Computer Science • 20(4) 2019 https://doi.org/10.7494/csci.2019.20.4.3179

Shuchi Dhir
Sumithra Devi K.A.

# CERTIFICATE-LESS DIGITAL SIGNATURE TECHNOLOGY FOR E-GOVERNANCE SOLUTIONS

**Abstract** *In spite of the fact that digital signatures are an essential requirement for the implementation of e-governance solutions in any organization, its use in large-scale Government ICT implementation is negligible in India. In order to understand the reasons for the low-level acceptance of the technology, the authors performed a detailed study of a famous e-governance initiative in India. The outcome of the study revealed that the reasons are related to the challenges concerning the use of cryptographic devices carrying private keys and the complicated process of generating, maintaining, and disposing Digital Signature Certificates (DSCs). A solution for the challenges understood from the case study required the implementation of a certificate-less technology, where private keys should be generated as and when required rather than storing them on cryptographic devices. Although many solutions exist that provide certificate-less technology, there have been no practical implementations to date for the use of biometrics in implementing the solution. This paper presents the first realistic architecture to implement identity-based cryptography with biometrics using the RSA algorithm. The solution presented in the paper is capable of providing a certificate-less digital signature technology to the users where public and private keys are generated on-the-fly.*

**Keywords** digital signatures, identity-based architecture, biometrics, fingerprint minutiae, RSA, private key, public key

**Citation** Computer Science 20(4) 2019: 431–452

## 1. Introduction

The limited use of digital signatures in governments and the corporate world remains an anathema in the digital era in which we live. "Usage of digital signatures across the world is very limited, theoretically we all should use smart cards, digital signatures, and Public Key Infrastructure (PKI) for authentication, but practically no one is willing to use them because of painful implementation procedures and difficulty in using them," says Peter Guttman [12]. We took a case-based approach to understanding the reasons for the low-level acceptance of digital signature technology by Indian government ICT implementations. The motivation of this research is to capture the challenges faced by end users, understand their needs, and provide a solution that is easy to use and implement. The results of our study show that the current certificate-based technology solution provided by PKI to implement digital signature technology has many limitations (such as the following):

a) It requires procedures for comprehensive management of Digital Signature Certificates (DSCs).
b) It requires assistance from third parties called Certifying Authorities (CAs) for creation, revocation, and management of DSCs.
c) Being a weak link in the security of PKI architecture, DSCs are prone to many attacks such as certificate substitution, public key substitution, parameter attacks, etc.
d) Private keys stored on cryptographic devices are prone to many attacks such as key separation.

These limitations cause significant roadblocks to the widespread adoption of digital signatures by governments and organizations across the world. In the course of our analysis, we inferred that the core issues with PKI are caused by two dependencies of the present architecture, which causes hindrances in the flexibility of implementing digital signature technology.

a) Requirement of third party agency or system for generating DSCs and associated key pairs.
b) Requirement of cryptographic devices for storing and managing private keys.

The challenges related to the dependency on third parties for generating DSCs was first understood by Shamir in 1984 [23]; he proposed an identity-based public key cryptosystem (ID-PKS). In his solution, he proposed using a user's existing public IDs (such as email addresses, official IDs, etc.) to derive his/her public key and proposed using a trusted private key generation center (PKG) for generating and distributing his/her private keys. ID-PKS is an appropriate replacement for the PKI architecture for closed groups of networks such as a government offices or organizations, where a limited number of users intend to seek cryptographic services.

Furthermore, to remove the second dependency of the PKI architecture, one should provide a robust solution for the on-the-fly generation of cryptographic keys. Cryptographic keys can be generated on-the-fly by using the characteristics of a user

that are always there with him/her, such as his/her fingerprint, voiceprint, etc. Thus, the biometrics of a person can be used for the generation of his/her cryptographic keys that can be generated as and when required, and they need not be stored on any device. Hence, a combination of the concept of Shamir's ID-PKS and cryptographic key generation through biometrics can eliminate the two major dependencies of PKI discussed above.

Even though ID-PKS seems to be a very promising architecture when compared to PKI, it has its own inherent limitations (such as the following):

a) There is no robust mechanism for key revocation. Unlike PKI (which uses certificate revocation lists [CRL]), keys are not stored in DSCs in ID-PKS, as such a key revocation becomes difficult.
b) Difficulty of message receiver authenticating sender.
c) Difficulty in authentication of users by PKG.
d) Generation and management of user's private key by PKG, which leads to key escrow problem.

Immense research efforts over a period of 15 years have provided solutions for removing the limitations of the ID-PKS architecture. To solve the first problem, Boneh and Franklin proposed a practical implementation of ID-PKS in 2001 that used Weil pairing and elliptic curve cryptography (ECC) [5]. They were the first to give the practical identity-based encryption scheme (IBE) for the revocation problem in ID-PKS. Thereafter, many researchers proposed solutions for the key revocation problem [3, 21]. Chen L. and Malone Lee J. were first to provide a practical solution for the second problem by introducing an ID-based signcryption scheme, which combined a digital signature and encryption in one logical step [6]. Currently, the major signcryption schemes are based on their work and Yuen T.H.'s blind ID-based signcryption scheme [28].

Jiang *et al.* gave a practical solution for the third problem [13,18]. They used the user's fingerprint and ECC to provide a two-party authentication scheme. In 2003, Boneh and Franklin proposed an approach to alleviate the key escrow problem using Shamir's threshold secret sharing [23]. Their approach required distributed PKGs to be online at all times (in contrast to Shamir's ID-PKS) in which PKG can be offline after private key generation. Based on their study, many researchers have provided solutions to the key escrow problem using secret sharing schemes [2, 25]. Although the key escrow problem has been mitigated to a great extent, further research is still desired to find a solution where the load of private key generation can be alleviated on PKG. Our solution of adding a biometric component where a user will generate his/her own private key will provide the solution for the fourth and most important limitation of ID-PKS. There are indeed solutions [13, 18] where biometrics is used with ID-PKS; however, a solution where biometrics is used for key generation does not exist.

Moreover, major works in ID-PKS use ECC as the basic algorithm for cryptographic key generation; however, recent studies have revealed some major disadvantages

of ECC, such as complexity and patent issues with curves. This has motivated researchers to identify the possibilities of implementing ID-PKS with RSA. Since the work done in this area is very limited, it is desirable to investigate and identify the schemes that can implement ID-PKS using RSA as a basic cryptographic algorithm. This paper proposes how biometrics and ID-PKS can be implemented together using the RSA algorithm.

Our research intends to provide a practical solution for implementing the identity-based architecture and biometrics together to resolve the practical limitations in the existing PKI architecture. This will relate the commonly used unique identity and biometrics of a person to derive a public-private key pair for the purpose of digitally signing documents. We have used unique user identification numbers to derive a public key. The paired private key is derived from the unique features extracted from the biometric details of the individual.

In the context of the discussion made above, we derive our research problem as such: "Combining the concept of identity-based cryptography with biometrics to derive a practical solution for implementing digital signature technology that will be used for the on-the-fly generation of public and private keys and implementing certificate-less digital signatures."

The main contribution in this paper can be summarized as follows:

a) Practical solution for combining ID-PKS with biometrics using RSA algorithm.
b) On-the-fly generation of user's private and public keys, which resolves inherent key escrow problem in ID-PKS architecture.
c) Complete solution to implement certificate-less digital signature technology.

This paper is organized as follows: Section 2 discusses the background and motivation behind the research; Section 3 explains the existing work done in this area; Section 4 explains the proposed solution; Section 5 presents the results of the experiments; and Section 6 concludes the paper.

