# An Approach to Improve the Match-on-Card Fingerprint Authentication System Security

Kishor Krishnan Nair
Council for Scientific and Industrial Research (CSIR)
Pretoria
South Africa
knair@csir.co.za

Albert Helberg
North-West University
Potchefstroom Campus
South Africa

Johannes van der Merwe
Council for Scientific and Industrial Research (CSIR)
Pretoria
South Africa

*Abstract*— **The combination of smart cards and biometrics results in a strong 2-factor authentication. It facilitates a trusted credential for authenticating an individual's identity through one-to-one biometric verification using the smart card token. Fingerprint Authentication Systems (FASs) embedded in smart cards are gaining massive acceptance as it offers a superior level of security when compared to the conventional Match-on-Device (MOD) and Match-on-Server (MOS) technologies. The fundamental difference between the conventional FAS authentication process and FAS using smart card process is all about the authentication location. FAS authentication using smart card token is possible through four authentication approaches. They are the Template-on-Card (TOC), Match-on-Card (MOC), Work-Sharing On-Card (WSOC), and System-on-Card (SOC). Out of these four approaches, the SOC is considered as the most secure and expensive, whereas the TOC is considered as the least secure and least expensive. The MOC scheme offers a higher level of security than the TOC scheme at an affordable cost when comparing to SOC and is thus the most practical solution available today. Although this is the case, the MOC scheme is also susceptible to the inherent security vulnerabilities associated with biometric modalities in general. The front runner of the biometric vulnerabilities being the compromise of the original finger templates. This paper focuses on this critical vulnerability aspect and conceptualizes a novel Match-on-Card Fingerprint Authentication System. The proposed system does not need to store or transfer the original fingerprint template to/from the smart card and therefore facilitates an enhanced security than the conventional MOC approach.**

***Keywords—ecc; FAS; MOC; smart card***

## I. Introduction

In the present electronically dependent and interconnected world, the need for superior security schemes persists more than ever. The conventional token based user authentication approaches using Personal Identification Numbers (PIN) and passwords are proved to be inadequate. Biometric technology uses a person's unique and permanent physical or behavioural characteristics to authenticate the identity of a person. Biometrics authentication systems based on the usage of fingerprints are gaining massive acceptance across the world and has become a synonym for biometric authentication [1]. In an FAS, the user first establishes an identity, in order to get added to the system. To do so, demographic information is provided and fingerprints are scanned to create a fingerprint template. The templates are then typically stored in a centralized server along with the user's demographics [2]. This process is called the biometrics enrollment. When the user is challenged to prove his/her identity, the fingerprints are scanned and sent to the server. The server validates the presented fingerprint template against the enrolled template to determine whether it is a match or a non-match. This type of matching is called Match-on-Server (MOS). The matching can be also performed on the device in which the user presents the biometric template and this scheme is called Match-on-Device (MOD) [1].

A challenge with MOS/MOD is its security. These authentication schemes are highly vulnerable to the compromise of the original fingerprint template and are also exposed to the interception along the communication channel. The original fingerprint templates are perpetually attached to the user and therefore the templates cannot be cancelled or revoked like a PIN or a password, once it is compromised. Since biometric data is permanent and each person has limited amount of choice (a user has a maximum of 10 fingers), having the biometric database information stolen is a serious implication to the actual owner and is the biggest risk of a biometric system in general. One of the countermeasures is to embed the fingerprint template into a smart card. A smart card is a plastic card with an option to include a microprocessor chip and internal memory chips. It meets the requirements for the necessary processing capability along with the inbuilt security features, data storage, and convenience. Hence, the combination of biometrics and smart card is expected to predominantly offer an enhanced security than MOS/MOD schemes [3].

Even though, fingerprint based Match-on-Card (MOC) approach can provide a superior secure authentication system when compared to MOS/MOD, the truth is that it is also vulnerable to the inherent security vulnerabilities associated to biometrics [3]. This paper emphases on this aspect and proposes countermeasures. The current work is structured as follows: Section II discusses the Conventional Match-on-Card

Fingerprint Authentication System (CMOCFAS). Section III conceptualizes the core idea of the Proposed Match-on-Card Fingerprint Authentication System (PMOCFAS). Section IV evaluates the PMOCFAS in conjunction with the CMOCFAS. Section V looks into the scope for future work and draws conclusions.

