2019

Developing an Algorithm for Securing the Biometric Data Template in the Database

Habibu, Taban

International Journal of Advanced Computer Science and Applications

Developing an Algorithm for Securing the Biometric Data Template in the Database

Taban Habibu*[1], Edith Talina Luhanga[2], Anael Elikana Sam[3]
School of Computational and Communication Sciences and Engineering (CoCSE)
The Nelson Mandela African Institution of Science and Technology (NM-AIST), Arusha, Tanzania[1,2,3]

***Abstract*—In the current technology advancement, biometric template provides a dependable solution to the problem of user verification in an identity control system. The template is saved in the database during the enrollment and compared with query information in the verification stage. Serious security and privacy concerns can arise, if raw, unprotected data template is saved in the database. An attacker can hack the template information in the database to gain illicit access. A novel approach of encryption-decryption algorithm utilizing a design pattern of Model View Template (MVT) is developed to secure the biometric data template. The model manages information logically, the view shows the visualization of the data, and the template addresses the data migration into pattern object. The established algorithm is based on the cryptographic module of the Fernet key instance. The Fernet keys are combined to generate a multiFernet key to produce two encrypted files (byte and text file). These files are incorporated with Twilio message and securely preserved in the database. In the event where an attacker tries to access the biometric data template in the database, the system alerts the user and stops the attacker from unauthorized access, and cross-verify the impersonator based on the validation of the ownership. Thus, helps inform the users and the authority of, how secure the individual biometric data template is, and provided a high level of the security pertaining the individual data privacy.**

Keywords*—Biometric template; template-database; multiFernet; encryption-algorithm; decryption-algorithm; Twilio SMS

I. Introduction

The biometric template is a digital sample of a distinct feature obtained from a biometric trait stored in the database, aimed at authenticating and recognizing an individual [1], [2]. The template is built on something you have (fingerprint, facial, iris and voice) as opposite to something you know, such as passwords or Personal Identification Number (PIN). It compares the individual's characteristic extracted to make a match score, the match score is computed, so that the resultant value is in the range (0, 1), where 0 means not matching and 1 means perfect match. If the matching fails, the person can repeat the verification attempt for the second time.

Numerous kinds of algorithmic methods are introduced by different scholars to transform the biometric traits into a template, for instance, the bio-hashing and concealable biometric [3]. The bio-hashing extracts, for example the fingerprint (minutiae point) and convert it into the mathematical file know as a biometric template. The template is then transformed and stored in the database, where matching is performed directly. However, despite the advantages of the biometric data template in verifying and authenticating the individual access, the storage template database can lead to high risk, such as template abuse, modification of the existing template, addition of a new template into the database, and stolen templates in the database [4], [5]. The stored template information can be substituted by an attackers pattern; the impostor can create the physical spoofing from the original pattern to gain unlawful access to legitimate individual's information i.e. Medical records, which may result in false accept or false reject, depending on the motive of the impostor or mount a denial-of-service (DoS) and counterfeit document. The impostor can inject or hijack the characters of the lawful person's template directly into the storage database and replace the original template with the fake template.

Lately, different approaches have been put in place to improve the protection of biometric templates, for instance, the hardware-based and software-based accesses. The hardware-based contain a closed recognition system such as the smart card or handheld device, where the template is securely laid in. The card or device makes up only the template information and the matcher scores (Match-on-card), that aid in mitigating an occurrence on the biometric templates. The software-based solution stores a revised template that do not disclose data about the original biometric traits. It ensures that the biometric data stored in the templates are coded (using a secret key) and practically infeasible to discover the encryption key or regenerate the original fingerprints of a user.

The purpose of this study is to identify the known attacks against the biometric data template in the database from the review of the literature and propose a solution to effectively protect the biometric data template in the database. The suggested solution is established on an encryption-decryption algorithm with a design pattern of model view template (MVT). The algorithm is based on the cryptographic module integrated with Fernet key instance, where two Fernet keys are combined to generate a multiFernet key (K) for the encryption. The Fernet keys guaranteed that, a template data encrypted can't be revealed or read without the secret key, making it unmanageable for an attacker to circumvent or breakthrough into the database server. The cryptographic module included the security tools such as Jinja2, Wtforms, SQLAlchemy [6]. Thus, securely prevented unauthorized access to sensitive template information in the database.

*Corresponding Author

The study, therefore, provided a useful insight on the current biometric technology vulnerability attacks and the mechanism to protect the biometric data template in the database for the everyday government or private registration application system.

It helps inform the users and the authority of, how secure the individual biometric data template in the database is, and provided a high level of the security pertaining the individual data privacy and integrity. Because any attempt in the database can inform the two parties. The key contribution of this article is:

- Providing insight of the possible vulnerability attacks on the biometric data template in the database.
- Implementation of the various sets of security tools and standards that facilitate the safety of the biometric data template in the database.
- The proposed framework architecture model of the encryption- decryption algorithm integrating the key management in securing the biometric data template information in the database.
- The Integrated biometric data template with the Twilio message in the database for security purposes.

Based on the above contributions, this article provides a simple mechanism to reduce the attackers from violating individual biometric data template in the database. The simple mechanism alleviates the database security performance as this paper does not require a complex computer algorithm for the biometric deployment, thus, minimizing overheads and costs.

II. Related Work

This section discusses about some previous works related to the attacks of the biometric data template in the database, and proposed techniques to secure the biometric data template. This is useful to map this paper with the current research trend about the biometric data template safety measure alongside the template database.

A. *Vulnerabilities Attacks of Biometric Template in the Database*

The vulnerability attacks of the biometric template in the database have evoke fears among the users of biometric applications. This is because of a person's privacy, public liberties infringements and trustworthiness [7], [8]. For instance, an impostors intention to corrupt and interchange a genuine template with a fake template, the deliberate alteration of an enrolled template by an impostors pattern or biometric operator, has deterred user's trustworthiness and fear in the technology itself [9].

The most vulnerability attacks of the biometric technology occur in the template database (see Table I). This is because the template database is responsible for comparing the individual's characteristic extracted to make a match score. Thus, a potential target of the attack. The attacker aims to pull out information related to the quality used in the encoding algorithm either directly or indirectly via an externally compromised system (if the server is online). The taken template can be used to attempt a man-in-the-middle attack, such as a replay attack or delete the stored templates to mount a denial-of-service attack [9], [10].

According to Arjunwadkar et al. [11] the impostor can intervene the stored template to compromise the biometric traits, breaks the security loopholes to replace, change or edit the existing template of the legitimate person [12]. Gobi and Kannan argued that the templates are tempered by adding the new user template or fake template to the database, altering the current template in the storage and deleting the existing genuine one [13]. Bindha et al. [14] indicated that the template data can be substituted by an attacker's template to attain illegitimate access to a system. Study of Mwema et al and Raju et al. [15], [16] observed that spoofing at the template database is the most determined epidemics experienced in biometric technology.

Manvjeet et al. [17] presented that the cross-matching is the possible exploitation of biometric templates, because the template data are used for extra aim than the proposed aim. Thus deprived of the individual's permission, for instance, fingerprint pattern taken from a bank's record can be used to search for an unlawful fingerprint database for criminal investigation [18]. In Uganda, the 15,277 million voters' fingerprint intended for National Identity Cards (NIC) was extracted from NIRA database, this provoked a serious concern of the citizens [19]. Yet, when a person's template is stolen, it is stolen forever and not easy to revoke, since every individual has (one face, ten fingers, two eyes etc.). In traditional password-based authentication systems, once a password is compromised, the new one is reissued [18]. If the biometric data is compromised, it can result into four vulnerabilities:

- The template being substituted or replaced by an attacker's template to gain illegal access.
- Physical spoofing being produced or generated from the template to gain illegal access to the organization as well as other schemes that employ similar biometric feature.
- The template being taken or reiterated to the matcher to acquire illegal access.
- If not correctly insured, the template can be used for cross-matching across different database records to covertly track an individual deprived of their consent.