## 2. Background and motivation

In order to appreciate the challenges associated with PKI, we decided to take a case--based approach. As part of this approach, we decided to study a financial management information system (FMIS) used by a provincial government in India. The case under consideration is the primary automation tool of the Treasury Department of the government; it integrates multiple financial institutions & external entities through a web of custom developed integrations. It is one of the most complex and challenging e-governance projects ever undertaken in India, making it an apt case to study for this research. The complexity of the project lies in its focus of fully automating the day--to-day functionalities of the department, which requires its secure integration with 170+ departments, banks, and other external agencies. Since the intra-departmental transactions deal with critical data related to the funds of the government, data security and the non–repudiation of the transmitted data is extremely important. The

project also envisages the provisioning of Digital Signature Certificates (DSC) to all of its users (numbering around 75,000) for signing transactions done over the system. The key challenge here is to generate, maintain, renew, and dispose the DSCs. Any loss or expiry of a DSC without timely replacement or renewal will put users out of action for the period until which a replacement arrives. We did a survey among the employees and stakeholders of the FMIS to identify their perspectives on substituting manual signatures with digital signatures. The results of the survey clearly explained the disinterest of the users in accepting the new technology. There were numerous reasons for the opposition elucidated by those surveyed, but the major finding was related to the procedural differences that the use of this technology brings between the traditional way of signing and the modern way of carrying digital signature certificates (DSC); i.e., holding e-token devices in a pocket. The key challenge pertained to the generation, maintenance, disposal, and security of DSCs and their carrying devices. Some fraud related to DSCs was also seen as a matter of concern. A few of the surveyed people were also worried about the statistics of the National Crime Records Bureau (NCRB) of India, which states that the cases of obtaining fake digital signatures is rising across the country year by year. Such challenges prevailing within the digital signature ecosystem has dissuaded users from accepting DSCs.

The present PKI architecture comes with many minuses, which may cause hindrances in the successful implementation and smooth functioning of the department. Thus, the proposed solution where identity-based cryptography is used to generate digital signatures on-the-fly is an ideal solution for this department as well as any other entity/organization that faces such challenges.

## 3. Literature survey

This paper provides a practical solution for combining ID-PKS and biometrics; thus, a brief literature review on both topics is given below.

### 3.1. Identity-based public key cryptosystem (ID-PKS)

In ID-PKS, public keys are generated using each user's public ID, and a trusted private key generation center (PKG) generates a private key. ID-PKS was first practically implemented using bilinear pairing on elliptical curves; since then, ECC has remained a prominent choice of researchers for implementing ID-PKS. Although RSA is a much simpler and more popular algorithm, distributing and storing modulus n was a challenging task that made researchers choose ECC over RSA for ID-PKS. In 2002, Boneh *et al.* proposed a solution to implement ID-PKS using mRSA (a modified version of RSA) and named it Mediated RSA (mRSA) [4]. mRSA is modified RSA algorithm in which a private key is split into two parts. The user possesses one part, and the other part is stored on a server called a security mediator (SEM). They were the first to propose a method for combining mRSA with identity-based cryptography. Later on in 2003, Ding and Tsudik proposed a slight modification of Boneh et al's scheme [9].

In their proposed architecture, they derive a user's public key from his/her email address, using a mapping function known as KG. This function performs a one-to-one mapping from identity strings to public keys. They use a single common RSA modulus n for all users, which is made public by storing it in a system-wide certificate.

The most important entity of their architecture is SEM, which they assumed would not be compromised throughout its lifetime. SEM is conceptualized as an online semi-trusted server that is responsible for performing the encryption and decryption of data as well as the revocation of a user's rights. The two major flaws in their architecture are the use of a single common RSA modulus n for all users and the assumption that SEM can never be compromised. In 2013, Elashry *et al.* presented the security vulnerabilities of Ding and Tsudik's scheme and proposed two solutions for the same [10]. They contradicted Ding and Tsudik's claim that public keys generated by a division intractable hash function are also division intractable and proved that their system can fail under some circumstances.

The PKG or SEM of ID-PKS and IBE hold an important place in the architecture, as they are the single point of trusted contact for private key generation and distribution. However, having a single trusted authority that is responsible for the generation and management of private keys leads to the problem of key escrow and opens the door to many security vulnerabilities. Boneh and Franklin proposed the use of verifiable secret sharing (VSS) to solve this problem. They used Shamir's secret sharing scheme [22] to distribute the master private key of PKG among n PKG nodes in an (n, t) manner such that a minimum t number of PKG nodes are required to regenerate the master private key. They were the first to discuss the concept of distributed PKG [2]; thereafter, many researchers have proposed practical applications using distributed PKG based on Shamir's secret sharing. In 2010, Kate and Goldberg proposed a practical architecture to implement a distributed PKG for use over the Internet [17]. They combined the concept of proactive secret sharing and forward secrecy for distributed PKG implementation.

## 3.2. Cryptographic key generation using biometrics

Biometrics offers a great scope into cryptographic key generation, so there is extensive work being done in this field. Many sophisticated mechanisms exist to convert fingerprint features to cryptographic keys. Since the proposed solution intends to derive a binary template from a fingerprint and, consequently, store it on a computer for further matching, we need to convert this template into a cancellable template to ensure security and revocability. Thus, we present a brief review of the work done to derive cancellable or revocable templates from a fingerprint. The methods used to derive cancellable or revocable templates from a fingerprint can be broadly divided into alignment based and non-alignment based. In contrast to non-alignment-based methods, alignment-based methods align a fingerprint image using the reference minutiae before further processing. Ratha *et al.* are among the pioneers in generating alignment-based non-invertible binary templates from fingerprint data [20]. Their method used three

non-invertible transform functions Cartesian, Polar, and Surface Folding, which were performed in a sequence, to generate a non-invertible binary template. Although they claimed that their transformation functions are non-invertible, Feng *et al.* [19] proved that their claim is wrong, even Shin *et al.* proved that the generated templates from their method are invertible [24].

In 2005, Ang *et al.* proposed a method for generating a revocable binary template from fingerprint minutiae [1]. They derived a binary template from an imaginary drawn line passing through the core point. The revocation of the template is achieved by changing the orientation of the line, which varies from $0^0$ to $180^0$. The orientation of the line is determined through a transformation function that takes a user's key as input. The major drawback of their method is the determination of a core point, which is not always present in a fingerprint.

In 2009, Jin, Z., *et al.* proposed an approach for generating secure binary templates from fingerprint minutiae using random triangle hashing [15]. After translating all minutiae based on random reference minutia, a binary template is derived from random triangles by calculating the number of minutiae contained in it. The performance of their proposed algorithm is very good, as they could achieve an EER of less than 1%. However, aligning the minutiae on the basis of random reference minutia will generate a different binary template each time from the same fingerprint. Moreover, a slight alteration in the reference point considerably affects the overall performance of the method; the authors did not discuss this issue in their work. They resolved this issue in a new method proposed in 2012 by generating revocable binary templates from a fingerprint using polar grid-based transformation and quantization [16].

Contrary to the alignment-based approach, the non-alignment-based approach does not align fingerprint images using a reference point. In 2007, Song and Beng proposed a non-alignment-based finger-hashing approach to transform a fingerprint into discrete binary values (called a finger-hash), which is XORed with a stabilized fingerprint to obtain a biometric key [26]. Although they used the Solomon and Reed correction method to stabilize the fluctuation in the value of the finger-hash, there are practical challenges in generating consistent keys using this method, and their scheme could not produce longer keys.

