




Article

# A Secure and Efficient Multi-Factor Authentication Algorithm for Mobile Money Applications

Guma Ali <sup>1,\*</sup> , Mussa Ally Dida <sup>1</sup>  and Anael Elikana Sam <sup>2</sup> 

<sup>1</sup> Department of Information Technology Development and Management (ITDM), Nelson Mandela African Institution of Science and Technology (NM-AIST), Arusha 447, Tanzania; mussa.ally@nm-aist.ac.tz

<sup>2</sup> Department of Communication Science and Engineering (CoSE), Nelson Mandela African Institution of Science and Technology (NM-AIST), Arusha 447, Tanzania; anael.sam@nm-aist.ac.tz

\* Correspondence: gumaa@nm-aist.ac.tz; Tel.: +255-779-59-7131

**Abstract:** With the expansion of smartphone and financial technologies (FinTech), mobile money emerged to improve financial inclusion in many developing nations. The majority of the mobile money schemes used in these nations implement two-factor authentication (2FA) as the only means of verifying mobile money users. These 2FA schemes are vulnerable to numerous security attacks because they only use a personal identification number (PIN) and subscriber identity module (SIM). This study aims to develop a secure and efficient multi-factor authentication algorithm for mobile money applications. It uses a novel approach combining PIN, a one-time password (OTP), and a biometric fingerprint to enforce extra security during mobile money authentication. It also uses a biometric fingerprint and quick response (QR) code to confirm mobile money withdrawal. The security of the PIN and OTP is enforced by using secure hashing algorithm-256 (SHA-256), a biometric fingerprint by Fast IDentity Online (FIDO) that uses a standard public key cryptography technique (RSA), and Fernet encryption to secure a QR code and the records in the databases. The evolutionary prototyping model was adopted when developing the native mobile money application prototypes to prove that the algorithm is feasible and provides a higher degree of security. The developed applications were tested, and a detailed security analysis was conducted. The results show that the proposed algorithm is secure, efficient, and highly effective against the various threat models. It also offers secure and efficient authentication and ensures data confidentiality, integrity, non-repudiation, user anonymity, and privacy. The performance analysis indicates that it achieves better overall performance compared with the existing mobile money systems.

**Keywords:** mobile money systems; 2FA; multi-factor authentication; PIN; OTP; biometric fingerprint; Twilio SMS; QR code; SHA-256; FIDO; Fernet encryption; mobile money



**Citation:** Ali, G.; Dida, M.A.; Elikana Sam, A. A Secure and Efficient Multi-Factor Authentication Algorithm for Mobile Money Applications. *Future Internet* **2021**, *13*, 299. <https://doi.org/10.3390/fi13120299>

Academic Editors: Kaushik Roy, Mustafa Atay and Ajita Rattani

Received: 20 October 2021

Accepted: 19 November 2021

Published: 25 November 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The advent and expansion of FinTech coupled with the availability of affordable high-speed internet and widespread usage of the latest smartphones have revolutionized the financial service industry. FinTech is defined as new financial innovation businesses used to enhance and automate financial services and perform forecast analysis to improve financial inclusion. FinTech is transforming all aspects of financial services, such as deposits, loans, money transfers, fundraising, clearing and settlements, leasing, wealth management, mobile banking, personal finance management, mobile payment, risk and investment management, virtual currencies, mobile insurance, customized consulting, and intermediate and direct finance [1–3]. The emergence and popularity of FinTech are mainly due to the limitations of the traditional banking system; the coming of new technologies (such as payment gateways, mobile banking, mobile wallets, payment apps, etc.) that offer fast and cheap financial services; higher demand for financial services; strict banking regulations; and the comfort of running the business and banking systems [2,4–6]. Sharma [7] reported that by the end of 2022, mobile service payments might reach up to \$3388 Billion.

Mobile money as a FinTech has been beneficial to subscribers, from a comfortable way of sending and receiving the money to improving the standard of living of the unbanked people [8–10]. Today, most mobile money systems authenticate their subscribers using a PIN and SIM. This 2FA scheme, though promising, is weak, thus increasing the number of security challenges [8,9,11] due to the short PIN length (4 or 5 digits) used, the same PIN being used to authenticate all mobile money services, the fact that there is no expiry date for the PIN, the fact that sharing the PIN among users takes place and the fact that the PIN is entered when unmasked during verification [12,13].

The rapid adoption of smartphones that comes with in-built biometric fingerprint sensors can enhance mobile money authentication. This study aims to develop a secure and efficient multi-factor authentication algorithm for mobile money applications where subscribers will be authenticated using a PIN, OTP, and biometric fingerprint. It also uses a fingerprint and QR code to confirm money withdrawal. Since mobile money systems contain confidential financial information, they must be protected from unauthorized users. Deploying a highly secure and efficient multi-factor authentication scheme will secure the mobile money systems' authentication and sensitive financial records.

There are related studies that focus on multi-factor authentication schemes for mobile money. For example, Mega [14] presented a framework that uses PIN and iris biometric to improve the security of mobile money services. Islam et al. [15] proposed a secure mobile money transfer system that uses PIN and iris biometric for Bangladesh's small and medium enterprises (SMEs). Chetalam [16] proposed a multi-factor authentication scheme for mobile phones that uses device-specific ID, voice biometric, and PIN to secure M-PESA transactions. However, the existing proposed algorithms, though promising, are vulnerable to several security attacks. Besides, none of those studies mentioned above implemented combining PIN, OTP, and biometric fingerprint for mobile money authentication where SHA-256 is used to secure the PIN and OTP, FIDO to secure biometric fingerprint, and Fernet encryption to secure a QR code and the records in the databases. This study develops a secure and efficient multi-factor authentication algorithm for mobile money applications. Prototypes known as Genuine mobile money (G-MoMo) applications were developed based on the proposed algorithm to achieve the security principles of confidentiality, integrity, availability, authentication, privacy, and non-repudiation.

The essential contributions of our proposed algorithm are as follows:

- We propose a secure and efficient multi-factor authentication algorithm for mobile money applications where PIN, OTP, and biometric fingerprints authenticate users. It also authorizes mobile money withdrawal by scanning the fingerprints of the mobile money users and the secure QR code of the mobile money agent that contains the unique mobile money agent code.
- We present the different ways of securing user authentication credentials such as PIN and OTP using SHA-256, biometric fingerprint by FIDO services that use the public-key cryptography technique (RSA), and a QR code and records in the databases by Fernet encryption.
- We design and implement the proposed secure and efficient multi-factor authentication algorithm for mobile money applications by developing G-MoMo application prototypes to prove that the algorithm is feasible and provides a higher degree of security.
- We present a comparative security and performance analysis of our proposed algorithm.

The relevant related work on mobile money authentication schemes, problem statements, and technologies is discussed in Section 2. Section 3 covers the proposed algorithm and system architecture, while Section 4 presents the system implementation. Security analysis is discussed in Section 5, performance analysis is discussed in Section 6, comparison of the proposed algorithm with other related works takes place in Section 7, and finally, Section 8 covers the conclusion and recommendation.

## 2. Related Work

Authentication is a necessity when users are accessing systems to prove that they are the correct person trying to gain access, and it also helps to protect users' confidential information. Thus, different authentication methods that use mechanisms such as knowledge factor, ownership factor, and biometric factor are proposed. However, there are few studies related to mobile money authentication schemes, among which are the following.

### 2.1. Studies Related to Mobile Money Systems

Mega [14] presented a framework that uses PIN and iris biometric to improve the security of mobile money services. The framework has registration and authentication phases. The mobile money user's biodata, ID number, PIN, and iris biometric data are obtained, verified, and saved in the database during the registration phase. The mobile money user first logs in using their PIN, selects a mobile money service (such as deposit money or send money), and enters the agent or customer reference number and the amount during the authentication process. Then, the system requests the mobile money user to authenticate with iris biometric. If the iris feature matches with the copy stored in the database, the transaction is allowed; otherwise, it is rejected. The proposed framework prevents unauthorized access to mobile money systems and ensures convenience. However, it is susceptible to PIN challenges; the non-match error rate of iris recognition is high due to the pupil stretching and poor quality of the camera used during registration as well as iris deformity due to diseases.

Islam et al. [15] proposed a secure mobile money transfer system that uses PIN and iris biometric for Bangladesh's SMEs. In the proposed work, the subscriber's biodata, national identification number (NIN), phone number, PIN, and iris biometric are captured, verified, and stored in the database during the registration phase. The subscriber's PIN and biometric iris are used for authentication. The proposed scheme is secure, accurate, achieves better productivity, reliability, and customer satisfaction, and is resistant to impersonation attacks. However, eyeglasses can reduce the quality of the iris images and the accuracy of performance of the iris recognition systems.

Chetalam [16] proposed a multi-factor authentication scheme for mobile phones that uses device-specific ID, voice biometric, and PIN to secure M-Pesa transactions. The user's biodata, phone number, specific ID, PIN, and voice biometrics are captured and stored in the database for authentication in the proposed model. A user logs in using their specific ID, voice biometric, and PIN during the authentication. The login credentials are verified, and if they match, the user is authenticated; otherwise, they are rejected. The proposed model has higher efficiency, convenience, accuracy, authentication level and security, and it cannot be impersonated. Moreover, users do not have to install additional software to participate in the mobile money authentication process. However, it is susceptible to replay attacks and man-in-the-middle (MITM) attacks; additionally, the human voice changes over time, which may cause some errors in voice recognition.

Osman and Nakanishi [17] proposed a mobile money authentication system with high correctness. The user's biodata, unique identity number, and iris biometric are captured, verified, and stored in the database during the registration phase. The mobile money user logs in during the authentication phase using the unique identity number and iris biometrics. If the iris biometric matches the copy stored in the database, the mobile money user can perform a transaction. The scheme also authorizes transactions using the iris biometric. The proposed system is secure, robust, and reliable. However, the poor quality of the mobile camera used during iris biometrics registration and sagging of eyelids due to age results in a high non-match error rate.

Okpara and Bekaroo [18] proposed a novel method of authenticating customers in an electronic wallet using a camera-captured fingerprint sample (Cam-Wallet). During the registration phase, the user's biodata, payment cards in virtual cards, and the visual template of the fingerprint are scanned and stored for authentication purposes. Then, the user's fingerprint is scanned and matched with the template stored in the secure element.

The uniqueness of the scheme is in the manner the fingerprint is captured for authentication by using the mobile device's camera, and the failure rate is minimized during fingerprint acquisition and authentication. However, the challenge with the scheme is that customer credentials are stored on the device, which can be a target point for the attackers.

Mtaho [13] discussed the current two-factor authentication that uses PIN and SIM for the mobile money system. The enrolment phase involves capturing and storing the subscriber's biodata, phone number, and PIN in the mobile network operators (MNO) database. During the verification phase, the user dials an unstructured supplementary service data (USSD) code, selects the service from the menu, and enters their mobile money PIN. If the PIN matches the copy stored in the MNO's database, the service requested is offered; otherwise, the user is asked to try the PIN up to three times. The model is user-friendly but susceptible to social engineering attacks, identity theft, shoulder-surfing attack, brute-force attack, replay attacks, MITM attacks, and PIN challenges.

In 2015, Mtaho [13] suggested a two-factor authentication system for enhancing mobile money security. The user biodata, PIN, and biometric fingerprint are collected and stored in the MNO database during the enrolment process. Then, the user is authenticated using a PIN and biometric fingerprint. The proposed scheme offers security against identity theft and shoulder-surfing attacks. However, it is susceptible to spoofing attacks, fake digital biometrics, Trojan horse attacks, matcher override or false matches, replay attacks, and intrusion attacks.

Coneland and Crespi [19] proposed a wallet-on-wheels using the vehicle's identity to secure mobile money. The scheme integrates machine-to-machine (M2M), mobile money, automotive communications, and advanced security procedures. The user's biodata and e-mail address, the international mobile equipment identity (IMEI) of the phone, SIM, and car serial number are captured and stored in MNO's system during enrolment. The service authorization and authentication are done using both car-based and SIM-based methods. The secret serial numbers (IMEI) and embedded SIM authentication enhance the method remarkably. The security hashing procedure uses both the car serial number and the MNO's international mobile subscriber identity (IMSI) number to produce authentication keys for both wallet-on-wheels and MNO communications. The proposed system provides improved car-related paid services and a higher security platform against increasing fraud and identity theft. However, a wallet-on-wheels is restricted to authorized car drivers and car-enabled applications. The scheme involves different technologies and stakeholders, thus increasing the complexity of offering a uniform service.

Hassan and Shukur [20] proposed a framework for electronic payment systems that use secure multi-factor user authentication. The user's biodata, password, and fingerprint are captured, verified, and stored in the database during the registration phase. In the authentication phase, the user must log in with the approved password and biometric fingerprint. The scanned biometric fingerprint is compared with the one stored in the database, and if it matches, the user signs in successfully and can go to the transaction phase. The user enters the amount to be transferred in the transaction phase and then authenticates by scanning the biometric fingerprint. If the biometric fingerprint matches, the system generates OTP and sends it to the user's registered phone number. The user is requested to enter the correct OTP. If the OTP matches, money is transferred, and the transaction is successful. The proposed framework increases the security of user authentication systems by offering multiple forms of protection to the authentication methods. It provides protection and trust in the electronic payment system. Besides, the security analysis demonstrated that the framework combats password-based attacks, dictionary attacks, brute-force attacks, phishing attacks, password-guessing attacks, shoulder-surfing attacks, and MITM attacks. However, they are susceptible to SIM-swapping attacks, wireless interception of SMS OTP messages, and malware attacks.

Vincent et al. [21] presented an improved security scheme for a mobile payment system using identity-based elliptic-curve cryptography (ECC). The system uses a payment gateway for registration and maps all input text to elliptic curve points using the American



standard code for information interchange (ASCII) values. Payment details are stored on the gateway, encrypted but decrypted only with the merchant's decryption key. The evaluation result for the proposed scheme has proven that the system is time-efficient in a resource-constrained environment. The computational power is efficient for low-resource mobile devices in battery life and faster encryption and decryption keys. IMEI with ECC provides integrity, confidentiality, privacy, non-repudiation and is resilient to identity theft. However, mobile phone data are encrypted on the payment gateway.

## 2.2. Problem Statement

Many algorithms have been developed to boost mobile money authentication security. However, to date, there are no strong security controls to suit all mobile money authentication security challenges. However, the existing proposed algorithms, though promising, require more work because they are vulnerable to impersonation attacks; USSD technology vulnerabilities; replay attacks; spoofing attacks; Trojan horse attacks; brute-force attacks; shoulder-surfing attacks; MITM attacks; insider attacks; identity theft; social engineering attacks; SIM-swapping attacks; malware attacks; agent-driven fraud; and privacy attacks [8,9,11,22–26]. Therefore, there is a need to develop a secure and efficient multi-factor authentication algorithm for mobile money applications where mobile money subscribers are authenticated using a PIN, OTP, and biometric fingerprints. It also uses a biometric fingerprint and QR code to confirm mobile money withdrawal. The authentication identifiers are protected using different mechanisms, such as SHA-256 for PIN and OTP, FIDO that uses RSA for Biometric fingerprint, and Fernet encryption for a QR code and the records in the databases.