Li and Kot [20] suggested a privacy safety (hiding scheme) for the weak-thinned fingerprint template. In their study, the user's identity is concealed in the thinned fingerprint image during enrollment and a fingerprint template stored in an online database for authentication. They claimed that applying such a system, it is impracticable for the attacker to expose the individuality of the user from stealing the template collected from a compromised online template database [21].

TABLE. I. POSSIBLE ATTACKS ON THE BIOMETRIC DATA TEMPLATE IN THE DATABASET

Targeted position	Possible attacks	Countermeasures	References
Data collection	Spoofing	Liveness detection, Challenge/response	[49],[50]
Raw data extraction and transmission	Eavesdropping	Data Transmitted over an encrypted path/secure channel Challenge/response	[51]–[54]
	Replay attacks	Communally authenticate/use symmetric key or asymmetric key; Digitally sign data; Utilize Timestamp (TTL) tag	[53]–[55]
	Man-in-the-middle	Bind biometric to PKI certificate; Data Transmitted over an encrypted path	[56]
Data processing, transmission	Brute force	Timeout/lock out policies	[51],[53],[54]
	Eavesdropping	Data Transmitted over an encrypted path/secure	[51]–[54]
	Replay attacks	Communally authenticate/use symmetric key or asymmetric key; Digitally sign data; Utilize Timestamp (TTL) tag	[53]–[55]
	Man-in-the-middle	Bind biometric to PKI certificate; Data Transmitted over an encrypted path	[56]
Template retrieval	Brute force	Timeout/lock out policies	[51],[53],[54]
	Eavesdropping	Data Transmitted over an encrypted path/secure	[51]–[54]
	Replay attacks	Communally authenticate/use symmetric key or asymmetric key; Digitally sign data; Utilize Timestamp (TTL) tag	[53]–[55]
	Man-in-the-middle	Bind biometric to PKI certificate; Data Transmitted over an encrypted path	[56]
Storage	Database compromise (reading template, replacing template(s))	Hardened server DB access controls; Sign and store encrypted templates; Store template on smart cards or other device.	[51],[53]
Matching scores transmission	Hill climbs	Trusted sensor (Mutual authentication); Secure channel	[53],[54],[57]
	Manipulation of match score	Secure channel; Communal authentication between matcher and decision components	[53]
	Component replacement (“yes machine”)	Significant components	[53]
Decision	Hill climbs	Communal Authentication; Secure channel	[53],[54],[57]
Communication to application	Eavesdropping	Data Transmitted over an encrypted path/secure	[51]–[54]
	Manipulation of match decision	Data Transmitted over an encrypted path	[53]

Liu et al. [22] considered a cryptographic technique of secret transmission encryption to encode a plaintext for multi recipients and hide the recipients’ identities. They suggested the cancellable biometrics and biometric cryptosystem to protect the template. In cancellable biometrics, instead of applying the original biometric data, a partial form is kept in the database, in this manner, an intruder cannot gain access to the unique pattern from the database. In biometric cryptosystem, the biometric data is encrypted before storing it in the database; this makes it rather hard for the attacker to decode the data and stealing the genuine template from the database.

Elkamchouchi et al. [23] suggested method of cryptography which uses the image as an open key and arbitrary integers as a private key to compute the image [24]. Jain et al. [25], suggested a steganography to hide biometric data (fingerprint minutiae) in multitude images (face), whose function is to transmit the data. The carrier image can be an artificial fingerprint image, a face image or any arbitrary image. The suggestion is useful in a distributed system, where the raw biometric data are transmitted over an insecure communication channel and prevented a skimmer from reading sensitive information. They also discussed a novel application wherein, the facial feature of a user (eigencoefficient) is embedded in a host fingerprint image of the user to increase the security, then stored them on a smart card. The fingerprint of the person is compared to the fingerprint on the smart card. The false information hidden in the fingerprint is recovered and used as an additional basis of validity either mechanically or by a humanoid in a controlled biometric application.

Emmanuel et al. [26] proposed the concealable biometric method to protect the template database. The cancellable biometrics involved repeatable distortion of an original biometric pattern intentionally based on a chosen non-invertible transform, to enroll and authentication the system from the stored template. This reduced the template compromise and resolved the legitimate substitution of a privacy-related issue for matching against transformed vector and prevented the system from storing the new biometric traits of the user.

Pratiba and Shobha [27] proposed a watermarking technique. The watermarking information on the biometric template data in the database allowed the legitimacy of the biometric contents to be verified, when retrieved for matching. The pixel value is used to hide the watermark information [28].

In case an impostor tried to replace or forge the secured biometric template, the system notifies the database manager signifying something is wrong with the biometric template in the database [29]. Although, the watermark information prevented invader from altering the template, there is a small alteration in the genuine template as well as insufficient changes in the pixel. Hence, resulted in insecure template database protection.

Nandakumar and Jain [30] proposed the fuzzy vault pattern using fingerprint and Iris. The study revealed that, multi biometric vault on thumbprints and irises achieved a Greater Accept Rate (GAR) of 98.2% at FAR of 0.01%. The matching GAR value of the person's irises and thumbprint vaults are 88% and 78.8% respectively. The safety of the system is at 41 bits and that of the thumbprint and irises offered 49 bits [31], [32]. In conclusion, the biometric vault provided improved recognition presentation and highest safety of the biometric data template.

Ashish et al. [33] suggested the usage of string re-arrangement to ease the protection of the template database. The biometric data is encrypted and discarded after constructing the comfortable template. During the verification, the stored data is deciphered using the secret key and matched against the captured query. The obstacle to the encryption-based policy is the unprotected key control that exposed the decryption secret to the machine for each authentication. The advantage is the matching process hired for maintaining the matching accuracy [34].

Rathgeb et al. [35] proposed an alignment free iris key-binding scheme with concealable transforms. They adopted Indexing-First-One (IFO) hashing to achieve non-invertible and cancelable transformation for biometrics and the cryptographic key-binding. The key-binding is separated into four levels cryptographic key generation, genuine and synthetic permutation, key-binding, hashed code generation and computer memory. The findings showed that the highest GAR of 96.37% at zero FAR with storage, record equal to 1.90 kB was achieved. They further proposed useful key retrieval metric KRR for implementing the security analysis. The proposed embraces the flexibility while maintaining significant accuracy, public presentation and protection layer. The quality preservation of the accuracy performance at higher security levels is achieved and the method requires no re-enrollment and storage [36].

Yang and Martiri [37] proposed honey template-based template protection scheme to detect the biometric template database leakage. In the protection scheme, machine learning based classification algorithms is utilized to produce the sugar and honey templates applied in face [38].

Hine et al. [39] introduced a zero-leakage biometric cryptosystem to measure the performance reachable when fusing the data from the four available fingers of each field at feature and score levels, utilizing the inverse of both L1 and L2 distance metrics as matching scores. The four classifiers give an equal error rate (EER) of 0:67%. The proposed system guarantees no information leakage and it allows achieving a trade-off between privacy and credit rates.

Dwivedi et al. [40] proposed a secrecy-protective cancelable irises template encoding and a new cancelable iris template on arbitrarily look-up table drawing. The method uses a number vector created from a changing-invariant character vector using 1-D Log Gabor filter usable to the iris picture. The experimentation is carried away on several iris databases to support the efficiency of the proposed attack. Equal Error Rate (EER) of 0.37%, 0.43% and 0.79% for CASIA-V 1.0, CASIA-V3-Interval and ICE 2005 iris databases are achieved [41].