In 2012, Conti *et al.* proposed a mechanism for generating a private key on-the-fly by fusing fingerprint traits with the RSA algorithm [7]. They proposed storing an encrypted biometric trait on a smart card and using it for authentication purposes and the generation of keys. They provided a detailed method for extracting cores and deltas from a fingerprint image and then using them to generate keys by using the Log-Gabor approach. Although their experimental results display a very promising FAR of 0% and FRR of 13.27 %, they are unable to provide robust results depending only on core and delta features.

In 2014, Wong *et al.* proposed a multi-scale bag of words paradigm to convert minutiae set to a bit string [27]. They used a minutiae vector (obtained by converting a minutiae set to a minutiae descriptor) and generated a codebook from it using

a histogram approach. This codebook is further used to produce a histogram representing the input fingerprint. They apply dynamic quantisation to assign more bits to the discriminative feature components to optimise the FAR and FRR. This process even helps in improving the entropy of the generated bit string, they have presented a comparison of the entropies of the bit strings generated by other methods too. The disadvantage of their method lies in the complexity of generating the bit string. Furthermore, it lacks the explanation of the cryptographic applications where this generated key can be used as the key length is short, approximately 420 bits.

## 4. Proposed solution

Our proposed solution derives the concept of mRSA from Ding and Tsudik's work [9] and the concept of distributed PKG from the work of Baek and Zheng [2]. An overview of the proposed solution is given in Figure 1.

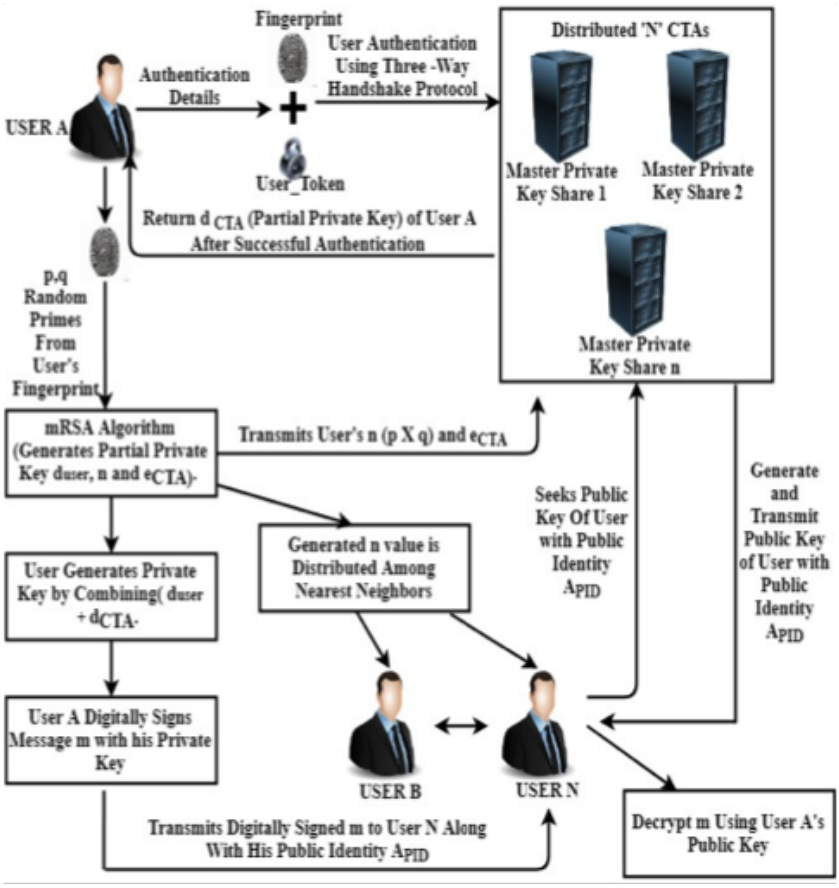



**Figure 1.** Overview of proposed system

The entities in our solution are the central trusted authority (CTA), sender (S) and receiver (R). The public and private keys of each entity are represented as follows:

a) $CTA :< M_{pub}, M_{pri} >$;
b) $S :< S_{pub}, S_{pri} >$ . The public and private key of S is divided into two parts (user and CTA) and is represented by $< S_{pub}\{e_{user}, e_{CTA}\}, S_{pri}\{d_{user}, d_{CTA}\} >$ ;
c) $R :< R_{pub}, R_{pri} >$ . The public and private key of R is divided into two parts (user and CTA) and is represented by $< R_{pub}\{e_{user}, e_{CTA}\}, R_{pri}\{d_{user}, d_{CTA}\} >$.

## 4.1. CTA

A CTA is a secure server that is responsible for the creation of trust in each entity of the system. Unlike the PKI architecture (where the trust authorities are usually third-party servers), any secure server or group of servers within the system can be promoted as a CTA, provided they implement stringent security policies and protocols. The CTA is the most important entity, as it provides a centralized control for authenticating users of the system, assisting them during the entire process of encryption, decryption, and digital signing. Unlike the PKG in ID-PKS that is responsible for the user's private key generation and distribution, the CTA will only assist users in providing them with their partial private and public keys using which user can generate its own private and public key as and when required.

During the initialization phase, the CTA generates a Master public key and a Master private key $< M_{pub}, M_{pri} >$ . $M_{pub}$ is stored in a system-wide certificate and is known to all publicly, while $M_{pri}$ is a secret key known only to the CTA. To mitigate the security vulnerabilities related to a single CTA, we propose distributing $M_{pri}$ among multiple CTAs using Shamir's secret sharing scheme [22]. Here, $M_{pri}$ is distributed among n CTAs in an (n, t) distributed manner such that a minimum of k CTA nodes are required to generate $M_{pri}$ (where k ¡ n).The distributed CTA approach renders robustness and alleviates the security vulnerability issues related to a single CTA.

The CTA generates partial private key $d_{CTA}$ for each user and stores it in the database after encrypting it with $M_{pub}$. During the private key generation process, the partial private key of the user is transmitted to the user using the Diffie-Hellman (D-H) key exchange algorithm [8]. The user then generates his/her private key by using the received partial private key.

Holding a user's partial private key provides a mechanism for CTA to manage revoked users. In such a case, if a user's rights are revoked or his/her authentication fails, it will not transmit the partial private key to him/her. The CTA is also responsible for generating and transmitting the user's public key by using its partial public key.

## 4.2. Authentication process

There are two types of authentication that would be required within the proposed system. The first one would involve the CTA authenticating a user and the message receiver authenticating a sender.

### 4.2.1. CTA authenticating a user

User authentication is a very important aspect of any system; while traditional methods of authentication like passwords, OTPs, etc. are proving to be insecure, biometrics remains the most reliable option for authentication. Nevertheless, handling biometric data is very complex, because it can vary due to several reasons; furthermore, storing biometric data for matching is perilous because, once it is compromised, it cannot be replaced. Keeping these difficulties in mind, we propose using the biometric authentication of a user, which would be further enriched using a secure three-way handshake protocol. The proposed methodology involves using a user's fingerprint features as biometric data for authentication. The entire process is divided into two phases:

**User Enrollment:** In this phase, a user's fingerprint features are captured and converted into unique code $U_C$ (a detailed explanation of this process is given in Subsection 4.3.1 of this document). This unique code is encrypted by $M_{pub}$ and stored by the CTA for future matching. $U_C$ is generated using a user-specified token. The change in token helps in generating a different $U_C$ for the user.

$$F(FingerprintFeatures, User_{Token}) = U_C \text{ (see Equation (5))}$$

Function $F$ transforms fingerprint features, and $User_{Token}$ to $U_C$ is a one-way function. It is nearly impossible to regenerate a user's fingerprint features from $U_C$.