The proposed algorithm implements FIDO, which uses a public-key cryptography technique (RSA encryption) to secure biometric fingerprints. There are two main phases in FIDO, i.e., registration and authentication. A mobile money user is requested to pick an available FIDO authenticator (e.g., smartphone) that matches the online service acceptance policy during the registration phase. While during the authentication phase, the user unlocks the smartphone by scanning the fingerprint using a fingerprint reader of the smartphone. A new public/private key pair unique to the smartphone, mobile money service, and user's account is created. The public key is encrypted and sent to the online FIDO database, associated with the user's mobile money account, and saved in a Keystore. The private key and biometric templates are encrypted and stored in the user's smartphone in a cryptographic Keystore. The G-MoMo applications challenge the users to log in with previously registered smartphones that match the service's acceptance policy and unlock the smartphones using biometric fingerprints. The smartphones use the users' account identifier to select the correct key and sign the service's challenge. The user's smartphone sends the signed challenge to the service, verifying it with the stored public key, and the user is logged in. In the proposed algorithm, the user's encrypted biometric template is not stored in the database but instead in the user's smartphone. Besides, the mobile money agent's serial number is encrypted using Fernet encryption to generate a mobile money agent code, which is then encoded in a QR code for the user to scan during money withdrawal. This helps to confirm the authenticity of the mobile money agents. The proposed algorithm helps to ensure secure and efficient authentication and provides data confidentiality, integrity, non-repudiation, user anonymity, privacy, and improved performance. It is resistant to shoulder-surfing attacks, social engineering attacks, phishing attacks, PIN-guessing attacks, and brute-force attacks. Likewise, it prevents replay attacks, insider attacks, impersonation attacks, identity fraud, and MITM attacks.

## 2.3. Technologies

The study used the following technologies to provide secure and efficient multi-factor authentication.

- Personal Identification Number (PIN)

A PIN in mobile money is a numeric password that can authenticate subscribers in an electronic transaction. The PINs used in mobile money authentications are often four or five digits [11]. The PIN value as a means of authentication depends entirely on its secrecy from the moment the PIN is created until entered into the system. Many systems and applications adopt the PIN-entry method due to its convenience, efficiency, reliability, better dependability, and customer satisfaction in mobile transactions [27]. Using PIN alone for authentication is risky and vulnerable to shoulder-surfing attacks, replay attacks, PIN leakage, guessing attacks, eavesdropping, phishing attacks, spoofing, MITM attacks, and malware attacks [9,11,28].

- One-Time Password (OTP)

A One-Time Password is a dynamic password that is effective only for a short period and valid for only one login session [28,29]. OTP can be time-based, pattern-based, and random key-based and is delivered via SMS, OTP application, RSA token, and e-mail [28]. The principle of OTP is to share a seed between a generator and a verifier where both can produce the same password. The generator is responsible for generating an appropriate OTP from the shared seed. The verifier is responsible for validating the OTP and storing the last valid OTP received and the current one. SMS-based OTP is most commonly used in multi-factor authentication for many different applications because it is cheap, fast, efficient, reliable, and convenient. Using OTP aims to prevent fraudulent attacks such as phishing attacks, shoulder-surfing attacks, replay attacks, eavesdropping, spoofing attacks, brute-force attacks, MITM attacks, and identity theft. OTP is also resilient to reverse engineering [9,11,28,30,31].

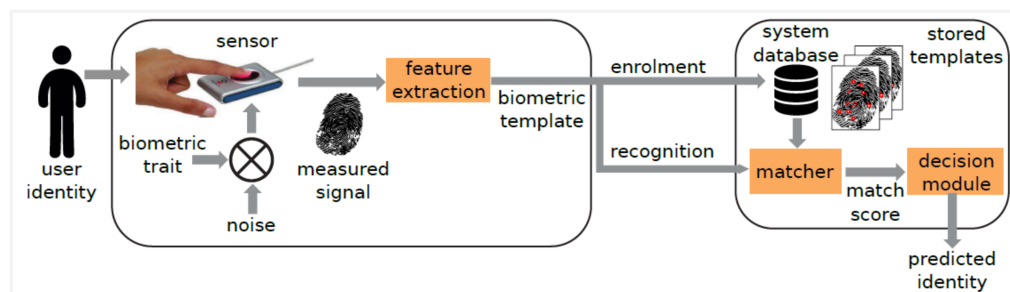
- Quick Response (QR) Code

A QR code is a two-dimensional, matrix populated, square-shaped, and machine-readable optical label used to store information about the item based on their allocation capacity [32–34]. It is represented by black and white square image blocks called modules. Each module describes some data, where the black square holds value 1 and the white square holds 0. Both black and white colors permit encoding and decoding of the data [35,36]. Each QR code stores vertical and horizontal strings of text comprised of standardized encoding modes, i.e., numeric characters, alphanumeric characters, letters, symbols, Byte characters, and Kanji/Kana [35,37,38]. It can encode 7089 numeric characters, 4296 alphanumeric characters, 2953 bytes [33,35,36]. The encoded information in the QR code can be decoded using a handheld scanner or a QR code scanner app installed on a smartphone [33]. QR codes are widely used in security-related operations such as authentication and identification. The QR code is rapidly being adopted in mobile payments because of its speed in terms of decoding, convenience, ease of use, security and privacy, cost-effectiveness, traceability, ample data storage, robustness against damages/error correction capability, flexibility, easy readability, the fact that it is easy to generate and manage, its user-friendly nature, its versatility, and its accuracy [38–45]. Likewise, they provide data confidentiality and integrity; ensure non-repudiation; and prevent identity theft, impersonation attacks, shoulder-surfing attacks, and brute-force attacks [46].

- Biometric Fingerprint

According to Dasgupta et al. [47], fingerprint recognition is a visual type of authentication that identifies individuals using the ridges and valleys found on the surface of fingers. The skin on the fingertip's surface consists of raised folds of skin, called ridges, and valleys separate these ridges. The pattern of ridges and valleys on a fingertip represents a fingerprint used in biometric recognition. Fingerprint recognition is commonly used for authentication on computerized systems [23]. Different schemes have used it to verify a user's identity, but it is not recommended as a standalone authentication approach. The biometric fingerprint system involves two distinct phases, i.e., enrollment and verification [48]. Three specific processes are performed in the enrollment phase, i.e., fingerprint

acquisition, fingerprint feature extraction, and template storage in the database. While in the verification phase, three operations are performed, i.e., fingerprint acquisition, fingerprint feature extraction, matching the extracted fingerprint with the templates previously created and stored in a database [48]. Depending on whether it matches, the user will be authenticated or rejected, as shown in Figure 1 below.



**Figure 1.** The fingerprint enrollment and verification phases [49].

Fingerprint technology is widely used because of its convenience, higher security, better efficiency, stability, higher system reliability, distinctiveness, greater robustness, better anti-counterfeiting performance, ease of use, the small size of fingerprint templates, and its small amount of power consumption [50–52].

#### ■ Secure Hashing Algorithm-256 (SHA-256)

SHA-2 is defined as a set of cryptographic hash functions designed by the United States National Security Agency (NSA) in association with the National Institute of Science and Technology (NIST), which was first published in 2001 as an enhancement to the SHA-1 algorithm [53]. Zhang et al. [54] added that SHA-2 is a one-way and collision-resistant cryptographic hash function used in cryptographic primitives to provide security to applications and protocols because they have variable digest lengths. SHA-2 follows the Merkle Damgård structure model and has six hash functions, such as SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256, and the bit size used for encrypting data differs though. The same algorithm is applied to different word lengths, with distinct constant parameters and values, and employs different initialization values [53,55–58]. The output length also differentiates them.

The SHA-256 algorithm has increasingly become a reliable algorithm in SHA-2 when considering factors such as security, efficiency, and implementation. Therefore, it is broadly used in information encryption, digital signature, message authentication, and blockchain [53]. SHA-256 produces a 256-bit hash value from messages of any length up to  $2^{64}$  bits. The message to be hashed is processed in fixed-length 512-bit blocks known as padded data blocks (PDBs), and the result of the processing of each block is fed as input to the processing of the next block [59]. The irreversible mathematical properties of hashing make it a unique mechanism to conceal the information. The hash functions are deterministic, making them vital for authentication since it guarantees that the message will produce the same hash. With the help of the SHA-256, integrity, confidentiality, authentication, and security are achieved. It is also one-way and fast; a message of any size can be compressed to a fixed-length hash; and it is resistant to collisions, brute-force attacks, dictionary attacks, and MITM attacks [54,55,58,60]. Figure 2 illustrates how messages are hashed using the SHA-256 algorithm to generate a unique fixed-length hash value.



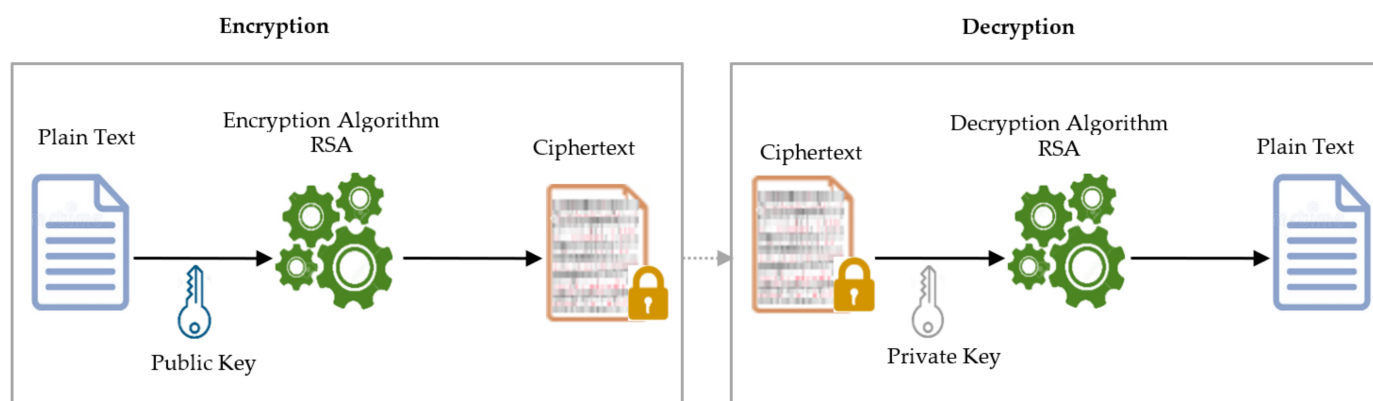
**Figure 2.** Illustrates how the message is hashed using the SHA-256 algorithm.

#### ■ Fast Identity Online (FIDO)

FIDO is a piece of authentication technology that improves original hardware-based solutions by using the mathematics of public-key cryptography. It supports multi-factor authentication and public key cryptography, where universal second-factor authentication (U2F), the universal authentication framework (UAF), and the client-to-authenticator protocol (CTAP) protocols are developed [61,62].

FIDO has two main phases, i.e., registration and authentication. The user is urged to pick an available FIDO authenticator (e.g., smartphone) that matches the online service acceptance policy during the registration phase. The user unlocks the smartphone by scanning the fingerprint using a fingerprint reader of the smartphone. A new public/private key pair unique for the smartphone, online service, and user's account is created. The public key is encrypted and sent to the online service, associated with the user's account, and saved in a Keystore. The private key and biometric templates are encrypted and stored in the user's smartphone in a cryptographic Keystore. The online service challenges users to log in with a previously registered smartphone that matches the service's acceptance policy during the authentication phase. The user unlocks the smartphone using the biometric fingerprint. The smartphone uses the user's account identifier to select the correct key and sign the service's challenge. The user's smartphone sends the signed challenge to the service, verifying it with the stored public key, and the user is logged in. FIDO architecture has been widely used in easy payment, money transfer, ATM withdrawal and savings, single sign-on, healthcare and insurance, enterprise organizations, government, etc.

The FIDO service uses RSA, a standard public-key cryptography technique. Ron Rivest, Adi Shamir, and Leonard Adleman invented RSA encryption in 1977. It is a public-key cryptosystem widely used to maintain a high level of security [63,64]. Key generation, encryption, and decryption are the three (3) steps involved in RSA [65]. RSA needs a pair of public and private keys, where a message is encrypted using the recipient's public key and decrypted using the recipient's private key [65]. Two large secret prime numbers together with an auxiliary value are used to create a public key. It uses an adjustable encryption block size and key size [65]. Figure 3 shows message encryption using the public key and decryption using the private key in RSA.



**Figure 3.** Illustrates message encryption and decryption in RSA.

RSA is used as the encryption algorithm because it is trusted as the most robust asymmetric encryption system that provides a high level of security, preserves data privacy, non-repudiation, data confidentiality, and data reliability. It also has quick encryption and authentication processes [63–67].

FIDO authentication helps establish a simple, robust, and secure authentication channel by creating a public–private cryptographic key pair, allowing a private key unique to each user to be stored on the smartphone. It offers convenience, scalability, universality, persistence and uniqueness, and it improves user experience and performance. FIDO also provides a uniform user login experience; mitigates phishing attacks, sniffing attacks, replay attacks, and MITM attacks; and creates a secure communication channel between the application server and the users using public-key cryptography [62,68–71].

- Fernet Encryption

Fernet encryption is a symmetric encryption method that ensures that the encrypted message cannot be read or updated without the key. Fernet encryption uses 128-bit advanced encryption standard (AES) symmetric encryption in cipher block chaining (CBC) mode with public-key cryptography standards 7 (PKCS7) padding and a hash-based message authentication code (HMAC) using SHA-256 for authentication [72]. It provides a secure means of generating keys, selects a secure encryption algorithm, randomly allocates a secure “salt” value to make the encryption safer, timestamps the ciphertext, and signs the message to detect any attempts to change it. Python was chosen to implement the Fernet encryption because it supports a cryptography package that helps encrypt and decrypt data in the database. It provides libraries that facilitate the generation of keys, and HMAC supports data integrity. The Fernet module of the cryptography package has inbuilt functions for key generation, encryption, and decryption using the encrypt and decrypt methods, respectively [72]. The Fernet encryption method has been chosen because of the multiple security mechanisms provided by its utilization [73]. Fernet offers higher security, trust, privacy, authenticity, authorization, data confidentiality, and integrity. It ensures non-repudiation, prevents impersonation attacks, shoulder-surfing attacks, phishing attacks, identity theft, and brute-force attacks [46,74–83].

### 3. Proposed Algorithm

This section describes the proposed secure and efficient multi-factor authentication algorithm for mobile money applications where users are authenticated using a PIN, OTP, and biometric fingerprint. It also uses a biometric fingerprint and a secure QR code to confirm money withdrawal. Besides, authentication factors (identifiers) such as PIN and OTP are protected using SHA-256, biometric fingerprint by FIDO, which uses the public-key cryptography technique (RSA encryption), and Fernet encryption to secure a QR code and the records in the databases.

The proposed algorithm for mobile money applications has three main phases, i.e., enrolment, authentication, and transaction. The symbols used in the algorithm are summarized in Table 1.

