CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter explains the literature review of the research scope which is to implement cryptography techniques to authenticate the user and the device in the proposed authentication model. This scope focuses more on smartphones since it is a common device used by the majority population around the world, thus, it is important to ensure that the data stored in smartphones are protected from intruders. Applying cryptography is one of the ways to implement security features in smartphone user and protect sensitive data from exposed. Implementation of cryptography in this research focuses more on authentication since it is the first step from the user side to access their data in the smartphone.

To support the development of the suggested authentication model and to identify the research gap, the main goals of this chapter are to identify the most pertinent criteria and requirements for authentication in smartphone applications and to identify the suitable structure of authentication in smartphone. Figure 2.1 illustrates the literature review in the context of relative objectives.

Figure 2. 1: General Structure of Literature Review Chapter

2.2 Cryptography

Developing, processing, and analyzing cryptographic algorithms with the aim of communication security is the field of study known as cryptology (Paar & Pelzl, 2010). As seen in Figure 2.2, the idea of cryptology frequently encompasses both Symmetric Cryptography and Asymmetric Cryptography. Cryptography focuses on designing a save algorithm to keep messages hidden.

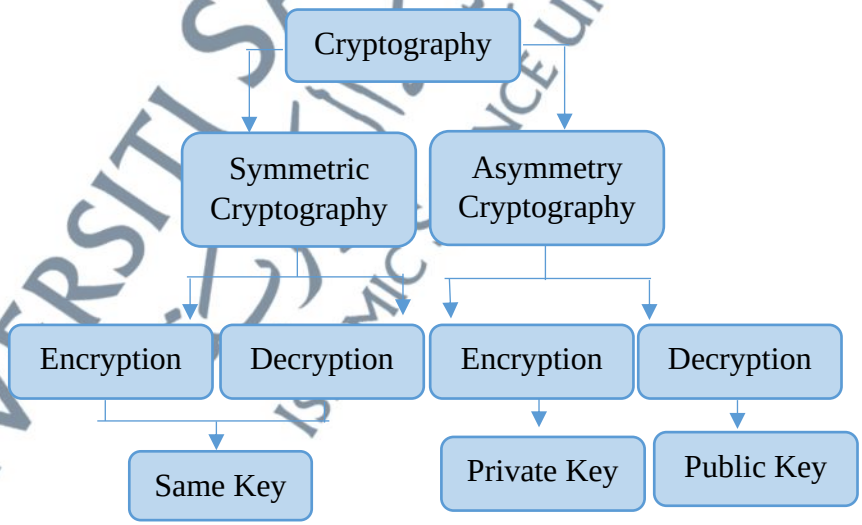


Figure 2. 2: Overview of Cryptography

Cryptography is the art of concealing or encoding data such that only the recipient of a communication can decipher it. For thousands of years, it has been used to encrypt messages, and it is being utilized today in e-commerce, bank cards, and computer passwords. By encrypting communicated communications with an algorithm and a key that is only known to the sender and receiver, cryptography maintains secrecy. Additionally, it protects surfing by using tools like virtual private networks (VPNs), which employ asymmetric encryption, public and private shared keys, and encrypted tunnels.

In cryptography, the sender and the recipient are the two persons or it can also involve one party which is the sender themselves as the sender and recipient involved in a typical communication. To prevent third parties from reading a communication, the sender uses cryptographic techniques (Rishu & Sinha, 2021). The communication is in plaintext P, while the unintelligible message is in ciphertext C. The secret key K is used in the encryption and decryption process. The transformation of P into C is called encryption. An encryption algorithm E_K contains the master key K is described as $E_K(P) = C$. On the other hand, the transformation of C into P is called decryption. E_{K-1} denotes the decryption algorithm described as $E_{K-1}(C) = P$.

Both symmetric and asymmetric encryption are used in modern times. As seen in Figure 2.3, symmetric cryptography, commonly referred to as secret-key cryptography, uses the same key to encrypt and decode data (Cordova *et al*., 2019). This sort of cryptography is broken down into five categories: block ciphers, stream ciphers, authenticated ciphers, hash functions, and message authentication codes.

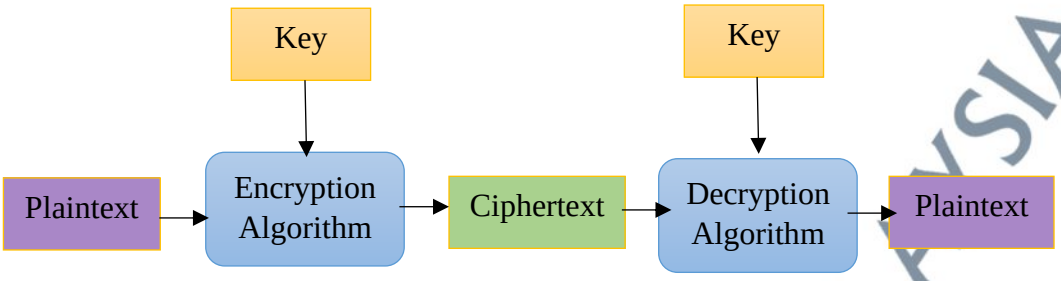


Figure 2. 3: Symmetric Cryptography

Asymmetric cryptography, sometimes known as public-key cryptography, is the study of cryptographic systems that use a public key and a private key, as depicted in Figure 2.4 (Saan *et al*., 2019). It is necessary to construct a pair of mathematically linked keys such that it is computationally impossible to separate the private key from the public key. While the public key may be shared with other parties, the private key must remain a secret. As shown in Table 2.1, public-key encryption and digital signatures are the two most well-known asymmetric cryptography applications.

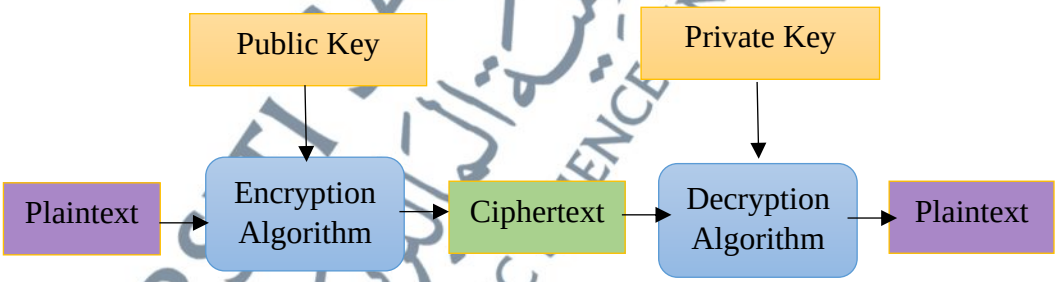


Figure 2. 4: Asymmetric Cryptography

Table 2. 1: Classification of Cryptography

Classification	Primitive	Method
Symmetric Cryptography	Block Cipher	Operates on a block of bits or bytes in plaintext to produce the equivalent block of ciphertext (e.g.: AES).
	Stream Cipher	Creates the ciphertext by utilizing a keystream to encrypt each bit of the plaintext (e.g.: K Cipher).
	Hash Function	Converts data of any length into a fixed-length digest (e.g.: SHA-1).
	Message Authentication code	Generates a fixed-length message authentication tag using a key and any plaintext.
	Authenticated Encryption	It provides secrecy, integrity, and authenticity all at once by combining a confidentiality mode and an authentication mode.
Asymmetric Cryptography	Public-Key	Utilizes the recipient's private key to decode the ciphertext and public key to encrypt the plaintext (e.g.: RSA)
	Digital Signature	Creates a signature by encrypting plaintext with the sender's private key.

Asymmetric cryptography, commonly referred to as public-key cryptography, is a type of encryption that encrypts and decrypts data using two keys: a public key and a private key (Saan *et al*., 2019). Asymmetric cryptography's main advantage is that it offers a safe means of information exchange without the necessity for a shared secret key.

Over symmetric encryption, asymmetries have several benefits. Because the private key is never shared with anyone else, it offers superior security, which is one of its biggest benefits. This implies that the communication cannot be decrypted even if the attacker manages to intercept the public key (Zhang, 2021). It also has the benefit of offering digital signatures, which may be used to confirm a message's legitimacy.

On the other hand, symmetric cryptography encrypts and decrypts data using a single secret key. Symmetric cryptography has the main advantage of being quicker and more effective than asymmetric cryptography since it uses fewer computer resources. Symmetric cryptography, however, has several drawbacks. Its requirement for a secure mechanism of sharing the secret key between the sender and receiver is one of its biggest drawbacks (Zhang, 2021). The secret key can be used to decode the message if it is intercepted by an attacker during transmission. Even though symmetric cryptography is quicker and more effective than asymmetric cryptography, but it needs a safe way for the sender and receiver to exchange the secret key.

In short, because asymmetric cryptography does not require a shared secret key, it offers greater security than symmetric encryption. It also offers digital signatures, which may be used to confirm a message's legitimacy. The value of cryptography resides in its capacity to safeguard people and data by assuring non-repudiation, secrecy, and integrity. To safeguard data in smartphone user, ensuring a strong authentication application is one of the ways to apply cryptography.

2.3 Public-Key Encryption

Asymmetric encryption, commonly referred to as public-key encryption, is a type of cryptography that encrypts and decrypts data using two keys: a public key and

a private key. The communication is encrypted using the public key, and it is decrypted using the private key (Easttom, 2022). Public-key encryption's main advantage is that it offers a safe way to exchange information without the requirement for a shared secret key.

Anyone can encrypt communications using the recipient's public key in public-key encryption, but only the owner of the paired private key can decode such a message (Assiri *et al*., 2019). Because symmetric cryptography requires the usage of a shared secret key, public-key encryption offers more security.

2.3.1 Rivest-Shamir-Adleman (RSA)

An asymmetric cryptography algorithm is the RSA formula. When something is asymmetric, it implies that it operates on two separate keys, such as a Public Key and a Private Key. The Public Key is distributed to everyone, as the name implies, while the Private Key is kept secret (Paar & Pelzi, 2010). Anyone can receive the public key, but the secret key needs to remain confidential. Given that prime numbers make up the integers and that it is challenging to discover the factors of a big composite number, the approach is based on the prime factorization issue. A crucial pair of public and private key generators is also included (Paar & Pelzi, 2014).

RSA is a valuable tool for secure communication because of a number of its characteristics. The fact that it is computationally challenging to factor huge integers into their prime factors is one of the most important characteristics of RSA. This indicates that factoring in the public key makes it challenging to defeat RSA encryption. Digital signatures, which may be used to confirm a message's validity, are another

feature of RSA. RSA can be implemented in Digital Signature. Figure 2.5 below shows the flow of RSA algorithms.

