generation of a secret key. Once an algorithm is randomly chosen, the particular algorithm is used to generate the secret key. Subsequently, users are prompted to input the desired size of the secret key in the range of 4 to 6 digits which are denoted as 'n'. The system then displays the first 'n' digits on the authorized device. Upon user input, the displayed secret key is entered on the device where the transaction is initiated. Validation of the entered secret key is performed if any invalid input entered results in the transaction being terminated. If valid input, the system proceeds to perform summation on the digits of the secret key until the length of the key is 1. The resulting summated value is then displayed on the user device and the user is prompted to enter the "summation key Bio value" where the user already stored his finger prints in hash code which is encrypted format and validates authentication. Upon wrong entry of Biometric entry transaction becomes failed and transaction becomes valid for successful entry of biometric value.

A. Process of Generating Secret Key

A random algorithm is chosen from the provided algorithms, which are used to create a secret key. By using the selected algorithm, a secret key is generated. Prompt the user to input the size of the secret key to be displayed, and assign the given value to 'n'. Display the first n digits of the secret key generated earlier. Prompt the user to enter the displayed secret key on the device where the transaction is being performed from the authorized device, where the secret key will be displayed in a form of a popup. Validate the entered secret key. If invalid, stop the transaction, else perform the next steps. Perform summation of the digits of the secret key until its length becomes 1 digit. Display the summated value.

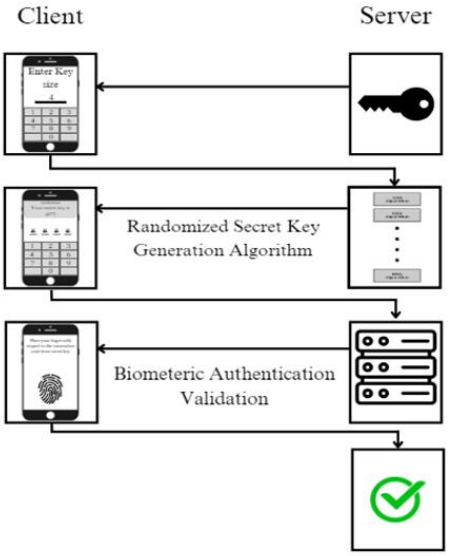

Fig 1: System Architecture

Algorithm:

crypto_algorithms = [list of crypto algorithms]

selected_algorithm = randomly select algorithm from crypto_algorithms

if selected_algorithm == algorithm_1:

```
secret_key = generate_secret_key_using_algorithm_1()
elif selected_algorithm == algorithm_2:
secret_key = generate_secret_key_using_algorithm_2()
...
# Repeat for each algorithm
# Use the secret key for cryptographic operations
Secret key generation
length = get_length_from_user()  # Ask the user for the length of the secret key
# Generate the secret key using the selected algorithm
secret_key = generate_secret_key_using(selected_algorithm)
# Divide the secret key into segments of the specified length
segments = divide_secret_key(secret_key, length)
# Store the segments of the secret key in the server
store_segments_in_server(segments)
```

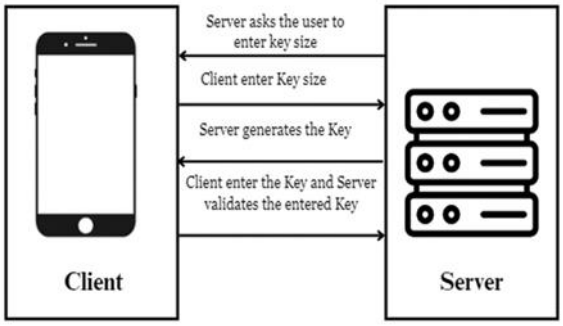

Fig 2: Block Diagram for Secret Key Generation

B. Process of Storing Bio Metric values with Hash Code Generation algorithm by using Haversine Distance:

This procedure outlines the step-by-step process for creating a hash code using a binary fingerprint image and a Haversine distance matrix. The algorithm's steps are described below. The pseudo code is also displayed by the algorithm.

Step 1. Read the fingerprint in grayscale from the input (input image)

Step 2. re_size image= image re_size (input, [256, 256] image) transforms the input image into a two-dimensional image with the dimensions 256 256.