Prasad et al. [42] applied a novel approach based on modulo operation. The method utilizes consistent bit vector generated from pre-aligned IrisCodes. These IrisCodes are created by applying 1-D Log Gabor filter on the iris images using different iris datasets. Equal Error Rate of 0.54% and 0.86% for CASIA-V 1.0 and CASIA-V3-Interval iris datasets are achieved. The method satisfies revocability, unlikability and irreversibility criteria and it is difficult to regenerate original IrisCode [43].

Lai et al. [44] proposed a novel cancellable iris system, coined as IFO hashing inspired from the Min-hashing. Two new mechanisms, namely Hadamard product code and modulo thresholding function are inserted to further enhance the system. The IFO hashing scheme endures numerous security and privacy attacks such as a single hash attack, multi-hash attack, attack via record multiplicity and pre-image attack. Thus, enjoys fast similarity search property inherited from Min-hashing and can potentially be drawn out to identification task and other binary biometric features [45].

Zhao et al. [46] proposed an iris template protection method based on local ranking. It is established from the resolutions that the method is able to give 0.57% EER value for CASIA-V 1.0 and 0.79% EER value for CASIA-V3-Interval and also cover all the security and revocable issues. Furthermore, Zhou and Ren [47] proposed a user-centric biometric validation system (PassBio) that allows end-users to encode personal patterns with light-weighted encryption scheme. The findings prove that no critical information of the templates can be revealed under both passive and dynamic approaches. It guarantees that only the comparison result is discovered and no key information about x and y can be memorized. It can be widely used in many interesting applications such as searching over encrypted data while assuring information protection and seclusion.

Mai et al. [48] presented an acceleration of the guessing entropy, which reflects the expected number of guessing trials in attacking the binary template in the biometric application. The results revealed that, rushing has more than 6x, 20x, and 200x speed upward lacking down the approximation accuracy in dissimilar system settings.

In conclusion, no single biometric system is enough to protect the biometric template database to its fullest. Thus, the study, suggested the encryption-decryption algorithm based on the cryptographic module incorporating the Fernet key instance. The cryptographic module integrated the biometric traits (fingerprint, and face image) with persons biodata, to produce an encrypted byte and a text file, these files are securely kept in the database incorporated with Twilio message. The approach is more secure and harder for an impostor to guess the key mixtures and suitable for use in many biometric software applications.

III. Materials and Method

This section explained the security tools and approach deployed in safeguarding the biometric data template in the database. The security tools installed are Jinja2, Wtforms, SQLAlchemy, Cryptography, Twilio programmable Short Message Service (SMS) and the encryption-decryption algorithm approach [58], [59].

The python flash is used as the development platform, because it contains inbuilt development server that provided simplicity, flexibility and fine-grained control, as considerably as a faster debugger to the network application. It allowed one to add security mechanisms to the application such as session-based certification, function management, password hashing, basic hypertext transfer protocol (HTTP), and token-based certification.

IV. Proposed Framework

In this section, the proposed framework is presented and classified into three. The first is the Model View Template (MVT) incorporating Helper Utilities File system (HUF), useful in controlling the models function for securing the biometric data template. The second is the encryption algorithm for encoding the biometric features and biodata. The third is the decryption algorithm for decoding the encrypted ciphertext.

A. The MVT-HUF Framework Architecture

The Model View Template (MVT), a modification of the Model View Controller (MVC) for securing the template database is used as the design implementation pattern. The model is an object that controls, data logically, the view signifies the imagining of the information a model contains, execute the business lucidity and relate with a model to transmit data and renders a template. The template controls the information movement into model entity and keeps the view and the model differently.

Fig. 1(a) and 1(b) shows the proposed framework architecture of the MVT-HUF System and the functional intent of the ePassport. The MVT is slightly modified to cater for security issues. For instance, the Wtforms was introduced for CSRF prevention, and helper, introduced to handle heavy processing between model and View. The biometric passport was used as an object acted as a model integrating fingerprint and facial image at the feature level with person's biodata, to derive a two byte and text file as a template. This template is securely kept in the database incorporated with the Twilio SMS. The encryption- decryption algorithm based cryptographic key management is applied to ensure the security of the template database. The algorithm accounted for the acceptable disparities in the biometric input. An impostor whose sample biometric is different from the enrolled biometric features, cannot break or recreate the private key. The biometric features encrypted stored a hashed value of the key as a template byte and text file and releases it only if the hashed value obtained for verification is the same. The hashed version can be functioned as a cryptographic key. With this invention, an attacker cannot obtain the original key outside the encoding scheme.

In Fig. 1(b), it's realizing that the encrypted fingerprint and face image is kept in the file system (including the ciphertext of second key). Meanwhile, user biodata is stored in relational tables. It is a significant to recognize that the computer memory in a file system is implemented using random integers (IDs) that hold less meaning to the user at presentation layers as the presentation level IDs are computed from helpers other than coming from the database.

B. The Encryption Algorithm

In encryption process, the users enter the credentials. The username and password are compared with a copy that is kept in the database. If the details do not match, he/she is requested to re-enter either a new username or password, else if it matches an authentication code (AC) is produced and sent to the user via SMS. Upon the user receiving the authentication code, he/she is requested to enter the received authentication code. The authentication code is matched with a copy that is kept in the database. If the authentication code does not match with the copy stored in the database, the user is guided back to login interface. If it matches, the database generates two Fernet keys (K_1 and K_2).

The Fernet keys are secret key of asymmetric implementation based on cryptography that supports key rotation in the form of byte key. The two keys are combined to further generate a multiFernet key (K) for encryption. The MultiFernet key (K) is integrated with biometric features (Face, Fingerprint) and biodata passing through the encryption algorithm to produce the biometric template as byte file and a text file. The two files are securely kept in the database. Fig. 2 summarized the proposed implementation of the encryption algorithm.

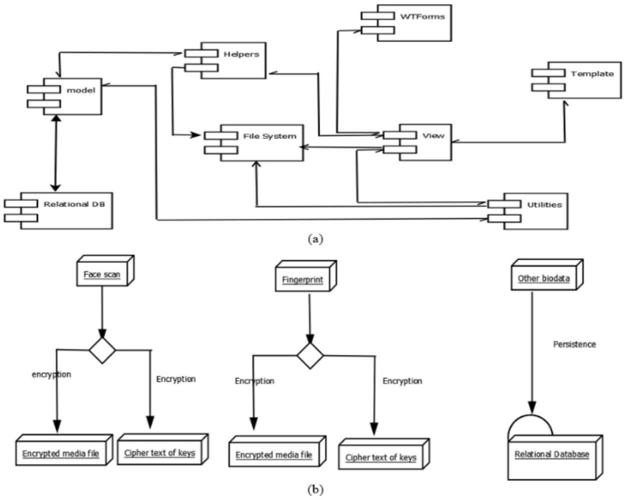


Fig. 1. (a) Overview of the Proposed Model of MVT-HUF System. (b) The Function Design of the E-Passport.