**Table 1.** The symbols used in the algorithm.

Symbols	Meaning
$U_i$	User
$FN_i$	User’s first name
$LN_i$	User’s last name
$SIM_{sn}$	SIM serial number
$SID_i$	Subscriber ID
$PN_i$	User’s phone number
$PIN_i$	User’s PIN
$PIN_j$	Re-entered PIN
$OTP_i$	User’s OTP
$BF_i$	User’s biometric fingerprint



Table 1. Cont.

Symbols	Meaning
$B_t$	Biometric template
$P_i$	User's public key
$F_i$	User's private key
$SP_i$	User's smartphone
$ID_i$	User's ID
$ID_{sp}$	Smartphone ID
$h(.)$	One-way hash function—SHA-256
$E_u(.) / D_u(.)$	Fernet Encryption/Decryption with key $u$
$E(.) / D(.)$	Public key Encryption/Decryption—RSA
$DB_m$	Main database
$DB_{fd}$	FIDO database
$Bal_i$	User's electronic balance
$A_a$	Agent
$QRcode_a$	Agent QR code
$Amt_i$	Amount

### 1. Enrolment Phase

Before enrolling a mobile money  $U_i$ , the  $U_i$ 's  $SIM_{sn}$  and the  $SID_i$  must be generated. The mobile money  $A_a$  then registers the mobile money  $U_i$  by following the steps below Algorithm 1:

- **Step 1.** The mobile money  $A_a$  must capture the  $U_i$ 's  $FN_i$ ,  $LN_i$ ,  $SIM_{sn}$ ,  $SID_i$ ,  $PN_i$ , i.e.,  $k_i = (FN_i, LN_i, SIM_{sn}, SID_i, PN_i)$ .
- **Step 2.** The mobile money  $A_a$  then verifies the  $k_i$  provided by the  $U_i$ . If the  $k_i$  is wrong, the  $U_i$  has three (3) attempts; otherwise, the information is saved in the  $DB_m$ .
- **Step 3.** The  $U_i$  must complete the registration process by entering a five-digit  $PIN_i$  and re-enter the five-digit  $PIN_j$ .
- **Step 4.** If the five-digit  $PIN_i$  entered and the re-entered  $PIN_j$  do not match, the  $U_i$  is required to enter the correct mobile money PIN; otherwise, it is hashed using the SHA-256,  $l_i = h(PIN_i, PIN_j)$  and saved in the  $DB_m$ .
- **Step 5.** The  $U_i$  must use the  $SP_i$  fingerprint sensor to scan their  $BF_i$ . If the  $BF_i$  is successfully captured, the  $SP_i$  creates a new  $(P_i, F_i)$  pair unique to the  $SP_i$  and  $U_i$ 's account. The  $P_i$  is encrypted using RSA,  $m_i = E(P_i)$ , sent to the  $DB_{fd}$  and saved in a Keystore. The  $F_i$  and the  $B_t$  are encrypted using RSA,  $n_i = E(F_i)$ ,  $o_i = E(B_t)$  and stored in the  $SP_i$  under cryptographic Keystore and  $SP_i$ , respectively.
- **Step 6.** The  $SP_i$  stores  $n_i, o_i$  and sends  $l_i$ ,  $m_i$ ,  $ID_i$ ,  $ID_{sp}$  to the  $DB_m$  and  $DB_{fd}$ .
- **Step 7.** The  $DB_m$  checks whether the  $ID_i$  or  $ID_{sp}$  exists. If yes, the  $U_i$  is requested to enter a new  $PN_i$  and  $SIM_{sn}$  for the registration; otherwise, the  $DB_m$  and  $DB_{fd}$  records are encrypted using the Fernet, i.e.,  $e_i = E_u(k_i, l_i, m_i, ID_i, ID_{sp})$ , save it in the  $DB_m$  and  $DB_{fd}$ , and a notification for successful registration is displayed to the  $U_i$ .

**Algorithm 1** Enrolment Phase

---

```

Input:  $FN_i, LN_i, SIM_{sn}, SID_i, PN_i, PIN_j$ 
START
1   $k_i \leftarrow$  Take input of the  $\{FN_i, LN_i, SIM_{sn}, SID_i, PN_i\}$ .
2  for ( $int\ i = 0; i \leq 3; i++$ ) {
3    if IsUserDataValid ( $k_i$ ) then
4      Save  $k_i$  in the  $DB_m$ 
5       $PIN_k \leftarrow$  Take input of the  $\{PIN_i, PIN_j\}$ 
6      if ( $(PIN_i.length == 5) \text{ AND } (PIN_j.length == 5) \text{ AND } (PIN_i == PIN_j)$ ) then
7        Hash the  $PIN_k$  using SHA-256,  $l_i = h(PIN_k)$  and save in the  $DB_m$ .
8      else
9        Enter a valid  $PIN_i$  and  $PIN_j$ 
10     end if
11     Retrieve the biometric fingerprint feature, and  $P_i/F_i$  pairs are created, i.e.,  $P_i/F_i$ .
12      $P_i$  is encrypted using RSA,  $m_i = E(P_i)$ , sent to the  $DB_{fd}$  and saved in a Keystore.
13     The  $F_i$  and the  $B_i$  are encrypted using RSA,  $n_i = E(F_i)$ ,  $o_i = E(B_i)$  and stored in the  $SP_i$  under cryptographic Keystore and  $SP_i$ .
14     The  $SP_i$  stores  $n_i, o_i$  and sends  $l_i, m_i, ID_i, ID_{sp}$  to the  $DB_m$  and  $DB_{fd}$ .
15     The  $DB_m$  checks whether the  $ID_i$  and  $ID_{sp}$  exists. If yes, the  $U_i$  is requested to enter a new  $PN_i$  and  $SIM_{sn}$  for the registration, else, the  $DB_m$  and  $DB_{fd}$  records are encrypted using the Fernet, i.e.,  $e_i = E_u(k_i, l_i, m_i, ID_i, ID_{sp})$ ; save it in the  $DB_m$  and  $DB_{fd}$ .
16     A notification for successful registration is displayed to the  $U_i$ .
17   else
18     Invalid user data
19   end if
20 }
21 Return
22 STOP

```

---

## 2. Authentication Phase

To log in to the system, the mobile money  $U_i$  must follow the steps below Algorithm 2:

- **Step 1.** The mobile money  $U_i$  must enter their five-digit  $PIN_i$  during the authentication phase to  $SP_i$ . Note that the  $U_i$  can only attempt the authentication three (3) times.
- **Step 2.** The  $SP_i$  sends  $w_i = ID_i, ID_{sp}, PIN_i, Request$  to the  $DB_m$  for validation.
- **Step 3.** The  $DB_m$  checks if the  $ID_i, ID_{sp}$ , and  $PIN_i$  match. If they do not match, the transaction is terminated; otherwise,  $OTP_i$  is generated and sent to the mobile money  $U_i$  via SMS, and the copy of the sent  $OTP_i$  is hashed using the SHA-256,  $b_i = h(OTP_i)$  and stored in the  $DB_m$ .
- **Step 4.** The  $U_i$  is requested to enter the received  $OTP_i$ . Note that the  $OTP_i$  is valid for only 60 s.
- **Step 5.** When the mobile money  $U_i$  enters the  $OTP_i$  and does not match with the copy stored in the  $DB_m$ , i.e.,  $b_i$ , the transaction is terminated. Otherwise, the  $U_i$  is required to scan the  $BF_i$  for recognition using the  $SP_i$ .
- **Step 6.** Once the  $U_i$  scans the  $BF_i$  and it matches the  $B_i$ , the protected resource informs the FIDO server component of the server-side application, which sends a challenge to the  $U_i$ 's  $SP_i$ . The FIDO authenticator (i.e.,  $SP_i$ ) locally verifies the  $U_i$ 's identity based on the  $BF_i$  by generating a  $(P_i, F_i)$  key pair and comparing the  $P_i$  with the copy stored under the Keystore in the  $DB_{fd}$ . The  $F_i$  and  $B_i$  are compared with the copy stored in the  $U_i$ 's  $SP_i$  under the cryptographic Keystore and  $SP_i$ , respectively.
- **Step 7.** If the  $(P_i, F_i)$  pairs do not match, the authentication will be terminated. Otherwise, the  $U_i$  is authenticated and can now perform transactions.

**Algorithm 2** Authentication Phase

---

```

Input:  $PIN_i, OTP_i, BF_i$ 
START
1   $PIN \leftarrow$  Take input of the  $U_i$ 's  $PIN_i$ .
2  for ( $int\ i = 0; i \leq 3; i++$ ) {
3    if IsPINValid ( $PIN$ ) then
4      Request for the generation of  $OTP_i$ 
5      Send  $OTP_i$  to the  $U_i$ 's  $SP_i$ .
6      Hash the sent  $OTP_i$  using the SHA-256,  $y_i = h(OTP_i)$  and store  $y_i$  in the  $DB_m$ 
7      Display the  $OTP_i$  for the  $U_i$  to read.
8    else
9      Invalid  $PIN_i$  and transaction terminated
10   end if
11    $OTP \leftarrow$  Take input of the  $U_i$ 's  $OTP_i$ .
12   if IsOTPValid ( $OTP$ ) then
13     Scan the  $BF_i$  for recognition using the  $SP_i$ 
14   else
15     Invalid  $OTP_i$  and transaction terminated
16   end if
17    $BF \leftarrow$  Take input of the  $U_i$ 's  $BF_i$  using the  $SP_i$  fingerprint sensor.
18   if IsBFValid ( $BF$ ) then
19      $P_i/F_i$  pair are created. The  $E_u(E(P_i))$  is decrypted,  $d_p = D_u(D(P_i))$ , and compared with  $P_i$ . The
20      $E_u(E(F_i))$  and  $E_u(E(B_t))$  are also decrypted,  $d_f = D_u(D(F_i))$  and  $d_t = D_u(D(B_t))$  and compared with  $F_i$  and  $B_t$ ,
21     respectively.
22     if ( $P_i == d_p$ ) AND ( $F_i == d_f$ ) AND ( $B_t == d_t$ ) then
23       The  $U_i$  is authenticated successfully and can proceed to transaction.
24     else
25       Invalid  $P_i, F_i$ , and  $B_t$ 
26     end if
27   else
28     Invalid  $BF_i$  and transaction terminated
29   end if
30 }
Return
STOP

```

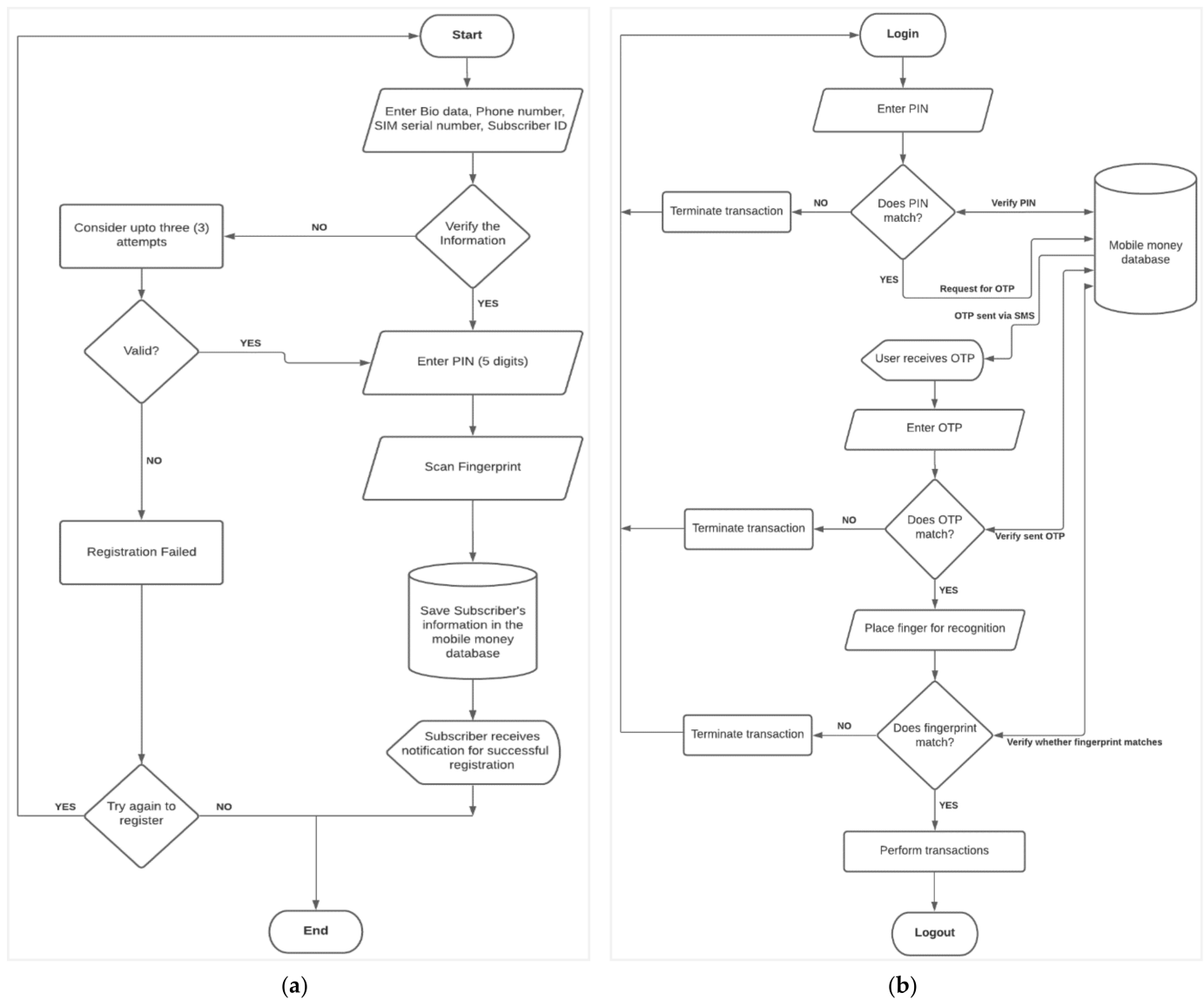
---

Figure 4a,b summarizes the proposed mobile money enrolment and authentication phase.

### 3. Transaction Phase

To withdraw money, the mobile money  $U_i$  must follow the steps below Algorithm 3:

- **Step 1.** The mobile money  $U_i$  begins the money withdrawal transaction by signing in to the application by entering their  $PIN_i, OTP_i$ , and  $BF_i$ .
- **Step 2.** If the  $PIN_i, OTP_i$ , and  $BF_i$  match, the  $U_i$  is logged in, and the system displays the available  $Bal_i$  the mobile money  $U_i$  has.
- **Step 3.** If the  $Bal_i$  is greater than or equals to 5000 and the mobile money  $U_i$  wants to withdraw money, they can enter an  $Amt_i$  less than or equal to the available  $Bal_i$ .
- **Step 4.** The system then requests the  $U_i$  to scan their  $BF_i$  using the  $SP_i$  fingerprint sensor.
- **Step 5.** If the scanned  $BF_i$  matches, the system requests the  $U_i$  to scan the secure  $QRcode_i$  of the mobile money  $A_a$  using the smart scanner for confirmation purposes.
- **Step 6.** If the scanned secure  $QRcode_i$  matches, the system displays a successful withdrawal message seeking the  $U_i$  to collect money from the mobile money  $A_a$ .