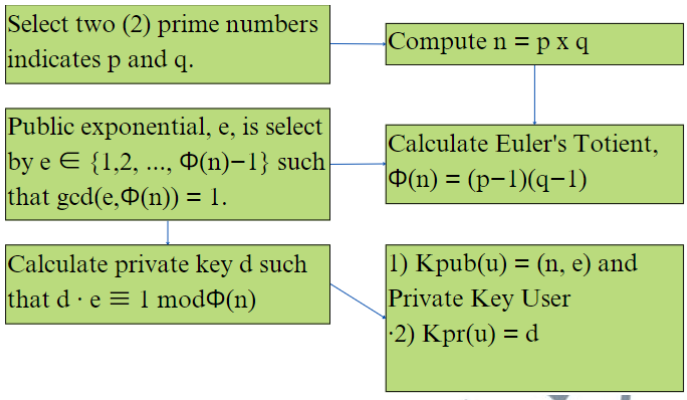


Figure 2. 5: Flow of RSA Algorithm

2.3.2 Elliptic Curve (ECC)

The same security is provided by ECC as by RSA, but it has a smaller computational footprint and uses less CPU power (Paar & Pelzi, 2014). ECC is a public key-based algorithm like RSA, but it is sort of expressed in an algebraic structure. An equation of this type defines an elliptic curve as a flat algebraic curve. It is non-singular, meaning it lacks self-intersections and cusps. It's really that easy, well, not really that easy. It's a curve where all the points (x, y coordinates) fulfill an equation.

A digital signature and key exchange both require an elliptic curve. By combining the key agreement with a symmetric encryption technique, they may be utilized for encryption (Lara-Nino *et al.*, 2018).

Elliptic curve cryptography generates keys using elliptic curve mathematics. It is the best algorithm to utilize in limited situations like mobile devices and embedded systems because it offers higher security with lower key sizes than other algorithms. Like other public-key encryption methods, ECC works similarly, with the sender

encrypting the message using the receiver's public key and the recipient decrypting it using their private key. Figure 2.6 below shows the flowchart of the ECC algorithm.

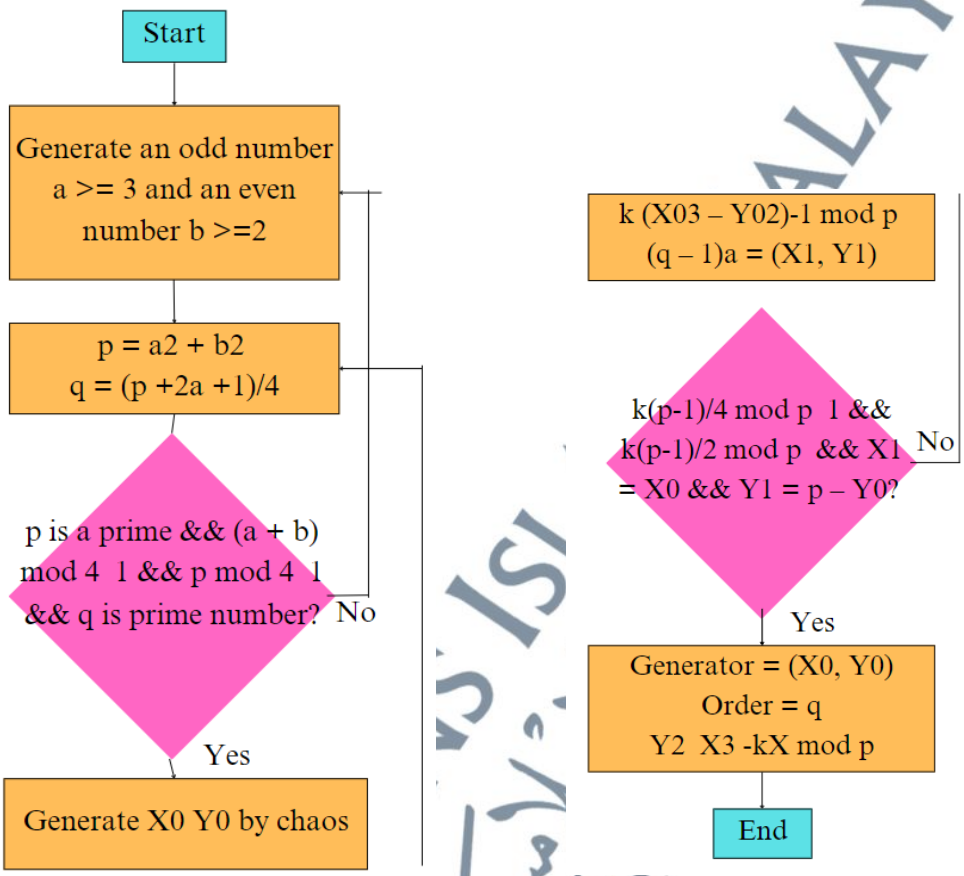


Figure 2. 6: Flowchart of ECC Algorithm

2.3.3 Diffie-Hellman Key Exchange (DHKE)

Using Diffie-Hellman, two persons can create a shared secret that cannot be observed by listening to their conversation (Lara-Nino *et al.*, 2018). In a nutshell, it is a technique for safely transferring cryptographic keys between two parties via a public channel. Using an unsecure channel, two parties with no prior knowledge of one another can construct a shared secret key together using the Diffie-Hellman key exchange

mechanism. Using a symmetric key cipher, this key may then be used to encrypt subsequent conversations. Table 2.2 below shows the comparison between RSA, Elliptic Curve, and Diffie-Hellman Key Exchange.

Table 2. 2: Comparison Between RSA, Elliptic Curve, and Diffie-Hellman Key Exchange.

Elements	Rivest-Shamir-Adleman (RSA)	Elliptic curve	Diffie-Hellman
Process	Key pair generation, Encryption and decryption, and creating a digital signature.	Key pair generation, Encryption and decryption, and creating a digital signature.	The process only involves key pair generation. It does not involve encryption and decryption or digital signature.
Transmission of public key	The public key is transmitted over an unsecured channel.	The public key is transmitted over an unsecured channel.	The public key is shared in the form of a secret key between two parties.
Size of CPU	Requires large CPU consumption	Requires small CPU consumption	Requires small CPU consumption
Key Size	1024, 2048, 3072, 7680	160, 224, 256, 384	1024, 2048
Rate of key generation	Slow	Fast	Slow
Digital Signature generation	Faster to generate digital signature compared to elliptic curve	Slower to generate digital signature compared to RSA	-
Implementation	Easy and widely use	Complicated and rarely use	-

Table 2.3 below shows the comparison of energy consumption between RSA and ECC to implement authentication for smartphone applications. Choosing the size of the keys is the first and most crucial step in the construction of an algorithm. In light of the fact that higher key sizes increase security but also increase cost, care should be taken while selecting keys to ensure the lowest key size and greatest level of security. To maintain enough cryptographic strength, the necessary RSA key size keeps growing from 1024 bit to 15360 bit. ECC can still provide the same degree of security and cryptographic stability with reduced key sizes. ECC reduces the amount of calculation required, hence raising safety. The following table compiles the range of key sizes.

Based on the table below, with the same security level, ECC requires less key size compared to RSA thus, requires less energy consumption to generate key size.

Table 2. 3: Comparison of Energy Consumption Between RSA and ECC**.**

Security Level		80	112	128	192
Key Size	RSA	1024	2048	3072	7680
	ECC	160-233	224-255	256-383	384-541
A Practical Evaluation on RSA and ECC-Based Cipher Suites for IoT High-Security EnergyEfficient Fog and Mist Computing Devices (Suarez-Albela et al., 2018)		Yes	Yes	Yes	Yes
A privacy-preserving smart parking system using an IoT elliptic curve-based security platform (Chatzigiannakis & Pyrgelis, 2016)		Yes	Yes	Yes	Yes
Elliptic Curve Cryptography for Real-Time Embedded Systems in IoT Networks (Dhillon & Kalra, (2016)		Yes	Yes	Yes	Yes
Comparison of ECC and RSA Algorithm in Resource-Constrained Devices (Bafandehkar et al., 2013)		Yes	Yes	Yes	Yes

2.4 Digital Signature and Digital Certificate

A digital signature, which is a mathematical procedure, may be used to verify the authenticity and integrity of a communication, piece of software, or digital document (Gallegos *et al.*, 2020). It is a type of electronic signature that uses public-key cryptography to assure authenticity and non-repudiation.

A message digest which is a predetermined-length summary of the message must first be encrypted using a private key for a digital signature to work. The sender's public key and the encrypted digest are then included with the message (Ganesh Kumar & Arivazhagan, 2014). The recipient can then use the sender's public key to decrypt the message by comparing the message's digest to one that was locally generated. If the two digests match, the recipient may be certain that the message was sent by the sender and was not changed.

Compared to traditional signatures, digital signatures provide a number of advantages. Because they are based on public-key cryptography, they are more secure than conventional signatures, which is one of their biggest advantages. Therefore, only the owner of the private key can generate a legitimate digital signature. They also provide non-repudiation, which makes it impossible for the sender to claim not to have sent the communication (Zhu et al., 2020).

Secure Socket Shell (SSH) and Secure Socket Layer/Transport Layer Security (SSL/TLS) are just a few secure communication methods that frequently include digital signatures (Diemert & Jager, 2021). To confirm the validity of communications and guarantee that they have not been tampered with, they are also employed in digital certificates and certificate authorities.

Compared to traditional signatures, digital signatures provide a number of benefits. Because they are based on public-key cryptography, they are more secure than conventional signatures, which is one of their biggest benefits. Therefore, only the owner of the private key can generate a legitimate digital signature. They also provide non-repudiation, which makes it impossible for the sender to claim not to have sent the communication. Figure 2.7 below shows the flow of the digital signature verification.

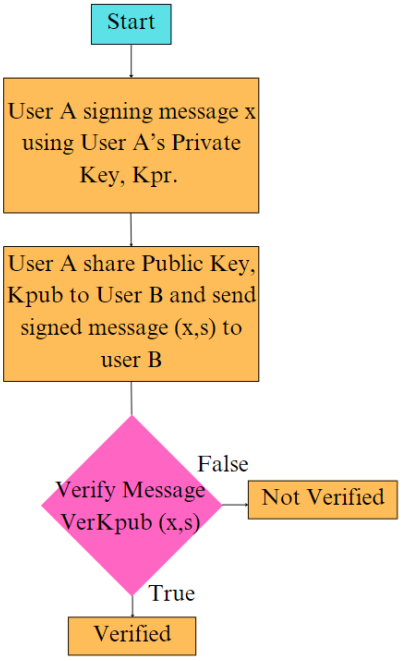


Figure 2. 7: Flowchart of Digital Signature Verification

2.4.1 Types of Digital Signature

There are a few types of digital signatures that are significant in the cryptography field and its applications. Table 2.4 below shows the types of the digital signature and the explanation of each of the types listed.

Table 2. 4: Types of Digital Signature

Types of Digital Signature	Explanation
Simple Electronics Signature (SES)	This is the simplest type of electronic signature, which involves the signer writing or drawing their name without any validation. This might be as simple as typing the user's name or pasting a copy of a written signature onto a document (Kumar & Sharma, 2022). Although simple to employ, this sort of signature has poor security and is not legally acceptable.
Basic Digital Signature (BDS)	Similar to SES, this sort of signature also encrypts data to protect it. However, it does not validate the authenticity of the document or the signature (Nia *et al*., 2015). Compared to SES, this sort of signature is a little bit more secure, but its legal validity is still quite low (Nia *et al*., 2015).
Qualified Electronic Signature (QES)	This signature type uses a qualified digital certificate issued by a recognized authority to confirm the signer's identity and the document's integrity, making it the most secure and legally acceptable type available (Krawczyk, 2014). This sort of signature can be used for any legal purpose and has the same legal effect as a handwritten signature.
Digital Certificate	This is a broad term for any sort of signature that creates a special code that identifies the signer and the document using cryptographic procedures. Three algorithms, including a key generation algorithm, a signing algorithm, and a verification method, normally make up a digital certificate scheme (Linden *et al*., 2017).