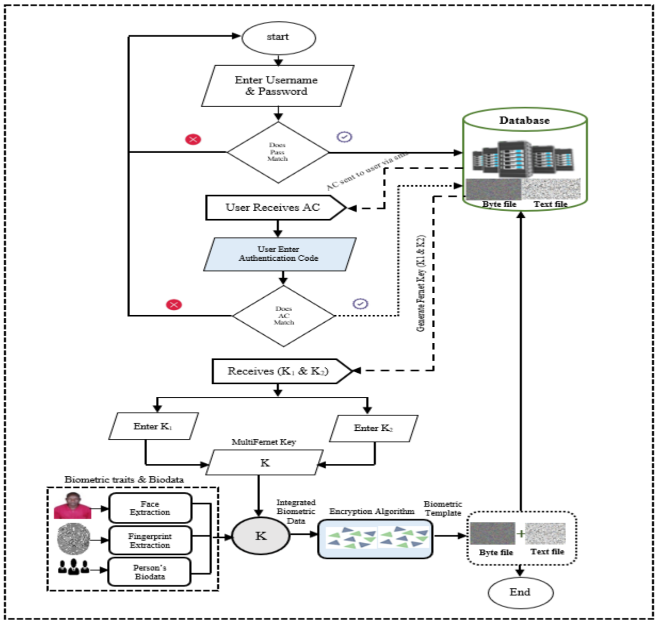


Fig. 2. Proposed Framework of the Encryption Algorithm.

C. The Decryption Algorithm

In decryption process, the administrator is requested to enter the credentials. The username and password are compared with a copy that is kept in the database. If the information does not tally, the administrator is requested to re-enter either a new username or password, else if it matches an authentication code is produced and sent to the administrator via SMS. Upon receiving the authentication code, the administrator is asked to enter the received authentication code. The authentication code that the administrator entered is matched with a copy that is kept in the database. If the authentication code does not match with the copy stored in the database, the administrator is led back to login interface. If it matches, the database generates two Fernet keys (K_1 and K_2). The two keys are combined to further generate a multiFernet key (K) for decryption. The MultiFernet key (K) is integrated with biometric template (byte file and a text file) passing through the decryption algorithm to produce the plain text. Fig. 3 summarized the proposed implementation of the decryption algorithm.

In case an attacker tries to access the biometric data template in the database, the system blocks the attacker from unauthorized access. Because the system cross-verify the user based on two dissimilar kinds of identification such as the knowledge base (something the user knows) and the possession factor (something the user owns) such as authentication code (AC). This is really important in securing up the biometric template information in the database. Even if the perpetrators are able to discover a user's password, they nevertheless lack the second kind of identification required to login to the application. Fig. 4 presented the suggested security measures in the encryption algorithm.

D. Database models

The SQLite3 is used as the proposed model for the development process, and the database switched to PostgreSQL, because of the object relational mapper (SQLAlchemy) for security purpose. The PostgreSQL has multi-value fields (aka arrays, aka nested tables) which can reduce the need for joins. Dramatically increase the performance of storing and retrieving the multi-dimensional data structures, and making it possible to write stored procedures in other programming languages such as C, Perl, Python and JavaScript V8 engine [60].

E. Comparison of the Current System

The proposed framework and the encryption-decryption algorithm based on the cryptographic module in the multiFernet key instance performed better. User data template in the database is securely protected. The imposter cannot easily break into the system or read or re-generate a key. Therefore, using the proposed approach, prevented data being compromised by an impostor, hence provided higher security of individual privacy data.

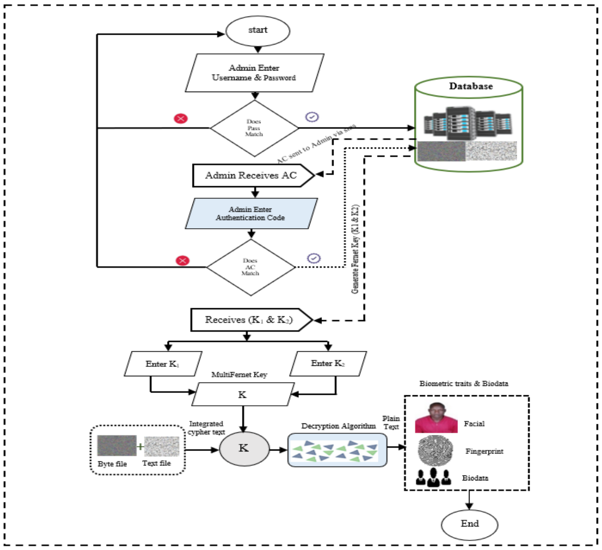


Fig. 3. Proposed Framework of the Decryption Algorithm.

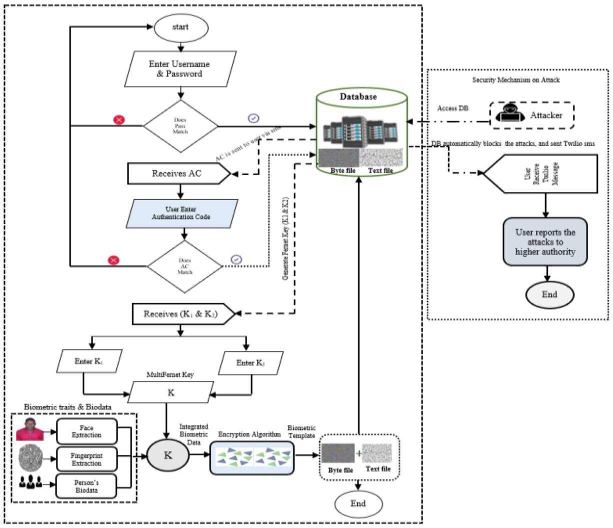

start
Enter Username & Password
Does Pass Match
Database
AC is sent to user via sms
Receives AC
User Enter Authentication Code
Does AC Match
Generate Fernet Key (K1 & K2)
Byte file
Text file
Security Mechanism on Attack
Access DB
Attacker
DB automatically blocks the attacks, and sent Twilio sms
User Receive Twilio Message
User reports the attacks to higher authority
End
Receives (K1 & K2)
Enter K1
Enter K2
MultiFernet Key
K
Biometric traits & Biodata
Face Extraction
Fingerprint Extraction
Person's Biodata
K
Integrated Biometric Data
Encryption Algorithm
Biometric Template
Byte file
Text file
End

Fig. 4. Proposed Framework of the Security Mechanism in the Encryption Algorithm.

V. The Implementation Design

This section discusses the security tools deployed for the implementation of the proposed encryption-decryption algorithm.

A. Jinja2 Implementation

Jinja2 is used as template engine containing variables and tags to direct the logic of the template. It provided a protected basis for mechanization of sampling the application and helped avoid cross-site scripting (XSS) occurrence through its powerful automatic Hypertext Markup Language (HTML) escaping system. The cross-site scripting (XSS) enabled the invaders to insert client-side scripts into web application sighted by different users. Fig. 5 presented the implementation code for Jinja2. Note that {%… %} is used to represent statements and {{…}} used to print the data.

The primary function of a template engine is to sort out the logic from the horizon. Thus, the template engines considered obeys the following principles:

1) A restricted set of command structures: such as Loop i.e. for, loop or while; Condition i.e. If, if else and else; Filter3 i.e. {{Variable filter}}; Setting of variables and Printing of a variable.

2) A mechanism to include other templates, to use inheritance of templates or to use macros, written entirely in the restricted instructions from above.

3) No way to write pure code in the language that is used for the backing (i.e. PHP, Python or Java) within the template.