**Figure 4.** (a) The flowchart for the enrolment phase. (b) The flowchart for the authentication phase.

**Algorithm 3** Transaction Phase—Withdraw Money

---

```

Input:  $PIN_i$ ,  $OTP_i$ ,  $BF_i$ ,  $Bal_i$ ,  $Amt_i$ ,  $QRcode_a$ 
START
1   $identifier \leftarrow$  Take input of  $U_i$ 's  $\{PIN_i, OTP_i, BF_i\}$ 
2  for ( $int\ i = 0; i \leq 3; i++$ ) {
3    if  $IsIdentifierValid(identifier)$  then
4      The user is successfully logged in, and the system checks for the  $U_i$ 's available  $Bal_i$ 
5    else
6      Invalid Login Credentials
7    end if
8  }
9  Return
10 if ( $Bal_i \geq 5000$ ) then
11   Enter the  $Amt_i$  to withdraw
12 else
13   Insufficient  $Bal_i$ 
14 end if
15  $Amt \leftarrow$  Take input of the  $Amt_i$  to withdraw
16 if ( $Amt_i \geq 5000$ ) then
17   Scan  $BF_i$  for authorization
18    $BF \leftarrow$  Take input of the  $U_i$ 's  $BF_i$  using the  $SP_i$  fingerprint sensor
19   if  $IsBFValid(BF)$  then
20      $P_i/F_i$  pair is created. The  $E_u(E(P_i))$  is decrypted,  $d_p = D_u(D(P_i))$  and compared with  $P_i$ . The
      $E_u(E(F_i))$  and  $E_u(E(B_t))$  are also decrypted,  $d_f = D_u(D(F_i))$  and  $d_t = D_u(D(B_t))$  and compared with  $F_i$  and
      $B_t$ , respectively.
21     if ( $(P_i == d_p) \text{ AND } (F_i == d_f) \text{ AND } (B_t == d_t)$ ) then
22       Scan the  $A_a$ 's secure  $QRcode_a$  for confirmation
23        $QR \leftarrow$  Take input of the  $A_a$ 's  $QRcode_a$  using the Smart Scanner
24       if  $IsQRCodeValid(QR)$  then
25         Withdraw money from  $U_i$ 's account
26         Update the remaining  $Bal_i$ 
27         Display successful money withdrawn message
28       else
29         Invalid  $QRcode_a$ 
30       end if
31     else
32       Invalid  $P_i$ ,  $F_i$ , and  $B_t$ 
33     end if
34   else
35     Invalid  $BF_i$ 
36   end if
37 else
38   Insufficient  $Amt_i$ 
39 end if
STOP

```

---

Figure 5 illustrates how a mobile money user can withdraw money from their account.



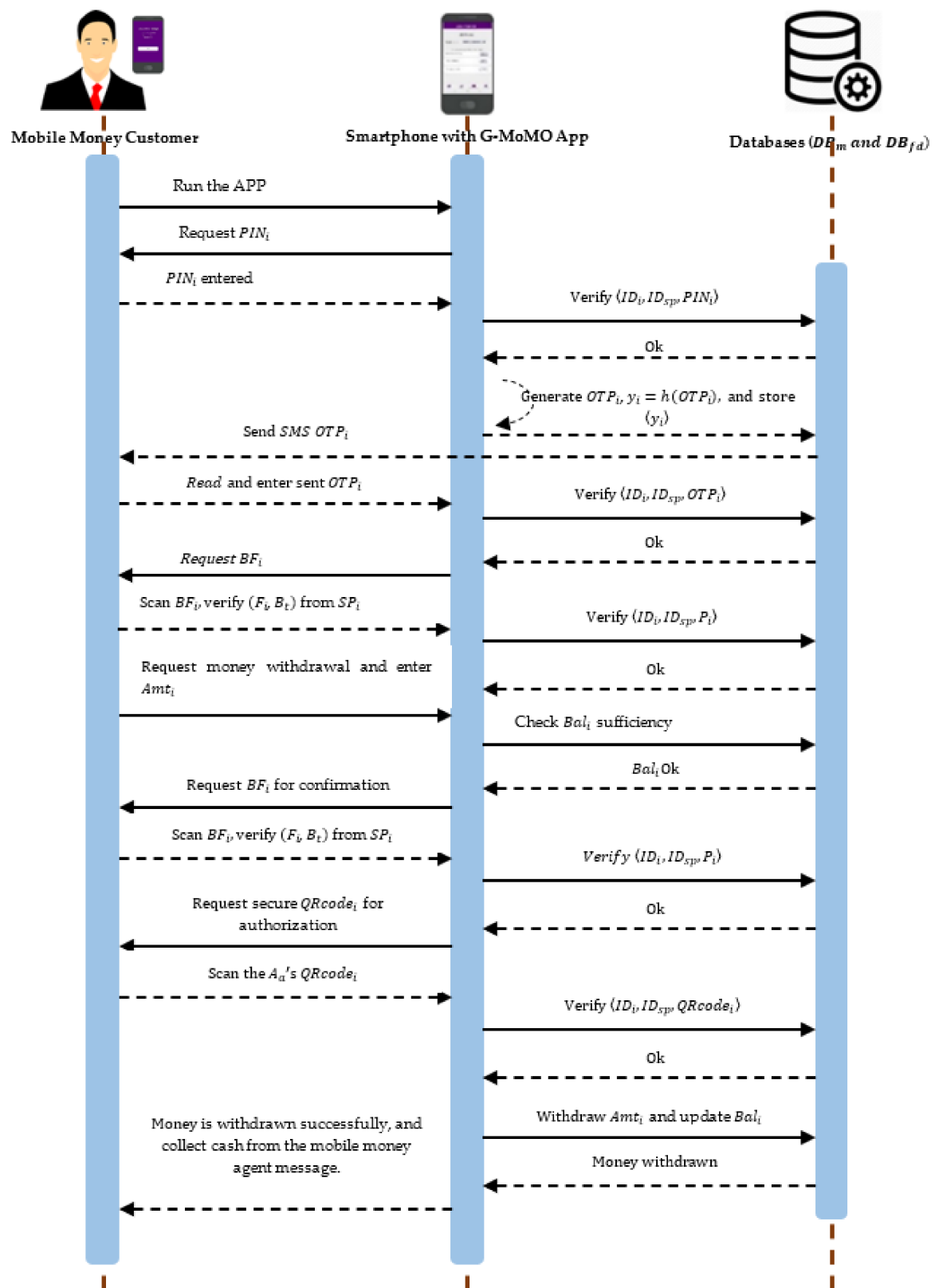


Figure 5. The algorithm for the transaction phase.

### System Architecture

The system architecture describes the interactions between different components of the proposed system. The main components are the mobile money applications (i.e., G-MoMo SIM Serial Number Generator Application, G-MoMo IT Support Application, G-MoMo Agent Application, and G-MoMo Customer Application), mobile money application server, mobile money databases (i.e., main database and FIDO database), network, mobile money services, mobile money IT support staff, mobile money agents, mobile money users, biller systems, payment service provider systems, point-of-sale devices and access devices (computers, tablets, and smartphones). The information about mobile money IT support staff, agents, and users is stored in the main database. The generated public keys are stored in the FIDO database, and private keys and biometric templates are stored in smartphones. The databases for the mobile money applications run on the mobile money application server to ensure security since the server checks requests to and from the databases. Mobile money services are used to transfer information between G-MoMo applications. With the G-MoMo applications, mobile money IT support staff can log in, register new mobile money agents, add new smartphones for the subscribers, add other new mobile money IT support staff, generate statistics showing the total number of registered subscribers, and log out. Mobile money agents can log in, register new mobile money users (customers), deposit money into customers' accounts, confirm money withdrawal through QR code scan, check balance, account management (change credentials such as PIN and fingerprint), and log out. Mobile money users can log in, withdraw money, send money, pay bills, check balance, check mini statements, account management (change credentials such as PIN and biometric fingerprint), and log out. The authentication security in G-MoMo applications is enforced on the application layer using a PIN, OTP, and biometric fingerprint. The security of PIN and OTP is ensured by SHA-256, biometric fingerprint by FIDO, and the QR code and records in the databases using Fernet encryption. All the components work together to attain the systems' goal. Figure 6 shows the system architecture for the proposed scheme.

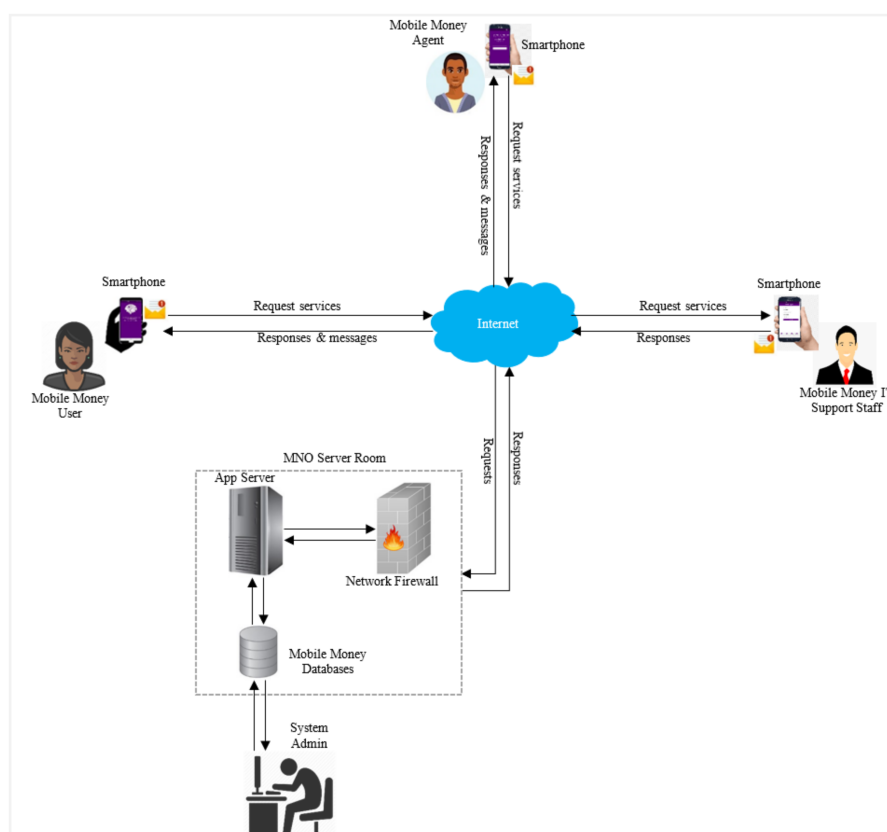


Figure 6. System architecture for the proposed system.

#### 4. System Implementation

The requirements for the proposed system are gathered through document reviews [11], surveys [9], and observation. The collected data were analyzed and used as essential inputs to define the necessary features of the proposed system.

Evolutionary prototyping was employed to develop the G-MoMo applications. Evolutionary prototyping is a software development method where a prototype is developed after receiving initial feedback from the customers [84]. Subsequent prototypes are produced, each with additional functionality until the customer is pleased. The last developed version can be engineered as the final product to be delivered [84]. It is similar to incremental development and has six steps of the prototyping process, namely, (1) requirements gathering and analysis, (2) quick design, (3) build a prototype, (4) initial user evaluation, (5) refine the prototype, and (6) implement the product and maintain [84]. Evolutionary prototyping allows mobile money IT support staff, agents, and users to involve themselves throughout the system development period, leading to user requirement satisfaction. It also allows changes in every phase, thus improving the prototype system; it minimizes severe and critical defects during the system testing and offers a better approach that saves time and effort [84].

In this research, the system requirements are grouped into functional and non-functional requirements, which provide guidelines for implementing the proposed algorithm. The functional requirements include user registration, user authentication, adding new smartphones for the subscribers, depositing money, withdrawing money, sending money, confirming money withdrawal through QR code scanning, checking balance, bill payments, checking the mini statement, statistics of subscribers, account management, and logout. However, the non-functional requirements are security, privacy, usability, maintainability, performance, reliability, interoperability, flexibility, robustness, availability, responsiveness, scalability, and look and feel.

The G-MoMo applications were designed using the unified modeling language (UML). UML is a modeling language that provides a standard way to visualize system designs through object-oriented diagramming schemes. The designs give an overview of the system, support the application's construction phase, help clarify the requirements to users, and explain the system to other stakeholders. The UML diagrams, such as use case diagrams, sequence diagrams, and flowcharts, were designed, and each is designed for specific modeling purposes.

##### 4.1. Software Development Tools

The software tools used in the development of the prototypes of the G-MoMo applications include the following:

- Vue JS Framework

The front-end of the native G-MoMo applications were developed using the Vue JS framework. Vue is a progressive JavaScript framework for building simplistic user-interface design. It is a model-view-view model (MVVM) patterned framework for building a lightweight front-end of the applications [85,86]. Vue JS enables the creation of fast and reliable applications, improves the application's development efficiency, maintains and integrates with other libraries and existing projects, and has a well-thought-out architecture to increase performance and reduce memory consumption [85–88].

- Python

The backend is implemented using Python. Python is an interpreted high-level, general-purpose, and object-oriented programming language designed to rapidly prototype complex applications. It has an extensive and comprehensive standard library, platform-independent, and automatic memory controls [89].

- MySQL

The backend databases are created using MySQL, an open-source database management system used to organize, store, retrieve and manage records. MySQL is chosen because of its compatibility with different operating systems, scalability, robustness, accessibility, reliability, backup and recovery utilities, flexibility to implement many features to improve system performance, encryption and decryption functions to protect sensitive data. It also handles large databases efficiently and delivers reliable and high-performance applications [89–91].

#### ○ Twilio programmable SMS

Twilio is a cloud communication platform that allows users to send and receive text messages with the help of its web service API [92]. The G-MoMo applications send mobile money IT support staff, agents, and the user's phone number to the Twilio API to send OTP messages. The OTP sent from Twilio via SMS to IT support staff, agents, and users during authentication are four (4) digits and is valid for only 60 s. The G-MoMo applications automatically detect the OTP upon receipt, and the Twilio API also keeps track of the sent OTP messages.

#### 4.2. Implementation of the G-MoMo Applications Prototypes

The researchers developed prototypes for the four native G-MoMo applications. In this section, we illustrate the native G-MoMo Customer Application prototype. The developed prototype for the G-MoMo Customer Application consists of registration, authentication, transactions, account management, and system logout as the main features. The transactions involve the mobile money customer using the G-MoMo Customer Application to send money, pay bills, check balance, and check mini statements. It only happens after a successful registration.

##### (a) Mobile Money Customer Registration

One of the main tasks of a mobile money agent is to register new mobile money customers. The mobile money customer registration process begins with the mobile money customer downloading and installing the G-MoMo SIM Serial Number Generator and G-MoMo Customer Applications in their smartphones. They then run the G-MoMo SIM Serial Number Generator Application to generate their SIM Serial Number and Subscriber ID, which the mobile money agent requires during customer enrolment. The mobile money agent will follow steps in Figure 7a–c to register mobile money customers by capturing their first name, last name, SIM serial number, subscriber ID, and phone number.

