A Secure Fingerprint Matching Technique

Shenglin Yang
UCLA Dept. of EE
Los Angeles, CA 90095
+1-310-267-4940
shengliny@ee.ucla.edu

Ingrid M. Verbauwhede
UCLA Dept. of EE
Los Angeles, CA 90095
+1-310-794-5209
ingrid@ee.ucla.edu

ABSTRACT
In this paper, we propose a novel robust secure fingerprint matching technique, which is secure against side channel attacks. An algorithm based on the local structure of the minutiae is presented to match the fingerprints. The main contribution is the careful division of the fingerprint recognition system into two parts: a secure part and a non-secure part. Only the relative small secure part, which contains sensitive biometric template information, requires realization in specialized DPA-proof logic. The rest of the system is running on LEON, which is a regular embedded platform.

Categories and Subject Descriptors
I.5.5 [**Pattern Recognition**]: Implementation – *Special architecture*; C.3 [**Special-purpose and Application-based Systems**] – *Real-time and embedded systems, Signal processing systems.*

General Terms
Algorithms, Performance, Design, Security.

Keywords
Fingerprint Recognition, Secure Matching, DPA-proof, Embedded System.

1. MOTIVATION
Biometric recognition systems offer greater security and convenience than traditional methods of personal recognition. Along with the rapid growing of this emerging technology, the system performance, such as accuracy and speed, is continuously improved. At the same time, the security of the biometric system itself is becoming more and more important.

One of the most significant disadvantages of the biometric recognition system is that they cannot be easily recalled. For example, if one of the fingers is used as a password, once it is compromised, it never can be used again since it is almost impossible that a fingerprint can be changed, which means it is compromised forever. Moreover, since one person only has a limited number of fingers, different applications might use the same fingerprint. A person's biometric stolen from one application could also be used in some other applications [12]. Therefore the secure storage of the biometric template is becoming extremely important. In a traditional biometric recognition system, the biometric template, such as fingerprint, voice, etc., is usually stored on a central server during enrollment. The input biometric signal captured by the front-end sensor is sent to the server and the processing and matching steps are performed on the server. In this case the safety of the precious biometric information cannot be guaranteed because attacks might occur during transmission or on the server. Embedded biometric recognition systems try to solve this problem by moving the signal processing and matching engines from the server to the embedded device. In these systems, the biometric signals are processed and matched on the embedded device and only the result is transmitted to the server. This approach can avoid the attacks on communication and server. It also avoids that the biometric data needs to be stored on multiple servers for multiple applications. However, it is very easy to compromise the plain-text storage of the template in the embedded device. To make the storage more secure, the biometric template is encrypted using a secret key before being stored. As soon as the input signal has come, the matcher decrypts the template and performs the comparison. However, some dedicated attacks can still extract the secure key, and in turn, the template. The reason for this is that the physical implementation of an algorithm provides attackers with some important information. Examples are variations in timing, power consumption and electromagnetic radiation, which can be used to link to the internal state, and hence to the secret data. These types of attacks are called Side Channel Attacks (SCA). Among the SCA, differential power Analysis (DPA) is the most powerful one. It relies on statistical analysis and error correction to extract information from the power consumption that is correlated to secret data [10].

To solve this problem, one possible way is that instead of storing the original biometric template on the embedded device, the system could store its noninvertible transformed version, for instance, a hash, in the enroll phase. During recognition, the input biometric information is first encrypted using the same noninvertible transform. Then matching is performed in the transformed space. Different applications can use different noninvertible transforms or different parameters of the same transform. Thus a template would be usable only by the application that created it. If a hacker ever compromises such a

WBMA '03, November 8, 2003, Berkeley, California, USA.

biometric template, the system can issue a new template using a different transform or a different parameter for the same transformation [12]. However, the implementation of this technique is very difficult since the noninvertible transforms are usually very weak in the information-theoretic sense. For example, the goal of a hash function is to convert the signal to a secure form whose change cannot reflect that of the original signal. This characteristic of hash function lowers the accuracy of the matcher or even makes it impossible because that the matcher requires strong relationship between the original signal and its transformed form. It cannot effectively deal with the biometric signal variations in this kind of transformed space.

In this paper we propose a novel fingerprint matching technique to address this problem. The basic idea is to construct the matching algorithm in two parts: a secure part and a non-secure part. The non-secure part is running on a regular embedded platform – LEON, which is a 32-bit highly configurable processor [5], and the secure part is designed to run on a DPA-proof platform. This paper is organized as following: section 2 briefly reviews some related work in fingerprint matching as well as the DPA proof technique, section 3 describes our proposed technique for secure fingerprint matching, section 4 presents the experimental results and the analysis of the proposed technique. Then, the conclusion is presented in section 5.

