Scientific Research Publishing

Journal of Computer and Communications, 2025, 13(7), 94-126
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

# A Reliable Multi-Factor Authentication Approach for Enhancing Security in Merchant Payment Transactions: Case Study of *Fapshi* Digital Wallet

**Tamo Karl Wilfried Djotchuang[1], Inès Raïssa Djouela Kamgang[1*], Théophile Fonzin Fozin[2*], Elie Fute Tagne[3]**

[1]Department of Computer Engineering, Faculty of Engineering and Technology, University of Buea, Buea, Cameroon
[2]Department of Electrical and Electronic Engineering, Faculty of Engineering and Technology, University of Buea, Buea, Cameroon
[3]Department of Mathematics and Computer Science, Faculty of Sciences, University of Dschang, Dschang, Cameroon
Email: *inesdjouela@gmail.com, *fozintheo@gmail.com

**How to cite this paper:** Djotchuang, T.K.W., Kamgang, I.R.D., Fozin, T.F. and Tagne, E.F. (2025) A Reliable Multi-Factor Authentication Approach for Enhancing Security in Merchant Payment Transactions: Case Study of *Fapshi* Digital Wallet. *Journal of Computer and Communications*, **13**, 94-126.
https://doi.org/10.4236/jcc.2025.137005

**Received:** May 20, 2025
**Accepted:** July 14, 2025
**Published:** July 17, 2025





## Abstract

The widespread availability and use of mobile phones and internet technologies have changed the way users purchase goods and services, revolutionising the merchant payments industry. Due to this expansion, more and more people are switching from cash to mobile money and digital wallet solutions. This shift unfortunately leaves more room for payment fraud attacks such as identity theft, application cloning and social engineering, to name but a few. To prevent or mitigate these attacks, researchers have proposed authentication methods based on one-factor, two-factor and even multi-factor authentication. However, these methods still present certain limitations in terms of the effectiveness of the proposed merchant payment security approaches, and the ability to implement them or integrate them into existing digital wallets. This paper addresses these challenges by proposing *FapshiSec*. It is a secure and efficient multi-factor authentication method that can be integrated into already-existing digital wallets to ensure the effective security of merchant payments. The method uses password, PIN, OTP, phone identification via phone ID, and biometric fingerprints to authenticate users and money withdrawals. The approach which has been implemented in the *Fapshi* mobile app, comprises two phases, namely the enrolment and authentication phases. The security analysis we have conducted through different scenarios shows that *FapshiSec* is efficient, and highly effective against payment fraud attacks such as Brute force, phishing and identity theft. It has equally revealed that *FapshiSec* ensures data confidentiality, integrity, non-repudiation, and privacy. Finally,

the performance analysis indicates that *FapshiSec* presents a smaller communication overhead, improved computational costs and security features when compared to four (04) existing relevant multi-factor authentication approaches.

## Keywords

Digital Wallet, Merchant Payment, Multi-Factor Authentication, Payment Fraud, Phone Identification

## 1. Introduction

The way people pay for goods and services has come a long way since the early days of bartering. Today, there is a plethora of payment options, ranging from cash to plastic cards and electronic payment techniques (ePayment), from which customers can choose the payment method that best suits their needs. Moreover, user behaviour has evolved rapidly in recent years, moving away from more traditional payment systems such as cash and bankcards towards alternatives such as payment via mobile devices (m-payment), most of which involve online payments [1]. For purchases in general, many consumers opt to pay directly from a digital wallet stored on their mobile devices or smartphones. For example, newer-generation iPhones come equipped with Apple Pay, which provides an additional option for convenient payments using facial recognition or PIN on newer-model iPhones and fingerprint recognition or PIN on older models [2].

The Global System for Mobile Communications Association (GSMA) report on the state of mobile financial services (MFS) shows that in 2015, there were about 203 million mobile money customers registered worldwide, 98 million of whom were located in sub-Saharan Africa [3]. By 2021, the number of registered mobile money users in sub-Saharan Africa had grown to 548 million, a 12% year-on-year increase, with a transaction value of US$490 billion and an annual transaction volume of 27.4 billion. In Cameroon, a significant positive impact has been observed with small and medium-sized enterprises (SMEs) where mobile money payment and receipt services contributed to the order of 73% of the total variance in the turnover of the SMEs in Douala after they had begun to use the technology [4]-[6]. This implies that most people use their phones and digital wallets for payment transactions. Unfortunately, this situation has led to an increase in various forms of payment fraud, depriving individuals and businesses of huge sums of money.

Recently, many people have complained that money was stolen from their digital wallets (Mobile money, Fintech apps, etc.). Most often, they either ignorantly approved the transaction or were hacked, and their PIN codes, passwords, or pattern codes were obtained. The following figures show just how serious the problem is. Datos Insights [7]-[9] revealed that US consumers lost nearly $8.8 billion

in 2022 due to identity theft and it is estimated to reach $23 billion in 2030. Stripe [10] reported that 44% of data breaches in the US in 2020 were from social engineering attacks (mainly phishing and smishing) and that businesses lose $1 billion yearly due to the same. The National Agency for ICT (ANTIC) [11] reported that Cameroon recorded nearly 6 billion CFA francs in losses related to bank and electronic money fraud in 2019. All these attacks have been able to proliferate because merchant payments, which are now mainly made via digital wallets, do not have sufficient security measures in place to prevent these frauds.

User authentication is the fundamental payment security measure to verify the identity of users attempting to access or complete a payment transaction and hence avoid those payment frauds. A wide variety of methods are available to authenticate payments, ranging from passwords and one-time passwords (OTPs) to biometric methods like fingerprint scanning and face authentication. Each of these methods relies on one or many different factors to establish trust, namely knowledge (something only the user knows), possession (something only the user has), and biometrics (something only the user is) [12] [13].

In recent years, organisations have started to move away from traditional knowledge-based authentication methods like identity documents and passwords because they are insufficient to combat identity theft or assign security. In fact, passwords are not secure because they can be shared, guessed, or stolen, and can also cause user frustration because they are easily forgotten [14]. This infuses the need to consider other modern and more secure methods for protecting payment channels used for merchant payment transactions. Several financial institutions have taken measures to increase the security of their users money. MTN, for example, launched the MTN Mobile Money app for Mobile Banking in January 2020, allowing customers to send, receive, save, and spend money as well as pay for goods and services using their mobile phones [15]. The app offers two-factor authentication, which adds an extra layer of security to the login process. This means customers need to provide a second form of authentication, such as a one-time PIN sent to their phone, in addition to their username and password [16]. The Orange Max it—Cameroon app [17] authenticates in the same way. However, analysing these most commonly used systems in Cameroon, we realize that consumers accounts are still very vulnerable because anyone who clones or has your phone number and knows your PIN can access your account and withdraw money. Likewise, if a hacker or scammer knows your phone number, he can spoof your account and access the sent OTP code and exclusive access to your account.

The constraints above raise a principal problem: The need for a stronger method to authenticate money withdrawals from consumers digital wallets.

Biometric authentication-based identity is vital in securing most e-businesses because biometric payment systems utilize unique human characteristics such as fingerprints, facial recognition, or voice patterns to verify identities and process transactions securely. This gives consumers a faster, more convenient checkout experience than traditional payment methods [2]. Biometric authentication, such

as fingerprint identification, offers maximum security for financial transactions but does not guarantee sufficient security when used solely. To guarantee adequate security, a trustworthy multi-factor authentication system is needed.

Related studies propose multi-factor authentication schemes for digital wallets. However, their proposed methods still present many shortcomings and security flaws. Some of them are not used because they are difficult to develop, some are solely framework-based, while others cannot be integrated into already-existing digital wallet systems without completely rebuilding the authentication module in the system [18]. Apart from these shortcomings, the following security flaws have been observed: Improvements are required for device identity-based authentication techniques that use IMEI since IMEIs can be easily phished or spoofed [19]. Also, the fingerprint authentication schemes that store fingerprints online can easily be breached [14], as well as those that do not authenticate fingerprints both locally and server-side [19] [20].

To address these shortcomings, this paper proposes *FapshiSec*. It is a biometric-based approach for securing money withdrawals from digital wallets. Unlike other approaches, *FapshiSec* can be integrated into already-existing digital wallets and can prevent and mitigate payment fraud attacks by implementing multi-factor authentication and an enhanced device identification method. The main contributions of this paper are summarised as follows:

- We survey the state of the art on biometric-based approaches to authenticate merchant payment transactions in digital wallets. The aim is to identify the methods ensuring the highest level of security.
- We model a two-phase multi-factor authentication method for money withdrawals from a digital wallet. The proposed method considers Password, PIN, OTP, Phone ID, and Biometric fingerprint in the authentication process.
- We assess the efficiency of the proposed method by demonstrating both its efficient integration into the existing Fapshi wallet app and its ability to prevent and mitigate payment fraud attacks using an Android phone. For this purpose, we conduct a security analysis through different scenarios.
- Finally, we conduct a performance analysis of *FapshiSec* in terms of communication overhead, computational costs and security features compared to four methods proposed in the state of the art.

The rest of this paper is organised as follows. Section 2 summarises relevant approaches related to securing merchant payment transactions based on a digital wallet. Section 3 presents the proposed *FapshiSec* system model, how the system works and the main algorithms used during the implementation. In Section 4, we present the tools used during the implementation and the results obtained thereafter. Section 5 evaluates the performances of the proposed approach by conducting both a security and a performance analysis, while Section 6 concludes the paper.

## 2. Related Works

Biometrics is a highly reliable technology that has become an indispensable addi-

tion to modern authentication systems, particularly for bank accounts, financial technologies (FinTech), Internet of Things (IoT) devices and all processes linked to money and privacy. Although effective, biometrics alone (SFA) cannot sufficiently secure payment transactions. A combination of two or more authentication methods provides a stronger form of authentication, *i.e.*, two-factor authentication (2FA) or multi-factor authentication (MFA). This section presents relevant related works on biometric-based authentication payment systems.

### 2.1. Biometric-Based Single and Two-Factor Authentication in Payment Systems

Okpara and Bekaroo [21], proposed an approach in which a camera-captured fingerprint sample is used for customer electronic wallet authentication. During the registration process, the users biodata, payment card details (stored as virtual cards), and a visual fingerprint template are captured and securely stored for authentication. In the authentication phase, the users fingerprint is scanned and matched against the template stored within the secure element. Although revolutionary, camera-captured fingerprints cannot accurately be likened to real fingerprints due to limited phone camera potential and image quality depletion (background images, distortions, etc.). The system is also susceptible to replay attacks since there is only one layer of security (using the fingerprint image).

Iqbal *et al.* [22], exploited fingerprint verification availability on mobile devices to provide a user-friendly and secure digital wallet payment facility for elderlies who are unable to use this facility due to the complex infrastructure of traditional authentication mechanisms. A novel digital payment mechanism is presented that uses Bluetooth technology on mobile devices for billing at the point of sale and fingerprint verification for user authentication. The technology provides enhanced security and ease of use for elderly people when making payments at a POS, since only a fingerprint and card details are necessary to validate a transaction (they do not have to remember PINs or passwords). Howbeit, the system only works via Bluetooth and, thus, will not work if the phone is not near the POS. In addition, the system requires the user to communicate with the POS operator, which gives the operator access to the users account, exposing it to impersonation threats.

Mega [23] introduced a framework to increase the security of mobile money services using iris biometrics and PIN. The framework has a registration and authentication phase. In the registration phase, the users biodata, ID number, PIN, and iris biometric are collected, verified, and stored in the database. In the authentication phase, the user logs in with their PIN, chooses a mobile money service (transfer or deposit) and enters the amount and the agent or customer reference number. The user is then prompted to authenticate using their iris biometric. The transaction is approved if the iris feature matches the stored copy in the database. Otherwise, it is rejected. The framework prevents unauthorised access to mobile money systems and is convenient, although vulnerable to PIN challenges.