## II. CMOCFAS

Fig. 1 illustrates a generic CMOCFAS. In this, the biometric template presentation and feature extraction is performed at the biometric terminal. During the enrolment process, the original biometric template captured at the terminal is stored inside the smart card. At the matching stage, the terminal will sent the presented template to the smart card for comparison and the final matching decision is calculated inside the smart card. The dotted line illustrated in Fig. 1 is the application or applet firewall which restricts the access to the matching application or applet to enquire the status of fingerprint authentication [4]. With MOC, the template is locked in the smart card (SC) and never leaves it [5]. To perform the authentication process, the user presents the card to the Card Acceptance Device (CAD) or the card reader. On the other end of this communications channel is a fingerprint scanner. Typically, this is an integrated fingerprint reader or peripherally attached fingerprint scanner. When the user presents the fingerprint to the scanner, it produces an image of the finger. The reader then extracts information from that fingerprint image in the form of minutiae points and is sent to the card for matching. The SC executes a fingerprint matching algorithm and produces a score revealing how similar the fingerprint sent to the SC is with the one stored in it. The SC then renders a decision as to whether or not it is belonging to the same user.

Even though, CMOCFAS offers a superior level of security when comparing to MOS/MOD, a potential problem with this method is that, in the case of offline matching, there is no central authority to dictate permissions. CMOCFAS would verify "the person is who they say they are", but without synchronizing with a central authority. Furthermore, the communication channel between the CAD and SC is vulnerable and the sensitive biometric information can be possibly compromised in the transit. In reality, the CMOCFAS is also susceptible to the key security vulnerabilities associated with biometric modalities in general, which are identified as follows.

1. The possibility of data being intercepted along the communication channel between the CAD and the SC leading to replay attacks.
2. The likelihood of compromising the original finger template locked in the SC.
3. The compromised template can be used to launch cross platform attacks.

The current study focuses on mitigating the above vulnerabilities in a MOC environment. For this purpose, a Match-on-Card Fingerprint Authentication System framework is conceptualized. The proposed frame work from here on will be known as PMOCFAS (which stands for Proposed Match-on-Card Fingerprint Authentication System).

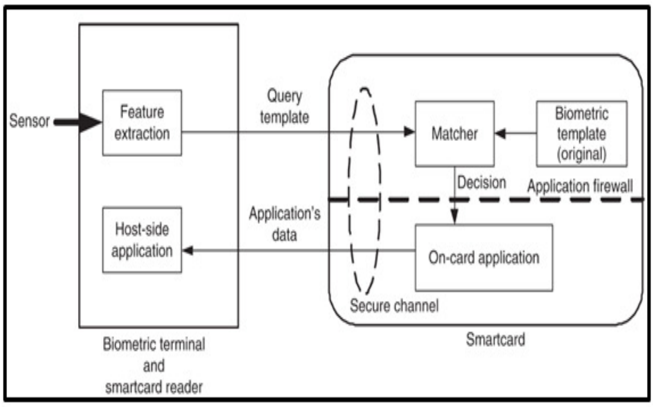


Fig. 1. Architecture of the CMOCFAS [4]

The security goals set for the PMOCFAS are formulated in what follows.

- Preserve the privacy and security of template data.
- Be resilient to the compromise of the template itself.
- Support revocation of the template.
- Be resilient to replay attacks.

## III. Conceptualization of the PMOCFAS

The PMOCFAS is conceptualized through sections A, B, C and D as follows.

### A. Prosposed security model

Ratha *et al.* introduced the concept of cancelable or replaceable biometrics and this approach is also known as feature transformation [6]. It consists of a deliberate and iterative distortion of a template based on the selected transformation function. The template is transformed in the same mode at each presentation, for every enrollment and authentication. In this scheme, if the transformed template is compromised, then the transformation function can simply be modified to create a new cancelable template. Biohashing is a derivative of the cancelable biometrics and is analogous to password "salting" in conventional crypto-systems. In this approach, before hashing, the *password P* of the user is concatenated with a pseudorandom *string or number S* and the resulting *hash H(P+S)* is stored in the database [7], [8]. The biohashing is based on the same principle and in this scheme, the template transformation is based on a function defined by a key or a password [9].