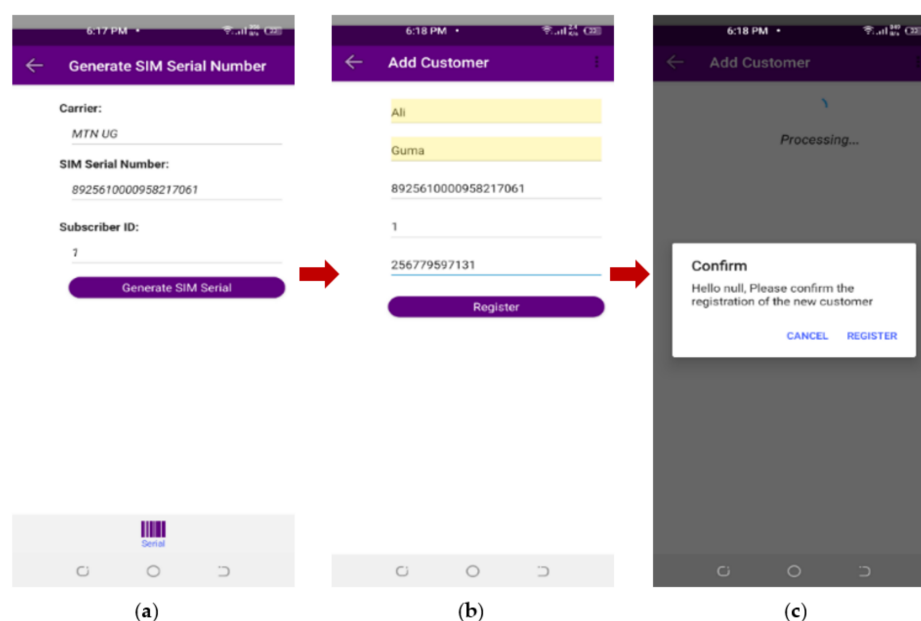
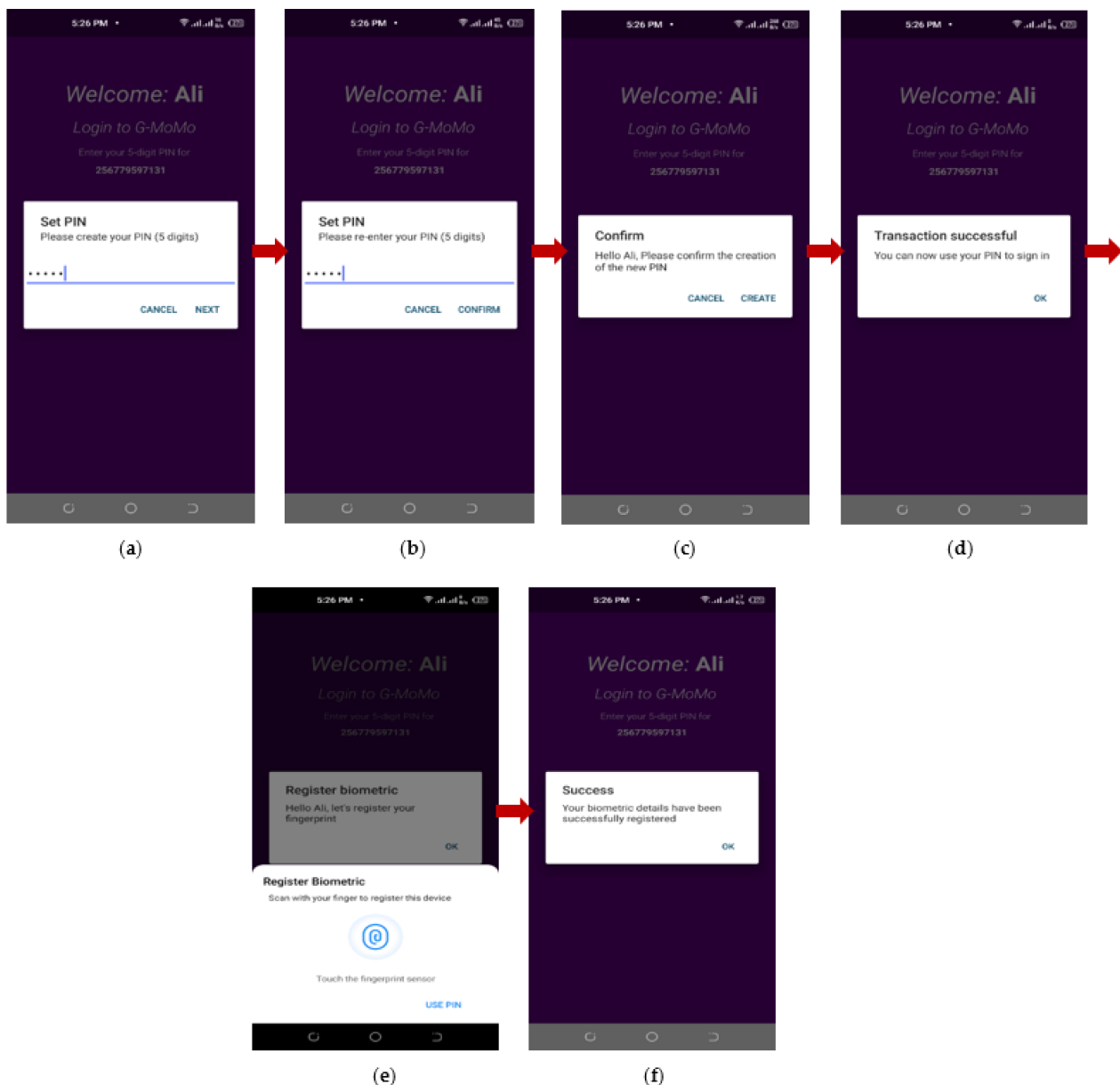


Figure 7. (a–c) Shows the mobile money customer registration process using the G-MoMo Agent Application.

After successful registration by the mobile money agent using the G-MoMo Agent Application, the mobile money customer must complete the registration process by running the G-MoMo Customer Application in their smartphone, entering their five-digit PIN, and re-entering the five-digit PIN again. If the five-digit PIN entered is the same as the re-entered PIN, the customer will be requested to confirm the creation of the new PIN. Once the mobile money customer approves, the system will ask him to register his biometric fingerprint by scanning it using the biometric fingerprint sensor of the smartphone. If the biometric fingerprint is successfully captured, the mobile money customer is enrolled and can now use the G-MoMo Customer Application. Figure 8a–f shows the process of completing mobile money customer registration using the G-MoMo Customer Application.



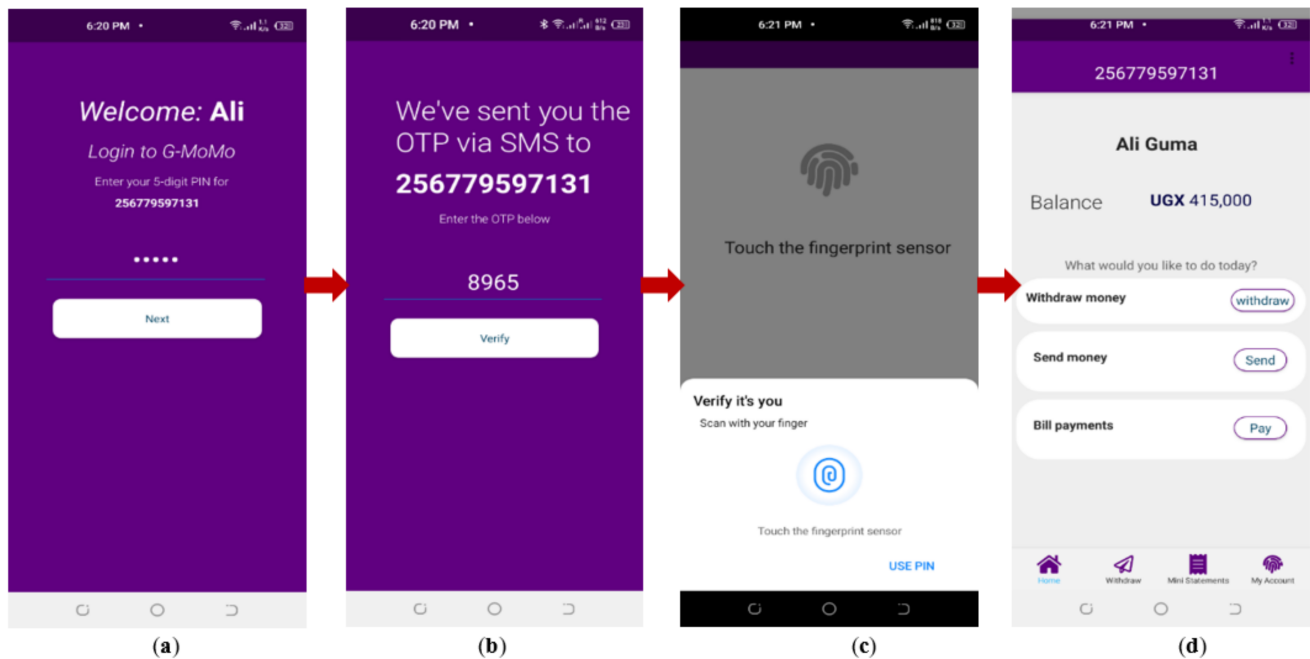
**Figure 8.** (a–f) Shows the process of completing mobile money customer registration using the G-MoMo Customer Application.

#### (b) Mobile Money Customer Authentication Process

Following the successful registration, the mobile money customer can log in to the system by entering their five-digit mobile money PIN. If the PIN matches, four-digit OTP is generated and sent to the customer via SMS. After receiving the OTP, it is automatically



detected and compared with the copy stored in the database. If it matches, the mobile money customer must scan the fingerprint for recognition. If the scanned fingerprint matches, the customer successfully logs in to the system and is presented with the menu. Figure 9a–d shows the mobile money customer authentication process.



**Figure 9.** (a–d) Shows the mobile money customer authentication process using the G-MoMo Customer Application.

#### (c) Money Withdrawal

To withdraw money using the G-MoMo Customer Application, the mobile money customer must register and log in to the application. He must check his balance and ensure that he has an electronic balance that is withdrawable. The system will request him to enter any amount less than or equal to the available balance. Once he enters the amount, the system will ask him to scan his fingerprint for authorization. If the scanned fingerprint matches, the customer is required to scan the secure QR code of the mobile money agent for confirmation. A successful withdrawal message seeking the customer to collect money from the mobile money agent is displayed. Figure 10a–f shows the customer money withdrawal process.

#### (d) Encrypted Records in the Tables

Figure 11 displays the transaction table, whose records are encrypted using the Fernet encryption algorithm.

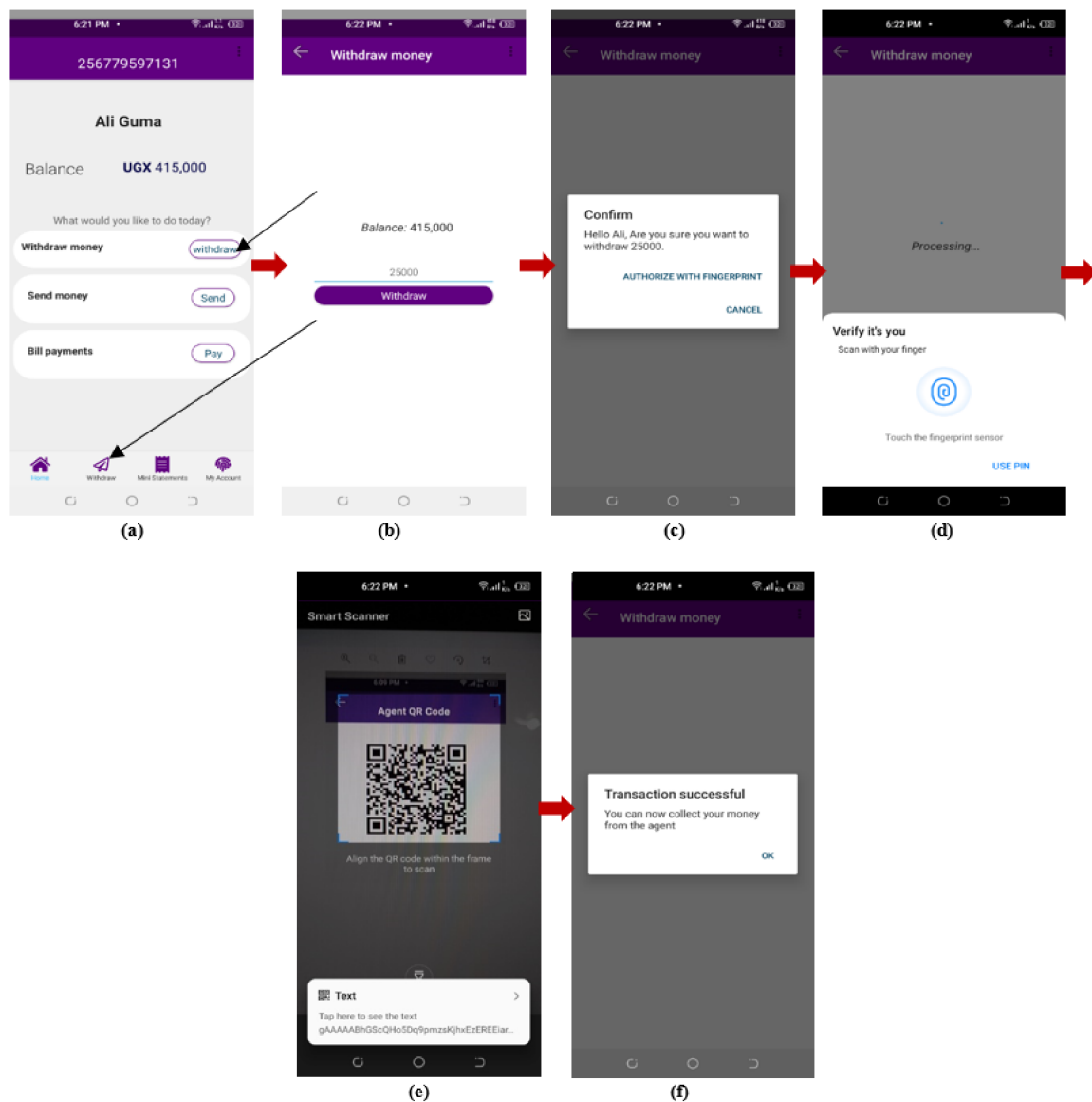


Figure 10. (a–f) Shows the customer money withdrawal process.

```
mysql> select * from transaction;
```

transaction_id	amount	user_id	date_of_record	transaction_type
74	gAAAAABhFSGsSLzpcnuLqCmPgEL36LFf6T8kPzkWxIuCNyJc3Ir2rIHoLnLU9h6VgnKyGfGJtgb4MfkfDa39gFu9r0fZ_sv0Q==	gAAAAABhFSGsvhNs0uTRGQjabJ	9   2021-08-12 13:27:09	
GULQaePdtQzvr	q8RJ2xbXuN0-3VFPtFj4sg8vhQzeF39YsqwnsgrLcws0LsdHPiJ4Rz2X-g==	9	2021-08-12 13:27:09	
75	gAAAAABhFSIJ6Icuuud0JVeSgWAS4zDQW-r92c8IGmTp04ipVopmaEoUMwiIGL-VCDZtIY4I6HafXttBNaSzJ7s1gx1ZsC27A==	gAAAAABhFSIJuy7f2v1eNGVhAB	9   2021-08-12 13:29:08	
FXxjBZzzSFj0yU-w0nQpnisdskB3HTIiftP1PD7zdhldtkdY30haD0R5YHVB9AdaRaPc3CQ==	9	2021-08-12 13:29:08		
76	gAAAAABhFSJ4F5dCqnELCmus8LTVBh10Uxgl9dCEg0Z5uU7a6XzdcPvLVjk1vqgRY3IoeLurFLGaS_y9v2y1_5pYgVwJTCm2IA==	gAAAAABhFSJ4_CAKXmzsW59wBJ	9   2021-08-12 13:30:33	
G9XGr3BHbuT1MkFyhCryFFoTxc72CKYjJPfVfu_vfiPls88SHsESpsWusHa9ZcC1sKfQZ_Sw==	9	2021-08-12 13:30:33		
77	gAAAAABhFSKs9qpxkCVPBzhmqR56SsZKwof6v3KxCzpxaEh3v1hB1z0Tn3_UC5RK2GkGMMD33gashV0Y-MJ0t8d0bggin5Avg==	gAAAAABhFSKetvXX_efYRrs3pn	9   2021-08-12 13:31:10	
RI4uZJyGD6vxc7jTi2S9khGnZTyZwYa2dAZ_AnaN16zIY-eanFPWqHbzzaLnC2cr1gQNViw==	9	2021-08-12 13:31:10		

4 rows in set (0.00 sec)

Figure 11. Records in the transaction table are encrypted using the Fernet encryption algorithm.