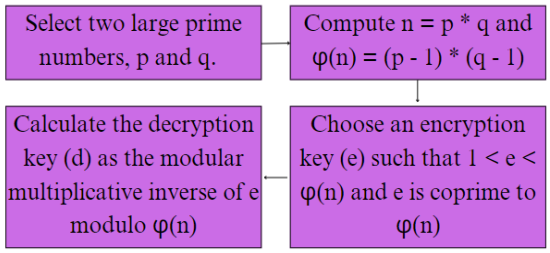
2.4.2 Algorithm Used in Digital Signature

In the digital age, digital signatures are an essential part of secure communication and authentication. An overview of the typical algorithms used in digital signatures is provided here.

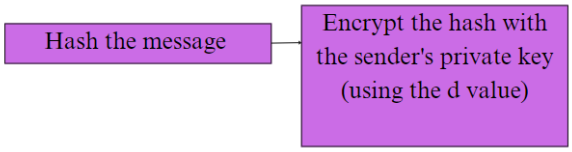
2.4.2.1 RSA (Rivest-Shamir-Adleman)

There are three processes using the RSA algorithm for Digital Signature which are Key Generation, Signing, and Verification. Figure 2.8 below shows the flow of the RSA Algorithm in Digital Signature.

i) Key Generation



ii) Signing



iii) Verification

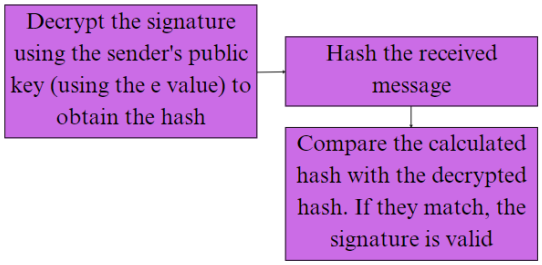
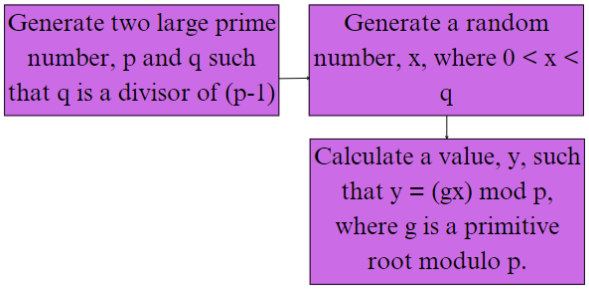


Figure 2. 8: The Flow of RSA Algorithm in Digital Signature

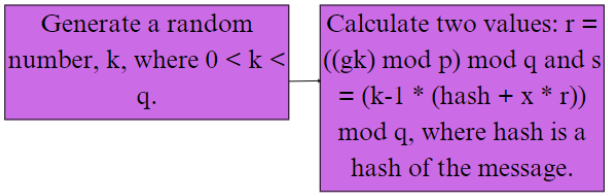
2.4.2.2 DSA (Digital Signature Algorithm)

There are three processes using the DSA algorithm for Digital Signature which are Key Generation, Signing, and Verification. Figure 2.9 below shows the flow of DSA Algorithm in Digital Signature.

i) Key Generation



ii) Signing



iii) Verification

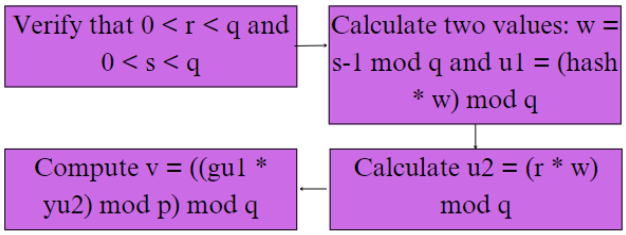


Figure 2. 9: The Flow of DSA Algorithm in Digital Signature

2.4.2.3 ECDSA (Elliptic Curve Digital Signature Algorithm)

ECDSA is an elliptic curve cryptography algorithm that works in the same way as DSA. ECDSA uses points on an elliptic curve and modular arithmetic to sign and verify digital documents and messages. These algorithms are essential for digital signatures and ensure that digital messages and documents are authentic and trustworthy (Kavin & Ganapathy, 2021). There are many factors to consider when choosing an algorithm, such as security requirements efficiency, and compatibility with different

applications and systems. RSA and ECDSA differ in terms of security, performance, and compatibility, but they are both widely used and considered safe when implemented correctly (Saho & Ezin, 2020). Table 2.5 below explains the differences between RSA Digital Signature and ECDSA in terms of security, performance, and compatibility.

Table 2. 5: Differences Between RSA Digital Signature and ECDSA

Element	RSA Digital Signature	ECDSA
Security	RSA's security relies on the fact that large composite numbers are extremely difficult to factor. Although RSA is safe as long as the key length is at least 1024 bits, it becomes vulnerable when quantum computers are powerful enough to factor in large numbers efficiently (Huang & Wang, 2015).	The key to ECDSA's security is based on the Elliptic Curve Discrete Logarithm Problem. Generally, ECDSA is considered safer per bit than RSA, meaning you can get the same security with smaller keys. This can be beneficial in terms of performance and storage (Kavin & Ganapathy, 2021).
Key length	To keep up with the ever-evolving threat landscape, RSA key sizes have been growing. This can lead to larger key sizes that can affect performance and storage requirements (Hannan *et al.*, 2020).	With smaller key sizes, ECDSA can offer the same level of security as RSA, making storage and calculation more efficient.
Performance of Signature generation and verification	In terms of generation signature and verification, RSA is proven faster because of its straightforward algorithm flow compared to ECC (Ullah *et al.*, 2023).	ECC takes a bit longer in signing and verifying a signature compared to RSA since the computation is complicated compared to RSA (Kashif *et al.*, 2023).
Compatibility	RSA is also more widely used in different crypto frameworks and protocols, which makes it a better fit in some situations and applications (Ullah *et al.*, 2023).	ECDSA is becoming increasingly popular, but it may not have the same level of adoption as RSA in every application and system (Kashif *et al.*, 2023).

In the end, RSA vs. ECDSA is a case-by-case comparison. While both are considered safe, RSA is still popular and will remain a dependable choice for many applications, particularly when compatibility issues arise especially using digital signature and verification of the signature. Table 2.6 the comparison between RSA and ECC based on signature generation and signature verification. Even in cryptographic systems the "security level" specifies the amount of protection afforded by the cryptographic algorithm in terms of computer time needed to crack the cipher. This effort is often quantified in bits of security. For instance, security level of 80 bits implies that an attacker would require about 2^80 operations to penetrate the encryption. The security level is compared between 80- bits, 112-bits, 128-bits and 192-bits. From the table below, it can be seen that to generate the signature and verify the signature, RSA takes a shorter time compared to ECC. Although ECC is favorable in terms of smaller key size, ultimately RSA is much more convenient since it is widely used.

Table 2. 6: Comparison in Terms of Signature Generation and Signature Verification Time Between RSA and ECC

Article	Security Level	Key Size		Signature Generation Time (s)		Signature Verification Time (s)	
		RSA	ECC	RSA	ECC	RSA	ECC
1. A Comparative Study of RSA and ECC and Implementation of ECC on Embedded Systems (Alam, 2016) 2. A Review: Security of Data in Cloud Storage using ECC Algorithm (Harsha & Patil, 2017) 3. Performance-Based Comparison Study of RSA and Elliptic Curve Cryptography (Sinha et al., 2013)	80	1024	163	0.0	0.15	0.01	0.23
	112	2240	233	0.15	0.34	0.01	0.51
	128	3076	283	0.21	0.59	0.01	0.86
	192	7680	409	0.53	1.18	0.01	1.80

2.4.3 Digital Certificate

A digital certificate (or digital key certificate or digital identity certificate) is an electronic document that verifies that a website, an individual, an organization, a user, a device, or a server is who they say they are. A digital certificate is issued by a Certification Authority (CA), which is a third-party organization that verifies who the certificate holder is.

Digital certificates are a way for people to securely communicate with each other over the internet. It's all thanks to Public Key Infrastructure or PKI. Basically, it's a bunch of protocols and tech that make it possible to send and receive data over the internet with just a public and private key (Zhu & Lin, 2016). The public key is what the certificate holder uses to encrypt the data they're sent, and the private key is what they use to decrypt it. Digital Certificate plays a critical role in verifying the authenticity of digital data and online transactions (Gobal, 2019). Table 2.7 below explains the elements of Digital Certificate and their roles.

Table 2. 7: Elements in Digital Certificate and its Roles

Elements	Explanation
Digital Identity Verification	A digital certificate is the primary form of digital identification in the digital environment. It assigns a cryptographic key pair, known as a public key or a private key, to a specific individual, device, or object. The public key is kept confidential, while the private key is widely disseminated (Liu *et al.*, 2020).
Issuing Authority	Digital certificates are issued and validated by Certificate Authorities (CA's). These CA's authenticate certificate applicants and validate their public keys (Ali *et al.*, 2021).
Key Pair	1) Public Key: This is the public part of a cryptographic key pair. It is used for encrypting data and verifying digital signatures generated with the private key. 2) Identity information: The Certificate contains information about the Certificate Authority (CA) such

	as the Name, Email Address, Organization, and other identifying information. 3) The Digital Signature: This is used to verify the authenticity and integrity of the certificate by digitally signing it using the CA's private key. Anyone can verify the authenticity of the certificate using the public key of the CA (Ali *et al.*, 2021).
Trust Hierarchy	Digital certificates are stored in a trust hierarchy called the PKI (Public Key Infrastructure). Root Certificate Authorities (Certificate Authorities) are the most trusted and well-known Certificate Authority (CA) in the digital certificate ecosystem. Root Certificate Authorities (CAs) issue intermediate Certificates to lower-level Certificate Authority (CAs) that issue end-user Certificates. The chain of trust makes it possible to validate digital certificates at different levels (Uahhabi & Bakkali, 2016).
Secure Communication	Digital certificates are mainly used to create secure connections over the internet using technologies such as TLS or SSL. When you visit an HTTPS website, your browser will use the server's digital certificate to confirm that you're on the right site and that your information is securely transferred (Dastres & Soori, 2020).
Authentication and Digital Signatures	Digital certificates are also used to verify users and create digital signatures. When a user digitally signs a document (such as an email), their certificate's private key creates a unique cryptographic signature. This signature can then be verified using the corresponding public key, ensuring the document's authenticity and the signer's identity (Afrianto *et al.*, 2020).
Revocation	Digital certificates have an expiration date. The certificate can be revoked if a private key is stolen or if the certificate holder's status (e.g., they leave an organization) changes. Certificate Authorities (CAs) keep a Certificate Revocation List (CRL) or use OCSP (Online Certificate Status Protocol) to notify users and systems of revoked certificates (Gu & Chen, 2020).