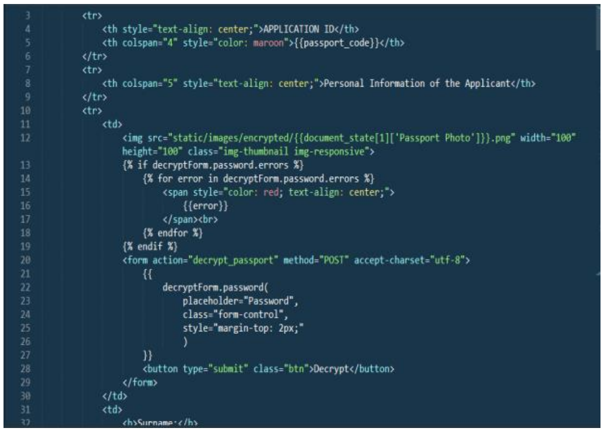
B. Wtforms

Wtforms generates applicant's passport forms, rather than coding Hypertext Markup Language (HTML). This helped protect the system from Cross Site Reference Forgery (CSRF) module. The CSRF implementation is pivoted around the exceptional token, put in a varied field on the form named csrf_token, rendered in the template, and passed from the browser back to the interface. The cryptography hashed function against the data enabled the attacker not to form the template database. Note that CSRF is a character of malicious exploitation of a website where unauthorized commands are transported from a user that the web application trusts. Through the Wtforms, the cross-site request forgery attack is prevented.

Notice that, when carrying out the web page form using Wtforms and python, the contours are represented as class representatives. This allowed clearer backend validations before data proceeds to the database, meaning in case the front-end is tempered with, the Wtforms validations can be capable to manage the authentication. Fig. 6 presented the implementation coding for the applicant detail on the template side using Wtforms.

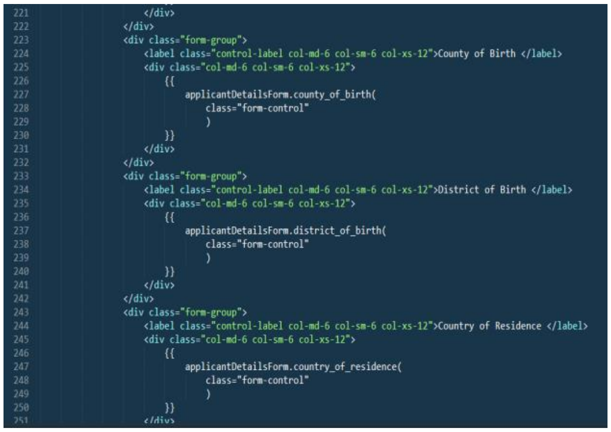
C. SQLAlchemy

The object relational mapper (SQLAlchemy) is applied to create database models instead of database drivers directly. The security advantage is to prevent the SQL- injection attack, zero-day attack for various databases plus other database exploited through the application, because the coercion is first practiced on every database transaction. With SQLAlchemy the user doesn't write SQL statements, instead make the class representative and the SQLAlchemy figures out the optimum and attack free SQL statement equivalence. The SQL Expressions can be applied independently of the ORM. When using the ORM, the SQL Expression language remains part of the public face API as it is used within object-relational configurations and queries. Notice that SQL injection is the location of malicious code in SQL statements via the web page input [60]. Fig. 7 presented a model of SQLAlchemy dependency layers. The SQLAlchemy helped in mapping this class to the corresponding table.



```
<tr>
    <th style="text-align: center;">APPLICATION ID</th>
    <th colspan="4" style="color: maroon">{{passport_code}}</th>
</tr>
<tr>
    <th colspan="5" style="text-align: center;">Personal Information of the Applicant</th>
</tr>
<tr>
    <td>
        <img src="static/images/encrypted/{{document_state[1]['Passport Photo']}}.png" width="100"
        height="100" class="img-thumbnail img-responsive">
        {% if decryptForm.password.errors %}
            {% for error in decryptForm.password.errors %}
                <span style="color: red; text-align: center;">
                    {{error}}
                </span><br>
            {% endfor %}
        {% endif %}
        <form action="decrypt_passport" method="POST" accept-charset="utf-8">
            {{
                decryptForm.password(
                    placeholder="Password",
                    class="form-control",
                    style="margin-top: 2px;"
                    )
            }}
            <button type="submit" class="btn">Decrypt</button>
        </form>
    </td>
    <td>
        <b>Surname:</b>
```

Fig. 5. Implementation of 'The Confirms Details' Template using Jinja2.



```
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-md-6 col-sm-6 col-xs-12">County of Birth </label>
        <div class="col-md-6 col-sm-6 col-xs-12">
            {{
                applicantDetailsForm.county_of_birth(
                    class="form-control"
                    )
            }}
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-md-6 col-sm-6 col-xs-12">District of Birth </label>
        <div class="col-md-6 col-sm-6 col-xs-12">
            {{
                applicantDetailsForm.district_of_birth(
                    class="form-control"
                    )
            }}
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-md-6 col-sm-6 col-xs-12">Country of Residence </label>
        <div class="col-md-6 col-sm-6 col-xs-12">
            {{
                applicantDetailsForm.country_of_residence(
                    class="form-control"
                    )
            }}
        </div>
```

Fig. 6. Implementation Code in the Template Side using Wtforms.

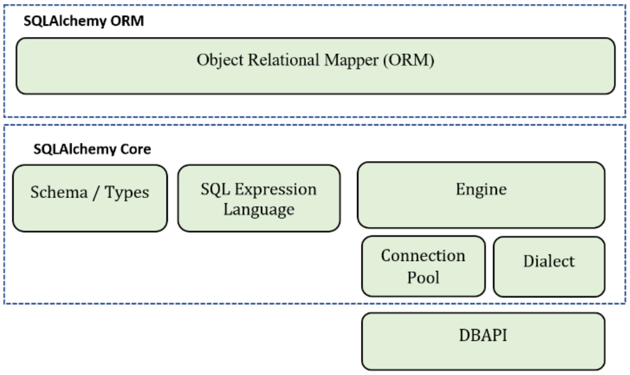


Fig. 7. SQLAlchemy Dependencies Layers.

D. Cryptography

In cryptography, plain text is encoded into coded text with the help of encryption algorithm, the coded text is decoded to plain text with the help of decryption algorithm. In both operations, the cryptographic key played a significant part. It limited the admission of the coded data so that the possessor of the key could decrypt cipher text properly. In this technique, it was expected that only the sincere user knows the decryption key. Therefore, cryptography, as a powerful tool in biometric technology, depends on the secrecy of cryptographic key and the key needed an efficient key management technique. The key management technique included the process of key generation, key modification and key sharing [61].

The cryptographic module encrypted the fingerprint and the face image based on a Fernet instance key. The key is categorized into two smaller keys: a 128-bit AES encryption key and a 128-bit SHA256 HMAC signing key. These keys are retained in a central source that keystone passes in a library to handle the encryption and decoding process. The Fernet key guaranteed that the message encrypted cannot be read missing the key. It involved the application of symmetric (secret key) authentication, that support Fernet key alternation via multiFernet key (class cryptography. fernet. Fernet(key)).

The multiFernet performed the cipher code using first key in the list of Fernet instance. Then decrypt each key in turn. The key alternation replaced the old key to add a new key Infront of the list. The new message was encrypted to discard the old key. The Token rotation was offered by meth (multiFernet. rotate) as a primary key to prevent mutilation and decreases the trouble of attack. Hence, preserved the timestamp originally saved with the token. The successful rotated token was returned while unsuccessful rotated token returned an exception error such as (cryptography. Fernet. InvalidToken). Token rotation as offered by MultiFernet, is the best practice and the manner of cryptographic hygiene, designed to fix damage in case of an undetected event and to increase the difficulty of attacks [58].

E. The Twilio Programmable SMS

The Twilio SMS is utilized to signal users and authority over SMS messages about unlawful entry to the system account and the database template. In case they are not the one accessing the record, then be able to identify. Twilio is a cloud communiqué system that offers SMS services to its users. The Twilio source fetches the logs for any outbound messages from the narrative, like the Sent folder in the email client. Utilize this data to update the customer relationship management (CRM) whenever a client gets a text message from the application. Or to see the recipients of an SMS message before it sends, to ensure they don't receive it before. It brings in any inbound messages to any of the Twilio numbers. This is like the email inbox.