However, iris recognition has a high error rate due to iris deformation caused by disease, stretched pupils and the low quality of the cameras used for registration. In addition, iris images can be captured and reused by an intruder to gain access to the system, making it vulnerable to replay attacks.

Pathan *et al*. [24], proposed a 2FA that collects username, password and fingerprints in the registration phase. The special feature of this approach is that the user must enrol at least two of their ten fingers during the enrolment phase. Each fingerprint has a unique password. The user has to register a password once, and each finger is assigned a password based on its position and the position number of the fingerprint. The problem of remembering passwords is solved by putting a password together with the finger location so that they are remembered automatically. To authenticate the user, he or she will have to provide the password and fingerprint of at least two of the stored fingers. The system offers greater security by using at least two fingers. However, it is susceptible to man-in-the-middle (MITM) and insider attacks since someone could obtain the fingerprints and because passwords can only be distinguished by the position of the fingers, it will be easy to guess and hack the system.

Mtaho [25] suggested a two-factor authentication system for enhancing mobile money security. During the enrolment phase, the user biodata, PIN, and biometric fingerprint are collected and stored in the database. Then, the user is authenticated using a PIN and biometric fingerprint. The proposed scheme offers security against shoulder-surfing attacks by providing two steps of authentication involving the biometric fingerprint. However, it is susceptible to spoofing attacks, fake digital biometrics, Trojan horse attacks, matcher override or false matches, replay, and intrusion attacks.

### 2.2. Biometric-Based Multi-Factor Authentication (MFA) in Payment Systems

Chetalam [26] presented a multi-factor authentication approach for mobile phones that combines voice biometrics, device-specific ID, and PIN to secure M-Pesa transactions. Here, the user's phone number, email address, unique ID, PIN, and voice biometrics are all recorded and kept in the database for authentication purposes. During login, the user is only authenticated if their unique ID, voice biometric and PIN match the stored samples. The proposed model has higher efficiency, convenience, accuracy, authentication level and security as compared to 2FA approaches. In addition, it cannot be impersonated, and users can authenticate without installing additional software. Nevertheless, the approach is vulnerable to replay and man-in-the-middle (MITM) attacks. Additionally, human voice changes over time which may cause errors in voice recognition.

Malathi and Raj [14] proposed a system to enhance the security of e-payments through banks by employing a combination of fingerprint, iris, and palmprint authentication. In their approach, users provide their biometric datafingerprint, iris, and palmprintto the bank via the banks server. This information is stored in the

banks database. When purchasing a product online, the user first enters their username and password. During the payment process, the merchant requests the user to provide their biometric data via a biometric service. The submitted biometric data is then compared with the records in the banks database. If a match is found, the merchant sends a confirmation message to the user and generates a One-Time Password (OTP). Once the user confirms the OTP, the bank processes the payment and transfers the funds to the merchant. The advantage of this system lies in its use of three distinct biometric methods, making account hijacking significantly more challenging. However, storing biometric data in an online database introduces vulnerabilities such as man-in-the-middle (MITM), insider, and spoofing attacks.

Similarly, Hassan [18] proposed a secure multi-factor authentication framework for electronic payment systems. During the registration phase, the users biodata, password, and fingerprint are collected, verified, and securely stored. For authentication, the user logs in with their password and biometric fingerprint. The scanned fingerprint is matched against the stored template, and upon successful verification, the user proceeds to the transaction phase. In this phase, the user enters the transaction amount and re-authenticates using their fingerprint. If the fingerprint matches, an OTP is sent to the users registered phone number. The transaction is completed once the OTP is correctly verified. This framework offers enhanced security by incorporating multiple authentication layers, effectively mitigating attacks such as password-based, dictionary, phishing, shoulder-surfing, and MITM attacks. However, it is still susceptible to SIM-swapping, wireless interception of SMS OTPs, identity fraud, advanced AI-driven attacks, and malware. Additionally, it requires a significant redesign to integrate with existing e-payment systems.

Ali *et al*. [27] proposed a secure and efficient multi-factor authentication algorithm designed for mobile money applications. This algorithm integrates PIN, OTP, and biometric fingerprint authentication to provide enhanced security. It also uses biometric fingerprint verification and quick response (QR) codes to facilitate secure mobile money withdrawals. The security measures implemented include the use of the Secure Hashing Algorithm-256 (SHA-256) to protect PINs and OTPs. Biometric fingerprint authentication is secured using Fast IDentity Online (FIDO), which leverages standard public key cryptography (RSA). Additionally, Fernet encryption is applied to secure QR codes and database records, ensuring a robust framework for safeguarding user data and transactions. This enforces the security of the approach as the approach can prevent several payment fraud attacks including MITM, replay attacks, and identity fraud, among others. However, because fingerprint authentication is only done locally and mobile money agents have access to users details, the approach is susceptible to app cloning fraud and identity theft.

Melendez *et al*. [19] introduced the SectraBank model, designed to mitigate cyber fraud attacks, specifically targeting SIM Swapping and Fake App schemes

in mobile banking users. In the enrolment phase, the users biodata is collected at the bank. After installing the app, the user registers his password, the app collects the device IMEI, and the user is prompted to validate his device fingerprint. In the final phase, the user defines three safe location points using the integrated Google Maps service in the app. The device IMEI and password are stored in the database. The user is granted access to the system after validating their password, IMEI, fingerprint, and geolocation. The model provided improved security and resistance to several payment attacks, specifically social engineering, insider fraud, and sim swapping. Nonetheless, fingerprints are only validated locally on the device, giving room for impersonation attacks and vulnerability to specialized AI attacks and identity fraud. Also, the system cannot be integrated into existing digital wallet systems unless their authentication processes are completely rebuilt.

### 2.3. Research Supporting Fingerprint-Based Authentication for Merchant Payments

Among the many biometric authentication methods, fingerprint authentication is the most widely accepted and used [24] [28]. They are secure to use and unique for every person as no two people have been found to have the same fingerprints. They are unique and do not change throughout ones lifetime [24]. Fingerprints offer an easy way to unlock smartphones, authorize payments, and confirm a person's identity. Compared to traditional payment options, fingerprint scanning is incredibly user-friendly and secure [29] [30].

Porubsky [28] investigated the level of end-user acceptance of fingerprint and face recognition authentication methods through a study involving 39 interview questions. These questions addressed topics such as prior use of these technologies, user preferences, perceived advantages and disadvantages, and opinions on features like ease of use, convenience, security, and overall experience. Additionally, a third authentication method was evaluated: two-factor authentication combining a biometric method (fingerprint) with a traditional method (PIN). The purpose of this evaluation was to assess the potential benefits of incorporating such technology into mobile payment systems and compare its acceptance to that of fingerprint and face recognition methods. The findings indicated that the majority of end-users viewed two-factor authentication for mobile payments as highly secure, with 90.4% strongly agreeing with this assessment. More so, 34.6% strongly agreed, and 30.8% agreed that implementing such technology would enhance the e-payments domain. Users also perceived mobile payments authenticated via face recognition or fingerprint as faster and more secure compared to traditional authentication methods [28].

Biometric approaches have proven to be more effective than traditional authentication methods. Al-Jarba and Al-Khathami [31] analysed existing biometric authentication techniques on mobile platforms, focusing on face recognition. Their study highlighted the feasibility and challenges associated with these methods, concluding that relying solely on biometric authentication is insufficient to con-

firm a user's authenticity based purely on biometric traits. The growing interest in biometric-enabled payment options is driven by benefits such as faster checkout times, enhanced data protection, and stronger authentication mechanisms [32].

**Table 1.** Summary of relevant related works.

| Ref. | Methods & models | Security Elements | Strengths | Limitations |
|---|---|---|---|---|
| [21] | SFA Camera captures fingerprint | Fingerprint | Cameras are available in almost all phones. Thus, the fingerprint method can be used by a larger population | Camera-captured fingerprints can be easily spoofed, making the authentication approach easy to bypass |
| [22] | Bluetooth, SFA | Fingerprint | Convenient for elderly users. | Impersonation threats since the user gives the operator access to their account |
| [24] | 2FA | Password, Fingerprint | Robust against brute force attacks since users register up to ten fingers | Cannot resist MITM and insider attacks since passwords are only distinguishable by their finger positions |
| [25] | 2FA | PIN, fingerprint | Security against shoulder-surfing by adding fingerprint authentication | Weak against fake digital biometrics, trojan horse or specialized AI attacks |
| [26] | MFA | Voice, biometrics, device ID, PIN | Resistance to impersonation and convenience to use | Vulnerable to replay and MITM attacks. The error rate is also high since voices change |
| [14] | MFA | Fingerpint, iris, palm print, OTP, password | Multiple authentication factors are difficult to crack collectively | Biometrics are stored in an online database exposing them to MITM, insider and spoofing threats |
| [18] | MFA | Password, OTP, fingerprint | Multiple forms and layers of protection provide resistance to brute force, MITM, shoulder-surfing, password-based, and phishing attacks | Cannot resist identity fraud, specialized AI, and malware attacks. Cannot be integrated without completely rebuilding the authentication of the existing system |
| [27] | MFA | PIN, OTP, fingerprint, QR Code for mobile money agent withdrawal | Database records, OTP and PINs are encrypted, local fingerprint authentication, thus preventing multiple attacks | Increased computation and communication costs. Fingerprints are authenticated locally and there is no phone identification, thus, room for app cloning fraud and identity theft |
| [19] | MFA cryptographic algorithm | Password, phone IMEI, fingerprint, geolocation | Resists multiple payment fraud attacks especially social engineering and sim-swapping | Local fingerprint authentication exposes it to identity fraud, and specialized AI and impersonation attacks |
| **Our Work** | MFA | PIN, *Phone ID* fingerprint, OTP, password | Both local and server-side fingerprint authentication, thus preventing multiple attacks including app cloning and identity fraud | The user can only use their account with one phone |

Table 1 shows the summary of the relevant state-of-the-arts on secured payment methods including strength and limits. One can observe that biometric ap-

proaches are more reliable and effective in securing payment transactions for merchants and, as more people have access to mobile phones with built-in features to support fingerprints, facial recognition, etc., biometric, authentication is becoming increasingly important. Moreover, fingerprints are the most widely adopted and accepted biometric authentication method to date, particularly among mobile phone users. [6] [22] [28] [33] [34] [35]. Fingerprints are believed to be unique for individuals, and even across fingers of the same individual. It has been proven that fingerprints vary even in identical twins who have similar DNA structures [35]. Although they ensure better security, biometric methods alone cannot sufficiently secure merchant payment transactions. Even a combination of biometrics and another authentication method (biometric-based 2FA) does not fully prevent payment fraud attacks. For all these reasons, we chose a multi-factor authentication approach combining password, PIN, OTP, phone ID, and fingerprint to validate payment transactions, and we demonstrate its effectiveness by focusing on payout transactions from the existing *Fapshi* mobile wallet.

## 3. System Model

Given that our approach is based on an existing digital wallet model, it is worth introducing that digital wallet (Fapshi) and how it works before diving into the *Fapshisec* approach. The following section gives us an overview of what *Fapshi* is.

### 3.1. What Is Fapshi and How Does It Work?

Fapshi [36] is a FinTech solution developed by young Cameroonians to tackle payment collection and management challenges prevalent in African markets. It offers a suite of tools designed to simplify payment processes, enabling users to make, manage, and collect payments effortlessly. These tools include, but are not limited to, payment links, invoicing, payment APIs, SDKs, plugins, product links, online stores, payouts, bulk payments, and event booking solutions.