2.5 Smartphone Architecture Layer

A smartphone's architecture is made up of layers. Each layer is responsible for a particular function and component of the phone. The layers work together for a smooth user experience. Figure 2.10 below shows the layers of a smartphone and explains each of the layers of the smartphone (Kim & Bahn, 2019).

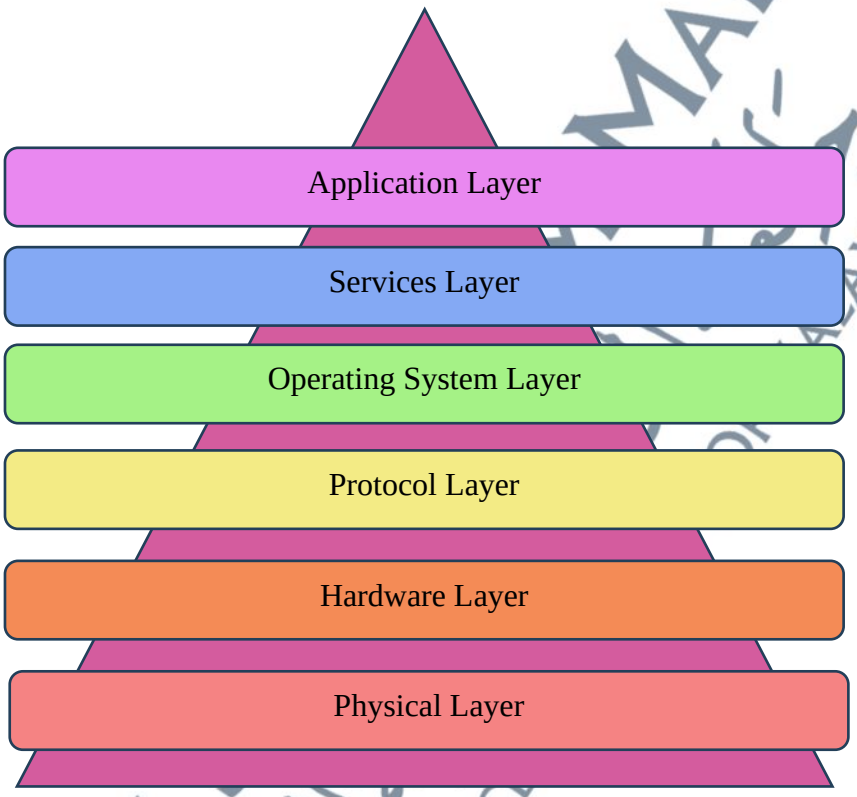


Figure 2. 10: Smartphone Layer

2.5.1 Physical Layer

The physical layer is made up of different elements that play a role in the shape, function, and overall appearance of a smartphone. Here are some of the most important elements of a smartphone's physical layer (Bai *et al.*, 2020) (Xie *et al.*, 2020):

1. Display: The screen of a smartphone, often referred to as a touch screen, is one of the most important physical components of a smartphone. It consists of the screen, along with the glass or plastic material used to protect it. Displays come in a variety

of sizes, resolutions, and technologies (such as LCD screens, OLED screens, and AMOLED screens).

2. Buttons and Controls: Smartphones have physical buttons and controls like power, volume, and camera buttons. Some smartphones also have touch-sensitive buttons or virtual navigation controls on the screen.

2.5.2 Hardware Layer

The hardware layer is made up of the physical parts and electronic devices that form the core hardware of a smartphone. The hardware layers work together to provide the functions and capabilities of the smartphone. Here are some of the most important elements of a smartphone hardware layer (Li *et al.*, 2015) (Jameel *et al.*, 2019):

1. Central Processing Unit (CPU): The central processing unit (CPU) is the main processing unit of the smartphone. It is responsible for executing the instructions and performing the calculations. Most modern smartphones have a multi-core processor for better performance and power efficiency.
2. Random Access Memory (RAM): RAM is used to store temporary memory for running programs and data. RAM enables the CPU to access data quickly and is essential for multitasking and device performance.
3. Storage: Smartphones are equipped with storage components (such as NAND flash memory), which store the smartphone's operating system, apps, and user data. The storage capacity of these components can differ, and some devices offer expandable storage capabilities via microSD cards.

2.5.3 Protocol Layer

The Protocol Layer of a Smartphone is the collection of protocols and standards that facilitate the connection and communication between the Smartphone and other Devices, Networks, and Services. These protocols enable the use of various communication modes, the exchange of data, and the networking of networks. Below are some of the key components of the Protocol Layer of Smartphones (Razi *et al.*, 2019).

1. Communication protocols such as Internet Protocol (IP) and HTTP/HTTPS protocols for internet and website connectivity.
2. Wireless Communication protocols such as Wi-Fi that enable local connectivity.

2.5.4 Operating System Layer

The OS (Operating System Layer) is an essential part of a smartphone. It controls the hardware and software of the device, enables user interaction, and supports a wide range of apps and services. The OS acts as a middleman between users, apps, and the hardware of the smartphone. Some examples are (Mrabet *et al.*, 2020) (Sudha & Jeyanthi, 2021):

1. Kernel: The kernel is what makes up the operating system of a smartphone. It is responsible for managing hardware resources such as the CPU, memory, and I/O. It also provides important services such as process management and memory management, as well as device drivers to talk to hardware components.
2. User Interface (UI): The UI layer contains the user interface elements such as the home screen, the app launcher, the notification system, and the navigation controls.

It allows the user to interact with the user interface through touch, voice, and keyboard gestures.

2.5.5 Service Layer

A smartphone's service layer is a collection of apps, services, and software that improve the device's performance and user experience. Services run on the phone's operating system and offer features, connectivity, and integrations with third-party services. Some examples such as app stores, cloud services, and social media interactions (Xu *et al.,* 2019).

2.5.6 Application Layer

The application layer is made up of the different software apps (applications) that you can install on your smartphone. These apps offer a variety of features, entertainment, and productivity applications, so you can customize your smartphone to suit your needs. Examples such as pre-installed system apps, and third-party apps downloaded from the app store and web browsers can be related to the Application layer in smartphone (Xu *et al.,* 2019).

2.6 Attacks on Smartphone Security

The security of smartphone authentication systems is subject to a variety of threats, and it is essential to comprehend these threats to create effective security measures. The following are some of the most common attacks that can arise in the context of smartphone authentication.

2.6.1 Attacks on Smartphones

This subtopic concentrates on attacks that compromise the authenticity of the application and the data stored in the user's smartphone. Most of the articles found in the literature cover a wide range of attacks, such as Man-In-the-Middle (MIM), Linkage, data manipulation, side-channel, unauthorized access, hash collision, and spoofing attacks. The following table 2.8 summarizes the articles that discuss the attacks on data confidentiality in smartphones.

Table 2. 8: Articles That Discuss the Attacks Related to Data Secrecy in Smartphone

Authors	Attacks	Attacks Mechanism
1. (Liu *et al.,* 2015) 2. (Brinkmann *et al.,* 2013) 3. (Grobauer *et al.,* 2011) 4. (Kaaniche & Laurent, 2017) 5. (Cherdantseva & Hilton, 2013) 6. (Aleisa & Renaud, 2017) 7. (Ahmed *et al.,* 2018) 8. (Abdulghani *et al.*, 2019) 9. (Claycomb & Nicoll, 2012) 10. (Grobauer *et al.,* 2011)	Man-in-the middle attack	The attacker or hacker obtains communication between the two systems and deceives the recipient into believing that they are still receiving a valid message.
1. (Kumar *et al.,* 2018) 2. (Miorandi *et al.,* 2016) 3. (Cherdantseva & Hilton, 2013)	Linkage attacks	The hacker then manipulates the captured information without accessing the real phone, revealing important data.
1. (Roman *et al.*, 2013) 2. (Williams *et al.*, 2016) 3. (Yu & Guo, 2016) 4. (Abdulghani *et al.*, 2019) 5. (Grobauer *et al.*, 2011) 6. (Miorandi *et al.*, 2016)	Data manipulations	The attacker attacks the data and application directly in the smartphone using SQL injection or cross-site scripting.
1. (Harnik *et al.*, 2017) 2. (Abdulghani *et al.*, 2019) 3. (Grobauer *et al.*, 2011)	Side-channel attacks	The attacker is indirectly exposing private information that has already been created

4. (Cherdantseva and Hilton, 2013) 5. (Aleisa & Renaud, 2017)		and analyzed by smartphone applications due to the lack of secure storage mechanisms for smartphone data.
1. (Kaaniche & Laurent, 2017) 2. (Kothmayr & Thomas, 2013) 3. (Abdulghani *et al.*, 2019) 4. (Williams *et al.*, 2016) 5. (Claycomb & Nicoll, 2012)	Unauthorized access	In the absence of effective encryption, unauthorized users can gain access to the data if it has not been encrypted.
1. (Rashid *et al.,* 2012) 2. (Aleisa & Renaud, 2017)	Hash Collision	Since the hash function takes different input lengths and outputs a short-fixed length, the hacker is indirectly exposing the private information that was previously collected and used by the phone app. This means that there's a chance that different inputs will give the same output.
1. (Hasan & Mohan, 2019) 2. (Kumar *et al.,* 2018)	Spoofing	The objective of an attack is to impersonate a legitimate device user to gain access to a smartphone to acquire information.

The most common type of MIM attack is when an attacker injects traffic between a device and a cloud-based application that may use unsecured communications or a poorly secured smartphone network. An attacker can break into two systems' communications by intercepting, delaying, or spoofing them. MIM attacks pose a serious threat to smartphones. Depending on the attacker's objective, the damage they can inflict can range from minor to catastrophic. A MIM attack can disrupt a smartphone and its applications, particularly if the authentication system is weak. In addition, a MIM attack can allow the attacker to collect personal information as well as login credentials. Lack of security in smartphones can increase the risk of MIM attacks,

as the attacker can control the instructions of smartphone applications to perform false output.

Maintaining a strong encryption mechanism between a client and the server is essential for the successful resolution of man-in-the-middle attacks, as described by Kaaniche and Laurent (2017). In this scenario, the server verifies a client's request through the presentation and validation of a Digital Certificate, after which the connection is established. As the number of interconnected data sources in a smartphone increases, the risk of unauthorized access to and leakage of sensitive data increases exponentially. Kumar et al. (2018) refer to this attack as a 'linkage attack', which involves the interception and cross-reference of multiple data sources to identify partial data. The attacker manipulates the intercepted data without affecting the smartphone and applications. Unauthorized access to the smartphone application reveals information and sensitive data. Additionally, the attacker can modify, erase, and copy the data in the smartphone application, which can compromise the confidentiality of data stored in the smartphone.

It is possible for unauthorized access to occur in multiple physical locations, thus leaving the data susceptible to physical attack. To ensure the security of smartphone data stored on a cloud or physical server, physical security solutions must be implemented (Williams et al., 2016). Currently, there are a variety of physical security solutions that can be implemented to protect the data stored in a smartphone, such as security guards; physical barriers; video monitoring; and locks. Furthermore, due to the use of connected sensors, actuators, and other physical security measures, it is recommended to integrate these physical security solutions with smartphone technology in order to enhance their effectiveness (Caycomb & Nicoll, 2012).