If you apply a single Twilio number to commit many types of messages, it can route the responses to the necessary people, founded along the sentiment score of the consistency of the message, who mailed it. Or what time it arrived in. It also sent lots of SMS messages while Parabola flow runs. This permitted one to send out custom or generic SMS messages to a list of recipients at scheduled times. Use the destination to send the weekly performance, remind occurrence of an event and threats coming up in the system, or constantly ping your details to remind you of any approval privileges to allow admittance to your certification. Once an approval of applicants is performed, the SMS confirmation message for biometrics scan is automatically sent to the applicant for the achievement. Fig. 8(a) and 8(b) illustrates the Twilio SMS sent to applicant for verification and biometric scan process.

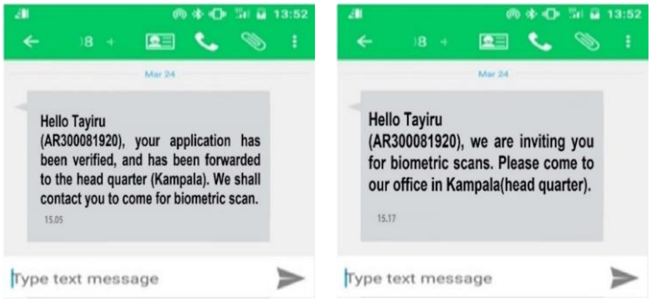
With the Twilio SMS, the system is non-traceable by the impostor, it also comprises safety mechanisms to circumvent hacking and despicable attacks to compromise the user's information and individual rights to secrecy during the personal information transmission.

F. The Cryptographic Fernet Keys

The cryptographic Fernet key is built on three criteria. The advanced encryption standard (AES) in coded block chaining (CBC) mode with a 128-bit key for encryption using the PKCS7 padding. The Hash-based Message Authentication Code (HMAC) uses the Secure Hash Algorithm (SHA) 256 for authentication. The Initialization vector to generate a random secret number using os. urandom () [58]. The AES provides advantages such as high-level security and implementation ability that does not expose unauthenticated bytes. It encrypts the data that easily fits in the memory.

It uses the parameter such as secret keys (byte) either in 128,192 or 256 bits long and the CBC mode using the padding for block ciphers. The parameters rest on the IV and secret key. The IV is a unique public information, randomly unpredictable at the encryption time to prevent data repetition, making it hard for a hack to get patterns to crack into the template database. It ensured that, information is not leaked by the cipher text itself and prevented identical plaintexts from producing identical cipher text. The secret key protected the encrypted information.

The HMAC is used to calculate the communication, validation using cryptographic hash functions, paired off with a private key. For example, class cryptography. hazmat. primitives.hmac.HMAC (key, algorithm, backend). This hashed algorithm randomly generated the bytes equal in duration to the digest_size of the secret hashed function kept.



Hello Tayiru
(AR300081920), your application has been verified, and has been forwarded to the head quarter (Kampala). We shall contact you to come for biometric scan.
15.05

Hello Tayiru
(AR300081920), we are inviting you for biometric scans. Please come to our office in Kampala(head quarter).
15.17

(a) (b)

Fig. 8. (a) Twilio Verification Message (b) Twilio Message for the Biometric.

G. Key Management for the Encryption Algorithm

The encryption algorithm used is based on the combination of two Fernet keys, i.e. the first key (K_1) and the second key (K_2). User inputs original biometric features Image (I) and K_2 to generate K_1-encoded (byte key). The K_1-encoded is further applied to generate K_1 decoded (string key) using K_2.

The K_1 encoded is combined with K_2 to generate multiFernet keys (K). The K is used in encrypting the Image (I) to realize the encrypted image file (I_0). In order to guarantee the safekeeping of the biometric data in the database, the encrypted image (I_0) is further re-encrypted with multiFernet key (K) to produce an encrypted byte and a text file (K_{10}). The two files are securely stored in the database as a template.

The encoding is the operation of transforming information (plaintext) into something that appears to be random and meaningless (ciphertext) so that it is unclear to anyone but to the intended receiver. Fig. 9 summarized the stepwise process for the key management of the encryption algorithm.

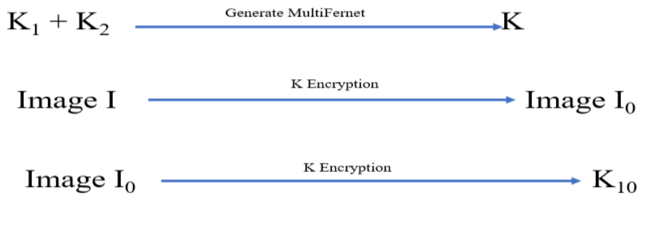
Presented below are the key management using encryption algorithm with multiFernet key.

H. Key Management for the Decryption Algorithm

In order to acquire the original image (I) from the encrypted byte and text file (K_{10}), the decryption process is simply the reversed engineering of the encryption step. K_{10} is decrypted using the multiFernet key (K) to realize the encrypted image (I_0). The K is generated from a combination of Fernet keys (K_1 & K_2). The multiFernet key (K) is further employed to decode the encrypted Image (I_0) to produce the original image (I). If the formatted token is successfully decoded, the original plain text (I) is received as the result, otherwise an exception error can be produced.

The decryption is the operation of changing encrypted information (secret code text) back to readable plaintext so that it is understandable again. Fig. 10 summarized the stepwise process for the key management of the decoding algorithm.

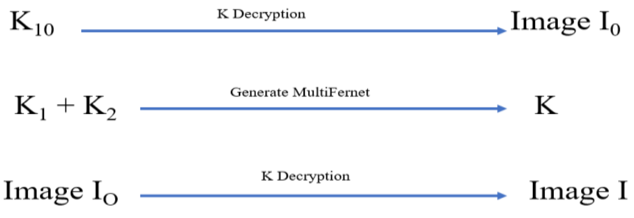
Given is the key management of the decryption algorithm with multiFernet key.



Such that

- i. **K_1:** First Fernet Key
- ii. **K_2:** Second Fernet Key
- iii. **K:** multiFernet key
- iv. **Image I:** Original face scan image or fingerprint Image
- v. **Image I_0:** Encrypted biometric image
- vi. **K_{10}:** Encrypted byte file and text file

Fig. 9. Key Management of the Encryption Algorithm.



Such that

- i. **K_{10}:** Encrypted byte file and text file
- ii. **K_1:** First Fernet Key
- iii. **K_2:** Second Fernet Key
- iv. **K:** multiFernet key
- v. **Image I_0:** Encrypted biometric image
- vi. **Image I:** Original face image or fingerprint Image

Fig. 10. Key Management of the Decryption Algorithm.

VI. Discussion of Results

The Twilio SMS is implemented for the validation over unlawful access to the system account and the template database. In case an attacker attempts to access the biometric data template in the database, the system blocks the attacker from unauthorized access. Because the system cross-verify the user based on something the user owns such as authentication code (AC). The Twilio fetches the login for any outbound messages from the report as well as any inbound messages to any of the Twilio numbers.

The Ubuntu 18.04 is used as a client server to provide an interface and allowed users to call for the services. Users are situated at workstations or on personal computers, while servers are located in the regional centers of the immigration offices, controlled in the powerful machines at the headquarter for the request and the response. The users and the server each have distinct jobs to perform. For example, in the biometric passport data processing unit, a user machine runs an application program, while the server mainframe runs another program that handles the database. Fig. 11 summarized the client-server architecture of the application system.