Figure 1 shows the service and payment on Fapshi digital wallet. Fapshi users typically collect payments by integrating Fapshi APIs into their websites or applications and get paid for their services. Users can also implore code-free solutions by creating events, product links or payment links on their Fapshi dashboard or app, and get paid via the payment links. Each payment made credits their Fapshi account and is reflected on their Fapshi wallet balance. They can withdraw the money (payout) either by initiating an MTN Mobile Money or Orange Money payout or by requesting a bank transfer. For security purposes, all Fapshi payments must be validated using the user's Fapshi PIN code. Currently, Fapshi has a password, a PIN and a two-step authentication system via OTP that can be activated or deactivated in the settings. However, to better address potential threats, we propose *FapshiSec*, a multi-factor authentication system that further strengthens the security of the current system by adding two additional layers of security, namely phone identification and fingerprint biometrics. The *FapshiSec*'s system model is presented hereafter.

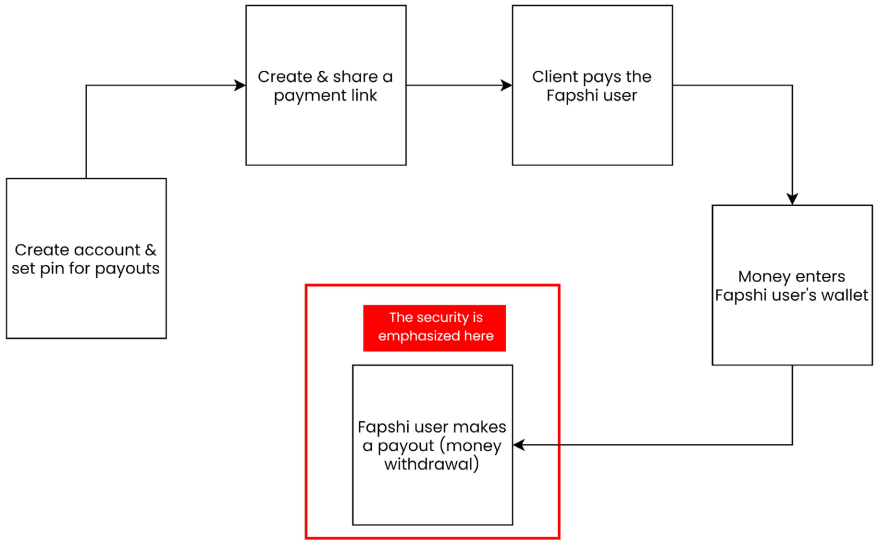



**Figure 1.** Fapshi service and payment flowchart.

### 3.2. The FapshiSec System Model

**Figure 2** presents the *FapshiSec* system model. It is made up of three main components, namely the Fapshi digital wallet app, the Fapshi server, and the Fapshi database.

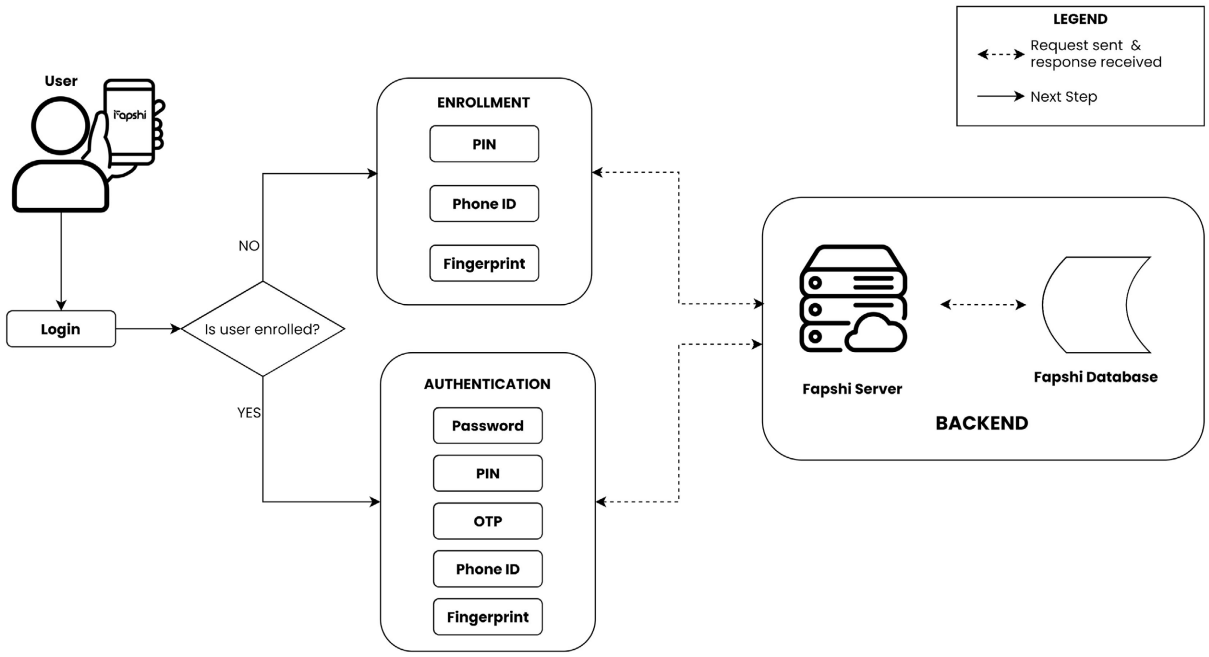



**Figure 2.** FapshiSec System model.

The Fapshi digital wallet app is located in the end users phone, while both the Fapshi server and the database, collectively called backend, are stored in the cloud. The Fapshi App and the backend communicate via API calls. Users can log into the app, check their balance, manage their account (modify account settings like PIN

and password), and perform other usual actions such as payment links creation, money withdrawal, etc. Due to the sensitivity and critical nature of money withdrawals, the security enhancement of Fapshi (*FapshiSec*) focuses on that process. For security, the database stores the users information (email, PIN code and password) and the generated public key, while the private keys and biometric templates are stored in the user's smartphones. The security of the password and PIN are ensured by asymmetric cryptography, that of the biometric fingerprint is ensured by the Fast IDentity Online (FIDO) protocol, while the security of the phone ID is managed by Android. All the components work together to attain the systems goal.

*FapshiSec* consists of two different phases, namely an enrolment phase and an authentication phase. Those phases are hereafter described.

### 3.2.1. The Enrolment Phase

The enrolment is done in-app, *i.e.*, the user is required to log in and then access the settings screen to activate MFA. The choice to activate the MFA is given to the user at this stage to comply with user consent and security regulations.

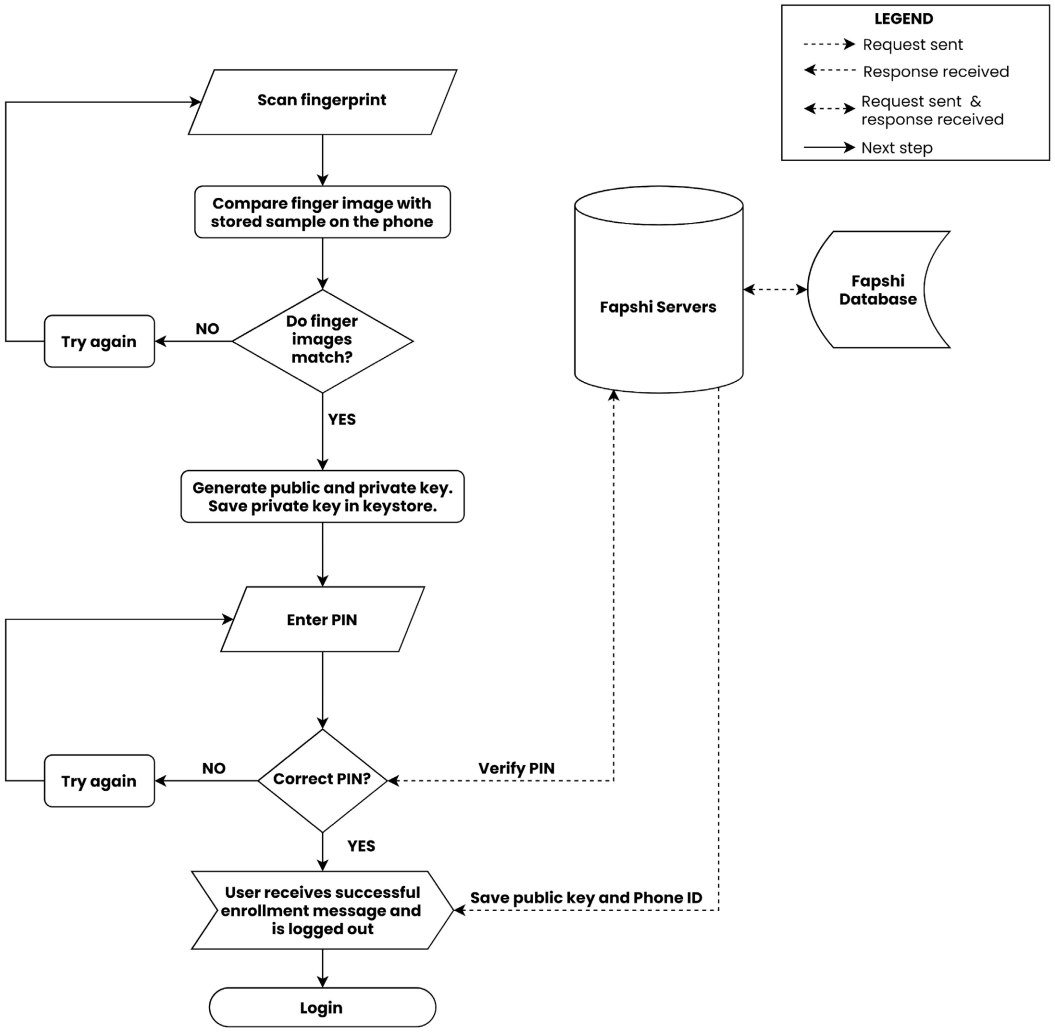



**Figure 3.** FapshiSec Enrolment flowchart.

Figure 3 presents the steps for the enrolment phase. The user scans their fingerprint by touching the fingerprint sensor on the phone. The captured fingerprint image is compared with the stored templates in the phone. If the two fingerprints match, an asymmetric key pair (public/private key) is generated via RSA and is saved in the phone's keystore. At the same moment, the user's phone ID is obtained and stored in a variable. The user is then prompted to enter their PIN code for verification. The PIN code entered, together with the public key and phone ID collected earlier are then sent to the server. If the two PINs (the one entered and the one previously stored) match, the public key and the user's phone ID are saved in the database as complementary user details. The user then receives a success message and is logged out. If not, they will be prompted to re-enter their PIN for verification. After enrolment, the user can login into the system and payouts via MFA will now be available. The authentication phase is described above.

### 3.2.2. Authentication Phase

The authentication phase begins when the user attempts to log into the system to make a withdrawal (payout). The aim is to ensure that the person accessing the app to withdraw money is indeed the account holder and that the phone they are using to access the account is their own, *i.e.*, the one they used to activate MFA. Figure 4 presents the flowchart for the authentication phase. It works as follows:

To log into the system, the user submits their username and password. The details are verified on the server and if correct, the user is granted access to the app. Otherwise, the user can try again up to four times after which the account will be suspended for an hour. While in the app, the user can access the payout screen to make withdrawals where he is asked to verify his identity. By tapping the button, an OTP is sent to the user via email. The user is prompted to enter the code which is sent to the server for verification. If the code is correct, the user is prompted to verify their fingerprint by touching the fingerprint sensor on the phone. Otherwise, they can restart the verification process to receive another OTP code. If the users fingerprint image matches the stored template, a payload which comprises the users email and phone ID is encrypted using the private key stored in the keystore and sent to the server for verification. If the fingerprint does not match, the user is redirected to the payout screen with a prompt that they are not authorized to make withdrawals from the account. On the server, the public key is used to decrypt the encrypted payload. If this process succeeds, the server sends a success message response. The user is then redirected to the amount details screen where they can input the amount, mobile money number and wallet PIN. The PIN is verified and if successful, the account balance is checked, and the amount is credited to the mobile money account. If the PIN verification fails or the account balance is insufficient, an appropriate error message is shown to the user.