Step 3. grayscale image of size 256 by 256 into a binary_image, image re_size = convert into binary (binary _image)

Step 4. Perform one's complement for binary_image's. This is done to obtain a new binary_ image, which is the

inverse of the original image.

Step 5. Calculate the image's Haversine distance. distance(binary_image) = Haversine image

Step 6. Discover the Haversine distance's distinct value using the formula distinct value (Haversine _image)

Step 7. Locate the unique value summation.

Step 8. size (distinct_ Haversine_value) for i=1 Haversine sum = Haversine _value is a separate value (i)

Step 9. stop for

Step 10. Identify the distinct's mean. Value of Haversine Mean (distinct_ Haversine value) = Haversine mean

Step 11. Find the distinct Haversine value's standard deviation using the formula: std deviation = standard deviation (distinct_ Haversine value)

Step 12. Summation of values from Steps 7 through 9 together.

Step 13. Combining Haversine sum, Haversine mean, and Standard Deviation yields the value combine.

Step 14. For the MD5 Hash function, provide the value from Step 10 as a parameter: hash value = MD5 DataHash(combine value).

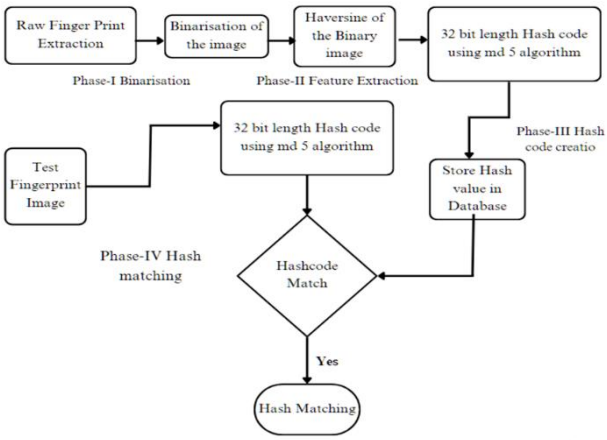

Fig 3: Fingerprint Hash code generation methodology

Maximum of work processes are listed below. Here MD5 algorithm receives the combined mean and standard deviation to strengthen and benefit from salting Haversine distance.

256 X 256 grayscale picture creation from the input image

Step 1. picture conversion to binary

Step 2. Finding a binary image's complement

Step 3. Haversine distance calculation

Step 4. determining the Haversine specific value

Step 5. calculating the total of the individual Haversine

Step 6. calculating the average of the unique Haversine

Step 7. calculating the distinct Haversine standard deviation

Step 8. using the total sum, mean, and standard deviation of individual Haversine values to create an MD5 hash code

The procedure for MD5 algorithm is described below.

Input: Extracted image Features

Output: Hash Code

Add cushioned pieces to the original input's length to the outcome of starting one before. Initialize the MD buffer with the values A, B, C, and D. The message digest was assessed using buffer. A, B, C, and D are all 32-bit registers. Message processing in 16-word chunks. Finally, we receive the output of the 32-bit hash code.