Biohashing schemes have been recommended for iris and palmprint modalities [10], [11]. Another example of biohashing is the cancelable face filter approach proposed for face recognition [12]. In the biohashing approach, it is a prerequisite that the key or the password needs to be securely stored and recalled by the user, and presented during authentication [12], [13]. The key is used to increase the security of the template and thus makes it difficult for the adversary to guess the template. Biohashing could protect the biometric system against any biometric fabrication and therefore the imposters can be easily identified. It also facilitates the revocation of the template in case if it is compromised. Furthermore, biohashing offers significant functional advantages such as zero error rate point and clean separation of the genuine and imposter populations, thereby allowing *False Acceptance Rate(FAR)*[1] elimination without suffering from increased occurrence of *False Rejection Rate (FRR)*[2] [9]. Biohashing approach achieved a zero *Equal Error Rate (EER)*[3] rate based on the hypothesis that the hash key will not be compromised [12], [13]. The hash key can be compromised and hence it is not a viable solution in addressing the vulnerabilities as it will lead to more security vulnerabilities. It is a very challenging task to develop a FAS that is secure, resilient and revocable; while maintaining a decent performance. To solve the above problem and to address the vulnerabilities, this paper proposes a security model that is based on the principles of Biohashing and One-Time Password (OTP) scheme for the MOC authentication framework. The OTP scheme is a recognized authentication scheme in payment transactions in that the password generated can be used one time and hence a strong security can be achieved. Thus, the proposal is to derive an unpredictable, one-time finger template by inheriting the principles of the Biohashing and an OTP scheme. The *One-Time Template* (OTT) will be generated based on the *Biohashing Key* derived for the current authentication session. During the authentication phase, the OTT derived for authenticating the user at the CAD is matched against the OTT generated at the SC and the decision is taken accordingly. The proposed security model is as illustrated in Fig. 2.

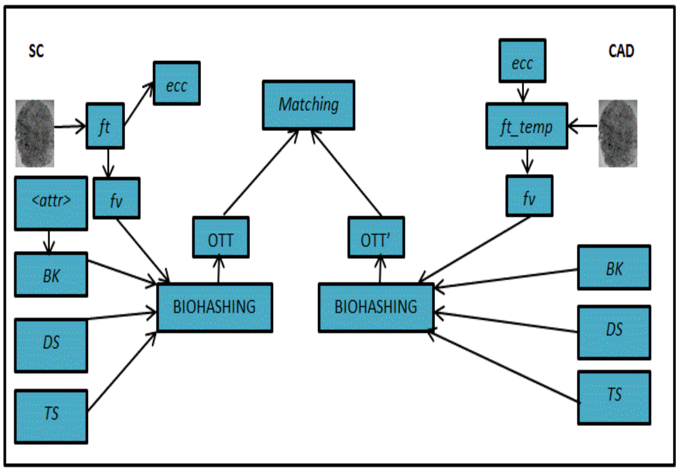


Fig. 2. Proposed security model

During the enrolment phase, the attribute list (<attr>) of the user is captured at the SC. The <attr> corresponds to the user specific information, such as email id, password, and identification number. In the current security model, the attribute list is limited to 8 as it is ample to generate the expected level of security, and at the same time it saves the enrollment time needed per user. The *<attr>* is then transformed to a *Biometric Key (BK)* and is stored in the SC. The *finger template (ft)* is subsequently captured from the user and the *error correction code (ecc)* is derived from the *ft*. The *ft* is then converted to *feature vector (fv)* and is then transformed to a master One Time Template known as *OTTm* by the BIOHASHING module and is stored in the SC. At this stage, the enrolment process is complete. The original finger template is dis-regarded and is never stored in the SC.

During the authentication phase, the SC is inserted into the CAD to start the authentication process and it receives the *Date Stamp (DS)* and *Time Stamp (TS)* from the CAD. The SC in turn returns the *ecc*, *BK* and *Authentication Session Counter (ASC)* to the CAD. For the first authentication attempt, the ASC will be 1 and will incremented by a factor of 1 on subsequent authentication attempts. The CAD at this stage prompts for the finger template for authentication, reads it and applies the *ecc* received from the SC and generates the *fv*. The *fv, BK, DS, TS,* and *ASC* is inputted to the BIOHASHING module at the CAD and the *OTT'* for the current authentication session will be generated. During the matching phase which happens at the SC, the OTT generated at the CAD is send to the SC. The SC compares the OTT generated by it with the OTT' received from the CAD. If the OTTs match, then the authentication is successful else the authentication process fails. All the communication between the CAD and SC happens in the encrypted domain.