## 4. FapshiSec's System Implementation

The system requirements are defined by the security measures we propose to prevent and mitigate payment fraud attacks. Based on an evaluation of the proposed

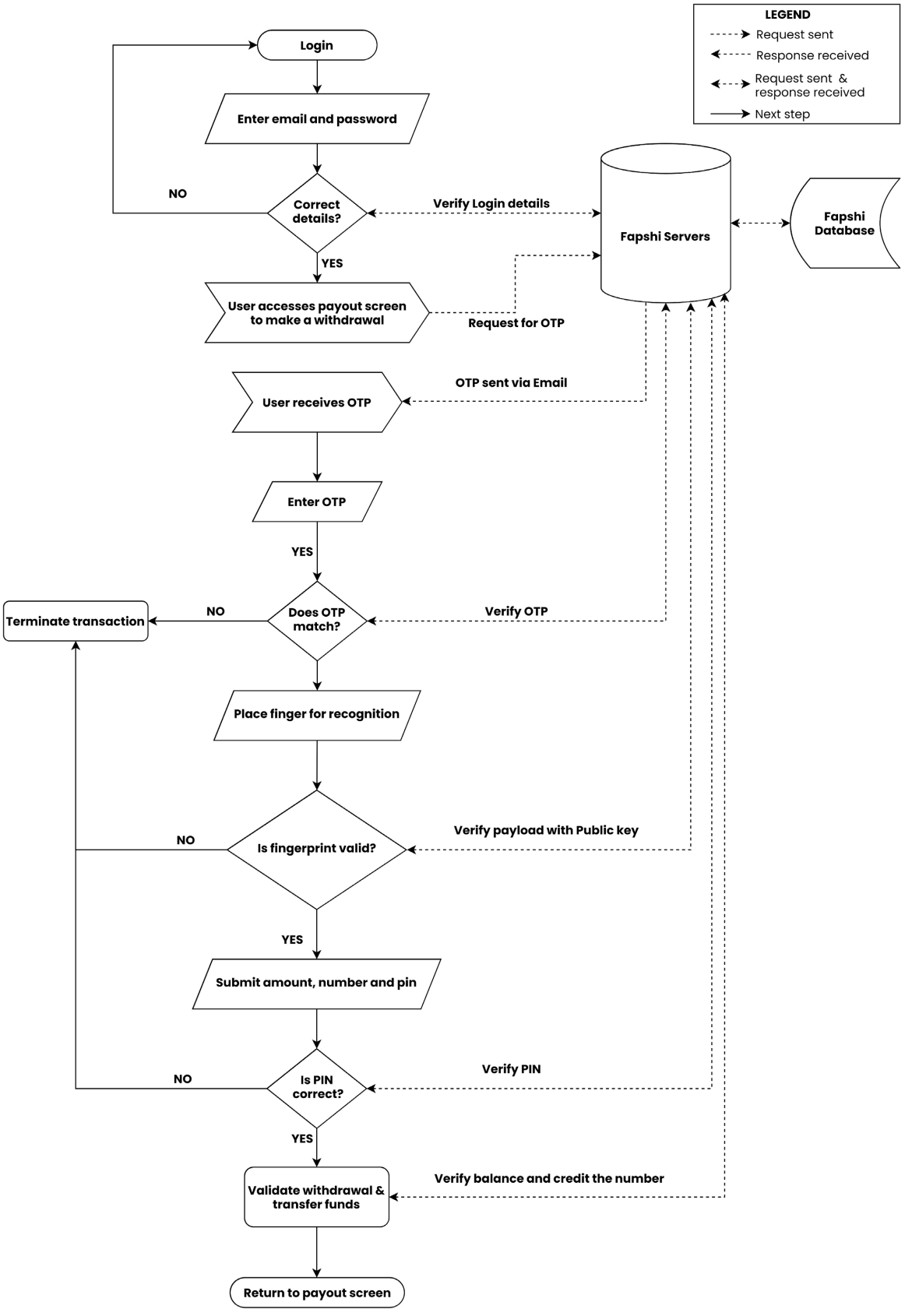



**Figure 4.** FapshiSec Authentication flowchart.

work and the current functioning of the Fapshi mobile wallet, Rapid Application Development (RAD) was employed to develop the solution. The RAD is mostly used for projects with tight deadlines that need prototyping and integrate high-level development tools and processes [37]. For the RAD, after gathering the requirements, we proceed to the design and application implementation directly. In the final stage, the testing is done, and the app is delivered. The following section highlights these different steps.

## 4.1. Requirement Gathering

The requirements are grouped into functional and non-functional requirements, which provide guidelines for implementing the proposed methodology.

### 4.1.1. Functional Requirements

The system functional requirements are listed as follows:

- The current user authentication process should not be completely altered. It should only be added to or improved upon.
- According to privacy policy and security compliances, users should be given the choice to activate the MFA. More concretely, users should be able to activate MFA in the security settings screen and when activated, they should not be able to deactivate it (unless they contact the Fapshi admins and justify why they need to deactivate it).
- Users should be able to withdraw their money via multi-factor authentication only if they have activated that option.
- Money withdrawals should be confirmed via multiple steps of authentication involving OTP, fingerprint authentication, phone identification and PIN code for users who have activated MFA.

**Algorithm 1** Enrollment Phase

**Data**:
$BF_i$ : User's biometric fingerprint;
$PIN_i$ : User's entered PIN
**Result**:
Enrolment status
**Begin**
BF $\leftarrow$ CaptureBF$_i$

**if** $IsValid(BF)$ **then**
  Generate and save $^{PK_i}/_{FK_i}$ in the keystore;
  $PID \leftarrow$ retrieve$PID_i$;
  $PIN \leftarrow$ Prompt user to insert $PIN_i$;
  **if** $(PIN_i.length() == 5)$ **then**
    Send $PIN, PK_i, PID$ to the server;
    **if** $IsValid(PIN)$ **then**
      Save $PIN$, $PK_i$, $PID$ in the $DB$;
      Send and display success response;
      Log the user out
    **else**
      Send and display Invalid Input error
    **end**
  **else**
    Display Incorrect PIN length
  **end**
**else**
  Rescan BF$_i$
**end**

#### 4.1.2. Non-Functional Requirements

Regarding non-functional requirements, one can underline the following two:

- The system should be secure. *i.e.*, prevents all forms of payment fraud attacks.
- The system should ensure privacy, usability, maintainability, performance, reliability, interoperability, flexibility, robustness, scalability, confidentiality, integrity, and availability.

Based on the requirements, the main algorithms used for implementing *FapshiSec* have been defined as follows.

### 4.2. Main Algorithms Used

Based on the flowchart presented in Figure 3 and Figure 4, we have been able to define the main algorithms implemented in *FapshiSec*. These algorithms are respectively the enrolment algorithm (see **Algorithm 1**) and the authentication algorithm (see **Algorithm 2**).

**Algorithm 2** Authentication Phase

**Data**:
$UE_i$ : User's Email;
$P_i$ : User's Password;
$OTP_i$: User's OTP;
$BF_i$: User's biometric fingerprint;
$PIN_i$: User's entered PIN;
**Result**:
Payout Status
Begin
$U_i \leftarrow$ User enters $UE_i$, $P_i$
**for** *int i=0; ij=5; i++* **do**
  **if** $IsValid(U_i)$ **then**
    Grant access to the app;
  **else**
    Display invalid credentials error;
  **end**
**end**
User requests $OTP_i$
$OTP_i \rightarrow$ sent to user's $UE_i$
$OTP \leftarrow$ User enters $OTP_i$
**if** $IsValid(OTP)$ **then**
  $BF \leftarrow$ Capture $BF_i$ **if** $IsValid(BF)$ **then**
    Retrieve and encrypt payload $P_{load} = UE_i + PID_i$ with private key, $FK_i$
    Send $E(P_{load})$ to the server for verification
    Use the public key,$PK_i$ to decrypt the payload, $P_{load}$
    **if** $IsSuccessfull(D(P_{load}))$ **then**
      **if** $UE_i + PID_i == P_{load}$ **then**
        Proceed to PIN verification
      **else**
        Display the "You are not allowed to do withdrawals on this device" error
      **end**
    **else**
      Display the invalid user error
    **end**
  **else**
    Display the "You are not allowed to make withdrawals from this account" error
    Return to payout screen
  **end**
**else**
  Display invalid $OTP$ error
**end**
User enters $Amt_i$, $M_i$, $PIN_i$ **if** $Bal_i \geq Amt_i$ *AND* $IsValid(PIN_i)$ **then**
  Send $Amt_i$ to $M_i$
  Display Payout success message
  Return to Payout screen.
**else**
  Display the respective error messages
**end**

After defining the different algorithms, it was important to select relevant tools to implement the solution. The different tools have been selected based on specific criteria which are emphasized below.

### 4.3. Software Development Tools Used

The following tools were used for the implementation of *FapshiSec*:

- React Native: React Native enables one to build cross-platform mobile apps for Android and iOS using React, a JavaScript framework for building single-page applications. React native biometrics [38] is a React native library that enables us to interact with the users stored fingerprint templates and perform local and server-side authentication of the biometrics. React native device info is a React native library that enables us to get the phone ID.
- Android phone: The phone lets us view the UIs and test the functionalities on a real device as we build. Three Android phones were used Redmi Note 10 Pro Max, Samsung Galaxy S8, and Samsung Note 9.
- MongoDB: MongoDB is a source-available, cross-platform, document-oriented database program. Classified as a NoSQL database product, MongoDB utilizes JSON-like documents with optional schemas. It is developed by MongoDB Inc. Fapshis database is MongoDB, thus, we used the same.
- Nodejs/Express: These were the backend technologies used to create the APIs. Nodejs is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts. Express.js, on the other hand, is a minimal and flexible web application framework that provides a robust set of features to develop Node.js based web and mobile applications such as templating, static file handling, connectivity with SQL and NoSQL databases.
- Visual Studio Code: Also commonly referred to as VS Code is a source-code editor developed by Microsoft for Windows, Linux, macOS and web browsers. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded version control with Git. This was our choice of code editor.

The tools selected allowed us to implement the *FapshiSec* solution. The results of implementation are described below.