The results are tested with user's biometric traits, containing 50 fingerprints and 50 face image templates incorporated with the personal biodata. The image size of fingerprint template extracted is 256X256 and resolution set to 72 dpi. The face image is uniformly illuminated and captured from the right mind with no rotation or tilting, no apparitions, with a plain background colour. The end product of the image is set to 600dpi with 120 pixels as the standard, recommended by ISO/IEC [62], [63]. The encrypted byte and text files are incorporated with Twilio programmable SMS. The Twilio SMS message is auto-generated directly from the database, to alert users in case an attacker tries to access the database. The text message is one of the security mechanisms successfully implemented. It helps inform the users and the authority of, how secure is the individual biometric data template in the database. How the users are indirectly involved in awarding or refusing access to the exercise of their biometric template information. Because any attempt in the database can inform the two parties.

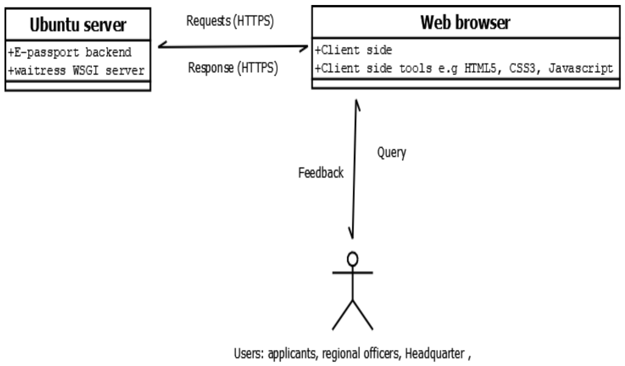


Fig. 11. Client-Server Architecture.

VII. Conclusions

In this article, various attacks against a biometric template database are highlighted and the techniques used to secure the contents of the biometric data template in the database are discussed. Since securing the biometric information in the database is one of the studies focuses, encryption-decryption algorithm approach is suggested based on the cryptographic module with the security tools such as Jinja2, Wtforms, SQLAlchemy and Twilio SMS.

The encryption-decryption algorithm is developed to encrypt the biometric data in the database. Two encrypted byte file and text file are generated, incorporated with Twilio message, securely stored in the database server. The database has security features that warn against any impostor attack alongside with persons information. The system can block the attacker from unauthorized access. This is really important in the security of privacy of the biometric template information in the database. Even if the perpetrators are able to discover a user's password, they nevertheless lack the second kind of identification required to login to the application.

Living by the proposed framework in Fig. 2 and 4, the same conclusion can be drawn near the model. It can be well-known that no biometric system is optimal. The decision as to which biometric is to be used can be prepared from the foundation of the kind of application and the degree of protection required. Therefore, the policymaker needs to design security-policies based on lawful ideologies and should include the input of various players to defend against the vulnerability of users' information. Because individual's data exist in every place online. Further experimental research on ways, to facilitate the database server with additional safety coding like hash functions and two factor authentications to prohibit data requests transmission and setting out principles and alertness session to all mediators at the start of the engagement before being granted access to personal information.

Acknowledgments

The authors would like to acknowledge Nelson Mandela African Institution of Science and Technology, Arusha-Tanzania, and Muni University Arua-Uganda, for the financial support and research resources to carry out this research.

References

[1] MacHado S, D'Silva P, D'Mello S, Solaskar S, Chaudhari P. Securing ATM Pins and Passwords Using Fingerprint Based Fuzzy Vault System. Proc - 2018 4th Int Conf Comput Commun Control Autom ICCUBEA 2018. 2018;1–6.

[2] MM A. Biometric: fingerprints protection. Biometrics Biostat Int J. 2018;7(2):156–61.

[3] Mm A, Gr S. Biometric Template Protection. J Biostat Biometric Appl. 2017;1(2):1–8.

[4] Riaz N, Riaz A, Khan SA. Biometric template security: an overview. Sens Rev. 2018;38(1):120–7.

[5] Ghouzali S, Lafkih M, Abdul W, Mikram M, El Haziti M, Aboutajdine D. Trace attack against biometric mobile applications. Mob Inf Syst. 2016;2016.

[6] Dey S, Ghosh R. A review of cryptographic properties of S-boxes with Generation and Analysis of crypto secure S-boxes . PeerJ-Preprints. 2018.

[7] Group UKBW. Biometric security concerns. Technical Report, CESG, September 2003, http://www. cesg. gov. uk/site/ast …; 2003.

[8] Habibu T, Luhanga ET, Sam AE. Evaluation of Users ' Knowledge and Concerns of Biometric Passport Systems. Data [Internet]. 2019;4(April):1–17. Available from: www.mdpi.com/journal/data.

[9] Ambalakat P. Security of biometric authentication systems. In: 21st Computer Science Seminar. Citeseer; 2005. p. 1.

[10] Habibu T, Sam AE. Assessment of vulnerabilities of the biometric template protection mechanism. Int J Adv Technol Eng Explor. 2018;5(45):243–54.

[11] Arjunwadkar M, Kulkarni R V, Shahu C. Biometric Device Assistant Tool: Intelligent Agent for Intrusion Detection at Biometric Device using JESS. Int J Comput Sci Issues. 2012;9(6):366–70.

[12] Poongodi P, Betty P. A Study on Biometric Template Protection Techniques. Int J Eng Trends Technol. 2014;7(4).

[13] Xi K, Hu J. Bio-Cryptography. Handb Inf Commun Secur [Internet]. 2010;129–57. Available from: http://www.springerlink.com/index/T28595L71117713Q.pdf

[14] Brindha VE, Natarajan AM. Multi-modal biometric template security: Fingerprint and palmprint based fuzzy vault. J Biometrics Biostat. 2012;3(3):100–50.

[15] Mwema J, Kimwele M, Kimani S. A simple review of biometric template protection schemes used in preventing adversary attacks on biometric fingerprint templates. Int J Comput Trends Technol. 2015;20(1):12–8.

[16] Raju S V, Vidyasree P, Madhavi G. Enhancing Security Of Stored Biometric Template in Cloud Compuuting Using FEC. Int J Adv Comput Eng Netw. 2014;2(2):35–9.

[17] Manvjeet Kaur, Sanjeev sofat DS. Template and Database Security in Biometrics Systems : A Challenging Task. Int J Comput Appl. 2010;4(5):2–6.

[18] Prabhakar S, Pankanti S, Jain AK. Biometric recognition: security and privacy concerns. IEEE Secur Priv Mag [Internet]. 2003;1(2):33–42. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1193209.

[19] Rindai Vava Chipfunde. Biometric voter registration : Lessons from Ugandan polls. 2016.

[20] Li S, Kot AC. Privacy protection of fingerprint database. IEEE Signal Process Lett. 2011;18(2):115–8.

[21] Yang W, Wang S, Hu J, Zheng G. SS symmetry Security and Accuracy of Fingerprint-Based Biometrics : A Review. Symmetry (Basel). 2019.

[22] Liu L, Li Y, Cao Z, Chen Z. One Private Broadcast Encryption Scheme Revisited. Int J Electron Inf Eng. 2017;7(2):88–95.

[23] Elkamchouchi HM. A New Image Encryption Algorithm Combining the Meaning of Location with Output Feedback Mode. In 2018.

[24] Nita SL, Mihailescu MI, Pau VC. Security and Cryptographic Challenges for Authentication Based on Biometrics Data. Cryptogr Artic. 2018.

[25] Jain AK, Ross A, Uludag U. Biometric template security: Challenges and solutions. In: Signal Processing Conference, 2005 13th European. Citeseer; 2005. p. 1–4.