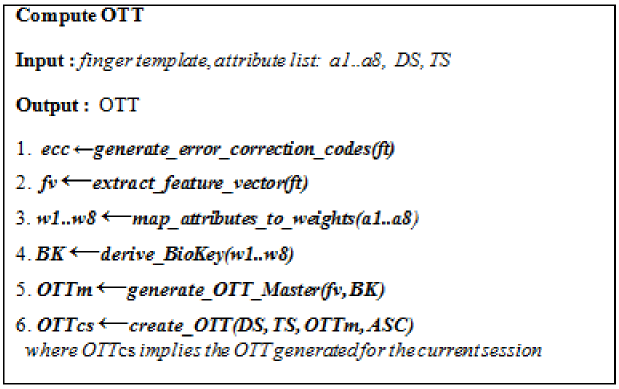
### *B. OTT generation at the SC*

The OTT generation at the SC is carried out based on the core OTT generation algorithm depicted in Fig. 3. In this algorithm, in step1, the finger template *ft* is inputted to the *generate_error_correction_codes* routine to generate the *ecc* corresponding to the captured finger template**.** The *ecc* will contain valid information such as the core minutiae point, the neighbors corresponding to the core and the alignment details of the captured finger template. The *ft* is also inputted to the *extract_feature_vector* routine to generate the corresponding *feature vector*; represented as *fv*. Step 3 encodes the attributes to weights using the *map_attributes_to_weights* routine. The attributes *a1..a8* are mapped to the corresponding weights *w1..w8*. The next step in the algorithm is to generate the *BK* and is done using the *derive_BioKey* routine. In this, the weights from *w1 to w8* are processed to derive the *BK.* Step 5 generates the *OTTm* and is produced using the *generate_OTT_Master* routine. The *OTTm* is derived using a transformation function that takes as input *BK* and the *fv* obtained in steps 2 and 3. The *DS, TS, OTTm,* and *ASC* are now passed as parameters to the *create_OTT_Set* routine. In this routine, the *OTT* necessary for performing the current authentication will be generated at the SC.

[1] FAR is the rate at which a biometric system authenticates an unauthorized person or an imposter.

[2] FRR is the rate at which a biometric system rejects an authorized person (i.e. the individual is not authenticated).

[3] The accuracy of a biometric system is usually measured by its EER and is the rate at which the FAR and FRR are equal.

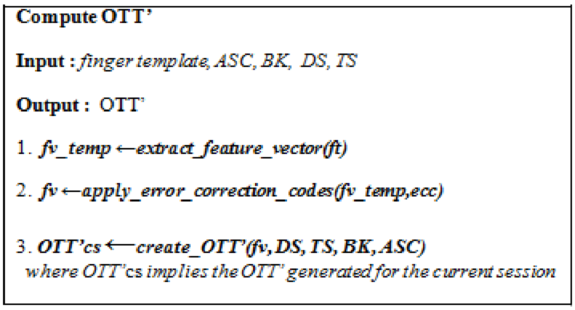


```
Compute OTT
Input : finger template, attribute list: a1..a8, DS, TS
Output : OTT
1. ecc ←generate_error_correction_codes(ft)
2. fv ←extract_feature_vector(ft)
3. w1..w8 ←map_attributes_to_weights(a1..a8)
4. BK ←derive_BioKey(w1..w8)
5. OTTm ←generate_OTT_Master(fv, BK)
6. OTTcs ←create_OTT(DS, TS, OTTm, ASC)
   where OTTcs implies the OTT generated for the current session
```

Fig. 3. OTT algorithm at SC

### C. OTT' generation at the CAD

The OTT' generation at the CAD is carried out based on the core OTT' generation algorithm, which is as illustrated in Fig. 4. At the CAD, the presented *ft* will be inputted to the *extract_feature_vector* routine. This will generate a temporary *feature vector*; represented as *fv_temp*. In the *apply_error_correction_codes* routine, the *ecc* will be applied over the *fv_temp* to generate the *fv*. The next step is to generate the *OTT'* set and is generated using the *create_OTT'* routine. In this routine, *fv* is transformed using the *DS, TS, BK* and *ASC* to derive the OTT' necessary for the current authentication session.



```
Compute OTT'
Input : finger template, ASC, BK, DS, TS
Output : OTT'
1. fv_temp ←extract_feature_vector(ft)
2. fv ←apply_error_correction_codes(fv_temp,ecc)
3. OTT'cs ←create_OTT'(fv, DS, TS, BK, ASC)
   where OTT'cs implies the OTT' generated for the current session
```

Fig. 4 OTT' algorithm at CAD

### D. Matching process

Each element of the OTT set generated at the SC will be represented as a binary bit string. They are represented as:

$$OTT = \{b1, b2, b3, \ldots, bn\}$$

Similarly each element of the OTT set generated at the CAD will be represented as a binary bit string. They are represented as:

$$OTT' = \{b1', b2', b3', \ldots, bn'\}$$

Now, each element of the OTT will be compared again each element of the OTT'. They are compared as follows:

$$b1 \rightarrow \{b1', b2', b3', \ldots, bn'\}$$

$$\cdot$$

$$bn \rightarrow \{b1', b2', b3', \ldots, bn'\}$$

Each local comparison operation will calculate the *hamming distance*[4] between the bit strings. The hamming distances of the local comparison elements under consideration is now summed up to generate a score for each local comparison, which is represented as follows.