The manipulation of data occurs in the context of smartphone applications, where modification attacks involve the alteration of records (Williams et al., 2016). If an unauthorized individual gains access to a data file and alters the data contained therein, the security of data stored in a smartphone device is at risk. The attacker modifies the data after obtaining or accessing the data for their gain (Grobauer et al., 2011). Additionally, the exploitation of multiple vulnerabilities in smartphone applications (e.g., SQL injection, Cross-site scripting) and the exploitation of weak security mechanisms (e.g., weak passwords) are two examples of unauthorized data alteration at rest (Miorandi et al., 2016).

A secure storage scheme can be used to resolve data manipulation. A secure storage technique can prevent a smartphone data breach by using a cryptographic scheme (Yu & Guo, 2016). Side-channel attacks are predicated on finding information by examining the algorithm implementation's accessible side features (e.g., processing timing or power consumption, accompanying sounds, etc.). In this type of attack, the attacker unauthorizedly extracts data from the smartphone applications. From there, the attacker manipulates the data (Harnik et al., 2016. This affects the accuracy and completeness of the data in smartphone applications, resulting in a loss of data integrity. This type of attack can occur because there are no secure smartphone data processors and storage mechanisms (e.g., unencrypted data stored in the cloud, or on the smartphone applications). Cascading Style Sheet (CSS) Data Leakage Attacks File Confirmation and Understanding the Content of Files For example, an adversary who is already familiar with the plain text contents of a file may use the file confirmation to check if a copy of the file is saved somewhere else in the CSS (Aleisa & Renaud., 2017) (Abdulghani et al., 2018). When an attacker learns a file's content, he or she can expose sensitive information because he or she already knows most of the file's contents and

attempts to guess or recognize the unknown parts by comparing the encrypted output to the observed encrypted content (Cherdantseva & Hilton, 2013).

Transient data storage (TDR) is one of the mitigation options in the event of a side-channel attack. TDR refers to the retention of data after the execution of such systems. However, only a few studies have been conducted on the management of transient smartphone data generated during the execution of a system (Harnik et al., 2017). The importance of transient data stems from the fact that data is processed during the execution of the system to create new versions of the data. These new versions can be saved in storage for use by users or destroyed, thus reducing the risks associated with such data (Cherdantseva & Hilton, 2013).

Spoofing is when an attacker pretends to be someone, they're not by pretending to be someone else. Basically, it's a way for someone to pretend to be someone else on the device so they can get into it or get more privileges than they're supposed to have. Basically, the attacker sends fake info to the person using the phone, making them think it's from the real deal. Since the attacker has full access to the phone, it's vulnerable (Hasan & Mohan, 2019).

Replay attacks involve forging a second “duplicate call” to repeat authorized commands on a service that has been authorized and completed. When a cybercriminal intercepts a secure network communication using a smartphone, he or she intercepts the message, delays or resends the message, and then tricks the recipient into doing what he or she wants (Kumar et al., 2018). The added risk in replay attacks is that once a message is decoded from a smartphone device or network, a hacker doesn’t need specialized skills to decrypt it (Hasan & Mohan, 2019). Since spoofing helps the

attacker gain information, the data exposed to an unauthorized person may lead to a loss of data privacy. The accuracy of the data may be compromised.

2.6.2 Security Requirements and Mechanisms for Smartphones

This section focused on the security requirements and mechanisms for addressing the attacks associated with authentication in smartphones. There are a total of 26 articles on the subject of security requirements and security mechanisms. The following Table 2.9 summarizes the articles that discussed the security requirements for the smartphone user and application: From the articles, there are five security requirements for the authentication of smartphone apps. These requirements for authentication of smartphone applications explain their functionalities to maintain the security policy in the smartphone application. This requirement explains the necessity of establishing authentication in the smartphone.

Table 2. 9: Security Requirements for Smartphone Users and Applications.

Authors	**Security Requirements**	**Description**
1. (Aswale *et al.*, 2019) 2. (Al-Fuqaha *et al.*, 2015) 3. (Gubbi *et al.*, 2013) 4. (Davoli *et al.*, 2019) 5. (Hameed *et al.*, 2019)	Lightweight mechanism	A lightweight security mechanism needs to strike a balance between the cryptographic techniques employed and device constraints such as power consumption, memory capacity, and processing power.
1. (Ahanger & Aljumah, 2019) 2. (Dhumane *et al.*, 2016) 3. (Liu *et al.*, 2017) 4. (Cai *et al.*, 2016)	End-to-End Security	Security provisioning must include Secure Storage, Secure Communications, Secure Content, and Authentication.

1. (Ahanger & Aljumah, 2019) 2. (Dhumane *et al.*, 2016) 3. (Liu *et al.*, 2017) 4. (Cai *et al.*, 2016)	End-to-End Security	Security provisioning must include Secure Storage, Secure Communications, Secure Content, and Authentication.
1. (Bansal & Kumar, 2020) 2. (Yaqoob *et al.*, 2019) 3. (Das *et al.*, 2018) 4. (Hammi *et al.*, 2017) 5. (Grobauer *et al.*, 2011)	Privacy	Users will also want to protect their privacy while receiving the services they require in a timely and correct manner.
1. (Alam *et al.*, 2020) 2. (Li *et al.*, 2020) 3. (Grobauer *et al.*, 2011) 4. (Sharma *et al.*, 2018)	Identity Management	Authentication helps to identify the identity of different users in smartphone devices via login, biometrics, etc.
1. (Sicari *et al.*, 2015) 2. (Deep *et al.*, 2019) 3. (Mohsen & Jha, 2016)	Mobility	In the context of mobility, acceleration enables devices to provide transparent services while minimizing interruptions or network disconnections for the user.

Lightweight solutions should consider resource constraints such as computational constraints, power constraints, and memory constraints. One example of resource constraints is computational limitations. These constraints limit the application's implementation of cryptographic algorithms and protocols supported (Aswale et al., 2015). Lightweight security systems must balance the use of cryptographic approaches with better energy efficiency mechanisms. The fact that smartphone processors and applications are small means that an algorithm that consumes less memory, uses less power, and executes faster is needed.

End-to-end Security is another practical requirement that needs to be implemented. Because of the large number and variety of smartphone applications, the communication between devices goes through many administrative domains and technologies, so provisioning for security needs to cover the entire span of a connection (Dhumane et al., 2016). Secure Storage, Secure Communication, Secure Content, and authentication are covered by this requirement (Cai et al., 2016). Privacy is when the size and nature of a smartphone necessitate a particular focus on privacy in all forms. Users will want to maintain their privacy while receiving the services they need quickly and accurately. This will guarantee the long-term security of smartphone applications.

A smartphone application must perform identification and anonymization verification, either at the device level or at the group level. Security must include reliable ways to manage device and user identity, and the ability to manage links between those identities flexibly (Alam et al., 2020). This will involve seamless integration of multiple services across multiple domains to connect devices and users. Flexible support for identity managing and mutual authentication of users, devices apps, and associated services will be required. Security solutions will need to recognize that predicting who will participate in an interaction isn't always possible and provide ways to address the size of number of identities within a smartphone (Sharma et al., 2018). Due to scalability issues, identity won't always be finely managed and identities will often need to be managed more flexibly, for example by using one identity to represent multiple entities. While identification may seem like a general security requirement, smartphone size will require novel ways to manage identity (Grobauer et al., 2011). This requirement helps in authenticating different users of smartphone apps by logging in to the apps, biometrics, and RFID.

Smartphones can operate on a huge scale and individual components are highly mobile. Therefore, mobility requirements for smartphone applications need to be extremely dynamic. Mobility can be classified into three categories (Deep et al., 2019): Dynamic infrastructure Location privacy Multiple jurisdictions Dynamic topology Data transmission routing Real-time environment Security solutions that allow seamless transition of jurisdictions Information exchange between interconnected devices, users, and things Connected devices, people and things go smoothly Different devices, users and stuff go smoothly Data stored in database needs to be synced with mobile to execute the exact result at any location (Sicari et al., 2015) (Sicari et al., 2015).

The above subtopic explains the attacks that are related to authentication in smartphone and the security mechanism to mitigate these attacks occur.

2.7 Related Authentication Model Compared to Proposed Model

The main focus of this research is the authentication of smartphone users especially the applications that contain sensitive information that must be protected from unauthorized individuals. This subtopic discusses related models and the proposed authentication model.

2.7.1 A Novel Dynamic Randomized Secret Key Model

This research has been proposed by Yaswanth and Reddy (2023) explaining more about using a time password rather than using static password to authenticate a user in smartphone application. The algorithm used is the RSA algorithm to create the

transaction password. The proposed approach involves the creation of a pin during the cryptographic key creation phase to obtain an OTP.

The framework for the transaction then sends an OTP encrypted and implanted at the server end with a time limit. The keys are exchanged between the client and the server for authentication in a secure channel. Extracting and translating the OTP is completed at the client's convenience, and the secret message is encrypted and transmitted as a single element to prevent potential attacks. This approach also adheres to the five security standards proposed in the document. The OTP is transmitted as encrypted text [SMS]. If the substance is altered halfway, the OTP cannot be retrieved for the certifiable client, and consequently, an attack is perceived. The degree of confidentiality is affected by the secrecy with which the restrictions are maintained. The pin must be protected on the server, and the client must not disclose their pin to anyone. The proposed strategy uses a pin and Secret Key for authentication. Whenever a transaction is initiated, the client must validate credentials. Figure 2.11 and 2.12 shows the proposed model (Yaswanth & Reddy, 2023).