[26] Emmanuel E, Edebatu D, Catherine N, Ngozi A. Vulnerability of Biometric Authentication System. Int J Innov Res Sci Eng Technol. 2016;2742–9.

[27] Pratiba D, Shobha G. A Novel approach for securing biometric template. Int J Adv Res Comput Sci Softw Eng. 2013;3(6):974–9.

[28] Malhotra S, Kant C. A Novel approach for securing biometric template. Int J Adv Res Comput Sci Softw Eng. 2013;3(5).

[29] Anitha P, Rao KN, Rajasekhar V, Krishna CH. Security for Biometrics Protection between Watermarking and Visual Cryptography. SSRG Int J Electron Commun Eng. 2017;(March):64–71.

[30] Nandakumar K, Jain AK. Biometric template protection: Bridging the performance gap between theory and practice. IEEE Signal Process Mag. 2015;32(5):88–100.

[31] Gomez-Barrero M, Maiorana E, Galbally J, Campisi P, Fierrez J. Multi-biometric template protection based on homomorphic encryption. Pattern Recognit. 2017;67:149–63.

[32] Khan SH, Akbar MA, Shahzad F, Farooq M, Khan Z. Secure biometric template generation for multi-factor authentication. Pattern Recognit. 2015;48(2):458–72.

[33] Ashish MM, Sinha GR. Biometric Template Protection. J Biostat Biometric App. 2016;1(2):202.

[34] Simoens K, Bringer J, Security HC-… F and, 2012 U. A framework for analyzing template security and privacy in biometric authentication systems. IEEE Trans Inf Forensics Secur [Internet]. 2012 [cited 2017 Dec 12]; Available from: http://ieeexplore.ieee.org/abstract/document/6129504/.

[35] Rathgeb C, Gomez-Barrero M, Busch C, Galbally J, Fierrez J. Towards cancelable multi-biometrics based on bloom filters: a case study on feature level fusion of face and iris. In: Biometrics and Forensics (IWBF), 2015 International Workshop on. IEEE; 2015. p. 1–6.

[36] Gomez-Barrero M, Maiorana E, Galbally J, Campisi P, Fierrez J. Multi-biometric template protection based on Homomorphic Encryption. Pattern Recognit [Internet]. 2017;67:149–63. Available from: http://dx.doi.org/10.1016/j.patcog.2017.01.024.

[37] Yang B, Martiri E. Using honey templates to augment hash based biometric template protection. Proc - Int Comput Softw Appl Conf. 2015;3:312–6.

[38] Martiri E, Gomez-Barrero M, Yang B, Busch C. Biometric template protection based on Bloom filters and honey templates. IET Biometrics. 2016;6(1):19–26.

[39] Hine GE, Maiorana E, Campisi P. A Zero-Leakage Fuzzy Embedder from the Theoretical Formulation to Real Data. IEEE Trans Inf Forensics Secur. 2017;12(7):1724–34.

[40] Dwivedi R, Dey S, Singh R, Prasad A. A privacy-preserving cancelable iris template generation scheme using decimal encoding and look-up table mapping. Comput Secur [Internet]. 2017;65:373–86. Available from: http://dx.doi.org/10.1016/j.cose.2016.10.004.

[41] Kumar MM, Prasad MVNK, Raju USN. Iris Template Protection using Discrete Logarithm. 2018;43–9.

[42] Prasad MVNK, Jyothi A, Lasya K. Cancelable iris template generation using modulo operation. Proc - 13th Int Conf Signal-Image Technol Internet-Based Syst SITIS 2017. 2018;2018-Janua:210–7.

[43] Rathgeb C, Breitinger F, Busch C. Alignment-free cancelable iris biometric templates based on adaptive bloom filters. Proc - 2013 Int Conf Biometrics, ICB 2013. 2013.

[44] Lai YL, Jin Z, Jin Teoh AB, Goi BM, Yap WS, Chai TY, et al. Cancellable iris template generation based on Indexing-First-One hashing. Pattern Recognit [Internet]. 2017;64(August 2016):105–17. Available from: http://dx.doi.org/10.1016/j.patcog.2016.10.035.

[45] Li C, Hu J, Pieprzyk J, Susilo W. A New Biocryptosystem-Oriented Security Analysis Framework and Implementation of Multibiometric Cryptosystems Based on Decision Level Fusion. IEEE Trans Inf Forensics Secur. 2015;10(6):1193–206.

[46] Zhao D, Fang S, Xiang J, Tian J, Xiong S. Iris Template Protection Based on Local Ranking. Secur Commun Networks. 2018;2018.

[47] Zhou K, Ren J. PassBio: Privacy-preserving user-centric biometric authentication. IEEE Trans Inf Forensics Secur. 2018;13(12):3050–63.

[48] Mai G, Lim MH, Yuen PC. On the guessability of binary biometric templates: A practical guessing entropy based approach. IEEE Int Jt Conf Biometrics, IJCB 2017. 2018;2018-Janua:367–74.

[49] Marasco E, Ross A. A survey on antispoofing schemes for fingerprint recognition systems. ACM Comput Surv. 2015;47(2):28.

[50] Hadid A, Evans N, Marcel S, Fierrez J. Biometrics systems under spoofing attack: an evaluation methodology and lessons learned. IEEE Signal Process Mag. 2015;32(5):20–30.

[51] Nandakumar K, Jain AK. Multibiometric template security using fuzzy vault. In: Biometrics: Theory, Applications and Systems, 2008 BTAS 2008 2nd IEEE International Conference on. IEEE; 2008. p. 1–6.

[52] Ross AA, Shah J, Jain AK. Toward reconstructing fingerprints from minutiae points. In: Biometric Technology for Human Identification II. International Society for Optics and Photonics; 2005. p. 68–80.

[53] Dürmuth M, Oswald D, Pastewka N. Side-Channel Attacks on Fingerprint Matching Algorithms. In: Proceedings of the 6th International Workshop on Trustworthy Embedded Devices. ACM; 2016. p. 3–13.

[54] Boult TE, Scheirer WJ, Woodworth R. Revocable fingerprint biotokens: Accuracy and security analysis. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2007. p. 1–8.

[55] Shelton J, Bryant K, Abrams S, Small L, Adams J, Leflore D, et al. Genetic & evolutionary biometric security: Disposable feature extractors for mitigating biometric replay attacks. Procedia Comput Sci. 2012;8:351–60.

[56] Lancelot Miltgen C, Popovič A, Oliveira T. Determinants of end-user acceptance of biometrics: Integrating the 'big 3' of technology acceptance with privacy context. Decis Support Syst. 2013;56(1):103–14.

[57] Miltgen CL, Popovič A, Oliveira T. Determinants of end-user acceptance of biometrics: Integrating the "Big 3" of technology acceptance with privacy context. Decis Support Syst. 2013;56:103–14.

[58] Contributors I. Cryptography Documentation. 2019.

[59] Chaudhary S. An Approach to Secure Database Templates in Multimodal Biometric Systems. IJCSC. 2013;4(2):268–73.

[60] Bayer M. SQLAlchemy Documentation [Internet]. SQLAlchemy Documentation Release 0.7.10. 2016. Available from: papers3://publication/uuid/5E97B936-E845-4995-92F5-EB7F0C39672B

[61] Stallings W. Cryptography and network security: principles and practice. Pearson Upper Saddle River; 2017.

[62] Griffin P, Ph D. Understanding The Face Image Format Standards. 2005.

[63] Sang J, Lei Z, Li SZ. Advances in Biometrics. Springer-Verlag [Internet]. 2009;5558(May 2014). Available from: http://link.springer.com/10.1007/978-3-642-01793-3.