Figure 4: Fingerprint Feature Extraction Process at User Side

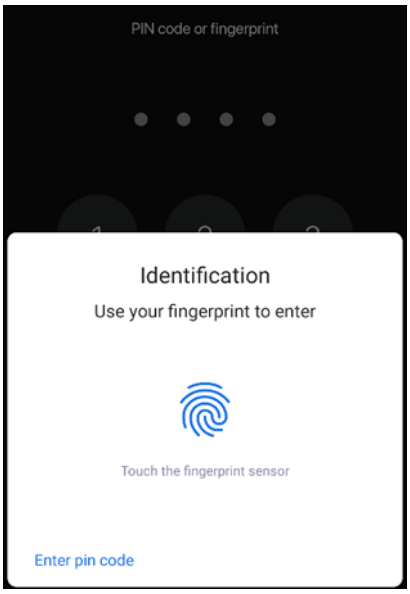

Fig 5: Screenshot of giving biometrics

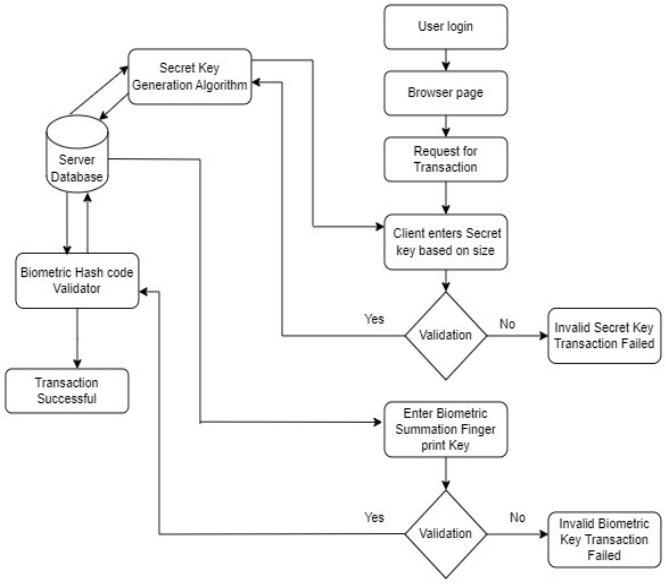

Fig 7: Dataflow Diagram of Proposed Multifactor Authentication

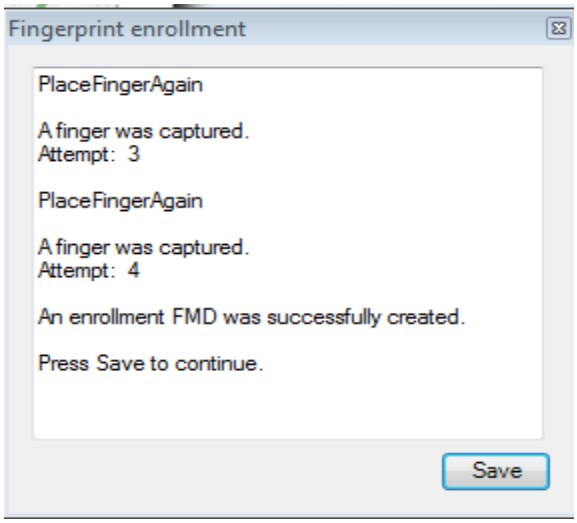

Fig 6: Status of fingerprint Process at server side.

V. Results And Discussions

A random algorithm is chosen from the provided algorithms, which are used to create a secret key. By using the selected algorithm, a secret key is generated. Prompt the user to input the size of the secret key to be displayed, and assign the given value to ‘n’. Display the first n digits of the secret key generated earlier. Prompt the user to enter the displayed secret key on the device where the transaction is being performed from the authorized device, where the secret key will be displayed in a form of a popup. Validate the entered secret key. If invalid, stop the transaction, else perform the next steps. Perform summation of the digits of the secret key until its length becomes 1 digit. Display the summated value. User is then prompted to input a biometric key, corresponding to the "displayed value”. Validation of the entered bio-key occurs against the stored hash code on the server of the bio-key is done. If the input does not match, the transaction is cancelled.

Algorithm for the proposed model

crypto_algorithms = [list of crypto algorithms]

selected_algorithm = randomly select algorithm from crypto_algorithms

```
if selected_algorithm == algorithm_1:
    secret_key = generate_secret_key_using_algorithm_1()
elif selected_algorithm == algorithm_2:
    secret_key = generate_secret_key_using_algorithm_2()
# Repeat for each transaction with choosing random algorithm
# Use the secret key for cryptographic operations
def get_length_from_user():
    # Prompt the user to input the desired length for the secret key
    length = input("Enter the length for the secret key: ")
    return int(length)
def generate_secret_key_using(selected_algorithm):
    # Generate the secret key using the selected algorithm
    # This function will depend on the specific algorithm being used
    # You need to implement generate_secret_key_using_algorithm_1(),
    # generate_secret_key_using_algorithm_2(), etc.
    secret_key = selected_algorithm.generate_secret_key()
    return secret_key
def divide_secret_key(secret_key, length):
    # Divide the secret key into segments of the specified length
    segments = [secret_key[i:i+length] for i in range(0, len(secret_key), length)]
    return segments
def store_segments_in_server(segments):
    # Store the segments of the secret key in the server
    # You'll need to implement how to store segments in the server
    # This could involve sending the segments to a server-side database or storage system
    pass
def validate_bio_key(bio_key, stored_bio_key):
    # Validate the bio key against the stored 'nth' bio key in the server
    # Return True if the bio key matches, False otherwise
    return bio_key == stored_bio_key
```