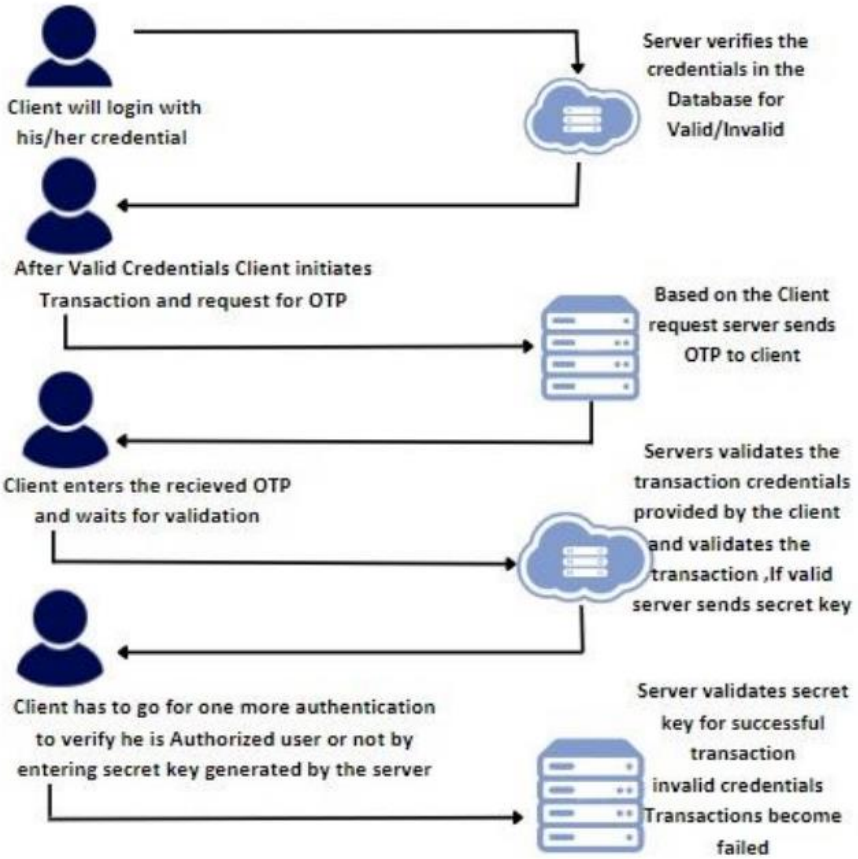


Figure 2. 11: System Architecture (Source: Yaswanth & Reddy, 2023)

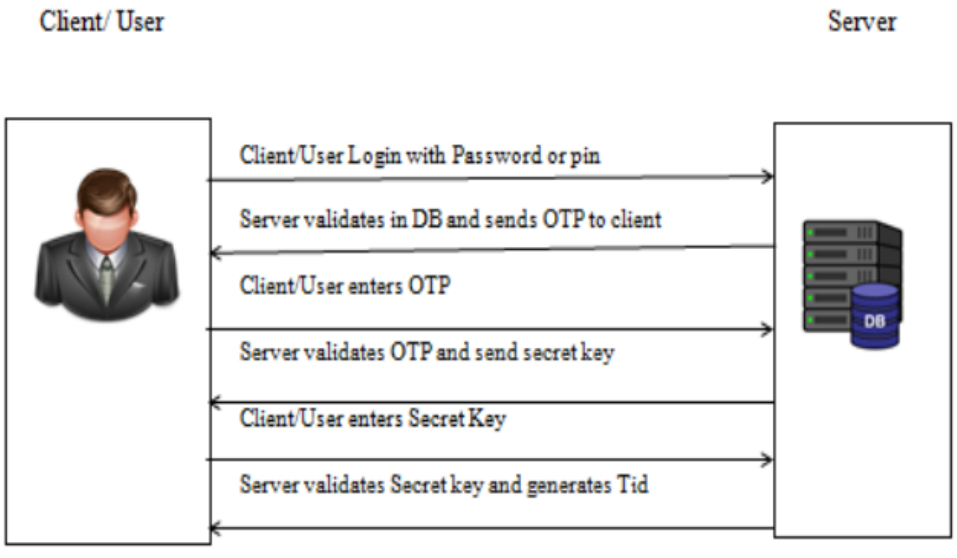


Figure 2. 12: Process of Generating OTP between Client and Server (Source: Yaswanth & Reddy, 2023)

Generating one-time passwords has a lot of advantages but the complexity to the user is one of the main concerns since the user needs to keep on receiving OTP and key in within a limited time frame. Besides, things can get worse if the attacker can retrieve the data from the user's mobile phone through phishing and thus can retrieve the OTP in order to successfully log in into user's account.

2.7.2 Lightweight Authentication Model for IoT Environments

Biometric authentication proposed by Oudah and Maolood (2022) focuses more on authenticating the user using a digital signature. The proposed system design involves the integration of multiple objects into the authentication model. Hybrid authentication infrastructure is composed of ECDSA and SSC stages, which are distributed across the system's objects to reduce computational complexity and communication overhead while providing a high level of protection against any vulnerabilities in standard algorithms. This model uses multiple devices such as smartphones, laptops, and desktops. These devices can be used to access IoT devices.

Users are required to register using a device used for IoT application, user name, and other necessary information. The server then verifies the accuracy of the received data before storing it in the database. Upon successful verification, the authentication certificate for the registered object is established. Figures 2.13 and 2.14 below show the proposed model flow for authentication process.

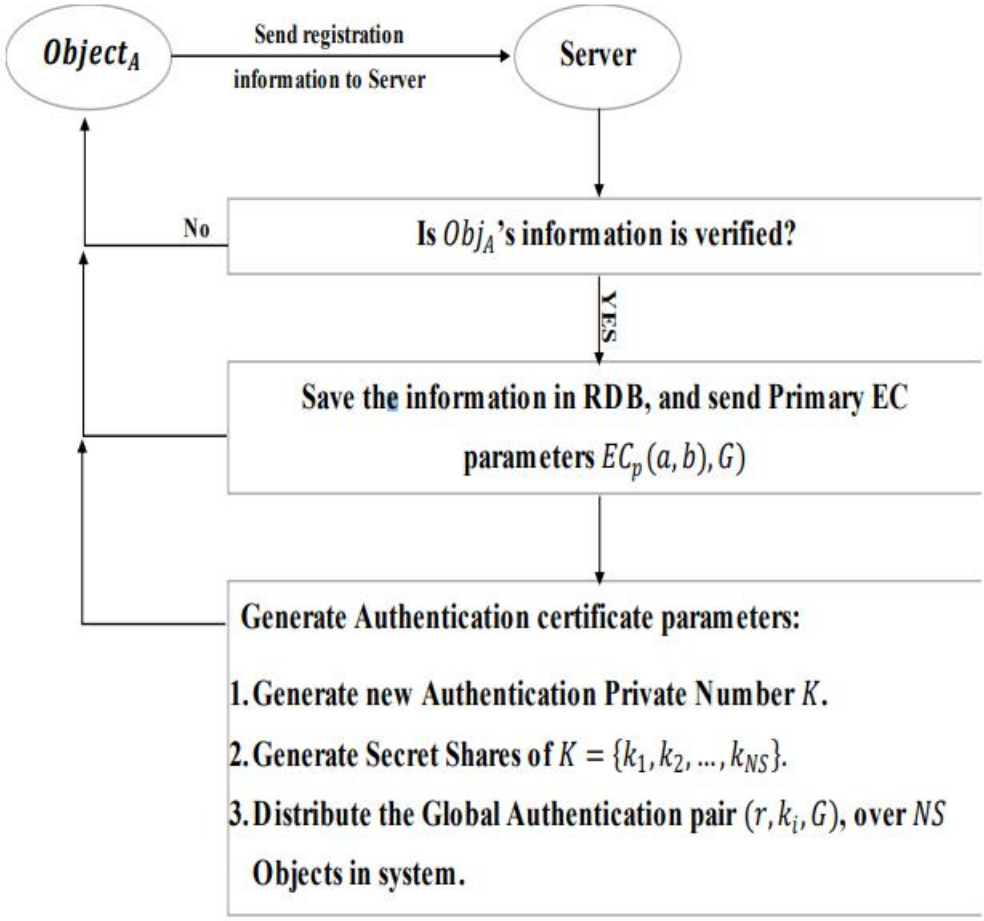


Figure 2. 13: Registration and Authentication Process (Source: Oudah & Maolood, 2022)

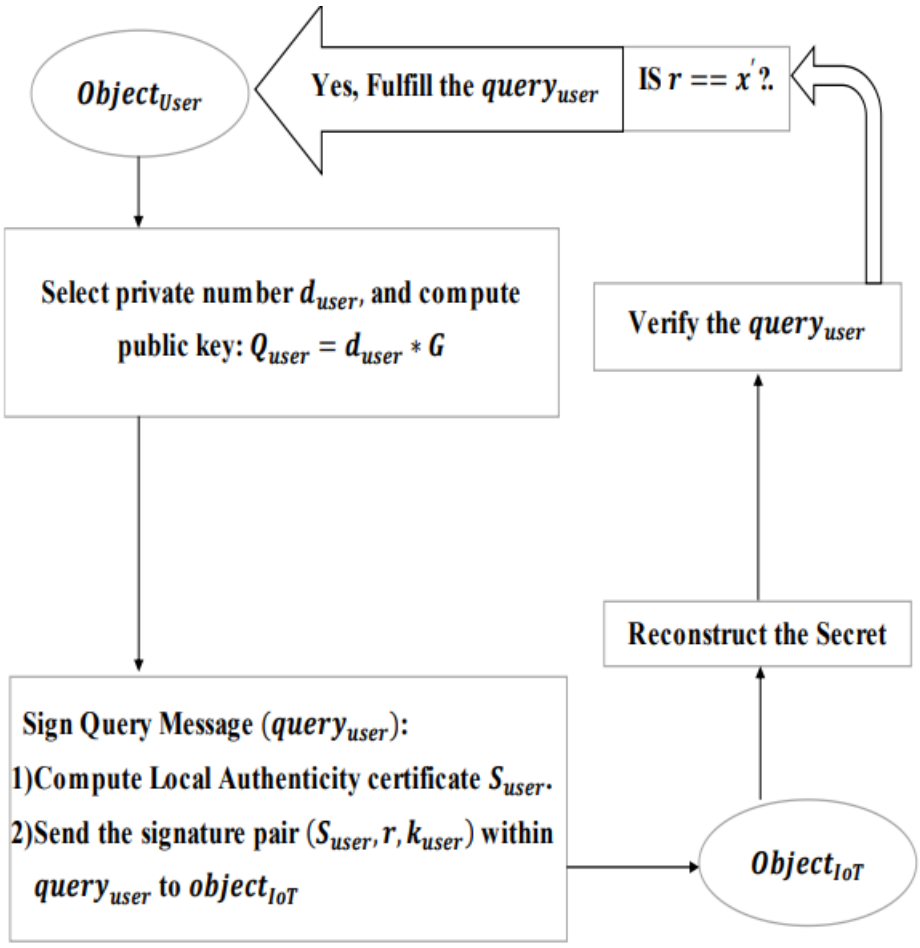


Figure 2. 14: Authentication and Verification Query (Source: Oudah & Maolood, 2022)

This model focuses more on authenticating user using digital signature. The proposed algorithm takes into account the constraints of storage and processing power in the Internet of Things (IoT) environments when addressing the underlying ECDSA deficiency and introducing a new iteration of the algorithm. Besides, since it only authenticates the user, phishing can occur where the intruder can have access to the device data and get the secret key stored in order to access the system.