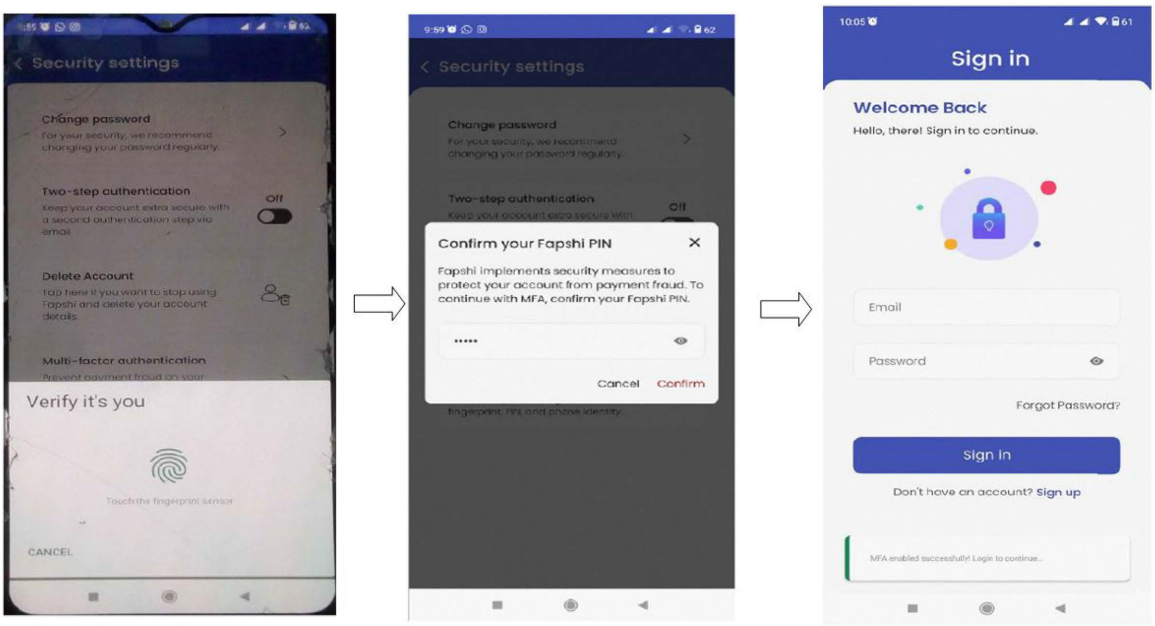
### 4.4. Outcome of the *FapshiSec*'s Implementation

The implementation was segmented into the enrolment and authentication phases. The outcomes are detailed below.

#### 4.4.1. Results of the Enrolment Process

**Figure 5** depicts the screenshots of the enrolment phase. It is assumed that the user already has an account and has defined a PIN code. When the user logs in and navigates to the security settings screen, they can activate MFA for payouts by pressing the "Activate MFA option" (*see* **Figure 5(a)**). This action initiates the fingerprint authentication process. After that, the user is prompted to verify their

PIN code (*see* Figure 5(b)). When the details are valid, the MFA option is removed from the settings screen. The user is logged out and a successful MFA activation message appears on the screen (*see* Figure 5(c)).




(a) (b) (c)

Figure 5. FapshiSec enrolment results.

#### 4.4.2. Results of the Authentication Process

Figure 6 depicts the authentication process. When attempting to authenticate, the user logs in (*see* Figure 6(a)) and accesses the multi-factor payout screen (*see* Figure 6(b)). By tapping the verify button, they receive an OTP (*see* Figure 6(c)) via email which when confirmed, initiates the fingerprint authentication process (*see* Figure 6(d)). When the fingerprint authentication is successful, they are presented with the payout details screen where they insert the amount, the method of withdrawal (phone number in this case), and PIN code (*see* Figure 6(e)). If all the information entered is correct, the user can withdraw their money. It is worth mentioning that Figure 5(a) and Figure 5(b) are unclear because the phones do not permit the fingerprint authentication process to be screenshotted. These images have therefore been taken by another phone's camera during the process. A video recording demo of the scenario is available as supplementary material at the following link:
https://drive.google.com/file/d/1pIXdHqeuFZQn3nk59sof0DxGpsJNEEPY/view?usp=sharing Authentication Process Video.

After implementing *FapshiSec*, we needed to evaluate its performance. The following section presents the evaluation.

## 5. Evaluation of *FapshiSec*

To evaluate *FapshiSec*, we conducted both a security analysis of the method

against existing fraud attacks and a performance analysis as compared to works in the state of the art. The security analysis of *FapshiSec* is presented below.

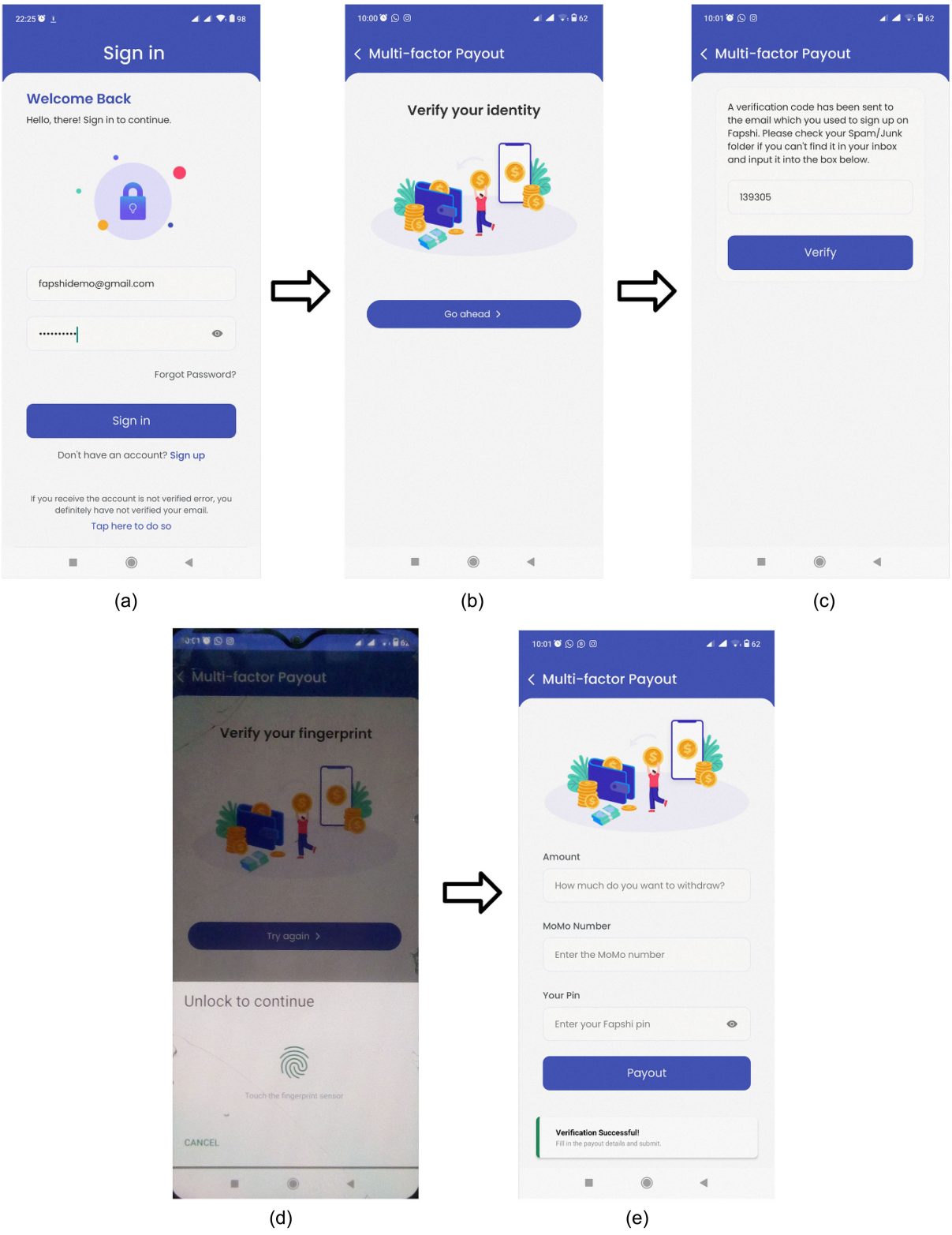



**Figure 6.** FapshiSec authentication results.

### 5.1. Security Analysis of *FapshiSec*

To conduct the security analysis of *FapshiSec*, it was important to first define the different attacks that are more recurrent in the FinTech industry, and afterwards, to evaluate our work based on those attacks.

#### 5.1.1. Types of Payment Fraud Attacks

According to the GSMA Mobile Money Fraud Typologies and Mitigation Plans report [39], mobile payment fraud schemes can be classified into four categories, namely impersonation, insider fraud, cyber fraud, and agent fraud.

*Impersonation* refers to the act of pretending to be another person, real or non-existent, and/or representing an entity to deceive others. Most common impersonation attacks on Fintech include *social engineering*, *identity fraud* and *SIM swapping*. Social Engineering relies on the psychological manipulation of human behaviour to disclose sensitive data, share credentials, grant access to a personal device or otherwise compromise their digital security [40]. Identity Fraud involves taking over the genuine identity of another or creating a fictitious, non-existent identity. This usually results in identity, ID card, information and biometrics thefts. Meanwhile, SIM Swap Fraud occurs when a fraudster tricks a provider into porting or transferring a victim's phone number to a new SIM card under the fraudsters control. This fraud is usually either to take over the users account or to impersonate the subscriber to validate another fraud scheme [41].

*Insider Fraud* involves employees within the mobile money system who exploit their position for illegal gains or act to the detriment of others. This is a type of fraud or threat that comes from the inside a current or former employee, contractor, or business partner can carry out a fraudulent scheme by taking advantage of knowledge, skill, experience, or access as an insider [42] [43].

*Cyber Fraud* involves using vulnerabilities in technological systems, software, or hardware components, networks, or the internet in general to gain unauthorized access to commit fraud. Subcategories include *Man-in-the-middle* (*MITM*), *denial-of-service* (*DoS*), *Malware*, *phishing* and *spoofing* [44]. MITM attack is the interception and possible alteration of the communication between two parties - usually the user and the mobile money service by a third-party. In this scenario, the attacker positions themselves between the users device and the mobile money system, allowing them to eavesdrop on or manipulate the data being exchanged. DoS are malicious activities that aim to disrupt or disable the normal functioning of mobile money services, making them temporarily or permanently unavailable to users. The goal is to overwhelm the targeted system with an excessive volume of traffic, requests, or other forms of malicious activity, causing it to become slow, unresponsive, or completely unavailable. *Malware* are malicious software designed to compromise the security and functionality of mobile devices and the mobile money applications or services they use. They are often used as a means of other schemes such as hacking, data theft, sabotage or blackmail. *Phishing* is a type of cyberattack carried out through various communication channels, such as emails, text messages, or fake websites where attackers use deceptive methods to

trick users into revealing sensitive information such as login credentials, PINs, and other confidential details related to mobile money accounts. *Spoofing* is a deceptive practice in which attackers manipulate information to falsely represent their identity or the identity of a legitimate entity such as a mobile money provider. The goal is to trick users into believing they are interacting with a trustworthy source when in reality they are engaging with a malicious actor.

*Agent Fraud* is a fraud committed by agents. An agent is a person or business that serves customers on behalf of a mobile money provider. They provide the mobile money provider with a wide distribution network and perform various functions including deposits (cash-in), withdrawals (cash-out), SIM registration and KYC verification for opening mobile money accounts. Agents are embedded within and mostly belong to the communities they operate in. This means they can easily build trust with customers which can be exploited to commit fraud.

Apart from these attacks classified by the GSMA, there are some noteworthy payment frauds usually performed by attackers to access mobile money accounts. These include *brute force*, *replay*, *and shoulder surfing attacks*.

*Brute force* is a malicious method employed by cybercriminals to infiltrate computer systems by making numerous login attempts with different password combinations. This approach relies on the premise that the correct password will eventually be found due to the exhaustive testing of all possible combinations. *Replay attack* is a form of network attack where an attacker intercepts and retransmits data that was previously exchanged between two parties. It fundamentally occurs when an attacker is able to capture data-in-transit in cleartext form, that is, after a MITM attack was successful. Finally, *shoulder surfing* describes a situation where the attacker can physically view the device screen and keypad to obtain personal information. It is one of the few attack methods requiring the attacker to be physically close to the victim to succeed.

After classifying and defining the different attacks that could appear in the mobile money environment, we need to conduct a security analysis of our approach to these attacks. In other words, we need to evaluate the resistance of our method when faced with these different attacks. In the following section, we emphasize the *FapshiSec*'s security analysis.