## 5. Security Analysis

The developed native G-MoMo applications were tested using a black-box testing approach. Both functional and non-functional testing was carried out to ensure that the applications met all functional and quality requirements to minimize the risk associated with the development process of the applications. After successfully testing the prototypes

of native G-MoMo applications, the security analysis proves that our proposed algorithm is safe and secure against well-known security attacks, as discussed below.

- ⇒ Provides secure and efficient authentication: The proposed multi-factor authentication algorithm uses a PIN, OTP, and biometric fingerprint. The security of the PIN and OTP are ensured by SHA-256 and the biometric fingerprint by FIDO, which provides secure and robust authentication [14,17,46,62,63,65,68–71]. Besides, using a mobile money agent's secure QR code in authorizing money withdrawal guarantees authenticity [81,83].
- ⇒ Provides data confidentiality: The proposed algorithm offers confidentiality by hashing PIN and OTP using SHA-256, securing the biometric fingerprint by FIDO that uses the RSA encryption algorithm, and Fernet encryption to secure the mobile money agent code and the records in the databases. At the same time, FIDO helps secure communication channels between the application server and the mobile money IT support staff, agent, and user using RSA. It helps to secure authentication credentials from being intercepted by attackers [21,46,63,65,66,93]. Furthermore, using Fernet encryption to secure QR codes in G-MoMo applications ensures data confidentiality [46,78].
- ⇒ Provides data integrity: The G-MoMo applications database records are protected using Fernet encryption. Likewise, the authentication credentials, such as PIN and OTP, are secured using SHA-256, and a biometric fingerprint using FIDO, which makes it difficult for the adversaries to modify or alter and insert new data and read their content both in storage and in transit, hence maintaining the integrity of the data [21,46,65]. Moreover, implementing secure QR codes in G-MoMo applications where the mobile money agent's serial number is encrypted using Fernet encryption to generate a mobile money agent code, which is then encoded in a QR code for the user to scan during money withdrawal, helps protect data integrity [46,78].
- ⇒ Ensures non-repudiation: During mobile money registration, the IT support staff, agents, and users avail their biodata, phone number, SIM serial number, PIN, and biometric fingerprint. The phone numbers uniquely identify the subscriber. When OTP is sent to the user's phone number during authentication, he should not deny receiving the sent OTP because a copy of the sent OTP is saved in the user's table, linked to the phone number. Likewise, a phone number ensures that no mobile money user can deny performing transaction(s) since every transaction is traced [21,65,66,93]. Still, using a secure QR code to store the mobile money agent code that the mobile money customer scans during mobile money withdrawal helps to ensure non-repudiation by the mobile money agent [46].
- ⇒ Ensures anonymity: Multi-factor authentication ensures user anonymity by entering a unique PIN and biometric fingerprints that uniquely identify them. There is no physical contact between mobile money agents and users and the mobile money service provider in mobile payments. Therefore, only the mobile money payment gateway has records that can trace and identify them [46,65,66].
- ⇒ Ensures privacy: The biometric fingerprint is protected using FIDO that uses RSA to protect public/private key pairs and biometric templates. The database records are also secured with Fernet encryption, thus protecting the credentials and transaction privacy and the privacy of the mobile money IT support staff, agents, and users [21,63,73]. In addition, the use of a secure QR code to store an encrypted mobile money agent code helps to ensure privacy [74].
- ⇒ Prevention of shoulder-surfing attacks: The current 2FA for mobile money uses only PIN and SIM to authenticate users, which is insufficient to provide strong security. Therefore, this approach is vulnerable to shoulder-surfing attacks because the PINs are entered when they are unmasked. However, using multiple identifiers, such as PIN, OTP, biometric fingerprint, and a secure QR code, in the proposed algorithm helps to prevent shoulder-surfing attacks [13,20,23,46].
- ⇒ Prevention of social engineering attacks: Social engineering is a typical security challenge in mobile money where attackers persuade mobile money agents and users

by calling them to reveal their mobile money PINs. This security challenge is solved in the proposed algorithm by implementing multi-factor authentication where users have to avail multiple identifiers such as PIN, OTP, and biometric fingerprint to verify them. Even if the attackers obtain the PIN, it will not be easy to guess the next OTP since it is generated randomly and only valid for 60 s. It is also challenging to get the biometric fingerprints because it is secured using FIDO, where public/private key pairs are generated. The private key and biometric templates are kept in the user's smartphone and the public key in the FIDO database.

- ⇒ Prevention of phishing attack: A phishing attack is where adversaries disguise themselves as the staff of the mobile money service provider by sending messages or voice calls to mobile money agents and users requesting them to avail their mobile money PINs. Once they succeed in obtaining the user's PIN, they can perform fraudulent transactions. The proposed algorithm solves this problem by implementing multi-factor authentication using a PIN, OTP, and biometric fingerprint. The OTP is random and unique and only valid for 60 s. The biometric fingerprint is secured using FIDO that uses RSA to generate public/private key pairs. The generated public key is encrypted and sent to the online FIDO database associated with the user's mobile money account and saved in a Keystore. The private key and biometric templates are encrypted and stored in the smartphone's cryptographic Keystore and smartphone [20,62,68–71]. Moreover, integrating a secure QR code in mobile money agent authorization helps prevent phishing attacks [77].
- ⇒ Prevention of PIN-guessing attack: The current 2FA scheme for mobile money uses four- or five-digit PINs to authenticate users, making it easy for the attacker to guess because of their simplicity. This attack is mitigated in the proposed algorithm using additional authentication factors such as OTP and biometric fingerprint. The OTP sent to users is four digits and is randomly generated, making it difficult for the attacker to guess the next OTP. Besides, the OTP is hashed using SHA-256 and encrypted using Fernet before saving it in the database. Furthermore, biometric fingerprints are secured using FIDO that uses RSA, making it complicated for the adversaries to guess the public and private keys used. The biometric template is encrypted and stored in the user's smartphone [20].
- ⇒ Prevention of brute-force attack: Attackers are familiar with PIN-based authentication systems, and most of the PINs used are easy to crack. Therefore, even if the PIN and OTP are cracked, it becomes difficult to break the FIDO system because it uses RSA to encrypt public/private key pairs and biometric templates. The encrypted private key and biometric template are stored in the user's smartphone, and the encrypted public key is stored in the FIDO database. Similarly, the secure QR code to store the encrypted mobile money agent code that the customer scans to confirm mobile money withdrawal helps prevent brute-force attacks [20,46].
- ⇒ Resistance to replay attacks: In the current 2FA scheme for mobile money, the adversaries delay or replay the authentication process several times to allow mobile money agents and users to enter their mobile money PIN several times until the attacker gets hold of the PIN. In the proposed algorithm, the researchers implemented a multi-factor authentication scheme where users must enter multiple identifiers, such as PIN, OTP, and biometric fingerprint, to verify themselves, which helps prevent replay attacks. In addition, the PIN and OTP are secured using SHA-256 and biometric fingerprints by FIDO, where the public/private key pair and biometric templates are encrypted using RSA, and the mobile money agent code stored in the QR code is secured by Fernet encryption. Protecting the authentication identifiers makes it difficult for the adversaries to perform replay attacks [62,66,68–71,93].
- ⇒ Resistance to insider attacks: Insider attacks are perpetrated by the employees and former employees of the mobile money service providers since they have inside information about the mobile money systems and access to mobile money agents and users' PIN and transaction data. They take advantage of being within the orga-

nization to perform fraudulent transactions at the expense of users. The proposed algorithm mitigates this attack by using Fernet encryption to secure records in the database [66,73].

- ⇒ Resistance to impersonation attacks: This attack is easy because attackers are familiar with the current 2FA scheme for mobile money. In the proposed algorithm, impersonation attacks are mitigated by (1) registering and identifying users with their phone number since no two persons will have the same phone number and (2) implementing multi-factor authentication involving PIN, OTP, and biometric fingerprints. Additionally, we secure the authentication identifiers, such as PIN and OTP, using SHA-256, biometric fingerprint by FIDO, a QR code, and records in the database using Fernet encryption [23,66].
- ⇒ Resistance to identity fraud: The current 2FA scheme for mobile money is prone to identity fraud. However, this threat model is mitigated in the proposed algorithm by implementing a solid and robust multi-factor authentication scheme using the PIN, OTP, biometric fingerprint, and secure QR [13,19,21,46]. Moreover, protecting the authentication identifiers, such as PIN and OTP, using SHA-256, biometric fingerprint by FIDO, a QR code, and records in the database using Fernet encryption helps to mitigate identity fraud [23,66].
- ⇒ Resistance to man-in-the-middle (MITM) attack: With the current 2FA scheme for mobile money, attackers can intercept communication between mobile money agents, users, systems, and banks. Attackers secretly steal the authentication credentials, such as a PIN or personal transaction information, spy on the user, or interrupt communications. This attack is prevented in the proposed algorithm by securing the PIN and OTP using SHA-256, biometric fingerprint by FIDO, QR code, and records in the database by Fernet encryption [20,62,68–71,93]. Likewise, the authentication identifiers, such as PIN and OTP, and public keys are hashed and encrypted before transmitting them online.

## 6. Performance Analysis

The proposed algorithm contains three main phases, namely, enrolment, authentication, and transaction. These phases are taken into consideration for performance comparison purposes. The performance of the proposed algorithm is evaluated by analyzing the communication overhead and computational cost with existing related algorithms, which helps to understand the efficiency of the proposed algorithm.

### 6.1. Communication Overhead

Communication overhead is associated with estimating the number of bytes in every communication message exchanged in the enrolment, authentication, and transaction phases. It is calculated by computing the number of bytes transmitted at the time as the message exchange process. Each packet size is calculated by summing the size of each message using the information in Table 2, and the results are shown in Table 3. In addition, it excludes the size of symmetric and asymmetric encryption.

As shown in Table 3, fifteen (15) messages are exchanged during user enrolment, authentication, and transaction phases in the proposed algorithm. Hence, the communication overhead is the sum of computing the bytes of these messages. The proposed algorithm has a high communication overhead, with a total of 744 bytes, as compared to the relevant related schemes in [15,16,19–21,93]. However, it provides strong security against many security attacks, as discussed in Section 5.



**Table 2.** Description and length in bytes of the different symbols used in the proposed algorithm.

Symbols	Meaning	Length (Bytes)
$U_i$	User	8
$FN_i$	User's first name	16
$LN_i$	User's last name	16
$SIM_{sn}$	SIM serial number	16
$SID_i$	Subscriber ID	8
$PN_i$	User's phone number	16
$PIN_i$	User's PIN	8
$PIN_j$	Re-entered PIN	8
$OTP_i$	User's OTP	8
$BF_i$	User's biometric fingerprint	16
$B_t$	Biometric template	16
$P_i$	User's public key	32
$F_i$	User's private key	32
$SP_i$	User's smartphone	8
$ID_i$	User's ID	8
$ID_{sp}$	Smartphone ID	8
$h(.)$	One-way hash function—SHA-256	32
$E_u(.) / D_u(.)$	Fernet Encryption/Decryption with key $u$	16
$E(.) / D(.)$	Public key Encryption/Decryption—RSA	256
$DB_m$	Main database	256
$DB_{fd}$	FIDO database	256
$Bal_i$	User's electronic balance	16
$A_a$	Agent	8
$QRcode_a$	Agent QR code	16
$Amt_i$	Amount	16

**Table 3.** Calculation of message sizes for messages exchanged during user enrolment, authentication, and transaction.

Phase	Message Content	Message Size (Bytes)
Registration	$\{FN_i, LN_i, SIM_{sn}, SID_i, PN_i\}$	$16 + 16 + 16 + 8 + 16 = 72$ bytes
	$\{PIN_i, PIN_j\}$	$8 + 8 = 16$ bytes
	$\{h(PIN_k), P_i, ID_i, ID_{sp}\}$	$40 + 32 + 8 + 8 = 88$ bytes
	$\{FN_i, LN_i, SIM_{sn}, SID_i, PN_i, h(PIN_k), P_i, ID_i, ID_{sp}\}$	$16 + 16 + 16 + 8 + 16 + 40 + 32 + 8 + 8 = 160$ bytes
Authentication	$\{ID_i, ID_{sp}, PIN_i\}$	$8 + 8 + 8 = 24$ bytes
	$\{ID_i, OTP_i\}$	$8 + 8 = 16$ bytes
	$\{ID_i, ID_{sp}, OTP_i\}$	$8 + 8 + 8 = 24$ bytes
	$\{BF_i(F_i, B_t)\}$	$16 + 32 + 16 = 64$ bytes
	$\{ID_i, ID_{sp}, P_i\}$	$8 + 8 + 32 = 48$ bytes
Transaction	$\{ID_i, Bal_i\}$	$8 + 16 = 24$ bytes
	$\{ID_i, Amt_i\}$	$8 + 16 = 24$ bytes
	$\{BF_i(F_i, B_t)\}$	$16 + 32 + 16 = 64$ bytes
	$\{ID_i, ID_{sp}, P_i\}$	$8 + 8 + 32 = 48$ bytes
	$\{ID_i, QRcode_i\}$	$8 + 16 = 24$ bytes
	$\{ID_i, ID_{sp}, QRcode_i\}$	$8 + 8 + 16 = 48$ bytes

## 6.2. Computational Cost

ElGhanam et al. [94] defined computational cost as “the time taken by the network entities to execute the different cryptographic techniques” (p. 12). The total computational cost of each phase in the proposed secure and efficient multi-factor authentication algorithm is analyzed based on message exchange sequences described in Figure 5 and compared with the computational cost of the scheme proposed in [93], as shown in Table 4. It is assumed that the notation  $T_h$  is the time for one-way hashing operation,  $T_{Re}$  and  $T_{Rde}$  are the time

durations to encrypt and decrypt messages using RSA, and  $T_{Fe}$  and  $T_{Fde}$  are the time durations to encrypt and decrypt the message using Fernet.  $T_h$  and  $T_{ECC}$  denote the times for a hash operation and the encryption/decryption operations in ECC-160, respectively, used in the algorithm proposed in [93]. It should be noted that the schemes [15,16,19,20] never used any cryptographic techniques to secure the data. In addition, the algorithm in [93] is mainly selected for comparison with the proposed algorithm because it uses SHA-1 as a one-way hash function and ECC for encrypting/decrypting the data to achieve data security. However, the enrolment, authentication, and transaction are different between the two algorithms, and the matching computational efficiencies are compared accordingly.