$$\mathrm{HD}b1 = \mathrm{HD}(b1,b1') + \mathrm{HD}(b1,b2') + \mathrm{HD}(b1,b3') + \ldots + \mathrm{HD}(b1,bn')$$

$$\cdot$$

$$\mathrm{HD}bn = \mathrm{HD}(bn,b1') + \mathrm{HD}(bn,b2') + \mathrm{HD}(bn,b3') + \ldots + \mathrm{HD}(bn,bn')$$

The HD score of the local comparison is represented as follows in " (1)".

$$HD_{score\_local} = \sum_{k=1}^{n} \sum_{k'=1}^{n'} HD\,(b_k, b_{k'}) \quad (1)$$

The local hamming distance values generated now will be used to derive a global hamming distance value. It is derived from the local hamming distance comparisons as follows:

$$\mathbf{HD}\boldsymbol{score\_global} = (\mathbf{HD}\boldsymbol{b1} + \mathbf{HD}\boldsymbol{b2} + \mathbf{HD}\boldsymbol{b3} + \ldots + \mathbf{HD}\boldsymbol{bn}) / \boldsymbol{n}$$

The HD score of the global comparison is represented as follows in "(2)".

$$HD_{score\_global} = \frac{HD_{score\_local}}{n} \quad (2)$$

Now the $HD_{score\_global}$ will be bench marked against the minimum *threshold*[5] and the maximum threshold to determine the outcome of the comparison which will make the decision. This is represented as follows in " (3)".

$$Tmin \leq HD_{score\_global} \leq Tmax \quad (3)$$

The next section focuses on the evaluation of the PMOCFAS

## IV. EVALUATION OF THE PMOCFAS

Matlab stands for Matrix Laboratory and is a dynamic numeric scripting language widely used by students, engineers, researchers, and scientists globally. Matlab is ideal for simulation and prototyping of a framework, an algorithm or an idea, because of its flexible syntax, rich set of built-in functions and language capabilities [15]. Therefore, the simulation models are developed in MATLAB for evaluating the PMOCFAS against the CMOCFAS.

The PMOCFAS will be benchmarked against the CMOCFAS by testing against the finger template images acquired in the FVC2002 Database. FVC2002 stands for the Second Fingerprint Verification Competition. The aim of FVC is to focus on the fingerprint verification software and

[4] Hamming distance is a number used to denote the difference between two binary strings.

[5] The threshold represents the degree of similarity required between the enrolled biometric template and the stored template before a match is declared between the two templates.

algorithm assessment and to provide up to date state-of-the-art in fingerprint technology.

The FVC started in 2000 and the competitions ran in 2002, 2004 and in 2006. The reason for choosing this database is because of its world-wide acceptance in benchmarking the performance of fingerprint authentication algorithms [16]. Furthermore, the FVC 2002 DB1_B database has been made freely available for both academia and industry and hence will be used for the evaluation. The database consists of 80 fingerprint images obtained from 10 subjects. Each subject has 8 images and the images are in *TIF*[6] format. The images are 560x296 pixels, with a resolution of 569 *dpi*[7]. The FVC2002-DB1_B database from here on will be known as test database, to avoid redundancy. The test cases identified for thoroughly testing the PMOCFAS and CMOCFAS is based on a regressive testing approach and the steps are in what follows.

1. Each finger template needs to be tested against all 80 finger templates in the database.

2. Step1 needs to be performed for all possible threshold values in the range of 0 to 1, starting from 0.01 until 0.99 incremented by a 0.01, on each iteration. In other words, the tests are performed on 0.01, 0.02, 0.03, and so on until 0.99. So, we have 100 threshold values or iterations to test against all 80 fingerprint templates.

3. This test case expects both FAR and FRR for each threshold and for each finger template. The FAR and FRR values corresponding to all 80 finger templates obtained in Step1 needs to be calculated and recorded in the FAR and FRR tables. The tables for FAR and FRR will be a 100 row by 80 column.

4. Based on the FAR and the FRR table generated in Step3, graphs needs to be plotted between the threshold or similarity score and the FAR/FRR Error Rates.

5. The EER, the ZeroFAR, and the ZeroFRR need to be calculated from the graphs obtained in Step4.

The test results obtained by simulating the CMOCFAS and PMOCFAS are plotted as graphs in Fig. 5 and Fig. 6 respectively.