#### 5.1.2. Security Analysis of *FapshiSec*: How Does Our Method Stand up to the Aforementioned Attacks?

As seen above (section 5.1.1), Fintech solutions are subject to many different types of attacks. To conduct this security analysis, we needed to assess how *FapshiSec* reacts when faced with these attacks. For this purpose, we have imagined some scenarios that simulate these different attacks. However, to keep the page limit of the paper, we only present three of these scenarios, one in each of the categories listed by the GSMA. Figure evidences the three attacks chosen. We have evidenced one cyber attack (phishing), one impersonation attack (identity fraud theft) and a brute force attack. These scenarios are described below.

**Scenario 1: Simulating a brute force attack.** Figure 7(a) simulates the brute

force attack. In this scenario, the hacker tried to login with the wrong credentials up to five times. It can be observed that after five false attempts, the system is blocked for one (01) hour. This way, we can prevent the brute force attack for at least an hour.

**Scenario 2: Simulating a Phishing attack.** Figure 7(b) simulates a phishing attack. In this scenario, we consider that the users login details were obtained probably via phishing, and his account is being spoofed. Since he had already activated MFA, the option is no longer available on the security settings screen. Thus, the intruder cannot replace his biometric details with his own, thus, cannot withdraw his money.

**Scenario 3: Simulating an identity theft attack.** Figure 7(c) shows how *FapshiSec* reacts when faced with an identity theft attack. This is one of the worst-case scenarios that could happen. We consider that the users email and password are known and his email has been compromised via phishing. In this case, the hacker can receive the OTP code. However, it can be seen that the hacker's details are rejected when he accesses the payout screen and gets to the point where he needs to validate via fingerprint.

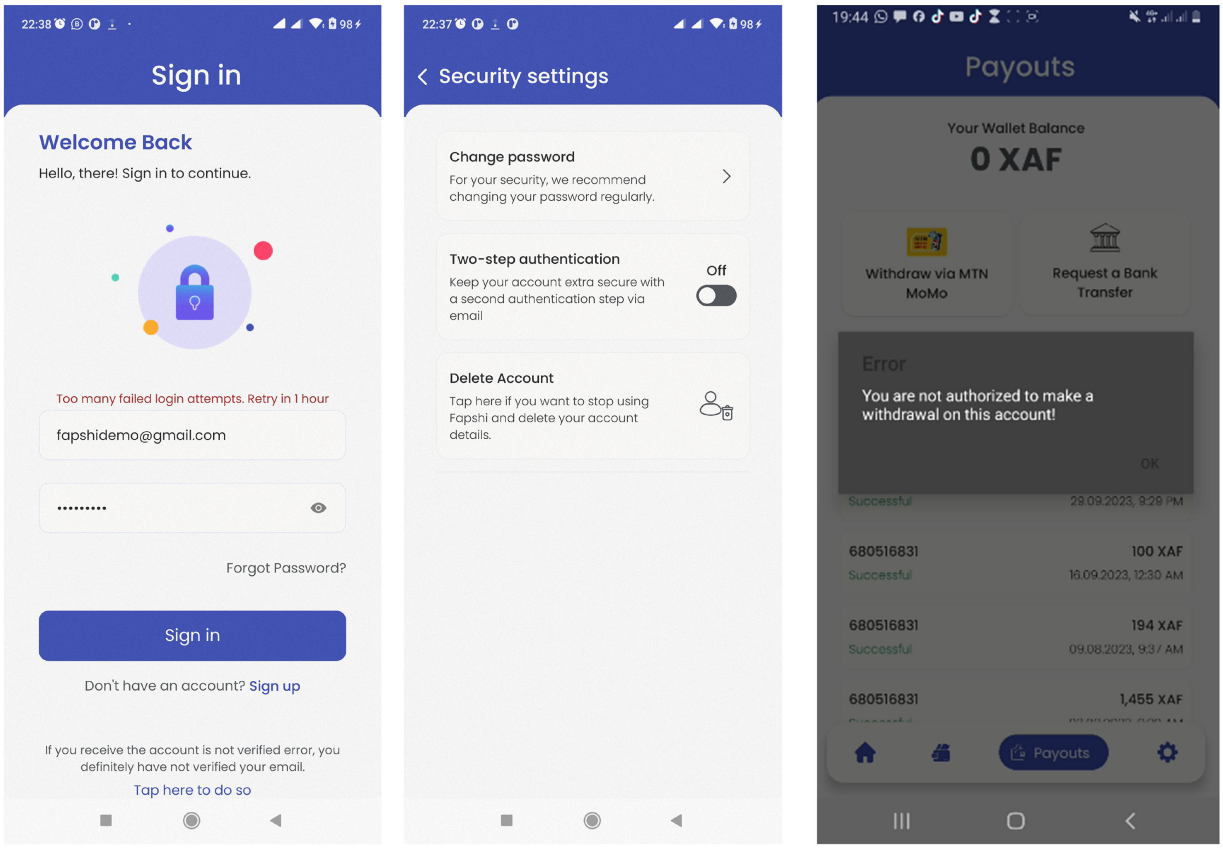



**Figure 7.** Security analysis of *Fapshisec*: Demonstrating the results of our approach when faced with brute force, phishing, and identity theft attacks.

Similar scenarios were conducted for other attacks and we were able to deduce the following observations:

***FapshiSec* provides secure and efficient authentication**: The proposed MFA algorithm offers multiple layers of security by incorporating five key elements: password, PIN, OTP, fingerprint, and phone ID. Together, these elements form a highly secure system that is difficult to compromise. The authentication process is swift, with an average time of approximately 20 seconds an acceptable increase given the significant enhancement in security it delivers.

**FapshiSec provides data confidentiality and integrity**: The proposed algorithm ensures confidentiality by hashing user passwords, PINs, and the payload while securing biometric fingerprint data through FIDO. Fingerprint images are stored in the phones keystore, preventing any transfer to external locations. Additionally, all communication between the application and the server is conducted over secure HTTPS channels. This robust approach makes it difficult for adversaries to modify, insert, or access data, whether in storage or during transmission, thereby preserving the integrity of the data.

***FapshiSec* ensures non-repudiation**: During mobile registration and enrolment, users provide their biodata, email, PIN, and biometric fingerprint. The email serves as a unique identifier for each user. During authentication, when an OTP is sent to the users email, the user cannot deny receiving it, as a copy is securely stored in the user table. Similarly, users cannot deny initiating transactions, as each transaction is meticulously tracked and recorded for accountability.

***FapshiSec* ensures anonymity**: Multi-factor authentication maintains user anonymity by requiring a unique PIN and biometric fingerprint, which serve to verify the users identity. Since there is no direct physical interaction between app agents and users, only the database holds records that can trace and identify the user, ensuring both security and privacy.

***FapshiSec* ensures privacy**: The biometric fingerprint is safeguarded by FIDO, which employs RSA to secure public/private key pairs and biometric templates. Additionally, database records are protected through the use of passwords and cryptographic schemes, ensuring the security of user credentials, transaction data, and overall user privacy.

***FapshiSec* prevents shoulder-surfing attacks**: The current two-factor authentication (2FA) for mobile money relies solely on a PIN and SIM for user authentication, which is inadequate for robust security. As a result, it is vulnerable to shoulder-surfing attacks, as both PINs and OTPs are entered in an unmasked form. In contrast, the proposed algorithm mitigates this risk by incorporating multiple identifiers, including a password, PIN, OTP, biometric fingerprint, and phone ID, thereby significantly reducing the likelihood of such attacks.

***FapshiSec* prevents social engineering attacks**: The proposed algorithm addresses this security challenge by implementing multi-factor authentication, requiring users to provide multiple identifiers such as a password, PIN, OTP, phone ID, and biometric fingerprint for verification. Even if attackers manage to obtain

the password and PIN, guessing the next OTP is difficult, as it is randomly generated and valid for only five minutes. Additionally, obtaining the biometric fingerprint is highly challenging, as it is secured by FIDO, which employs public/private key pairs to protect biometric data.

***FapshiSec* prevents phishing attacks**: The OTP is randomly generated, unique, and valid for only five minutes. The biometric fingerprint is protected by FIDO, which utilises RSA to generate public/private key pairs. The public key is encrypted before being transmitted to the server, while the private key and biometric templates are encrypted and stored in the smartphones cryptographic keystore. Since even the user does not have access to these keys, it is exceptionally difficult for attackers to obtain them through phishing attempts.

***FapshiSec* prevents brute-force attack**: After five unsuccessful login attempts, the user is blocked for an hour, significantly reducing the likelihood of successful brute-force attacks. Attackers are well-acquainted with PIN-based authentication systems, and many commonly used PINs are easy to crack. However, even if the PIN and OTP are compromised, breaching the FIDO system remains highly challenging. FIDO employs RSA encryption to secure public/private key pairs and biometric templates, and the computational effort required to break the asymmetric key pair would far outweigh any potential gain, rendering such attacks impractical.

***FapshiSec* resists replay attacks**: In the current two-factor authentication (2FA) scheme for mobile money, adversaries can delay or replay the authentication process, forcing mobile money agents and users to repeatedly enter their PINs, increasing the risk of PIN compromise. In contrast, our approach requires users to provide multiple identifiers, including a PIN, OTP, and biometric fingerprint, for verification. This multi-layered authentication significantly reduces the risk of replay attacks, as attackers will need to bypass all authentication factors simultaneously, making such attacks far more difficult to execute.

***FapshiSec* resists insider attacks**: Insider attacks, often carried out by current or former employees with access to system information and user transaction data, are mitigated in our approach through multi-step authentication. Additionally, users biometric data are not stored in the database. Instead, the biometric fingerprint is securely stored on the users device. Therefore, an insider would need both access to the users phone and their physical fingerprint to carry out an attack, which is highly unlikely in most scenarios, thereby enhancing the systems security.

***FapshiSec* resists impersonation attacks**: Impersonation attacks are mitigated in our approach by multi-factor authentication and by registering and identifying users by their e-mail address and phone ID because no two people can have the same e-mail address, just as no two phones can have the same ID.

***FapshiSec* resists identity fraud**: This threat model is mitigated in our approach through the implementation of a robust multi-factor authentication system. What is more, users' wallets are linked to their phones, so hackers will have

to steal the user's phone to get hold of their identity. However, stealing the phone does not give the attacker access to the user's fingerprint data. Even if he deletes the fingerprint data, he will not be able to reactivate MFA with his own data.

***FapshiSec* resists Man in The Middle attacks**: In the current two-factor authentication (2FA) scheme for mobile money, attackers can intercept communication between users, systems, and banks to steal authentication credentials, such as PINs or transaction details, spy on users, or disrupt communications. The proposed algorithm mitigates this threat by securing data transmission through robust cryptographic schemes. Furthermore, all communication between the app and the server occurs over HTTPS, making interception significantly more difficult. Additionally, biometric fingerprint data is never transmitted over the server, rendering it inaccessible to attackers.

***FapshiSec* resists app cloning**: More recently, attackers have found a new way to scam users by creating app clones such that the users believe they are using the real app. In such cases, they can get all the users details and even have them approve their biometric data. However, in our approach, we prevent this attack by using the phone ID. Unlike the IMEI, the phone ID represents the users IMEI, their Google ID, and the app signing key. Thus, the phone ID is never the same even for apps that look the same because Google generates a signing key for every app. Also, fingerprint authentication in our approach is done server-side. Thus, if the backend is unable to decrypt the payload as described in Section 3.2.2, the transaction will fail.

After conducting a security analysis of our approach, we need to conduct its performance analysis and compare it to works from the state of the art. The following section presents this performance analysis.