**User Authentication:** In this phase, the query fingerprint is converted into $U_C$. This generated $U_C$ is matched with the $U_C$ stored by the CTA (a detailed explanation is given in Step 4 of Sub-section 4.3.1). Upon authentication, the CTA generates a temporary session key and transmits it to the user using a secure three-way handshake protocol. Any data that is transmitted between the CTA and the user upon authentication is encrypted using the session key.

The three-way handshake protocol for user authentication is explained below.

**STEP 1:** The user generates temporary session key $K_S$ and initiates a connection request to the CTA by transmitting encrypted query $U_C^q$ as $M_{pub}$ $(U_C^q||T_i, K_S)$. The user encrypts $U_C^q$ by using the $M_{pub}$ key and attaches a timestamp value and session key along with it.

**STEP 2:** The CTA decrypts the data sent by the user using the $M_{pri}$ key ($M_{pri}$ is generated by contacting k CTA nodes) and matches the $U_C^q$ sent by the user with the enrolled $U_C^e$ stored in the database. Upon matching, the CTA encrypts the partial private key of the user's $d_{CTA}$ using $K_S$ and transmits it to the user.

**STEP 3:** Upon receiving the partial private key, the user generates his/her own private key. A detailed explanation of this process is provided in Sub-section 4.3.2 of this document.

#### 4.2.2. Message receiver authenticating a sender

An ideal communication scenario where sender S wants to send message m to receiver R is given below:

a) S generates temporary session key $K_S$. S computes hash of $m$ using function $h$, as $h(m)$. S digitally signs $h(m)$ by encrypting $h(m)$ using its private key $S_{pri}$ as $S_{pri}(h(m))$. S encrypts digitally signed message including $K_S$ using receiver's public key $R_{pub}$ as $R_{pub}$ $(S_{pri}$ $(h(m)||$ $S_{UniquePublicID}$ $), K_S)$. S also includes its unique public id to ensure that the public key that receiver R obtains from the CTA is respective to that ID. S transmits the encrypted message to R.

b) R can decrypt the message only if:

   1) It can decrypt m using its private key $R_{pri}$. Thus, S can authenticate that data reaches the authentic R.
   2) R can decrypt m using $S_{pub}$. R obtains $S_{pub}$ from CTA for $S_{UniquePublicID}$. Thus, R can authenticate and non-repudiate S.

c) Upon successful decryption, R obtains session key $K_S$ transmitted by S encapsulated in the encrypted message. R uses $K_S$ for securing further communication with S. R generates the hash of m and compares it with the hash sent by S. if both hashes are the same, R infers that m has not been altered in transit.

#### 4.2.3. Generation of public and private keys using mRSA algorithm from fingerprint

We convert the features extracted from a user's fingerprint into a binary template – we call this unique code $U_C$. $U_C$ is a cancellable binary template representation of the fingerprint data that is used for validating a user's authenticity during signing as well as for generating his/her digital signatures. The user's partial private key is derived from $U_C$, and the public and private keys are generated from it using the mRSA algorithm. The entire process is given in detail in the following sub-sections.

#### 4.2.4. Conversion of fingerprint data to unique code $U_C$

$U_C$ is a binary template of a user's fingerprint that carries its unique biometric traits; and its value is unique for each fingerprint. Each fingerprint image has many local features (called minutiae) and global features (such as cores, deltas, etc.); these features altogether render uniqueness to an individual's fingerprint. After performing a series of steps (such as image enhancement, binarization, thinning, etc.), our method generates a template binary code called $U_C$ from the extracted minutiae (local features). The standards prescribed by the National Institute of Standards and Technology (NIST) through its fingerprint minutiae viewer (FpMV) has been used for minutiae extraction (see Fig. 2).

The proposed method is comprised of the following steps and is an improvised version of the method proposed by Jin *et al.* [14].

**STEP 1: Translation and rotation of extracted minutiae based on a eference minutia.** To produce a rotation and translation invariance, we select a reference minutia $m_r$and transform the rest of the minutiae on the basis of it. We have not considered a core point as $m_r$, as a few fingerprints do not have a core point; hence, we cannot rely on it. The selection of a reference point is quite critical for the generation of a binary template from the fingerprint features. Since the features of the same fingerprint vary from scan to scan, sticking to one reference point will mislead the results; thus, we propose using M reference points that will generate M binary templates from the same fingerprint (where M is the total number of minutiae extracted from the fingerprint).


**Figure 2.** Screenshot Of FpMV software used for minutiae extraction (local features)

Let us denote $m_i$ as the $i^th$ minutia extracted from the fingerprint, where $m_i = [x_i, y_i, \theta_i]$, $x_i$ and $y_i$ is the position of $m_i$, and $\theta_i$ is its orientation $[0, 2\pi]$.We choose $m_i$ as reference minutia $m_i^r$ and transform all other minutiae on the basis of it.

Transformed minutia $m_i^t = [(x_i^t, y_i^t, \theta_i^t)]$is obtained from the following equations:

$$\begin{bmatrix} x_i^t \\ y_i^t \\ \theta_i^t \end{bmatrix} = \begin{bmatrix} cos\theta_r & -sin\theta_r & 0 \\ sin\theta_r & cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_i - x_r \\ -(y_i - y_r) \\ \theta_i - \theta_r \end{bmatrix} \tag{1}$$

After all minutiae are transformed according to Equation (1), they are shifted to new coordinates using the following equation:

$$\begin{bmatrix} x_i^n \\ y_i^n \\ \theta_i^n \end{bmatrix} = \begin{bmatrix} x_i^t \\ y_i^t \\ \theta_i^t \end{bmatrix} + \begin{bmatrix} W \\ H \\ 0 \end{bmatrix} \tag{2}$$

$W$ and $H$ in Equation (2) are the width and height of the new coordinate system, respectively, and are set to be twice the size of the fingerprint image; this ensures that $m_r$ is located in the center.

**STEP 2: Orientation angle-based quantization.** We tessellate the new coordinate system into equal-sized squares (see Fig. 3). The number of squares is determined experimentally; if we increase this number, we get better results; however, the computational performance decreases, so we have considered 16 squares in our experiment. The quantization level is determined by orientation angle $\theta$, where $\theta \in [0, ...., 360]$. The levels for quantization are determined experimentally, having a too-large or too-small number of levels can reduce the discriminability of the features.