**Table 4.** Computational cost calculation for an authentication phase.

Proposed Algorithm	Authentication Phase	Transaction Phase (Cash Withdrawal)
Our Algorithm	$3T_h + 1T_{Fde} + 3T_{Rde} + 1T_{Fe}$	$2T_{Fde} + 1T_{Rde} + 1T_{Fde} + 1T_{Fe}$
[93]	$2T_h + 1T_{ECC}$	$2T_h + 3T_{ECC}$

It was observed that the proposed algorithm provides better security than the schemes in [15,16,19–21,93]. The result also shows that the proposed algorithm has a higher computation cost than the related scheme [93] because of the cryptographic functions (i.e., SHA-256, RSA, and Fernet encryption/decryption operations). Contrastingly, the algorithm in [93] only uses SHA-1 and ECC and stores the user's biometric template in the database, which takes time to retrieve compared to our proposed algorithm. The encrypted private key and biometric template are stored in the smartphone. Therefore, the proposed algorithm design in securing mobile money proves to be a reliable algorithm because of its efficiency.

The proposed algorithm has a high communication overhead and computational cost compared to the existing schemes but provides strong security against many security attacks. This is because it uses multiple identifiers, such as PIN, OTP, biometric fingerprint, and a QR code, for user authentication compared to existing schemes that only use two factors to authenticate mobile money users. Further, the proposed algorithm uses cryptographic techniques such as SHA-256 to secure PIN and OTP, FIDO, which uses the public-key cryptography technique (RSA encryption) to secure biometric fingerprint, and Fernet encryption to secure a QR code and the records in the databases. These cryptographic techniques take extra time during hashing and encrypting/decrypting data. However, most of the existing schemes, such as [13–19], did not implement cryptographic techniques.

The user enrolment, authentication, and transaction processes are simple and easy to use for novice users since the interfaces for the native G-MoMo application prototypes are simple, much like the existing schemes.

## 7. Comparison with Other Related Works

### 7.1. Security Features

The security features of the proposed algorithm are compared with other schemes, as summarized in Table 5.

### 7.2. Performance Comparisons

The enrolment and authentication phases of the proposed algorithm are compared with existing schemes in terms of calculation costs, i.e., the time for one-way hashing operation ( $T_h$ ), the time durations to encrypt and decrypt messages using RSA ( $T_{Re}$  and  $T_{Rde}$ ), the time durations to encrypt and decrypt the message using Fernet ( $T_{Fe}$  and  $T_{Fde}$ ), and the time durations to encrypt and decrypt messages using ECC ( $T_{ECC}$ ), as shown in Table 6.

**Table 5.** Comparison of the security features of the proposed algorithm with other related schemes.

S/No	Security Feature	[15]	[16]	[19]	[20]	[21]	[93]	Our Algorithm
1	Provides efficient authentication	No	Yes	No	No	No	No	Yes
2	Provides data confidentiality	No	No	No	No	Yes	Yes	Yes
3	Provides data integrity	No	No	No	No	Yes	No	Yes
4	Ensures non-repudiation	No	No	No	No	Yes	Yes	Yes
5	Ensures anonymity	No	No	No	No	No	No	Yes
6	Ensures privacy	No	No	No	No	Yes	No	Yes
7	Prevention of shoulder-surfing attacks	No	No	No	Yes	No	No	Yes
8	Prevention of social engineering	No	No	No	No	No	No	Yes
9	Prevention of phishing attack	No	No	No	Yes	No	No	Yes
10	Prevention of PIN-guessing attack	No	No	No	Yes	No	No	Yes
11	Prevention of brute-force attack	No	No	No	Yes	No	No	Yes
12	Resistance to replay attacks	No	No	No	No	No	Yes	Yes
13	Resistance to insider attacks	No	No	No	No	No	No	Yes
14	Resistance to impersonation attacks	Yes	Yes	No	No	No	No	Yes
15	Resistance to identity fraud	No	No	Yes	No	Yes	No	Yes
16	Resistance to a MITM attack	No	No	No	Yes	No	Yes	Yes

**Table 6.** Computational cost comparison.

Proposed Algorithm	Authentication Phase	Transaction Phase (Cash Withdrawal)	Total
<b>Our Algorithm</b>	$3T_h + 1T_{Fde} + 3T_{Rde} + 1T_{Fe}$	$2T_{Fde} + 1T_{Rde} + 1T_{Fde} + 1T_{Fe}$	$3T_h + 2T_{Fde} + 4T_{Rde} + 2T_{Fe}$
[93]	$2T_h + 1T_{ECC}$	$2T_h + 3T_{ECC}$	$4T_h + 4T_{ECC}$

## 8. Conclusions

Advancements in smartphone technology and mobile telecommunication networks with easy access have increased mobile money services in many developing countries. Such advancements have also introduced the financial sector to a new form of banking using technology, thus promoting and widening mobile money service coverage. However, there are many security challenges related to mobile money authentication schemes because the current systems use only PIN and SIM to authenticate mobile money users. The researchers proposed a secure and efficient multi-factor authentication algorithm for mobile money applications to address these issues. It uses PIN, OTP, and biometric fingerprints as authentication identifiers. It also involves mobile money users scanning the biometric fingerprint and secure QR code of the mobile money agents to confirm mobile money withdrawal. The security of the PIN and OTP is ensured by SHA-256, a biometric fingerprint by FIDO that uses RSA and Fernet encryption to secure QR codes and the records in the databases.

Four prototypes of native G-MoMo applications were developed, i.e., G-MoMo SIM Serial Number Generator Application, G-MoMo IT Support Application, G-MoMo Agent Application, and G-MoMo Customer Application. The G-MoMo applications were developed for mobile money IT support staff, agents, and users to prove the feasibility of the proposed algorithm and provide a higher degree of security. The security analysis confirmed that the proposed algorithm offers secure and efficient authentication; ensures data confidentiality, integrity, non-repudiation, user anonymity, and privacy; and improves performance. It provides prevention to shoulder-surfing attacks, social engineering attacks, phishing attacks, PIN-guessing attacks, and brute-force attacks. Likewise, it is

resistant to replay attacks, insider attacks, impersonation attacks, identity fraud, and MITM attacks. This study, therefore, recommends heuristic evaluation and usability testing of G-MoMo applications to identify any usability problems that arise so that recommendations for improvements can be made to achieve effectiveness, efficiency, and user satisfaction. The proposed algorithm will benefit mobile money IT support staff, agents, users, mobile money service providers, government, researchers, mobile application developers, decision-makers, and policymakers who wish to see a secure mobile money transaction.

**Author Contributions:** Data curation, G.A.; formal analysis, M.A.D. and A.E.S.; investigation, G.A.; methodology, G.A.; supervision, M.A.D. and A.E.S.; writing—review and editing, G.A., M.A.D. and A.E.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Not applicable, the study does not report any data.

**Acknowledgments:** The authors are grateful to the anonymous reviewers for their constructive and valuable comments. We are also thankful to NM-AIST and Muni University for helping us conduct this research.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Hendershott, T.; Zhang, X.; Zhao, J.L.; Zheng, Z. FinTech as a Game Changer: Overview of Research Frontiers. *Inf. Syst. Res.* **2021**, *32*, 1–17. [\[CrossRef\]](#)
- Barbu, C.M.; Florea, D.L.; Dabija, D.-C.; Constantin, M.; Barbu, R. Customer Experience in Fintech. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *16*, 1415–1433. [\[CrossRef\]](#)
- Dharmadasa, P.D.C.S. “Fintech Services” and the Future of Financial Intermediation: A Review. *Sri Lanka J. Econ. Res.* **2021**, *8*, 21–38. [\[CrossRef\]](#)
- Langley, P.; Leyshon, A. The Platform Political Economy of FinTech: Reintermediation, Consolidation and Capitalisation. *New Polit. Econ.* **2020**, *26*, 376–388. [\[CrossRef\]](#)
- Lu, B.; Hao, S.; Pinedo, M.; Xu, Y. Frontiers in Service Science: Fintech Operations—An Overview of Recent Developments and Future Research Directions. *Serv. Sci.* **2021**, *13*, 19–35. [\[CrossRef\]](#)
- Cornelli, G.; Frost, J.; Gambacorta, L.; Rau, R.; Wardrop, R.; Ziegler, T. Fintech and Big Tech Credit: What Explains the Rise of Digital Lending? *CESifo Forum* **2021**, *22*, 30–34.
- Sharma, Y. Mobile Payments Market—Global Opportunity Analysis and Industry Forecast, 2014–2022. 2017. Available online: [www.alliedmarketresearch.com/mobile-payments-market](http://www.alliedmarketresearch.com/mobile-payments-market) (accessed on 10 May 2020).
- Jakhiya, M.; Bishnoi, M.M.; Purohit, H. Emergence and growth of mobile money in modern India: A study on the effect of mobile money. In Proceedings of the 2020 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, 4 February–9 April 2020; pp. 1–10.
- Ali, G.; Dida, M.A.; Sam, A.E. Evaluation of key security issues associated with mobile money systems in Uganda. *Information* **2020**, *11*, 309. [\[CrossRef\]](#)
- Egami, H.; Matsumoto, T. Mobile money use and healthcare utilization: Evidence from rural Uganda. *Sustainability* **2020**, *12*, 3741. [\[CrossRef\]](#)
- Ali, G.; Dida, M.A.; Sam, A.E. Two-factor authentication scheme for mobile money: A review of threat models and countermeasures. *Future Internet* **2020**, *12*, 160. [\[CrossRef\]](#)
- Basigie, A.; Mtaho, L. Securing Mobile Money Services in Tanzania: A Case of Vodacom M-Pesa. *Int. J. Comput. Sci. Netw. Solut.* **2014**, *2*, 1–11.
- Mtaho, B.A. Improving Mobile Money Security with Two-Factor Authentication. *Int. J. Comput. Appl.* **2015**, *109*, 9–15.
- Mega, B. Framework for Improved Security on Usage of Mobile Money Application Based on Iris Biometric Authentication Method in Tanzania. Master’s Thesis, The University of Dodoma, Dodoma, Tanzania, 2020.
- Islam, I.; Munim, K.M.; Islam, M.N.; Karim, M.M. A proposed secure mobile money transfer system for SME in Bangladesh: An industry 4.0 perspective. In Proceedings of the 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 24–25 December 2019; pp. 1–6.
- Chetalam, J.L. Enhancing Security of MPesa Transactions by Use of Voice Biometrics. Master’s Thesis, The United States International University—Africa, Nairobi, Kenya, 2018.
- Osman, F.; Nakanishi, H. High Correctness Mobile Money Authentication System. *Int. J. Psychosoc. Rehabil.* **2020**, *24*, 3544–3556. [\[CrossRef\]](#)