2.7.3 Authenticating Data Transfer Using RSA-Generated QR Codes

This research proposed by Pangan *et al*. (2022) explained the main goal was to explain in detail the use of RSA crypto in an online authenticated data transfer. The important properties of this research are a cryptographic algorithm is employed to generate a key pair by combining the encryption and decryption functions. The public key is used by users/s to encrypt plaintexts to encrypted texts, while the private key is only used by the authenticated recipient to decrypt the encrypted text. The encryption algorithm is highly complex, making it difficult to deduce plain text from encrypted text. This provides an unbreakable form of single-directional communication, as the private key cannot be calculated from the public key. Figure 2.15 explains the conceptual framework of the VacciFied, net

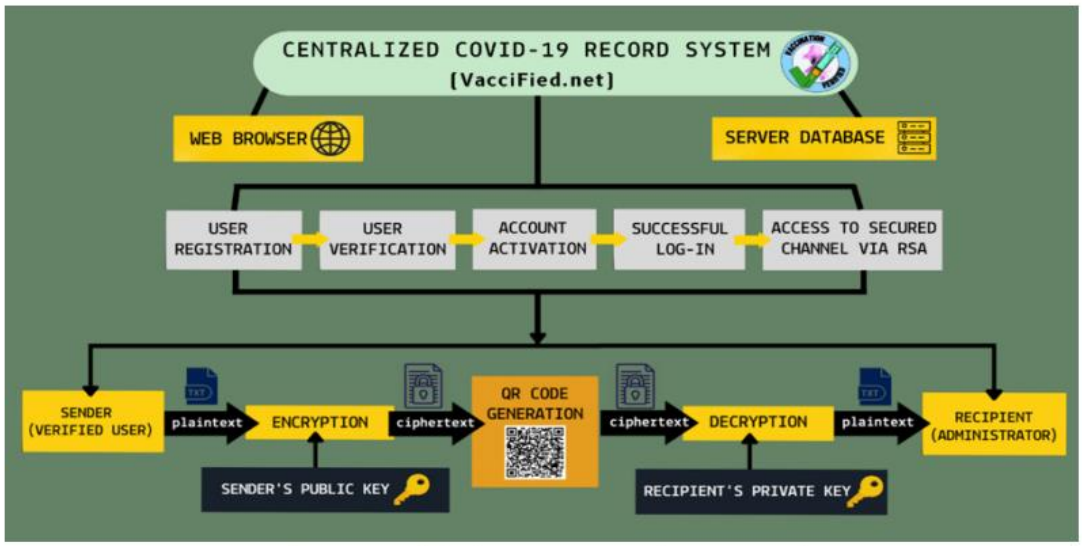


Figure 2. 15: VacciFied.net Conceptual Framework (Source: Pangan *et al*., 2022)

The parameters of this model are generating an RSA pair key where the user needs to register their ID. Once the user has registered, the QR authentication code will be given to the user. For the verification process, the public key is the one the user uses to encrypt plaintexts to encrypt texts. The private key is only used by the authenticated

recipient to decrypt the encrypted text. A cryptographic algorithm is used to generate a key pair by combining the encryption and decryption functions. The encryption algorithm has a high level of complexity. Before the admin can gain access to the user's profile after registration, they must first verify the newly registered user's verification status. The admin can do this by first verifying the personal and vaccine details provided by the user on the "Individual List" page. If the given information passes, the admin will be able to grant the user's registration status to "Verified" from "Unverified". Without the admin's approval, the registration status of the user will remain pending and inaccessible due to the complexity of the deduction of plain text from cipher text. It provides an unbreakable form of one-way communication since the private key cannot be calculated from the public key. Upon verification of the newly registered user by the administrator, the QR code will be used to direct the user to their profile page. The system is capable of detecting fraudulent or unauthenticated QR codes and will terminate the login process immediately. Additionally, it can identify verified QR codes and redirect them to the corresponding user profile page.

This model focuses on authenticating the user using a QR code with the implementation of the RSA algorithm. Since it only authenticates the user, phishing can occur where the intruder can have access to the device data and get the QR code in order to access the system.

2.7.4 Development of Two-factor Authentication Login System

This research proposed by Iyanda and Fasasi (2022) explains authentication using an OTP password via an SMS Gateway. The password was generated using a PHP OTP generator function. The Dynamic password is delivered to the mobile device

via a Bulk SMS provider. The Dynamic password generator ensures that no similar passwords are generated twice, and the generated password is automatically removed from the database. The two-factor authentication method was based on a Static Password derived by the user and a Dynamic Password. In this method, the user provided their personal information such as their username, email address, and phone number as their password. The goal of the system is to create a one-of-a-kind code that is undetectable by third parties and to encrypt the provided password to prevent any kind of database attack in order to scale the system. Figure 2.16 below shows the system architecture model.

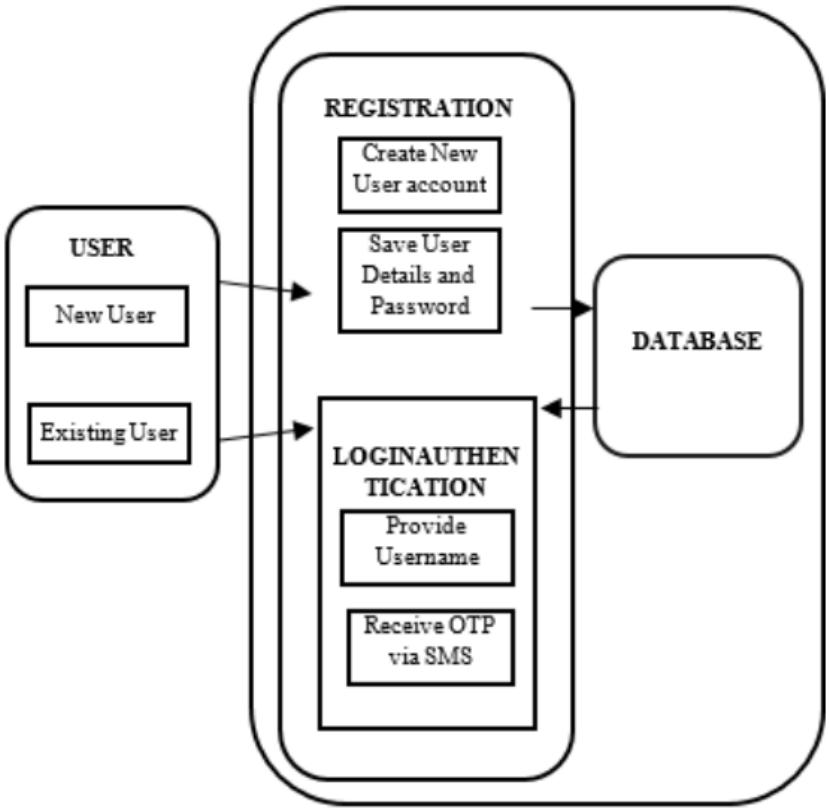


Figure 2. 16: System Architecture of Two-Factor Authentication using Dynamic Password Via SMS (Source: Iyanda & Fasasi, 2022)

One-time password generation has a lot of benefits but the complexity for the user is a major concern as the user has to keep receiving OTPs and key in in a short period of time. Furthermore, it can be even more problematic if the attacker is able to get the OTP data from the user's mobile phone via phishing, so they can get the OTP to log in to the user's account.

2.7.5 A Secure and Efficient Multi-Factor Authentication Algorithm

This research proposed by Ali *et al*. (2021) explains authentication where they propose a secure and effective multi-factor authentication system for mobile money applications, which verifies users with a combination of PIN, OTP, and biometric fingerprints. Additionally, it authorizes mobile money withdrawals by scanning the user's fingerprints and the QR code of the associated mobile money agent, which contains the user's mobile money agent code. Figures 2.17 and 2.18 below explain the flow of the authentication proposed by Ali et al. (2021).

Having multiple authentications can ensure a secure authentication process. While modern fingerprint recognition technology has advanced significantly, there are still concerns about the potential for fingerprint spoofing. Techniques using high-quality fake fingerprints or even 3D-printed replicas have been demonstrated to fool some fingerprint recognition systems. Besides, sometimes people have trouble entering their fingerprints into the system because they have physical issues like skin conditions or injuries that affect the quality of their fingerprints. In some cases, once the intruders have access to all data in the mobile phone, they can still obtain all the information such as the OTP.

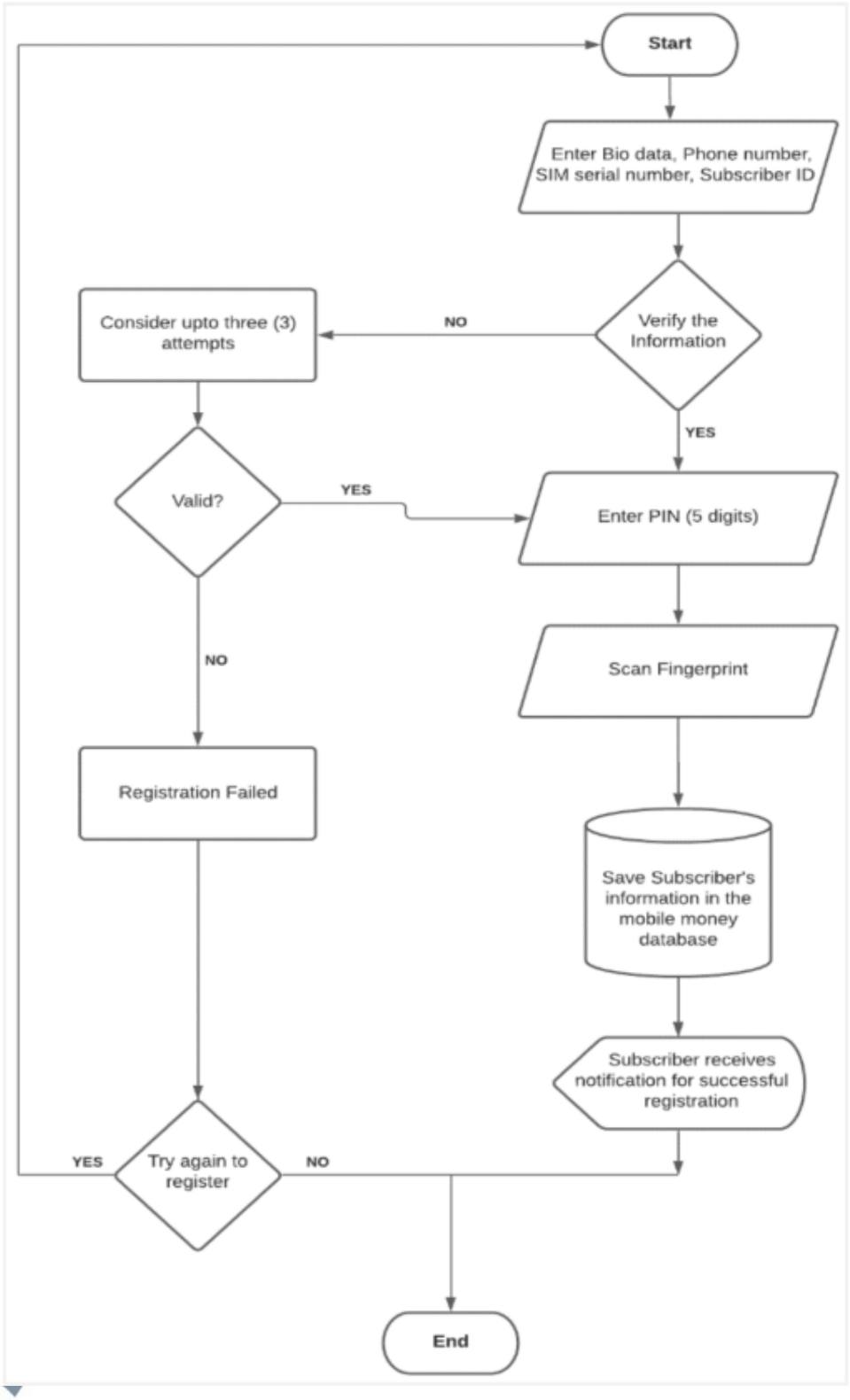


Figure 2. 17: Registration Phase (Source: Ali et al. (2021)