Fig 8.1 User Authentication

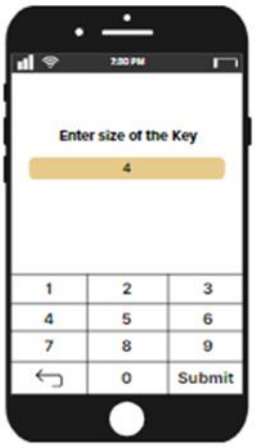

Fig 8.2 Secret Key Generation

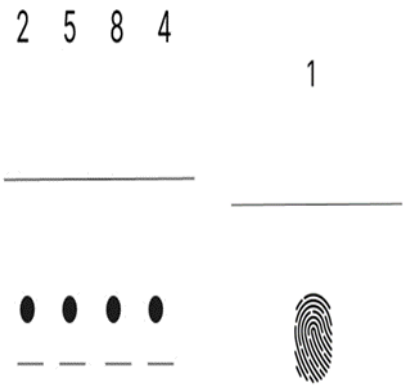

Fig 8.3 Secret Key Validation Fig 8.4 Biometric Authentication

Fig 8.5 Transaction Successful

VI. CONCLUSION

Fingerprint authentication frameworks have produced low false acceptance and denial rates and other advantageous conditions like a straightforward usage technique. If passwords are compromised or stolen, fingerprints are the next best thing because they can be revoked using a different password. However, a static biometric like a fingerprint that doesn't change significantly over time is difficult to change in a biometric security system that uses biometric features. This study covered the topic of creating fingerprint hash codes using Haversine. The fingerprint hash code utilized in the multifactor authentication approach serves as an identity-key or index-key to uniquely identify particular people. Combining a Secret key and fingerprint hash code, creates a solid and highly secure authentication mechanism. The fingerprint picture is sufficiently salted and hashed through the double-folded layer. The multifactor authentication model utilized in this study can be implemented in smart gadgets in mobile as well as in computer for secure transactions. This study's client-server architecture required by the multifactor authentication model makes it suitable for independent systems.

REFERENCES

[1] S. Babkin and A. Epishkina, "Authentication Protocols Based on One-Time Passwords," 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Saint Petersburg and Moscow, Russia, 2019, pp. 1794-1798.

[2] S. Roy, M. Rutherford and C. H. Crawshaw, "Towards designing and implementing a secure one time password (OTP) authentication system," 2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC), Las Vegas, NV, USA, 2016, pp. 1-2.

[3] H. S. Elganzoury, A. A. Abdelhafez and A. A. Hegazy, "A new secure one-time password algorithm for mobile applications," 2018 35th National Radio Science Conference (NRSC), Cairo, 2018, pp. 249-257.

[4] N. Sukma and R. Chokngamwong, "One time key Issuing for Verification and Detecting Caller ID Spoofing Attacks," 2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), NakhonSiThammarat, Thailand, 2017, pp. 1-4.

[5] S. Ma et al., "Fine with "1234"? An Analysis of SMS One-Time Password Randomness in Android Apps," 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), Madrid, ES, 2021, pp. 1671-1682.

[6] K. Sharma, N. Baghel and S. Agarwal, "Multiple Degree Authentication in Sensible Homes basedon IoT Device Vulnerability," 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, India, 2020, pp. 539-543.

[7] R. Shibayama and H. Kikuchi, "Vulnerability Exploiting SMS Push Notifications," 2021 16th Asia Joint Conference on Information Security (AsiaJCIS), Seoul, Korea, Republic of, 2021, pp. 23-30.

[8] N. sukma and R. Chokngamwong, "Increasing the efficiency of One-time key Issuing for The First Verification Caller ID Spoofing Attacks," 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE), Nakhonpathom, Thailand, 2018, pp. 1-6.

[9] K. T. Kumar, N. H. Kumar, A. Ittadi, M. Akila and K. Bhanushree, "Secure strategic mail application with hardware device," 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India, 2016, pp. 887-892.

[10] Mandalapu, Daffney Deepa V, L. D. Raj and Anish Dev J, "An NFC featured three level authentication system for tenable transaction and abridgment of ATM card blocking intricacies," 2015 International Conference and Workshop on Computing and Communication (IEMCON), Vancouver, BC, Canada, 2015, pp. 1-6.