## 5.2. Performance Analysis: Comparison of the *FapshiSec* Method with Other Approaches from the State of Art

Given that the proposed approach comprises an enrolment and an authentication phase, the evaluation of the performances will consider these two phases at each stage. Performance is assessed by analysing the communication overhead, computation costs, the added value of our approach and the security features as compared to related approaches. This helps to understand the effectiveness of the proposed approach.

### 5.2.1. Communication Overhead

Communication overhead is associated with estimating the number of bytes in every communication message exchanged in the enrolment and authentication phases. Using the method of *Ali et al.* [27], we were able to compute the communication overhead of *FapshiSec*. Each packet size is calculated by summing the size of each message using the information in **Table 2**. As shown in **Table 3**, seven (07) messages are exchanged during the global process, two (02) messages during the enrolment phase and five (05) messages during the authentication phase, giv-

ing a communication overhead of 304 bytes. By comparing these results obtained by *FapshiSec* to those of *Ali et al*. [27], it can be seen that *FapshiSec*'s communication overhead is almost two times smaller than that of *Ali et al*. [27] 704 bytes. This is mainly because unlike *Ali et al*., we do not encrypt records in the database with Fernet encryption and our approach does not have a transaction phase. Hence, only seven (07) messages are transmitted during the authentication and enrolment phases compared to their fifteen (15) messages.

**Table 2.** Description and Length in bytes of the different symbols.

| Symbols | Meaning | Message size |
|---|---|---|
| $U_i$ | User email and password | 32 |
| $UE_i$ | Users Email | 16 |
| $P_i$ | Users Password | 16 |
| $PIN_i$ | Users entered PIN | 8 |
| $OTP_i$ | Users OTP | 8 |
| $PID_i$ | Users phone ID | 16 |
| $BF_i$ | Users biometric fingerprint | 16 |
| $PK_i$ | Users fingerprint public key | 32 |
| $FK_i$ | Users fingerprint private key | 32 |
| $Amt_i$ | Withdrawal Amount | 16 |
| $M_i$ | MoMo account to withdraw into | 16 |

**Table 3.** Calculation of message sizes for messages exchanged during the phases.

| Phase | Message Content | Message size (Bytes) |
|---|---|---|
| Enrolment | $\{BF_i, FK_i\}$<br>$\{PID_i, PK_i, PIN_i\}$ | $16+32=48$,<br>$16+32+8=56$ |
| Authentication | $\{U_i\}$<br>$\{OTP_i - 2times\}$<br>$\{BF_i, FK_i\}$<br>$\{UE_i, PID_i, PK_i\}$<br>$\{Amt_i, M_i, PIN_i\}$ | 32,<br>$8\times 2=16$<br>$16+32=48$<br>$16+16+32=64$<br>$16+16+8=40$ |

### 5.2.2. Computational Cost

The most common method to analyse computation cost is by measuring the time it takes for the necessary operations to finish processing. One of the more common methods of analysing the computational cost is by counting the number of different operations (e.g. hash, encryption, secret generation, etc.) that need to be performed and, commonly, comparing the results to those of other schemes [45].

The total computational cost of our approach (*FapshiSec*) as well as those of *Ali et al*. [27] and *Melendez et al*. [19] are presented in **Table 4**. In this table, $T_h$ and $T_{ed}$ are the durations of a one-way and two-way hash operation respectively. $T_{Re}$ and $T_{Rde}$ are the durations required to encrypt and decrypt messages using RSA, while $T_{Fe}$ and $T_{Fde}$ are the durations required to encrypt and decrypt messages using Fernet. It is worth mentioning that among the relevant works, only *Ali et al*. [27] and *Melendez et al*. [19] use cryptographic techniques to secure the data, which is why they have been chosen as a means of comparing our approach.

**Table 4.** Computational costs for the approaches.

| Proposed Algorithm | Enrolment Phase | Authentication Phase | Transaction (Cash withdrawal) Phase | Total |
|---|---|---|---|---|
| Ref. [27] | $2T_h + 1T_{Fe} + 1T_{Re}$ | $3T_h + 1T_{Fde} + 3T_{Rde} + 1T_{Fe}$ | $2T_{Fde} + 1T_{Rde} + 1T_{Fe}$ | $5T_h + 3T_{Fde} + 4T_{Rde} + 3T_{Fe} +$ |
| Ref. [19] | $2T_h + 1T_{ed}$ | $1T_{ed}$ | None | $2T_h + 2T_{ed}$ |
| Our work: FapshiSec | $1T_{Re}$ | $1T_{ed} + 1T_{Re} + 1T_{Rde}$ | None | $1T_{ed} + 1T_{Re} + 1T_{Rde}$ |

It can be seen that our approach has a relatively low computational cost compared to those schemes. The cryptographic techniques used by [27] take extra time during hashing and data encryption/decryption. In [19], only two-way encryption is performed during the registration and authentication processes; thus, the computational cost is equally low. However, fingerprint authentication is only performed locally, exposing the system to app cloning fraud. In our approach, fingerprint authentication is performed both locally and server-side. Thus, the approach is effective in resisting these attacks.

#### 5.2.3. Our Approach vs the State of the Art

**Figure 8** shows a comparison between the existing relevant MFA methodologies and our approach. The items highlighted in red background are our contributions while those with the red borders and texts are ameliorations to the existing methods.

It can be seen that in existing approaches, authentication is mostly only carried out during login. It is, therefore, difficult to integrate the approach into existing digital wallets without completely rebuilding the system, whereas, in our approach, authentication is activated in the application parameters, so there is no need to completely modify the existing system. Furthermore, unlike other approaches where identification is carried out via the IMEI - the IMEI is the same for each application and is susceptible to phishing, identity theft, identity fraud attacks and application cloning - our approach uses a phone identifier that is unique for each user and for each application, and is therefore resistant to fraud. As for fingerprints, some existing approaches store them online, which exposes them to MITM and insider attacks, whereas in our approach, fingerprints are only stored in the phone's keystore. Finally, none of the existing methods authenticate fingerprints both locally and on the server, unlike our approach where fingerprints are authenticated on both sides.

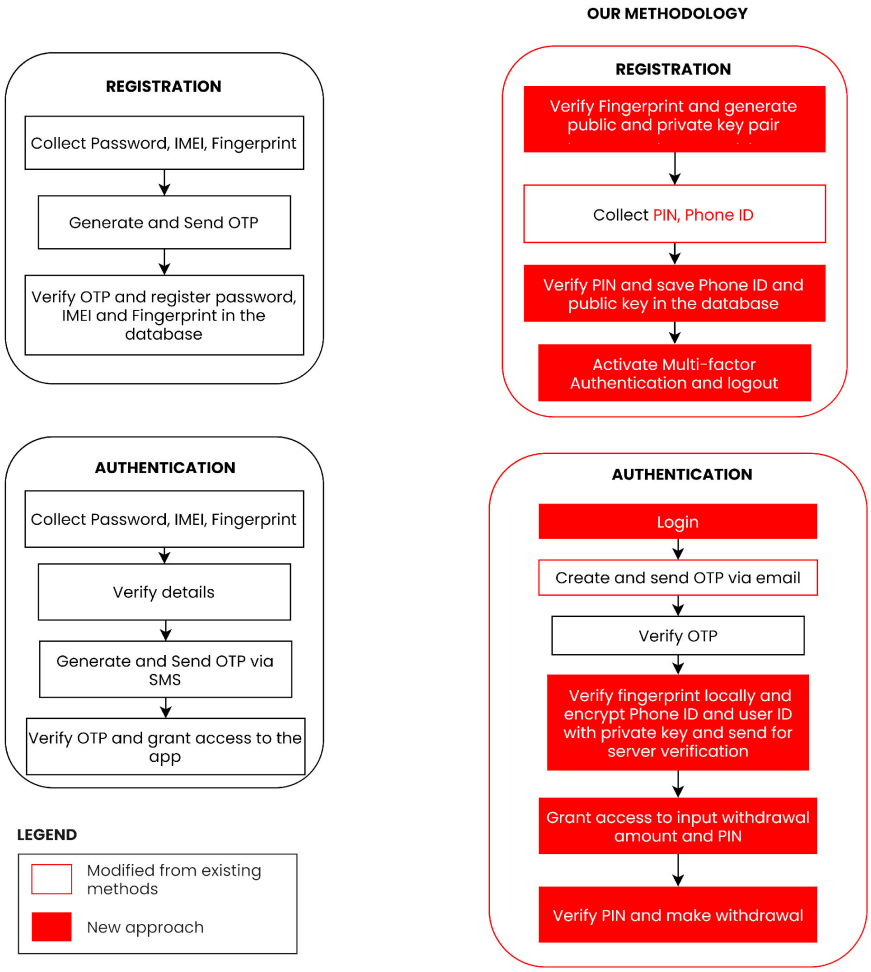



**Figure 8.** Comparison between the existing methodology and that of FapshiSec proposed in this work.

### 5.2.4. Security Features

The security features of the proposed algorithm are compared with other relevant works as summarised in **Table 5**. It can be observed that our approach globally provides better security than the schemes in [18] [19] [26] [27]. This is because of the improvements we propose in terms of phone identification (use phone ID instead of IMEI) and fingerprint authentication (authenticating both locally and server-side).

**Table 5.** Comparison of security features of our approach with related works.

| S/N | Security Feature | Ref. [26] | Ref. [18] | Ref. [27] | Ref. [19] | FapshiSec |
|---|---|---|---|---|---|---|
| 1 | Efficient authentication | Yes | No | Yes | Yes | Yes |
| 2 | Data confidentiality | No | No | Yes | Yes | Yes |
| 3 | Data integrity | No | No | Yes | Yes | Yes |
| 4 | Ensures non-repudiation | No | No | Yes | Yes | Yes |
| 5 | Ensures anonymity | No | No | Yes | Yes | Yes |

**Continued**

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | Ensures privacy | No | No | Yes | Yes | Yes |
| 7 | Prevents phishing attacks | No | Yes | Yes | Yes | Yes |
| 8 | Prevents other forms of social engineering attacks | No | No | Yes | Yes | Yes |
| 9 | Prevents brute force attacks | No | Yes | Yes | Yes | Yes |
| 10 | Resists replay attacks | No | No | Yes | Yes | Yes |
| 11 | Resists insider attacks | No | No | Yes | Yes | Yes |
| 12 | Resists masquerade attacks | Yes | No | Yes | Yes | Yes |
| 13 | Resists MITM attacks | No | Yes | Yes | Yes | Yes |
| 14 | Resists app cloning | No | No | No | No | Yes |
| 15 | Resists identity fraud | No | No | No | Yes | Yes |
| 16 | Resists specialized AI | No | No | No | No | Yes |

## 6. Conclusion

In this paper, we have proposed an efficient multi-factor authentication approach to secure digital wallets in which password, PIN, OTP, Phone ID, and biometric fingerprint authenticate users and money withdrawals. The proposed method, which comprises an enrolment phase and an authentication phase, is described and the algorithms used are implemented using various software development tools. The results presented show that the proposed approach achieves the set objectives and satisfies the functional and non-functional requirements. The security analysis reveals that the approach provides improved security, addressing the limitations of the related works. In terms of performance, our approach has a communication cost of 304 bytes, better than the 744 bytes of [27]. The approach also shows improved performance in terms of the computational cost since only the enrolment and authentication phases are involved. Globally, it has been shown through scenarios that *FapshiSec* prevents and mitigates payment fraud attacks such as brute force, phishing, and identity theft. We have also demonstrated that our approach can resist shoulder-surfing, social engineering, replay, insider, MITM, app cloning, identity fraud, and specialized AI attacks, unlike most state-of-the-art approaches. In future works, we intend to lay more emphasis on the security analysis of the proposed solution by presenting other possible scenarios that could happen.

## Acknowledgements

The authors are grateful to Dr. Tsague Aline and Dr. Sop Deffo Lionel Landry from the department of Computer Engineering of the University of Buea for all the helpful discussions and readings.