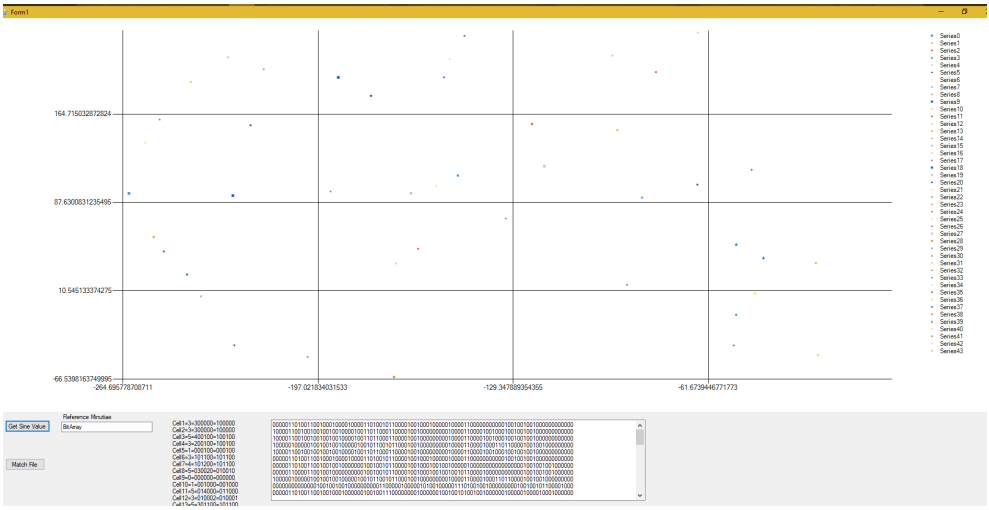



**Figure 3.** Screenshot from implemented solution where new coordinate system is tessellated into equal-sized grids

Mathematically, the histogram generation after quantization is given in Equation (3) as follows:

$$M = \sum_{i=1}^{n} C_i = \sum_{i=1}^{N} h_i \tag{3}$$

where $M$ is the total number of minutiae extracted, $n$ is the total number of squares, $C_i$ is the $i^th$ square, and $N$ is the total number of quantization levels of the orientation angle.

For quantizing the minutiae features, we count the number of minutiae in each square and then place them in predefined equal intervals generated from the orientation angle. Following this procedure, we obtain n different templates from n squares.

These templates are binarized using the rule given below:

$$\sum_{i=1}^{m} = b_i = \begin{bmatrix} 1 & if h_i \geqslant 1, \\ 0 & otherwise \end{bmatrix} \tag{4}$$

where m is the total number of bits in $C_i$. Thus, we get n Binary Templates (BT), which are m bits long, at the end of this step. All n binary templates are concatenated together to obtain the $(n \times m)$ bit $BT_i$ of the $i^th$ reference minutia $m_i^r$. We choose $m_{i+1}$ as new reference minutia$m_{i+1}^r$ and repeat Equations (1)–(4) given in Steps 1 and 2.

We repeat the steps until all of the extracted minutiae are used at least once as reference minutiae to obtain M binary templates from the fingerprint.

**STEP 3: $U_C$ generation using user-specific tokenized permutation.** Storing the M binary templates directly into the database is not safe, as it reveals the information about a user's fingerprint. We pass $M$ binary templates to a function $F$, which transforms the BT and combines them together to generate a $U_C$ that is $(n \times m \times M)$-bits long.

$$U_C = \sum F(BT_i, User\_Token) \tag{5}$$

**STEP 4: Similarity score calculation and matching.** M binary templates $BT^q$ are generated from the query fingerprint using Equations (1)–(5) given above. Each of the templates in $BT^q$ is brute-force matched, with each template stored during enrollment $BT^e$ (see Fig. 4).

Individual binary templates are derived from $U_C$ by applying the reverse function in Equation (5). Ideally, the binary templates generated from the same reference minutiae should be same; however, the reference minutiae in the query and enrolled fingerprints will vary in reality. Thus, we compare each binary template generated during the query with each one stored during enrollment and calculate a similarity score. The value of this score ranges from 0 to 1; a value of 1 implies a perfect match.

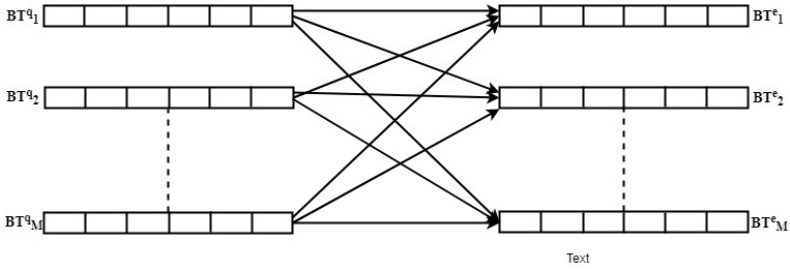



**Figure 4.** Comparison between enrolled binary templates and query binary templates

We store the similarity-matching scores in two-dimensional array $S(i,j)$ as:

$$s(i,j) = \frac{(\sum_{k=1}^{m} BT_{j,k}^{q} + \sum_{k=1}^{m} BT_{i,k}^{e}) \sum_{k=1}^{m} (BT_{j,k}^{q} \wedge BT_{i,k}^{e})}{(sum_{k=1}^{m} BT_{j,k}^{q})^2 + (sum_{k=1}^{m} BT_{i,k}^{e})^2} \tag{6}$$

where $\sum_{k=1}^{m} BT_{i,k}^{e}$ and $\sum_{k=1}^{m} BT_{j,k}^{q}$ count the total number of 1's in the enrolled and query templates, respectively; they are used in the equation to normalize the matching score. $BT_{j,k}^{q} \wedge BT_{i,k}^{e}$ performs a bitwise AND operation and counts the positions with values of 1 in both binary templates.

$S(i,j)$ contains the matching scores; to identify the perfect match, we select the maximum similarity score. We calculate the mean of the maximum distance for each column in $S(i,j)$ and the maximum distance for each row in $S(i,j)$. We choose the maximum between the mean column and mean row as our final score. Once the fingerprints are matched, the CTA derives $d_{CTA}$from $U_C$ and transmits it to the user for private key generation. Figure 5 displays a screenshot of the file that stores the generated similarity scores.

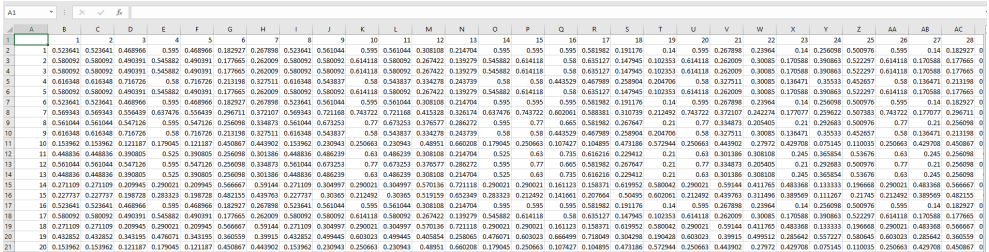


**Figure 5.** Screenshot of file storing similarity scores

### 4.2.5. Generation of public and private keys using unique code $U_C$ and mRSA algorithm

The mRSA algorithm exploits the concept of a threshold cryptosystem. In a threshold cryptosystem, a certain number of parties are involved in cryptography. Within our solution, two parties (one user and another CTA) are involved in private and public key generation.

In the proposed solution, the private key has two parts; one is generated by the user and known only to him/her, and the other is a user-specific partial private key generated by the CTA and transmitted secretly to the user. A two-part private key is required so that a user never digitally signs a document without the CTA knowing this.

In the proposed solution, the public key also has two parts; the first part is generated on-the-fly using the user's public identity, while the second part is generated during the fingerprint enrollment process and stored in the CTA. Both parts are generated in such a manner that, whenever a public key is generated by combining them on-the-fly, it remains relatively prime to $\phi(n)$.

The success of the RSA algorithm critically depends upon the selection of two random prime numbers p and q, as the factorization of large prime numbers takes a long time. Shkodran (2008) proposed a method for generating these numbers through a fingerprint and proved that the generated prime numbers have very good entropy [11].