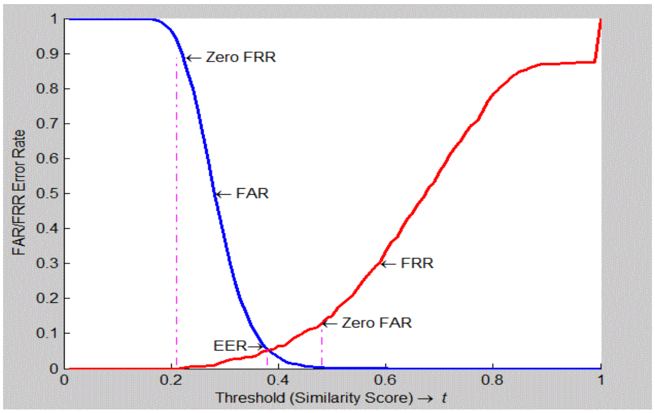


Fig. 5. CMOCFAS test results

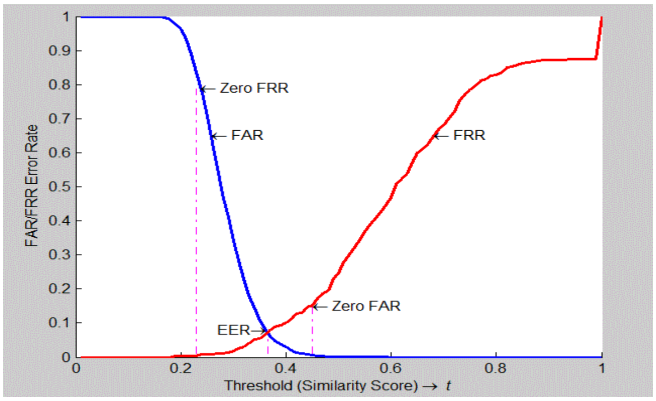


Fig. 6. PMOCFAS test results

The equations necessary to bench mark the PMOCFAS and the CMOCFAS error rates based on the test case are in what follows [17].

$$\boldsymbol{FAR} = \frac{\textit{Falsely Accepted Individuals}}{\textit{Total Number of Wrong Matchings'}} \qquad (4)$$

$$\boldsymbol{FRR} = \frac{\textit{Falsely Rejected Individuals}}{\textit{Total Number of Correct Matchings'}} \qquad (5)$$

$$\boldsymbol{EER} = \boldsymbol{FAR}\ (\boldsymbol{t}) = \boldsymbol{FRR}\ (\boldsymbol{t}) \qquad (6)$$

$$\boldsymbol{ZeroFAR} = \boldsymbol{min}\{\boldsymbol{FRR}\ (\boldsymbol{t}) \mid \boldsymbol{FAR}\ (\boldsymbol{t}) = \boldsymbol{0}\} \ (7)$$

$$\boldsymbol{ZeroFRR} = \boldsymbol{min}\{\boldsymbol{FAR}\ (\boldsymbol{t}) \mid \boldsymbol{FRR}\ (\boldsymbol{t}) = \boldsymbol{0}\} \ (8)$$

FAR implies the probability that an intruder gaining access to an authentication system and FRR implies the probability that an authorized person not gaining access to an authentication system. The term "*Total Number of Wrong Matchings'* " in "(4)" implies the total number of incorrect matchings' that are possible and it represents the finger templates that are not belonging to a valid or a genuine user. The term "*Total Number of Correct Matchings'* " in "(5)" implies that the total number of correct matchings' that are possible and it denotes the finger templates of valid or genuine

[6] Tag image bitmap file (TIF) is an image format file for high-quality graphics.

[7] dpi or DPI stands for Dots Per Inch, it's basically a measure of how many pixels can be placed in a span of 1 inch.

users. EER in “(6)” is defined as the point or threshold *t*, where FAR and FRR intersects. The EER is easy to understand and has found wide acceptance as it is often desirable to characterize the performance of a biometric system by a single parameter in order to benchmark system comparisons. ZeroFAR in “(7)” is defined as the lowest FRR at which no False Acceptances occur; whereas ZeroFRR in “(8)” is defined as the lowest FAR at which no False Rejections occur. The CMOCFAS and the PMOCFAS are now compared qualitatively based on the key security objectives set in this research and also based on the final test results obtained from “(4)” to “(8)”. Table I depicts a comparison of the CMOCFAS and the PMOCFAS.