## Author Contribution

All the authors contributed equally.

## Data Availability Statement

The data that support the findings of this study are not openly available due to reasons of sensitivity and are available from the corresponding author upon reasonable request. Data (source codes) are located in controlled access data storage at Fapshi.

## Conflicts of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

## References

[1] Liébana-Cabanillas, F., Kalinic, Z., Muñoz-Leiva, F. and Higueras-Castillo, E. (2024) Biometric M-Payment Systems: A Multi-Analytical Approach to Determining Use Intention. *Information & Management*, **61**, Article 103907. https://doi.org/10.1016/j.im.2023.103907

[2] Zarco, C., Giráldez-Cru, J., Cordón, O. and Liébana-Cabanillas, F. (2024) A Comprehensive View of Biometric Payment in Retailing: A Complete Study from User to Expert. *Journal of Retailing and Consumer Services*, **79**, Article 103789. https://doi.org/10.1016/j.jretconser.2024.103789

[3] Douanla Meli, S., Fosso Djoumessi, Y. and Djiogap, C.F. (2022) Analysis of the Socio-Economic Determinants of Mobile Money Adoption and Use in Cameroon. *Telecommunications Policy*, **46**, Article 102412. https://doi.org/10.1016/j.telpol.2022.102412

[4] Muzam, J. and Tambi, M.D. (2023) The Relationship between Mobile Money Services and Small and Medium-Sized Enterprise Growth in Bamenda, Cameroon: A Probit Model Approach. *Journal of African Business*, **25**, 394-408. https://doi.org/10.1080/15228916.2023.2196173

[5] Gyamerah, K.K. and Tetteh, F.K. (2024) Examining the Impact of Mobile Money on Financial Inclusion in Sub-Saharan Africa: The Role of Institutions and Governance. *SAM Advanced Management Journal*, **89**, 315-339. https://doi.org/10.1108/samamj-08-2024-0048

[6] Noumba, I. (2024) An Overview of the Impact of Mobile Money in Cameroon. GEFONA.

[7] Gupta, P. (2024) Securing Tomorrow: The Intersection of AI, Data, and Analytics in Fraud Prevention. *Asian Journal of Research in Computer Science*, **17**, 75-92. https://doi.org/10.9734/ajrcos/2024/v17i3425

[8] Antony, A. (2024) Fintech Fraud Detection and Prevention. *IUP Journal of Accounting Research & Audit Practices*, **23**, 239-251.

[9] Cameron, S. (2024) Top 5 Fraud Trends in 2024 and How to Mitigate Them. https://complyadvantage.com/insights/top-fraud-trends/

[10] Stripe (2024) Fraud Prevention Guide: Recognize and Stop Payment Scams. Stripe.

[11] Cameroun, P. (2023) Following Complaints of Scams, the Operator Orange Cameroon Strengthens the Security of Mobile Money Transactions.

[12] Idrus, S.Z.S., Cherrier, E., Rosenberger, C. and Schwartzmann, J.-J. (2013) A Review on Authentication Methods. *Australian Journal of Basic and Applied Sciences*, **7**, 95-107.

[13] Shaliyar, M. and Mustafa, K. (2024) Watermarking Approach for Source Authentication of Web Content in Online Social Media: A Systematic Literature Review. *Multimedia Tools and Applications*, **83**, 54027-54079. https://doi.org/10.1007/s11042-023-17559-0

[14] Malathi R., and Jeberson Retna Raj R., (2016) An Integrated Approach of Physical Biometric Authentication System. *Procedia Computer Science*, **85**, 820-826. https://doi.org/10.1016/j.procs.2016.05.271

[15] Waseem (2019) MTN Announces the Go Live Date of Mobile Money Service, MoMo, in South Africa. https://www.mtn.com/mtn-announces-the-go-live-date-of-mobile-money-service-momo-in-south-africa/

[16] Security Measures in Place for Mtn Mobile Banking. https://fastercapital.com/topics/security-measures-in-place-for-mtn-mobile-banking.html

[17] Max It Orange Cameroun. https://www.orange.cm/fr/orange-max-it.html?srsltid=AfmBOooN7iNKIFCpq-MTVVNG1Kxz9txrLoQyANJ-a_QuJcX03qe0Z0Gl

[18] Hassan, M.A. and Shukur, Z. (2021) A Secure Multi Factor User Authentication Framework for Electronic Payment System. 2021 3*rd International Cyber Resilience Conference* (*CRC*), Langkawi Island, 29-31 January 2021, 1-6. https://doi.org/10.1109/crc50527.2021.9392564

[19] Melendez, J., Noriega, J., Tiznado, J., Calderon, P., Benites, Y., Rivera, L., Herrera, J., and Mayhuasca, J. (2024) Sectrabank Model to Mitigate Computer Fraud in Electronic Operations through Banking Applications on Android Devices. *Computer Science and Mathematics*. Preprints. https://doi.org/10.20944/preprints202404.0238.v1

[20] Ali, G., Ally Dida, M. and Elikana Sam, A. (2020) Two-Factor Authentication Scheme for Mobile Money: A Review of Threat Models and Countermeasures. *Future Internet*, **12**, Article 160. https://doi.org/10.3390/fi12100160

[21] Okpara, O.S. and Bekaroo, G. (2017) Cam-Wallet: Fingerprint-Based Authentication in M-Wallets Using Embedded Cameras. 2017 *IEEE International Conference on Environment and Electrical Engineering and* 2017 *IEEE Industrial and Commercial Power Systems Europe* (*EEEIC / I&CPS Europe*), Milan, 6-9 June 2017, 1-5. https://doi.org/10.1109/eeeic.2017.7977654

[22] Iqbal, S., Irfan, M., Ahsan, K., Hussain, M.A., Awais, M., Shiraz, M., *et al*. (2020) A Novel Mobile Wallet Model for Elderly Using Fingerprint as Authentication Factor. *IEEE Access*, **8**, 177405-177423. https://doi.org/10.1109/access.2020.3025429

[23] Mega, B. (2022) Framework for Improved Security on Usage of Mobile Money Application Based on Iris Biometric Authentication Method in Tanzania. The University of Dodoma.

[24] Pathan, A.Q.M.S.U., Chakraborty, A., Kabir, M. and Thakur, K. (2019) Fingerprint Authentication Security: An Improved 2-Step Authentication Method with Flexibility. *International Journal of Scientific and Engineering Research*, **10**, 438-442.

[25] Mtaho, A.B. (2015) Improving Mobile Money Security with Two-Factor Authentication. *International Journal of Computer Applications*, **109**, 9-15. https://doi.org/10.5120/19198-0826

[26] Chetalam, L.J. (2018) Enhancing Security of MPESA Transactions by Use of Voice Bio-Metrics. https://www.academia.edu/72587262/Enhancing_Security_of_Mpesa_Transactions_by_Use_of_Voice_Biometrics

[27] Ali, G., Dida, M.A. and Elikana Sam, A. (2021) A Secure and Efficient Multi-Factor Authentication Algorithm for Mobile Money Applications. *Future Internet*, **13**, Article 299. https://doi.org/10.3390/fi13120299

[28] Porubsky, J. (2020) Biometric Authentication in M-Payments: Analysing and improving Endusers Acceptability.

[29] Syed, W.K., Mohammed, A., Reddy, J.K. and Dhanasekaran, S. (2024) Biometric Authentication Systems in Banking: A Technical Evaluation of Security Measures. 2024 *IEEE 3rd World Conference on Applied Intelligence and Computing* (*AIC*), Gwalior, 27-28 July 2024, 1331-1336. https://doi.org/10.1109/aic61668.2024.10731026

[30] Alrawili, R., AlQahtani, A.A.S. and Khan, M.K. (2024) Comprehensive Survey: Biometric User Authentication Application, Evaluation, and Discussion. *Computers and Electrical Engineering*, **119**, Article 109485. https://doi.org/10.1016/j.compeleceng.2024.109485

[31] Al-Jarba, F. and Al-Khathami, M. (2021) Review of Biometrics-Based Authentication Techniques in Mobile Ecosystem. *International Journal of Computer Science & Network Security*, **21**, 321-327.

[32] (2024) Biometric Payment System: 5 Benefits, Types and Working-NTT Data Payment Services India. https://in.nttdatapay.com/blog/biometric-payment-system/?utm_source=web-story&utm_campaign=biometric+payment

[33] U. Group (2023) How Does Biometric Authentication Work for Digital Payments?

[34] Umamaheswari, M., Chennai, D.C.O.E. and Harish, K.B. (2010) Online Credit Card Transaction Using Finger Print Recognition. *International Journal of Engineering and Technology*, **2**, 320-322.

[35] Chelliah, B. and Geetha, S. (2014) Enhancing E-Payment Security through Biometric Based Personal Authentication Using Steganography Scheme-B-Pass. In: Martínez Pérez, G., Thampi, S.M., Ko, R. and Shu, L., Eds., *Communications in Computer and Information Science*, Springer, 461-472. https://doi.org/10.1007/978-3-642-54525-2_41

[36] F. Inc. (2025) Fapshi—Collect Payments through Fapshi's APIs or Use Prebuilt Solutions.

[37] Leong, J., May Yee, K., Baitsegi, O., Palanisamy, L. and Ramasamy, R.K. (2023) Hybrid Project Management between Traditional Software Development Lifecycle and Agile Based Product Development for Future Sustainability. *Sustainability*, **15**, Article 1121. https://doi.org/10.3390/su15021121

[38] (2022) React-Native-Biometrics. https://www.npmjs.com/package/react-native-biometrics

[39] Wambugu, W. (2024) Mobile Money Fraud Typologies and Mitigation Strategies. https://www.gsma.com/solutions-and-impact/connectivity-for-good/mobile-for-development/wp-content/uploads/2024/03/GSMA-Fraud-Typologies-04-03-24.pdf

[40] Birthriya, S.K., Ahlawat, P. and Jain, A.K. (2024) A Comprehensive Survey of Social Engineering Attacks: Taxonomy of Attacks, Prevention, and Mitigation Strategies. *Journal of Applied Security Research*, **20**, 244-292. https://doi.org/10.1080/19361610.2024.2372986

[41] Chatterjee, P., Das, D. and Rawat, D.B. (2024) Digital Twin for Credit Card Fraud Detection: Opportunities, Challenges, and Fraud Detection Advancements. *Future Generation Computer Systems*, **158**, 410-426. https://doi.org/10.1016/j.future.2024.04.057

[42] Manoharan, P., Yin, J., Wang, H., Zhang, Y. and Ye, W. (2024) Insider Threat Detection: A Review. 2024 *International Conference on Networking and Network Applications* (*NaNA*), Yinchuan, 9-12 August 2024, 147-153. https://doi.org/10.1109/nana63151.2024.00031

[43] Whitelaw, F., Riley, J. and Elmrabit, N. (2024) A Review of the Insider Threat, a Practitioner Perspective within the U.K. Financial Services. *IEEE Access*, **12**, 34752-34768. https://doi.org/10.1109/access.2024.3373265

[44] Jain, S., Sharma, J., Sharma, S., Kaushik, A. and Rajawat, N. (2024) Bibliometric Analysis of Literature on Fintech and Cyber Frauds in Banking. In: Tripathi, A., Birla, S., Soni, M. and Sahariya, J., Eds., Monica Sharma *Multidisciplinary Approaches for Sustainable Development*, CRC Press, 138-144. https://doi.org/10.1201/9781003543633-22

[45] Kompara, M. and Hölbl, M. (2018) Survey on Security in Intra-Body Area Network Communication. *Ad Hoc Networks*, **70**, 23-43. https://doi.org/10.1016/j.adhoc.2017.11.006