[11] L. J. Hong Sim, S. Q. Ren, S. L. Keoh and K. M. Mi Aung, "A cloud authentication protocol using One-Time Pad," 2016 IEEE Region 10 Conference (TENCON), Singapore, 2016, pp. 2513-2516.

[12] S. M. Abdullahi, H. Wang and T. Li, "Fractal Coding-Based Robust and Alignment-Free Fingerprint Image Hashing," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 2587-2601, 2020.

[13] D. Zhong, H. Shao and X. Du, "A Hand-Based Multi-Biometrics via Deep Hashing Network and Biometric Graph Matching," in IEEE Transactions on Information Forensics and Security, vol. 14, no. 12, pp. 3140-3150, Dec. 2019.

[14] A. Singh, P. Gaurav, C. Vashist, A. Nigam and R. P. Yadav, "IHashNet: Iris Hashing Network based on efficient multi-index hashing," 2020 IEEE International Joint Conference on Biometrics (IJCB), Houston, TX, USA, 2020, pp. 1-9.

[15] X. Chen, M. Yu, F. Yue and B. Li, "Orientation Field Code Hashing: A Novel Method for Fast Palmprint Identification," in IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 5, pp. 1038-1051, May 2021.

[16] Y. Li, H. Zhao, Z. Cao, E. Liu and L. Pang, "Compact and Cancelable Fingerprint Binary Codes Generation via One Permutation Hashing," in IEEE Signal Processing Letters, vol. 28, pp. 738-742, 2021.

[17] [17] S. Shi, J. Cui, X. Zhang, Y. Liu, J. Gao and Y. Wang, "Fingerprint Recognition Strategies Based on a Fuzzy Commitment for Cloud-Assisted IoT: A Minutiae-Based Sector Coding Approach," in IEEE Access, vol. 7, pp. 44803-44812, 2019.

[18] [18] Y. Zhang, Y. -b. Huang, D. -h. Chen and Q. -y. Zhang, "Long Sequence Biohashing Speech Authentication Based on Biometric Fusion and Modified Logistic Measurement Matrix," 2021 International Conference on Computer Engineering and Application (ICCEA), Kunming, China, 2021, pp. 426-434.

[19] Y. K. Jang and N. I. Cho, "Deep Face Image Retrieval for Cancelable Biometric Authentication," 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 2019, pp. 1-8.

[20] Y. Li, L. Pang, H. Zhao, Z. Cao, E. Liu and J. Tian, "Indexing-Min-Max Hashing: Relaxing the Security-Performance Tradeoff for Cancelable Fingerprint Templates," in IEEE Transactions on Systems, Man, and Cybernetics: Systems.

[21] X. Wang and H. Li, "One-Factor Cancellable Palmprint Recognition Scheme Based on OIOM and Minimum Signature Hash," in IEEE Access, vol. 7, pp. 131338-131354, 2019.

[22] X. Dong, K. Wong, Z. Jin and J. -l. Dugelay, "A Cancellable Face Template Scheme Based on Nonlinear Multi-Dimension Spectral Hashing," 2019 7th International Workshop on Biometrics and Forensics (IWBF), Cancun, Mexico, 2019, pp. 1-6.

[23] A. N. Carey and J. Zhan, "A Cancelable Multi-Modal Biometric Based Encryption Scheme for Medical Images," 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 2020, pp. 3711-3720.

[24] V. Talreja, M. C. Valenti and N. M. Nasrabadi, "Deep Hashing for Secure Multimodal Biometrics," in IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1306-1321, 2021.

[25] Y. -L. Lai, J. Y. Hwang, Z. Jin, S. Kim, S. Cho and A. B. J. Teoh, "Secure Secret Sharing Enabled b-band Mini Vaults Bio-Cryptosystem for Vectorial Biometrics," in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 1, pp. 58-71, 1 Jan.-Feb. 2021.

[26] S. K. Jami, S. R. Chalamala and A. K. Jindal, "Biometric Template Protection Through Adversarial Learning," 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2019, pp. 1-6.

[27] D. Sadhya, Z. Akhtar and D. Dasgupta, "A Locality Sensitive Hashing Based Approach for Generating Cancelable Fingerprints Templates," 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS), Tampa, FL, USA, 2019, pp. 1-9.