We use the randomness generated through the fingerprint data as the source for generating the prime numbers. During fingerprint enrollment, two random prime numbers ($p$ and $q$, each of a length of 1024 bits) are derived from $U_C$ using a user-specific secret key. Once $p$ and $q$ are generated from the unique code, the value of $n$ is calculated as $n = p \times q$. The user distributes the value of $n$ publicly to all of the nearest neighbors and the CTA; thus, each node maintains a copy of its $n$ neighbors. Whenever the user changes the value of $n$, the old value of $n$ is overridden; this ensures the security of $n$ against any illegal modifications. Upon generation of the random prime numbers, the public and private keys are generated as given below.

#### Public Key

In the RSA algorithm, the public key is defined as {e,n}, where e is the public component, and e is coprime to $\phi(n)$ and $1 < e < \phi(n)$. In our solution, this e is split into two parts: $e_{user}$ and $e_{CTA}$. The process of public key generation is given below:

a) During fingerprint enrollment, prime random numbers p and q are derived using the user's fingerprint features (see Section 4.3). Using the values of p and q, $\phi(n)$ is derived as $\phi(n) = (p-1)(q-1)$.

b) We derive the user part of the public key from a collision resistant function R (x) as $e_{user}$ = R (user_publicID) in such a way that R (x) maps each user identity to a unique value and for two different public identities (x & y) R(x) ≠R(y).

c) Another part of the user's public key $e_{CTA}$ that is stored by the CTA is derived using the values of $\phi(n)$ and $e_{user}$ in such a manner that gcd $(e_{user} + e_{CTA}, \phi(n)) = 1$. Thus, the value of $e_{CTA}$ is chosen in such a manner that the sum of $e_{CTA}$and $e_{user}$ is relatively prime to $\phi(n)$.

d) Once partial public key component $e_{CTA}$ is derived, it is stored in the CTA for generating the user's public key as and when required.

e) The values of p, q, and $\phi(n)$ are discarded immediately after the user's fingerprint enrollment process.

f) Thus, the public key of a user is $\{(e_{user} + e_{CTA}), n\ \}$.

g) Storing $e_{CTA}$instead of a public key will ensure that simply changing the user id can result in a change of the public keys.

**Private Key**

According to the RSA algorithm, a private key is defined as $\{d, n\}$, where $d$ is the private component, and $d = e^{-1}(mod(\phi(n))$. However, in the mRSA algorithm, private component d is split into two parts:$d_{user}$ and $d_{CTA}$. The relationship between them is given as $d = d_{CTA} + d_{user} mod \phi(n)$. The process of private key generation is given below:

a) The user uses his/her fingerprint to generate $U_C$ using Equations (1)–(5).
b) Using $U_C$, the user calculates the values of p and q and subsequently generates $\phi(n)$.
c) The user requests the CTA to generate and transmit his/her $e_{user}+ e_{CTA}$ and his/her partial private key $d_{CTA}$.
d) The user computes his/her part of private key $d_{user}$ using two equations:

$$d = d_{cta} + d_{user} mod \phi(n) \tag{7}$$

$$d = e^{-1}(mod \phi(n)) \tag{8}$$

e) Subsequently, the private key of the user is derived using $\{\ (d_{user} + d_{CTA}), n\}$.

## 4.3. Process of signing

The following sequence of steps will take place if S wants to digitally sign message m and transmit it to R.

a) S requests the CTA for his/her public key and a partial private key.
b) Simultaneously, S computes the $d_{user}$ part of his/her private key using his/her fingerprint and public component e.
c) The CTA first confirms that S is not revoked and then fetches S's partial private key stored in the database, updates it with the current time key, computes $d_{CTA}$ as $d_{CTA}||t_i$, and transmits it securely (by encrypting it with the session key).
d) S computes his/her private key.
e) S uses the steps mentioned in Section 4.2.2 to digitally sign the data.

## 5. Experiment results

Our experiments focused on validating the following:

a) There is a one-to-one relationship between a fingerprint and the generated unique code; i.e., no two fingerprints should generate the same unique code.
b) The entropy of the unique code is very high.
c) The random numbers generated from the unique code have high entropy.
d) There is a one-to-one relationship between the public and private key pair; i.e., the data encrypted using the private key can only be decrypted by using its paired public key and vice versa.
e) No two fingerprints should generate the same set of public-private key pairs.

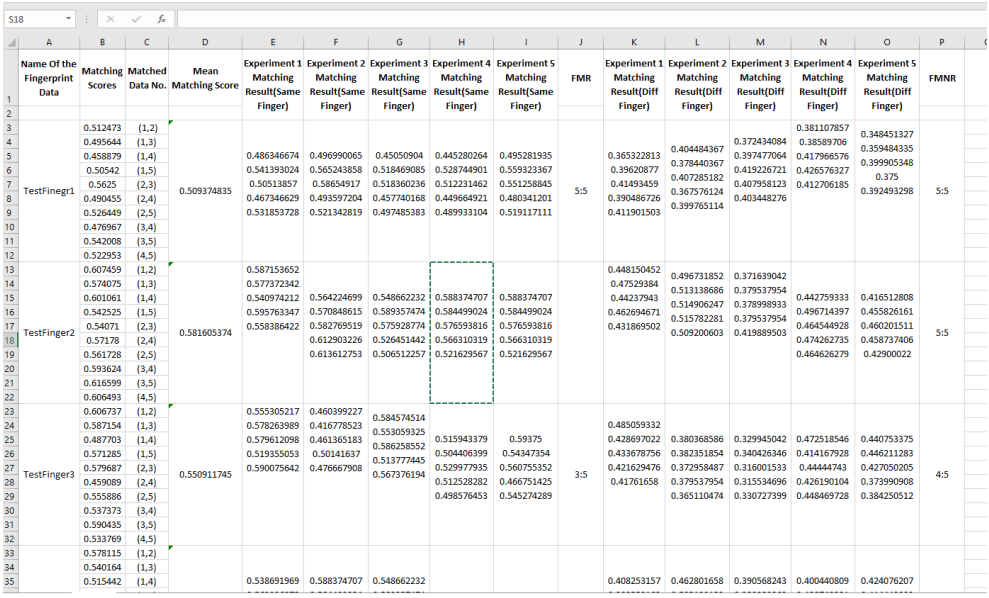
Keeping the above points in mind, we performed various experiments to check the validity of the proposed solution. We implemented the proposed solution in C#. For experimental purposes, we used public fingerprint databases FVC2002 DB1, DB2, DB3, and DB4. Each database contains eight impressions per finger from ten different people; thus, we generated digital signatures for 40 individuals.

The average length of the unique code (in bits) generated for each database is given in Table 1.