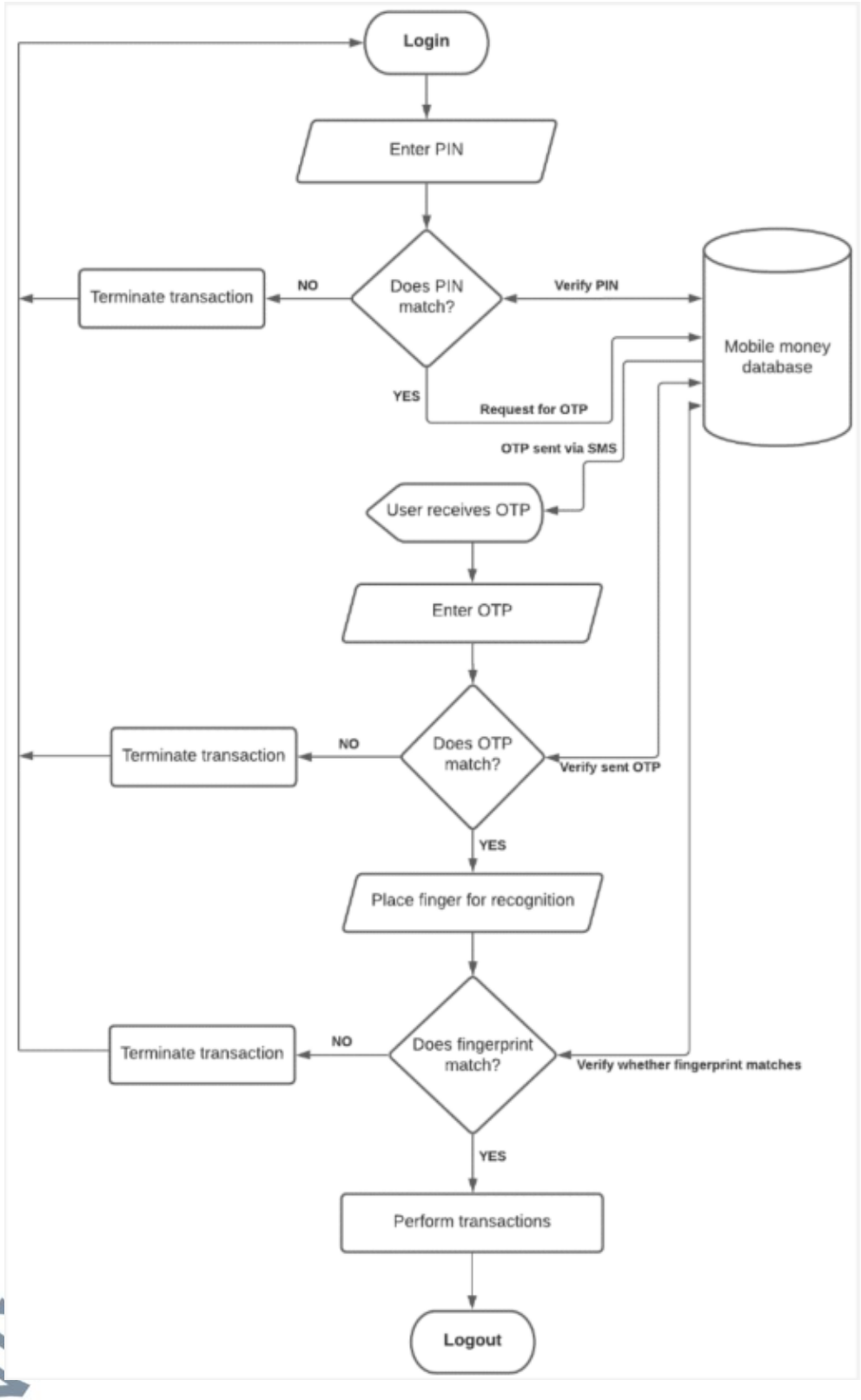


Figure 2. 18: Authentication Phase (Source, Ali et al, 2021)

2.8 Proposed Model: User-Device Authentication Model with Digital Certificate for Smartphone User

Based on the previous research explained in sections 2.7.1 until 2.7.5, the limitation that the researcher found is the authentication involves only the user even though there are multi-factor authentication involved. In this section, the researcher proposed an authentication model named User-Device Authentication Model with Digital Certificate for Smartphone User. This proposed mode includes the authentication of both the user and the device together for the user to be able to log in successfully to the system. The figure and the flow of the proposed model are explained further in Chapter 4. In summary, there are several features used to authenticate the user in a smartphone using the proposed model. Table 2.10 explains the elements used in the proposed model.

Table 2. 10: Elements in the User-Device Authentication Model with Digital Certificate for Smartphone User.

Elements	Explanation
Authentication	There are two parties involved in the authentication process which are the user and the device which is the smartphone of the user.
Algorithm	The Algorithm used in the proposed model are RSA algorithm with an implementing Digital Signature for the authentication process.

2.8.1 Conceptual Direction of the Proposed Authentication Model

One of the current challenges for smartphone user is the authentication concerns faced while logging into applications that require high data confidentiality. Thus, the main aim of this research is to propose an authentication model that can authenticate

both user and device together using digital certificate for smartphone user. Authentication in smartphone applications offers several advantages such as data confidentiality and protect sensitive data from unauthorised access. The idea that both user and device need to authenticate together will ensure that the authentication is more valid than single authentications to reduce the chances of intruders imitating the authenticated user since they do not have access to the authenticated user's device.

Reviews from various related works on the issue of authentication security in smartphone users have identified many conceptual directions. Firstly, the security of the authentication model should be focused on four levels which are the security of the smartphone devices, the security of the control server, the security of the certificate authority, and the security of the network between devices, the control server, and certificate authority.

Secondly, the security of the authentication model includes privacy and security processes. While security guards against assaults on the data and services, privacy ensures that users may use them legally. The usage of a username and password to access a service is an illustration of privacy. Encryption of data and services is an illustration of security.

Thirdly, the main security and privacy processes that could be conducted to enhance authentication security in smartphones are as follows:

1. At smartphone devices level: Data transferred using the smartphone can be secured using two actions (Liu *et al.,* 2020). First, for login users and devices into an application on a smartphone using private accessing methods such as the password for the user and IMEI number for devices. Second, installing antivirus in

smartphones to reduce malware that can attack any application especially those that require authentication.

2. At connection level: The network between the device, control server, and certificate authority can be secured through two events, effective transferring speed of data and compatibility between network and smartphone (i.e., transfer data following the received/sent speed of the device (Jeyakumar *et al.,* 2019).

Based on the sections above, the conceptual directions for the authentication model are clarified. This section explains the development activities of the proposed User-Device Authentication Model implementing Digital Certificates for smartphone user. The User-Device Authentication Model with Digital Certificate for Smartphone User is constructed based on data collection through literature review. The implementation of security in the model is the authentication of both the user and device, generating digital certificate and connection between the device, control server, and certificate authority.

At the smartphone level, it is better to authenticate the user by associating the user with their device to allow the user to have access to an application. The authentication requires the user to register the user ID as well as password. Meanwhile, the user is required to register the device using the phone number and device IMEI Number. Once the certificate authority generates certificates for both the user and the device, the signatures are used to authenticate the user and device. The password registered by the user must be a strong password where the chances for the intruder to retrieve the password through Brute Force is small. Besides, since digital signature is required for the authentication process, the smartphone needs to have a good processor speed and storage capacity for the smooth generation of the public and private keys as well as to generate the digital certificate. At the connection level, the speed of data

transfer should be effective, and there should be an effective agreement between the amount and speed of gathered data and the smartphone specifications including scheduling the transferred data as blocks based on smartphone specifications.

The primary security operations that might be performed on smartphone application authentication are listed in Table 2.11. The user-device authentication is one of the security-related tasks. The application on a smartphone can be secured through a variety of actions, including assigning a username and password to the user, retrieving the IMEI number from the device, generating digital certificates, and installing antivirus software to prevent attacks on the smartphone before, during, and after data transfer. By ensuring compliance between mobile specifications (i.e., processor) and conveyed data types and capacities, the connections between the device, control server, and certificate authority may be made safe.

Table 2. 11: Main Security Activities of User-Device Authentication in Smartphone User.

Elements	Security Method
Device	Username, password, phone number, and retrieved IMEI Number of the device.
	Antivirus
Connection between device, control server, and certificate authority	Compatibility between amounts of gathered data and mobile specifications (i.e., capacity and speed).

Table 2.12 compares the authentication models from subsection 2.7.1 until 2.7.5 with the proposed User-Device Authentication Model with Digital Certificate for Smartphone User.

Table 2. 12: Comparison Between the Related Model and the Proposed Model.

Research	Party Involve	Algorithm	Fulfill
A Novel Dynamic Randomized Secret Key Model Based on One-Time Password Authentication (Yaswanth & Reddy, 2023)	User using OTP password generated in for authentication.	RSA Algorithm is used for generating the OTP.	The degree of confidentiality and the secrecy with which the restrictions are maintained.
Lightweight Authentication Model for IoT Environments Based on Enhanced Elliptic Curve Digital Signature and Shamir Secret Share (Oudah & Maolood, 2022)	User using biometric information and digital signature for authentication.	ECDSA and SCC are used for generating the digital certificate.	The objects in the system are designed to cut down on the amount of time it takes to process data and the amount of effort it takes to communicate with each other, while also making sure there are no weak spots in the standard algorithms.
Authenticating Data Transfer Using RSA-Generated QR Codes (Pangan et al., 2022)	Users will be given a QR Code to authenticate into the system.	RSA is used for generating encryption of the user data to QR Code.	Protects the users' authentication information by changing from string to QR code and eases

			the authentication process.
Development of Two-factor Authentication Login System Using Dynamic Password with SMS Verification (Iyanda & Fasasi, 2022)	User using OTP password generated in for authentication.	MySQL and PHP are used for generating the OTP.	The scope of the restriction and the level of discretion with which the restriction is implemented.
A Secure and Efficient Multi-Factor Authentication Algorithm for Mobile Money Applications (Ali et al., 2021)	The user is authenticating by multiple factors such as the OTP, PIN, and Biometric Information.	RSA and SHA-236 are used for authenticating fingerprints and producing PIN and OTP respectively.	It protects the data integrity and non-repudiation.
Proposed Model: User-Device Authentication Model with Digital Certificate for Smartphone User	User and device need to authenticate using digital signature.	RSA and RSA Digital Signature is used for the authentication process.	To ensure secrecy when both user and device are authenticated.

2.9 Summary

Security products in electronic devices such as smartphones require a good authentication scheme to authenticate the genuine user before they can access into the system.

One way of implementing security mechanisms in smartphone devices is by applying cryptography. There are two types of cryptography mainly symmetric and asymmetric cryptography. In this research, the researcher focuses on asymmetric cryptography where the researcher used the RSA algorithm in the proposed model.

It is important to authenticate the user but to authenticate both the user and the device is much more applicable since it can enhance the security when the user and device authentication provide an additional layer of protection. Even if the user's credentials (i.e., username and password) are compromised, an attacker would still need to access the authenticated device to gain access. This greatly reduces the chances of unauthorized access.