[28] X. Dong, M. K. Khan, L. Leng and A. B. J. Teoh, "Co-Learning to Hash Palm Biometrics for Flexible IoT Deployment," in IEEE Internet of Things Journal, 2022.

[29] D. Sadhya and B. Raman, "Generation of Cancelable Iris Templates via Randomized Bit Sampling," in IEEE Transactions on Information Forensics and Security, vol. 14, no. 11, pp. 2972-2986, Nov. 2019.

[30] H. Wang, X. Dong, Z. Jin, A. B. J. Teoh and M. Tistarelli, "Interpretable security analysis of cancellable biometrics using constrained-optimized similarity-based attack," 2021 IEEE Winter Conference on Applications of Computer Vision Workshops (WACVW), Waikola, HI, USA, 2021, pp. 70-77.

[31] X. Duan and B. Niu, "A change password attack resistant scheme for remote user authentication using smart card," 2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS), Chongqing, China, 2016, pp. 269-272.

[32] X. Zhan, H. Guo, X. He, Z. Liu and H. Chen, "Authentication Algorithm and Techniques Under Edge Computing in Smart Grids," 2019 IEEE International Conference on Energy Internet (ICEI), Nanjing, China, 2019, pp. 191-195.

[33] S. Monfared, D. Andrade, L. Rodrigues and J. N. Silva, "BioALeg - Enabling Biometric Authentication in Legacy Web Sites," 2016 IEEE 35th Symposium on Reliable Distributed Systems Workshops (SRDSW), Budapest, Hungary, 2016, pp. 25-30.

[34] J. Thomas and R. H. Goudar, "Multilevel Authentication using QR code based watermarking with mobile OTP and Hadamard transformation," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 2018, pp. 2421-2425.

[35] D. K. Purwar, D. Vishwakarma, N. Singh and V. Khemchandani, "One v/s All SVM Implementation for Keystroke based Authentication System," 2019 4th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2019, pp. 268-272.

[36] K. Zhang, M. Spanghero and P. Papadimitratos, "Protecting GNSS-based Services using Time Offset Validation," 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), Portland, OR, USA, 2020, pp. 575-583.

[37] K. Sasa and H. Kikuchi, "Impact Assessment of Password Reset PRMitM Attack with Two-Factor Authentication," 2018 IEEE Conference on Dependable and Secure Computing (DSC), Kaohsiung, Taiwan, 2018, pp. 1-8.

[38] S. Salamatian, W. Huleihel, A. Beirami, A. Cohen and M. Médard, "Why Botnets Work: Distributed Brute-Force Attacks Need No Synchronization," in IEEE Transactions on Information Forensics and Security, vol. 14, no. 9, pp. 2288-2299, Sept. 2019.

[39] Z. Ahmed, I. Nadir, H. Mahmood, A. Hammad Akbar and G. Asadullah Shah, "Identifying Mirai-Exploitable Vulnerabilities in IoT Firmware Through Static Analysis," 2020 International Conference on Cyber Warfare and Security (ICCWS), Islamabad, Pakistan, 2020, pp. 1-5.

[40] Poniszewska-Marańda, Ł. Rogoziński and W. Marańda, "Security Library for Safe Data Storage on Android Platform," 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Shenyang, China, 2021, pp. 309-316.

[41] Zhang and L. Chen, "OTP_SAM: DHCP security authentication model based on OTP," 2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Nanchang, China, 2016, pp. 346-350.

[42] S. Desai and D. P. Gaikwad, "Real time hybrid intrusion detection system using signature matching algorithm and fuzzy-GA," 2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT), Pune, India, 2016, pp. 291-294.

[43] N. Chakraborty, J. Li, S. Mondal, F. Chen and Y. Pan, "On Overcoming the Identified Limitations of a Usable PIN Entry Method," in IEEE Access, vol. 7, pp. 124366-124378, 2019.

[44] Le, A. M. Grande, A. Carmine, J. Thompson and T. Khan Mohd, "Analysis of Various Vulnerabilities in the Raspbian Operating System and Solutions," 2022 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 2022, pp. 01-06

[45] Yaswanth, A. ., & Reddy, K. T. . (2023). A Novel Dynamic Randomized Secret Key Model Based on One-Time Password Authentication . International Journal of Intelligent Systems and Applications in Engineering, 11(3), 850–858.