**Table 1**
Experimental results

| S. No | Name of the DataBase | Average length of Uc [Bits] | Average metric entropy of Uc | Average time taken for key generation | Average FRR FRR values [%] |
|---|---|---|---|---|---|
| 1 | FVC2000DB1 | 10363 | 0.0001 | 0.002 | FAR:0 FRR:0 |
| 2 | FVC2000DB2 | 14780 | 0.00007 | 0.001 | FAR:0 FRR:0 |
| 3 | FVC2000DB3 | 6259 | 0.00016 | 0.003 | FAR:0 FRR:0 |
| 4 | FVC2000DB4 | 8347 | 0.00012 | 0.002 | FAR:0 FRR:0 |

The length of the unique code varies largely because the number of features collected varies from fingerprint to fingerprint. Similarly, investigating the entropy of the unique code was necessary because we are using it to render randomness in the random numbers generated from it. We used Shannon's Entropy theorem to calculate the entropy of the unique code; the average entropy value for each database is given in Table 1. The entropy of the unique code is very high mainly because of fusing fingerprint features with the 4-bit unique code of the feature lookup table. We also calculated the metric entropy; its average value for each database is also given in Table 1. Metric entropy is a measure of the randomness of information. Figure 6 displays the similarity score calculation for the matching unique codes of query and enrolled fingerprint.



| Name Of the Fingerprint Data | Matching Scores | Matched Data No. | Mean Matching Score | Experiment 1 Matching Result(Same Finger) | Experiment 2 Matching Result(Same Finger) | Experiment 3 Matching Result(Same Finger) | Experiment 4 Matching Result(Same Finger) | Experiment 5 Matching Result(Same Finger) | FMR | Experiment 1 Matching Result(Diff Finger) | Experiment 2 Matching Result(Diff Finger) | Experiment 3 Matching Result(Diff Finger) | Experiment 4 Matching Result(Diff Finger) | Experiment 5 Matching Result(Diff Finger) | FMNR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.512473 | (1,2) | | | | | | | | | | | 0.381107857 | 0.348451327 | |
| | 0.495644 | (1,3) | | | | | | | | | | 0.372434084 | 0.38589706 | 0.359484335 | |
| | 0.458879 | (1,4) | | 0.486346674 | 0.496990065 | 0.45050904 | 0.445280264 | 0.495281935 | | 0.365322813 | 0.404484367 | 0.397477064 | 0.417966576 | 0.399905348 | |
| | 0.50542 | (1,5) | | 0.541393024 | 0.565243858 | 0.518469085 | 0.528744901 | 0.559323367 | | 0.39620877 | 0.378440367 | 0.419226721 | 0.426576327 | 0.375 | |
| TestFinegr1 | 0.5625 | (2,3) | 0.509374835 | 0.50513857 | 0.58654917 | 0.518360236 | 0.512231462 | 0.551258845 | 5:5 | 0.41493459 | 0.407285182 | 0.407958123 | 0.412706185 | 0.392493298 | 5:5 |
| | 0.490455 | (2,4) | | 0.467346629 | 0.493597204 | 0.457740168 | 0.449664921 | 0.480341201 | | 0.390486726 | 0.367576124 | 0.403448276 | | | |
| | 0.526449 | (2,5) | | 0.531853728 | 0.521342819 | 0.497485383 | 0.489933104 | 0.519117111 | | 0.411901503 | 0.399765114 | | | | |
| | 0.476967 | (3,4) | | | | | | | | | | | | | |
| | 0.542008 | (3,5) | | | | | | | | | | | | | |
| | 0.522953 | (4,5) | | | | | | | | | | | | | |
| | 0.607459 | (1,2) | | 0.587153652 | | | | | | 0.448150452 | | | | | |
| | 0.574075 | (1,3) | | 0.577372342 | | | | | | 0.47529384 | 0.496731852 | 0.371639042 | | | |
| | 0.601061 | (1,4) | | 0.540974212 | 0.564224699 | 0.548662232 | 0.588374707 | 0.588374707 | | 0.44237943 | 0.513138686 | 0.379537954 | 0.442759333 | 0.416512808 | |
| | 0.542525 | (1,5) | | 0.595763347 | 0.570848615 | 0.589357474 | 0.584499024 | 0.584499024 | | 0.462694671 | 0.514906247 | 0.378998933 | 0.496714397 | 0.455826161 | |
| TestFinger2 | 0.54071 | (2,3) | 0.581605374 | 0.558386422 | 0.582769519 | 0.575928774 | 0.576593816 | 0.576593816 | | 0.431869502 | 0.515782281 | 0.379537954 | 0.464544928 | 0.460201511 | 5:5 |
| | 0.57178 | (2,4) | | | 0.612903226 | 0.526451442 | 0.566310319 | 0.566310319 | | | 0.509200603 | 0.419889503 | 0.474262735 | 0.458737406 | |
| | 0.561728 | (2,5) | | | 0.613612753 | 0.506512257 | 0.521629567 | 0.521629567 | | | | | 0.464626279 | 0.42900022 | |
| | 0.593624 | (3,4) | | | | | | | | | | | | | |
| | 0.616599 | (3,5) | | | | | | | | | | | | | |
| | 0.606493 | (4,5) | | | | | | | | | | | | | |
| | 0.606737 | (1,2) | | 0.555305217 | 0.460399227 | 0.584574514 | | | | | | | | | |
| | 0.587154 | (1,3) | | 0.578263989 | 0.416778523 | 0.553059325 | | | | 0.485059332 | | | | | |
| | 0.487703 | (1,4) | | 0.579612098 | 0.461365183 | 0.586258552 | 0.515943379 | 0.59375 | | 0.428697022 | 0.380368586 | 0.329945042 | 0.472518546 | 0.440753375 | |
| | 0.571285 | (1,5) | | 0.519355053 | 0.50141637 | 0.513777445 | 0.504406399 | 0.54347354 | | 0.433678756 | 0.382351854 | 0.340426346 | 0.414167928 | 0.446211283 | |
| TestFinger3 | 0.579687 | (2,3) | 0.550911745 | 0.590075642 | 0.476667908 | 0.567376194 | 0.529977935 | 0.560755352 | 3:5 | 0.421629476 | 0.372958487 | 0.316001533 | 0.44444743 | 0.427050205 | 4:5 |
| | 0.459089 | (2,4) | | | | | 0.512528282 | 0.466751425 | | 0.41761658 | 0.379537954 | 0.315534696 | 0.426190104 | 0.373990908 | |
| | 0.555886 | (2,5) | | | | | 0.498576453 | 0.545274289 | | | 0.365110474 | 0.330727399 | 0.448469728 | 0.384250512 | |
| | 0.537373 | (3,4) | | | | | | | | | | | | | |
| | 0.590435 | (3,5) | | | | | | | | | | | | | |
| | 0.533769 | (4,5) | | | | | | | | | | | | | |
| | 0.578115 | (1,2) | | | | | | | | | | | | | |
| | 0.540164 | (1,3) | | | | | | | | | | | | | |
| | 0.515442 | (1,4) | | 0.538691969 | 0.588374707 | 0.548662232 | | | | 0.408253157 | 0.462801658 | 0.390568243 | 0.400440809 | 0.424076207 | |

**Figure 6.** Screenshot of experiment table

## 5.1. Security and privacy analysis

It is a well-known fact that systems where cryptographic keys are directly exposed are more vulnerable to security breaches. Moreover, if the security of the cryptographic keys, is dependent on the software and hardware which are used to store them, any small flaw in them can prove disastrous for the security of the entire system.

Our solution has taken these major facts into consideration by implementing the following:

a) Use of biometric derived key pairs consisting of two-part private and public keys, with one part stored remotely.
b) Private key generation at the user's end, which reduces the chance of stealing the private key during transmission over a network.
c) The entire solution is implemented using asymmetric key cryptography, which does not require additional security mechanisms for key sharing.
d) A user's biometric data is not vulnerable because it is not stored on any device or computer or at the CTA.
e) A user's private and public keys are generated on-the-fly.
f) Various weak links such as DSC, cryptographic token for storing private keys, etc. (which are prone to many attacks, as discussed in Section 1) are not found in our solution.
g) Security of the private key is very high because it is generated on-the-fly.

The following sections provide a case-driven explanation of the security features of our solution.