TABLE I. COMPARISON OF THE CMOCFAS AND PMOCFAS

| **FAS** | ***PS*** | ***RC*** | ***RS*** | ***RRA*** | ***Zero FAR*** | ***Zero FRR*** | ***EER*** |
|---|---|---|---|---|---|---|---|
| CMOCFAS | L | L | N | L | at FRR = 0.13 | at FAR = 0.89 | 0.68 at *t* = 0.38 |
| PMOCFAS | H | M | H | H | at FRR = 0.15 | at FAR = 0.79 | 0.68 at *t* = 0.365 |

The abbreviations in Table I correspond to the entities listed as follows:

***FAS***– *Fingerprint Authentication System,* ***PS-*** *Privacy and security,* ***RC-*** *Resilience to compromise of finger template* ***RS-*** *Revocation support,* ***RRA-*** *Resilience to replay attacks* ***ZeroFAR-*** *Zero False Acceptance Rate,* ***ZeroFRR-*** *Zero False Rejection Rate,* ***EER-*** *Equal Error Rate,* ***CMOCFAS-*** *Conventional Match-on-Card Fingerprint Authentication System,* ***H-*** *High,* ***L-*** *Low,* ***M-*** *Medium,* ***N-*** *Nil,* ***FAR-*** *False Acceptance Rate,* ***FRR-*** *False Rejection Rate,* ***t-*** *threshold,* ***PMOCFAS-*** *Proposed Match-on-Card Fingerprint Authentication System*

As illustrated in Table I, the *privacy and security (PS)* property is marked as low in the CMOCFAS and high in the PMOCFAS. In the CMOCFAS, the original finger templates are captured and stored in the SC during enrollment. This is highly risky because if the finger template gets compromised; the finger template will be lost forever. In the PMOCFAS, the original finger template will be deleted after the OTT*m* generation process. As a result, even if the finger template is compromised, the attacker will not be able to reverse engineer the finger templates in its original form. Furthermore, all the communications between the SC and CAD are encrypted. As a result, the privacy and security is high in the PMOCFAS, where as it is low in the CMOCFAS.

The *resilience to compromise of finger template* (RC) property in the CMOCFAS is marked as low, as the original finger templates are stored in the SC. Furthermore, due to the low *privacy and security*, the finger templates are susceptible to compromise. In the PMOCFAS, the original finger templates are never stored in the SC and hence the *resilience to compromise of finger template* is high. Even if the finger template is compromised in the PMOCFAS, the authentication will not succeed as the OTT generated is unique for each authentication session. In the PMOCFAS, the *revocation support (RS)* property is high owing to the fact that even if the template is compromised, it is still revocable. The original finger template is intentionally distorted to an OTT*m* during the enrollment phase and stored in the SC. In the event of a compromise, a new transformed finger template can be easily generated. In the CMOCFAS, the support for revocation is absent. The compromise of a finger template in the CMOCFAS implies that the finger template is lost forever.

The *resilience to replay attacks* (RRA) property is high in PMOCFAS due to the fact that the usage of a compromised OTT to launch a replay attack will not succeed, as the OTT is unpredictable and distinctive for each authentication session. In the case of CMOCFAS, this is not the case, as the finger templates are not distinctive for each authentication session and are predictable. A vigilant reuse of an old instance of a finger template to launch a replay attack will succeed. Hence, the *resilience to replay attack* is low in the CMOCFAS.

It is observed from the Table I that the ZeroFAR of the CMOCFAS is at FRR 0.13. This means that the CMOCFAS will not be allowing the authentication of any imposters, when it has an FRR of 13% (an error rate of not allowing the authentication of a 13% of the genuine users). In the case of the PMOCFAS, the ZeroFAR is at FRR 0.15. It implies that the PMOCFAS will not be allowing the authentication of any imposters, when it has an FRR of 15% (an error rate of not allowing the authentication of a 15% of the genuine users). The ZeroFRR of the CMOCFAS is at FAR 0.89. This means that the CMOCFAS will not be rejecting the authentication of any genuine users, when it has an FAR of 89% (an error rate of allowing the authentication of 89% of imposters). In the case of the PMOCFAS, the ZeroFRR is at FRR 0.79. It implies that the PMOCFAS will not be rejecting the authentication of any genuine users, when it has an FAR of 79% (an error rate of allowing the authentication of 79% of imposters). The EER of the CMOCFAS is at 0.068 when the threshold *t* is at 0.38. In other words, the FAR and the FRR are equal when threshold *t* is at 0.38. In the case of the PMOCFAS, the EER is at 0.07 when the threshold *t* is at 0.365 and it is acceptable to keep the same threshold as the EER varies by a margin of only 0.002 when comparing to the EER of the CMOCFAS.

From the analysis, conducted in this section, it is observed that the PMOCFAS offers a higher security when compared to the CMOCFAS. Moreover, the FAR of the PMOCFAS is much lesser than the CMOCFAS when achieving the ZeroFRR. This is definitely an improvement over the CMOCFAS, as the FAR is reduced by a margin of 10%. The EER of the PFAS is on par with the EER of the CMOCFAS, as it varies only by a margin of 0.002. The FRR of the PMOCFAS is higher that the FRR of the CMOCFAS by a margin of 2% when achieving ZeroFAR. After taking all these factors into consideration, the PMOCFAS will definitely offer a superior security than the CMOCFAS, with a reasonably good performance in its attempt to achieve its security objectives.