18. Okpara, O.S.; Bekaroo, G. Cam-Wallet: Fingerprint-based authentication in M-wallets using embedded cameras. In Proceedings of the 2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I & CPS Europe), Milan, Italy, 6–9 June 2017; pp. 1–5.
19. Coneland, R.; Crespi, N. Wallet-on-wheels—Using a vehicle’s identity for secure mobile money. In Proceedings of the 2013 17th International Conference on Intelligence in Next Generation Networks (ICIN), Venice, Italy, 15–16 October 2013; pp. 102–109.
20. Hassan, M.A.; Shukur, Z. A secure multi factor user authentication framework for electronic payment system. In Proceedings of the 2021 3rd International Cyber Resilience Conference (CRC), Langkawi Island, Malaysia, 29–31 January 2021; pp. 1–6.
21. Vincent, O.R.; Okediran, T.M.; Abayomi-Alli, A.A.; Adeniran, O.J. An Identity-Based Elliptic Curve Cryptography for Mobile Payment Security. *SN Comput. Sci.* **2020**, *1*, 1–12. [\[CrossRef\]](#)
22. Castle, S.; Pervaiz, F.; Weld, G.; Roesner, F.; Anderson, R. Let’s talk money: Evaluating the security challenges of mobile money in the developing world. In Proceedings of the 7th Annual Symposium on Computing for Development (ACM DEV’16), New York, NY, USA, 18–20 November 2016; pp. 1–10.
23. Sharma, L.; Mathuria, M. Mobile banking transaction using fingerprint authentication. In Proceedings of the 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 19–20 January 2018; pp. 1300–1305.
24. Phipps, R.; Mare, S.; Ney, P.; Webster, J.; Heimerl, K. ThinSIM-based attacks on mobile money systems. In Proceedings of the COMPASS ’18: ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS), New York, NY, USA, 20–22 June 2018; pp. 1–11.
25. Alhassan, N.S.; Yusuf, M.O.; Karmanje, A.R.; Alam, M. Salami attacks and their mitigation—An overview. In Proceedings of the 5th International Conference on Computing for Sustainable Global Development, New Delhi, India, 14–16 March 2018; pp. 4639–4642.
26. Altwairqi, A.F.; AlZain, M.A.; Soh, B.; Masud, M.; Al-Amri, J. Four Most Famous Cyber Attacks for Financial Gains. *Int. J. Eng. Adv. Technol.* **2019**, *9*, 2131–2139.
27. Binbeshr, F.; Mat Kiah, M.L.; Por, L.Y.; Zaidan, A.A. A systematic review of PIN-entry methods resistant to shoulder-surfing attacks. *Comput. Secur.* **2021**, *101*, 102–116. [\[CrossRef\]](#)
28. AbouSteit, M.H.S.; Tammam, A.F.; Wahdan, A. A novel approach for generating one-time password with secure distribution. In Proceedings of the 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, UK, 27–28 July 2020; pp. 461–466.
29. Anusha, N.; Sai, A.D.; Srikar, B. Locker security system using facial recognition and One Time Password (OTP). In Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 22–24 March 2017; pp. 812–815.
30. Zadeh, M.J.; Barati, H. Security improvement in mobile banking using hybrid authentication. In Proceedings of the 3rd International Conference on Advances in Artificial Intelligence, Istanbul, Turkey, 26–28 October 2019; pp. 198–201.
31. Iftikhar, J.; Hussain, S.; Mansoor, K.; Ali, Z.; Chaudhry, S.A. Symmetric-key multi-factor biometric authentication scheme. In Proceedings of the 2nd International Conference on Communication, Computing and Digital Systems (C-CODE), Islamabad, Pakistan, 6–7 March 2019; pp. 288–292.
32. Devendra, S. The Significant Role of Smartphones in Improving Consumer’s Quality of Life. *Int. J. Adv. Res. Innov. Ideas Educ.* **2021**, *7*, 578–586.
33. Shaik, C. Preventing Counterfeit Products using Cryptography, QR Code and Webservice. *Comput. Sci. Eng. Int. J.* **2021**, *11*, 1–11. [\[CrossRef\]](#)
34. Kurniawan, I.; Sudaryanto, S.; Sukarno, H. The Shifting of or Code-Based Payment Method to Improve the Competitive Advantage (Ca) at Bank Jatim through Tam Model Approach. *IOSR J. Bus. Manag.* **2021**, *23*, 22–27.
35. Sabri, P.N.A.A.; Abas, A.; Din, R. Enhancing Data Storage of or Code Using C3M Technique. *Eur. J. Mol. Clin. Med.* **2020**, *7*, 3805–3813.
36. Cho, J.; Seo, G.W.; Lee, J.S.; Cho, H.K.; Kang, E.M.; Kim, J.; Chun, D.-I.; Yi, Y.; Won, S.H. The usefulness of the QR code in orthotic applications after orthopaedic surgery. *Healthcare* **2021**, *9*, 298. [\[CrossRef\]](#)
37. Chou, G.J.; Wang, R.Z. The Nested QR Code. *IEEE Signal Process. Lett.* **2020**, *27*, 1230–1234. [\[CrossRef\]](#)
38. Din, M.M.; Anwar, R.M.; Fazal, F.A. Asset tagging for library system-does QR relevant? In Proceedings of the International Conference on Applied and Practical Sciences ICAPS (2021), Kuala Lumpur, Malaysia, 18–19 February 2021; pp. 1–11.
39. Onyinyechi, O.P.; Ifeanyi, O.A.; Nnabuchi, E.N.; Nwakaego, I.P. Enhanced Business Marketing for Small Scale Enterprises Via the Quick Response Code Technology. *Frontiers* **2021**, *1*, 7–13.
40. Sun, L.; Liang, S.; Chen, P.; Chen, Y. Encrypted digital watermarking algorithm for quick response code using discrete cosine transform and singular value decomposition. *Multimed. Tools Appl.* **2021**, *80*, 10285–10300. [\[CrossRef\]](#)
41. Suebtimrat, P.; Vonguai, R. An Investigation of Behavioral Intention Towards QR Code Payment in Bangkok, Thailand. *J. Asian Financ. Econ. Bus.* **2021**, *8*, 939–950.
42. Kosim, K.P.; Legowo, N. Factors Affecting Consumer Intention on QR Payment of Mobile Banking: A Case Study in Indonesia. *J. Asian Financ. Econ. Bus.* **2021**, *8*, 391–401.
43. Widaningsih, S.; Suheri, A. Design of Waste Management System Using QR Code for Effective Management in Wastebank. *J. Phys.* **2021**, *1764*, 1–6. [\[CrossRef\]](#)



44. Chaveesuk, S.; Piyawat, N. Use of QR code technology in eastern Thailand: Entrepreneur perspective. *Utopia Prax. Latinoam.* **2021**, *26*, 76–88.
45. Tao, Y.; Cai, F.; Zhan, G.; Zhong, H.; Zhou, Y.; Shen, S. Floating quick response code based on structural black color with the characteristic of privacy protection. *Opt. Express* **2021**, *29*, 1–11. [\[CrossRef\]](#)
46. Ximenes, A.M.; Sukaridhoto, S.; Sudarsono, A.; Albaab, M.R.; Basri, H.; Yani, M.A.; Islam, E. Implementation QR code biometric authentication for online payment. In Proceedings of the 2019 International Electronics Symposium (IES), Surabaya, Indonesia, 27–28 September 2019; pp. 676–682.
47. Dasgupta, D.; Roy, A.; Nag, A. Biometric authentication. In *Advances in User Authentication*; Infosys Science Foundation Series; Springer: Cham, Switzerland, 2017.
48. Priya, S.P. Biometrics and Fingerprint Payment Technology. *Int. J. Adv. Res. Comput. Sci. Technol.* **2017**, *5*, 114–118.
49. Jain, A.K.; Nandakumar, K.; Ross, A. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognit. Lett.* **2015**, *79*, 80–105. [\[CrossRef\]](#)
50. Buciu, I.; Gacsadi, A. Biometrics Systems and Technologies: A survey. *Int. J. Comput. Commun. Control* **2016**, *11*, 315–330. [\[CrossRef\]](#)
51. Faridah, Y.; Nasir, H.; Kushsairy, A.K.; Safie, S.I.; Khan, S.; Gunawan, T.S. Fingerprint Biometric Systems. *Trends Bioinform.* **2016**, *9*, 52–58. [\[CrossRef\]](#)
52. Fingerprints. Biometric Technologies. 2017. Available online: <https://www.fingerprints.com/asset/assets/downloads/fingerprints-biometric-technologies-whitepaper-2017-revb.pdf> (accessed on 16 May 2021).
53. Wang, J.; Liu, G.; Chen, Y.; Wang, S. Construction and Analysis of SHA-256 Compression Function Based on Chaos S-Box. *IEEE Access* **2021**, *9*, 61768–61777. [\[CrossRef\]](#)
54. Zhang, Y.; He, Z.; Wan, M.; Zhan, M.; Zhang, M.; Peng, K.; Song, M.; Gu, H. A New Message Expansion Structure for Full Pipeline SHA-2. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 1553–1566. [\[CrossRef\]](#)
55. Al-Odat, Z.; Abbas, A.; Khan, S.U. Randomness analyses of the secure hash algorithms, SHA-1, SHA-2 and modified SHA. In Proceedings of the 2019 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 16–18 December 2019; pp. 316–321.
56. Nassr, D.I. Secure Hash Algorithm-2 formed on DNA. *J. Egypt. Math. Soc.* **2019**, *27*, 1–20. [\[CrossRef\]](#)
57. Aradhana, S.; Ghosh, S.M. Review Paper on Secure Hash Algorithm with Its Variants. *Int. J. Tech. Innov. Mod. Eng. Sci.* **2017**, *3*, 43–48.
58. Martino, R.; Cilaro, A. A Flexible Framework for Exploring, Evaluating, and Comparing SHA-2 Designs. *IEEE Access* **2019**, *7*, 72443–72456. [\[CrossRef\]](#)
59. Martino, R.; Cilaro, A. Designing a SHA-256 processor for blockchain-based IoT applications. *Internet Things* **2020**, *11*, 1–13. [\[CrossRef\]](#)
60. Sghaier, A.; Zeghid, M.; Massoud, C.; Machout, M. Design and implementation of low area/power elliptic curve digital signature hardware core. *Electronics* **2017**, *6*, 46. [\[CrossRef\]](#)
61. Panos, C.; Malliaros, S.; Ntantogian, C.; Panou, A.; Xenakis, C. A Security Evaluation of FIDO's UAF Protocol in Mobile and Embedded Devices. *Commun. Comput. Inf. Sci.* **2017**, *766*, 127–142.
62. Feng, H.; Li, H.; Pan, X.; Zhao, Z.; Cactilab, T. A formal analysis of the FIDO UAF protocol. In Proceedings of the Network and Distributed Systems Security (NDSS) Symposium 2021, San Diego, CA, USA, 21–25 February 2021; pp. 1–15.
63. Purnomo, A.T.; Gondokaryono, Y.S.; Kim, C.S. Mutual authentication in securing a mobile payment system using encrypted QR code based on Public Key Infrastructure. In Proceedings of the 2016 6th International Conference on System Engineering and Technology (ICSET), Bandung, Indonesia, 3–4 October 2016; pp. 194–198.
64. Sharma, N.; Bohra, B. Enhancing online banking authentication using the hybrid cryptographic method. In Proceedings of the 2017 3rd International Conference on Computational Intelligence and Communication Technology (CICT), Ghaziabad, India, 9–10 February 2017; pp. 1–8.
65. Hassan, M.A.; Shukur, Z.; Hasan, M.K. An efficient secure electronic payment system for e-commerce. *Computers* **2020**, *9*, 66. [\[CrossRef\]](#)
66. Mohit, P.; Amin, R.; Biswas, G.P. Design of secure and efficient electronic payment system for mobile users. In *International Conference on Mathematics and Computing*; Springer: Singapore, 2017; Volume 1, pp. 34–43.
67. Susanna, A.; David, S.; Kathrine, J.W.; Esther, A.G. Enhancing user authentication for mobile wallet using cryptographic algorithm. *J. Adv. Res. Dyn. Control Syst.* **2018**, *10*, 891–897.
68. Kim, H.; Jung, Y.; Jun, M. A Study on Secure Mobile Payment Service for the Market Economy Revitalization. *J. Korea Acad. Ind. Coop. Soc.* **2017**, *18*, 41–48.
69. Han, Z.; Yang, L.; Wang, S.; Mu, S.; Liu, Q. Efficient multi-factor two-server authenticated scheme under mobile cloud computing. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 1–14.
70. Shin, Y. Review of the suitability to introduce new identity verification means in South Korea: Focused on Block Chain and FIDO. *J. Conver. Inf. Technol.* **2018**, *8*, 85–93.
71. Canales, C. FIDO Alliance Overview. 2020. Available online: <https://novugens.com/wp-content/uploads/2020/03/ID37-FIDO-Alliance-2.pdf> (accessed on 2 April 2021).
72. Singh, A.; Singh, C.; Mishra, S. Enhanced Honey Encryption Algorithm on e-mail with Increased Message Space. *Int. J. Res. Eng. Sci. Manag.* **2020**, *3*, 453–456.

73. Dijesh, P.; Suvanamsasidhar, B.; Yellepeddi, V. Enhancement of e-commerce security through asymmetric key algorithm. *Comput. Commun.* **2020**, *153*, 125–134.
74. Chang, Y.-Y.; Yan, S.-L.; Lin, P.-Z.; Zhong, H.-B.; Marescaux, J.; Su, J.-L.; Wang, M.-L.; Lee, P.-Y. A mobile medical QR-code authentication system and its automatic FICE image evaluation application. *J. Appl. Res. Technol.* **2015**, *13*, 220–229. [\[CrossRef\]](#)
75. Asok, A.; Arun, G. QR Code Based Data Transmission in Mobile Devices Using AES Encryption. *Int. J. Sci. Res.* **2016**, *5*, 1116–1120.
76. Mittra, P.; Rakesh, N. A desktop application of QR code for data security and authentication. In Proceedings of the 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–27 August 2016; pp. 1–5.
77. Goel, N.; Sharma, A.; Goswami, S. A way to secure a QR code: SQR. In Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 5–6 May 2017; pp. 494–497.
78. Husny, H.R.M.; Binti, N.A.N.; Nizar, N.A.; Abdullah, N.Y.; Ismail, W.H.W. Encrypted QR Code System. *J. Comput. Technol. Creat. Content* **2017**, *2*, 82–92.
79. Soltani, M.; Bardsiri, A.K. Designing a Novel Hybrid Algorithm for QR-Code Images Encryption and Steganography. *J. Comput.* **2018**, *13*, 1075–1088. [\[CrossRef\]](#)
80. Ghodke, A.V.; Dagade, R.V. Electronic secure vehicle verification system using advanced Digi-locker system. In Proceedings of the 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, India, 6–8 April 2018; pp. 1–4.
81. Arief, A.T.; Wirawan, W.; Suprpto, Y.K. Authentication of printed document using quick response (QR) code. In Proceedings of the 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 28–29 August 2019; pp. 228–233.
82. Wahsheh, H.A.M.; Luccio, F.L. Security and privacy of QR code applications: A comprehensive study, general guidelines and solutions. *Information* **2020**, *11*, 217. [\[CrossRef\]](#)
83. Pramusinto, W.; Sartana, B.T.; Mulyati, S.; Amini, D.S. Implementation of AES-192 Cryptography and QR Code to Verify the Authenticity of Budi Luhur University Student Certificate. *J. Pendidik. Teknol. Kejuru.* **2020**, *3*, 209–215.
84. Carter, R.A.; Anton, A.I.; Dagnino, A.; Williams, L. Evolving beyond requirements creep: A risk-based evolutionary prototyping model. In Proceedings of the Fifth IEEE International Symposium on Requirements Engineering, Toronto, ON, Canada, 27–31 August 2001; pp. 94–101.
85. Song, J.; Xie, H. Design and Implementation of a Vue.js-Based College Teaching System How to Work with This Template. *Int. J. Emerg. Technol. Learn.* **2019**, *14*, 59–69. [\[CrossRef\]](#)
86. Yun, Q. Design and implementation of E-commerce platform based on Vue.js and MySQL. In Proceedings of the 3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019), Wuhan, China, 30–31 May 2019; pp. 449–454.
87. Kyriakidis, A.; Maniatis, K.; You, E. *The Majesty of Vue.js 2*; Lean Publishing: Victoria, BC, Canada, 2017.
88. Macrae, C. *Vue.js: Up and Running—Building Accessible and Performant Web Apps*; O'Reilly Media: Sebastopol, CA, USA, 2018.
89. Vyas, H.A.; Virparia, P.V. Template-Based Transliteration of Braille Character to Gujarati Text—The Application. *Rising Threat. Expert Appl. Solut.* **2021**, *1187*, 437–446.
90. Mpawe, N.M.; Mussa, A.D. A Web-based Monitoring and Evaluation System for Government Projects in Tanzania: The Case of Ministry of Health. *Eng. Technol. Appl. Sci. Res.* **2020**, *10*, 6109–6115.
91. Sadeq, M.J.; Rayhan, K.S.; Akter, M.; Forhat, R.; Haque, R.; Akhtaruzzaman, M. Integration of blockchain and remote database access protocol-based database. In Proceedings of the Fifth International Congress on Information and Communication Technology, London, UK, 20–21 February 2020; pp. 533–539.
92. Geetha, V.; Anbumani, V.; Selvi, T.; Sindhuja, C.S.; Vanathi, S. IoT based well-organized hostel power consumption and attendance administration system. In Proceedings of the International Virtual Conference on Robotics, Automation, Intelligent Systems and Energy (IVC RAISE 2020), Perundurai, India, 15 December 2020; pp. 1–9.
93. Ray, S.; Biswas, G.P.; Dasgupta, M. Secure Multi-Purpose Mobile-Banking Using Elliptic Curve Cryptography. *Wirel. Pers. Commun.* **2016**, *90*, 1331–1354. [\[CrossRef\]](#)
94. ElGhanam, E.; Ahmed, I.; Hassan, M.; Osman, A. Authentication and billing for dynamic wireless EV charging in an internet of electric vehicles. *Future Internet* **2021**, *13*, 257. [\[CrossRef\]](#)