**Case 1:** Consider a malevolent user (A) who intends to modify the digitally signed communications of another user (B). To achieve this, A has must obtain the digital signature of B. In order to do so, A needs the fingerprint image of B, which itself will be difficult to capture and will be prohibitive enough to prevent A from obtaining B's digital signature. Assuming that A obtains the fingerprint image of B, he/she still needs access to the second-factor authentication, that is B's user token. This two factor authentication will fail the intentions of the malevolent user.

**Case 2:** Assume that a CTA is compromised and A gains access to B's partial private key. In this case, the key is of no use to A since it is just a random number and one part of the private key; he/she cannot generate B's private key using this partial component.

**Case 3:** A performs a sniffing attack during the user authentication process between B and the CTA. Assume A gains knowledge of B's encrypted $C_T$; to decrypt it, A requires $M_{pri}$, which is with the CTA and never shared publicly. Moreover, A cannot replay this information again at some other time to gain unauthenticated access, as an encrypted biometric template is generated along with timestamp value as $M_{pub}$ $(C_T||T_i)$.

The cases discussed above clearly explain that it is very difficult for any malevolent user to obtain the private key of some other user.

## 6. Conclusion and future scope

We have proposed a practical solution for combining biometrics with ID-PKS, which provides complete certificate-less technology for implementing digital signatures. This study is quite unique since it uses biometrics for the on-the-fly generation of public-private key pairs using ID-PKS. The uniqueness of our work lies in on-the-fly generation of public and private keys and, hence, digital signatures. Future work will be aimed at proposing the security architecture of a CTA.

## References

[1] Ang R., Safavi-Naini R., McAven L.: Cancelable key-based fingerprint templates. In: *Australasian Conference on Information Security and Privacy*, pp. 242–252, Springer, 2005.

[2] Baek J., Zheng Y.: Identity-Based Threshold Decryption. In: *International Workshop on Public Key Cryptography*, pp. 262–276, Springer, 2004.

[3] Boldyreva A., Goyal V., Kumar V.: Identity-based encryption with efficient revocation. In: *Proceedings of the 15th ACM Conference on Computer and Communications Security*, pp. 417–426, ACM, 2008.

[4] Boneh D., Xuhua D., Tsudik G.: Identity-based mediated RSA. In: *3rd Workshop on Information Security Application, Jeju Island, Korea*, vol. 12, Citeseer, 2002.

[5] Boneh D., Franklin M.: Identity-Based Encryption From the Weil Pairing. In: *Annual International Cryptology Conference*, pp. 213–229, Springer, 2001.

[6] Chen L., Malone-Lee J.: Improved Identity-Based Signcryption. In: *International Workshop on Public Key Cryptography*, pp. 362–379, Springer, 2005.

[7] Conti V., Vitabile S., Sorbello F.: Fingerprint Traits and RSA Algorithm Fusion Technique. In: *2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems*, pp. 351–356, IEEE, 2012.

[8] Diffie W., Hellman M.: New Directions in Cryptography, *IEEE Transactions on Information Theory*, vol. 22(6), pp. 644–654, 1976.

[9] Ding X., Tsudik G.: Simple Identity-Based Cryptography with Mediated RSA. In: *Cryptographers' Track at the RSA Conference*, pp. 193–210, Springer, 2003.

[10] Elashry I., Mu Y., Susilo W.: Identity-based mediated RSA revisited. In: *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 728–735, IEEE, 2013.

[11] Gerguri S.: *Biometrics used for random number generation*, Ph.D. thesis, Masarykova univerzita, Fakulta informatiky, 2008.

[12] Gutmann P.: Engineering security, 2014. https://www.cs.auckland.ac.nz/˜pgut001/pubs/book.pdf.

[13] Jiang W.Q., Huang Z.Q., Yang Y.X., Jie T., Liang L.: ID-based authentication scheme combined with identity-based encryption with fingerprint hashing, *The Journal of China Universities of Posts and Telecommunications*, vol. 15(4), pp. 75–120, 2008.

[14] Jin Z., Ong T.S., Tee C., Teoh A.B.J.: Generating revocable fingerprint template using polar grid based 3-tuple quantization technique. In: *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1–4, IEEE, 2011.

[15] Jin Z., Teoh A.B.J., Ong T.S., Tee C.: Secure Minutiae-Based Fingerprint Templates Using Random Triangle Hashing. In: *International Visual Informatics Conference*, pp. 521–531, Springer, 2009.

[16] Jin Z., Teoh A.B.J., Ong T.S., Tee C.: Fingerprint template protection with minutiae-based bit-string for security and privacy preserving, *Expert Systems with Applications*, vol. 39(6), pp. 6157–6167, 2012.

[17] Kate A., Goldberg I.: Distributed private-key generators for identity-based cryptography. In: *International Conference on Security and Cryptography for Networks*, pp. 436–453, Springer, 2010.

[18] Li L., Jiang W.Q., Tian J., Yang Y.X., Jiang C.P., Wu Z., Yang X.: A Networking Identity Authentication Scheme Combining Fingerprint Coding and Identity Based Encryption. In: *2007 IEEE Intelligence and Security Informatics*, pp. 129–132, IEEE, 2007.

[19] Quan F., Fei S., Anni C., Feifei Z.: Cracking Cancelable Fingerprint Template of Ratha. In: *2008 International Symposium on Computer Science and Computational Technology*, vol. 2, pp. 572–575, IEEE, 2008.

[20] Ratha N.K., Chikkerur S., Connell J.H., Bolle R.M.: Generating Cancelable Fingerprint Templates, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29(4), pp. 561–572, 2007.

[21] Seo J.H., Emura K.: Revocable Identity-Based Encryption Revisited: Security Model and Construction. In: *International Workshop on Public Key Cryptography*, pp. 216–234, Springer, 2013.

[22] Shamir A.: How to share a secret, *Communications of the ACM*, vol. 22(11), pp. 612–613, 1979.

[23] Shamir A.: Identity-Based Cryptosystems and Signature Schemes. In: *Workshop on the Theory and Application of Cryptographic Techniques*, pp. 47–53, Springer, 1984.

[24] Shin S.W., Lee M.K., Moon D., Moon K.: Dictionary Attack on Functional Transform-Based Cancelable Fingerprint Templates, *ETRI Journal*, vol. 31(5), pp. 628–630, 2009.

[25] Susilo W., Guo F., Mu Y.: Efficient dynamic threshold identity-based encryption with constant-size ciphertext, *Theoretical Computer Science*, vol. 609, pp. 49–59, 2016.

[26] Thian Song O., Teoh Beng Jin A., Connie T.: Personalized biometric key using fingerprint biometrics, *Information Management & Computer Security*, vol. 15(4), pp. 313–328, 2007.

[27] Wong W.J., Wong M.L.D., Kho Y.H., Teoh A.B.J.: Minutiae set to bit-string conversion using multi-scale bag-of-words paradigm. In: *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, IEEE, 2014.

[28] Yuen T.H., Wei V.K.: Fast and Proven Secure Blind Identity-Based Signcryption From Pairings. In: *Cryptographers' Track at the RSA Conference*, pp. 305–322, Springer, 2005.

## Affiliations

**Shuchi Dhir**
Research Scholar, Bharathiar University, Coimbatore, Tamil Nadu, India,
shuchi23_dhir@yahoo.com, ORCID ID: https://orcid.org/0000-0002-8780-011X

**Sumithra Devi K.A.**
Dean(Academics), Dayananda Sagar Academy of Technology and Management, Bangalore, India, sumithraka@gmail.com

**Received**: 12.02.2019
**Revised**: 16.09.2019
**Accepted**: 16.09.2019