## V. Future Work and Conclusions

The MOC technology using fingerprints holds great promise in offering good security and privacy protection when compared to the MOS/MOD authentication schemes. Although this is the case, the usage of it in the present form may lead to the vulnerabilities that are showcased in this paper. To alleviate the existing vulnerabilities, a novel PMOCFAS was conceptualized in this study. The core idea of the PMOCFAS is that the original finger template of the user is never stored in SC or on the CAD, and that the original fingerprint template is never transmitted during authentication.

The PMOCFAS finds its usage in numerous uses case of which the most significant ones are included in this section. It can be deployed by the banks across their ATMs and mobile device platforms to bring robustness on the payment using their SCs. The idea of the authentication performed using the OTT can be deployed in use cases where the existing fingerprint authentication schemes are used. The idea can also be well adapted to other biometric platforms. The proposed research can be used as catalyst for further researches involving biometric template protection and for MOC authentication approaches.

## References

[1] H. Lasisi and A. A. Ajisafe, “Development of stripe biometric based Fingerprint Authentications Systems in Automated Teller Machines,” 2012, IEEE, ISBN. 978-1-4673-2488-5, pp. 172–175.

[2] A. S. Shinde and V. Bendre, “An Embedded Fingerprint Authentication System,” 2015, IEEE, DOI. 10.1109/ICCUBEA.2015.45, pp. 205–208.

[3] B. Vibert, C. Rosenberger, and A. Ninassi, “Security and performance evaluation platform of biometric match on card,” 2013, IEEE, ISBN. 978-1-4699-0460-0, pp. 1–6.

[4] C. T. Pang, Y. W, Yun, and J. Xudong, “On-Card Matching,” 2009, Springer.

[5] Y. W. Yun and C. T. Pang, “An Introduction to Biometric Match-on-Card,” 2009.

[6] N. K. Ratha, J. Connell, R. M. Bolle, and S. Chikkerur, “Cancelable Biometrics: A Case Study in Fingerprints ,” 2006, IEEE, ISSN. 1061-4651, pp. 370–373.

[7] N. K. Ratha, J. Connell, R. M. Bolle, and S. Chikkerur, “Generatig Cancelable Fingerprint Templates ,” 2007, IEEE, ISSN. 0162-8828, pp. 561–562.

[8] U. Uludag, S. Pankanti, S. Prabhakar, and A. K. Jain, “Biometric Cryptosystems: Issues and Challenges ,” 2004, IEEE, ISSN. 0018-9219, pp. 948–960.

[9] A. K. Jain, K. Nandakumar, and A. Nagar, “Biometric Template Security,” 2008, ACM, , DOI. 10.1155/2008/579416.

[10] A. Nagar, K. Nandakumar, and A.K. Jain, “Biometric Template Transformation: A Security Analysis,” 2010, SPIE, DOI: 10.1117/12.839976.

[11] T. Connie, A. B. J. Teoh, M. Goh, and D. C. L. Ngo, “PalmHashing: A Novel Approach for Cancelable Biometrics,” Information Processing Letters,” 2005, ScienceDirect, DOI: 10.1016/j.ipl.2004.09.014, vol. 93, pp. 1-5.

[12] M. Savvides and B. V. K. Vijaya Kumar, “Cancellable Biometric Filters for Face Recognition,” 2004, IEEE, ISSN. 1051-4651, vol. 3, pp. 922–925.

[13] Y. Lee, Y. Chung, and K. Moon, “Inverse Operation and Preimage Attack on BioHashing,” 2009, IEEE, ISBN, 978-1-4244-2773-4, pp. 92-97.

[14] K.K. Nair, A. Helberg, and J. V. D. Merwe, “Intrusion detection in Bluetooth enabled mobile phones,” 2015, IEEE, ISBN. 978-1-4799-7754-3, pp. 1-8.

[15] X. Li, L. Hendren, “Mc2 FOR Demo: A Tool for Automatically Translating MATLAB to FORTRAN 95,” 2014, IEEE, doi. 10.1109/CSMR-WCRE.2014.6747218, pp. 458-463.

[16] H. Yoshimura, “Fingerprint Templates with High Recognition Accuracy and High Security Generated by Discrete Fractional Sine Transform,” 2014, IEEE, ISBN. 978-1-4577-0884-8, pp. 185-190.

[17] G. Vitello, V. Conti, A. Gentile, S. Vitabile, and F. Sorbello, “Design and Implementation of an Efficient Fingerprint Features Extractor,” 2014, IEEE, doi. 10.1109