2. RELATED WORK

2.1 Fingerprint Matching

There are two basic types of fingerprint matching techniques: graph based and minutiae based. For modern embedded fingerprint recognition systems, the minutiae-based matching is popular because, on the one hand, the minutiae of the fingerprint are widely believed the most discriminating and reliable features, and on the other hand, the template size of the biometric information based on minutiae is much smaller and the processing speed is higher than that of graph-based fingerprint matching. These characteristics are very important for saving memory and energy on the embedded devices. Lots of work has been done for minutiae-based fingerprint matching. Some of them use the local structure of the minutiae to describe the characteristics of the minutiae set [7]. This approach has high processing speed and robustness to rotation and partial prints. However, the local structure usually has less distinct features because it only represents some parts of the whole minutiae set. Prints from different fingers may have quite a few similar local structures by coincidence while prints from the same finger may only have very few similar structures due to the presence of false minutiae and the absence of genuine minutiae. Alignment-based matching algorithms take use of the shape of the ridge connected to minutiae [8]. This might improve the system accuracy. However, this approach results in a larger template size because the associated ridges for each minutia must be saved. Some other researches combine the local and global structures [9]. The local structure is used to find the correspondence of two minutiae sets and increase the reliability of the global matching. The global structure of minutiae reliably determines the uniqueness of a fingerprint. The approach in [16] is similar to our work. However we propose a new definition of the local structure of a minutia, which is proven efficient for low quality input fingerprints and a low accurate minutiae extraction.

2.2 DPA Proof Technique

Along with the growing of the SCA techniques, countermeasures against Differential Power Analysis have been proposed at different levels of abstraction. Yet, advanced versions of DPA are able to greatly reduce their effects. For example, Random Process Interrupts [4] can be synchronized by integration techniques [3] and Modified DPA [11] can handle masking techniques [2]. Random power consuming operations on the other hand merely lower the side channel information and might be disabled through tampering.

The former countermeasures attempt to conceal the power variations at the architectural or algorithmic level, while they originate at the logic level. Implementing the sensitive parts of a crypto processor in a logic style, whose power consumption is independent of the signal transitions, removes the foundation of DPA. One such logic style available is Sense Amplifier Based Logic (SABL) [13][14]. A logic gate in SABL charges a total capacitance with a constant value in every cycle. Hence SABL consumes the same constant energy independent of the input values and is an effective countermeasure against DPA.

3. SECURE MATCHING TECHNIQUE

3.1 Algorithm

In this paper, the image processing stage to extract a minutiae set from the fingerprints is based on the NIST Fingerprint Image Software [15]. The architectural modifications to obtain a high-speed and memory efficient implementation for an embedded platform are discussed in [17]. From the result of the minutiae detection step, information such as x, y co-ordinates and local ridge direction are available for each minutia. Of course this minutiae information could be used directly to match the fingerprints. However, to separate the secure part of the matching algorithm easily and lower the secure cost of the system, our newly proposed technique is based on a derived local structure. A detailed discussion will be given in section 4.

Generally, given one minutia M, the local structure of it is described as a feature vector:

$$L_M = \{d_1, d_2, ..., d_N, \varphi_1, \varphi_2, ..., \varphi_N, \vartheta_1, \vartheta_2, ..., \vartheta_N, \Psi\} \quad (1)$$

Where N is the number of neighbors taken into consideration during matching. Ψ is the local ridge direction of minutia M. $d_n \ (n = 1,2,...N)$ describes the distance between the selected minutia M and its n^{th} nearest neighbor, $\varphi_n \ (n = 1,2,...N)$ is the related radial angle between M and its n^{th} nearest neighbor, and $\vartheta_n \ (n = 1,2,...N)$ represents the related position angle of the n^{th} nearest neighbor. One example with $N = 2$ is shown in figure 1.

Figure 1 describes the local structure of a minutia with its two nearest neighbors. All the elements in the local structure can be calculated from the information obtained from the minutiae extraction following equation (2).

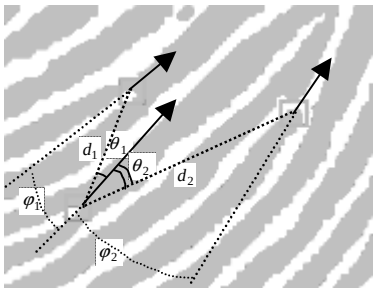


Figure 1. Local structure of the minutia (N=2).

$$\begin{cases} d_n = \sqrt{(x_n - x_0)^2 + (y_n - x_0)^2} \\ \varphi_n = diff(\Psi_n, \Psi) \\ \vartheta_n = diff\left(\arctan\left(\frac{y_n - y_0}{x_n - x_0}\right), \Psi\right) \end{cases}, n = 1,2,..., N \quad (2)$$

The function $diff(\ )$ calculates the difference of two angles and converts the result to range $[0,2\pi)$.

The proposed matching algorithm calculates how similar the neighborhood of one minutia in the input fingerprint is to that of one in the stored template. If it is similar enough, then these two minutiae are taken as a "matched" minutiae pair. After each minutia in the input fingerprint is checked, the total number of "matched" minutiae pair is used to calculate the final matching score.

The selection of the number of neighbors is very important for the system performance. In this paper, the neighborhood is defined as 6 nearest neighbors for each minutia as shown in figure 2. The detailed selection methodology will be presented in section 4.

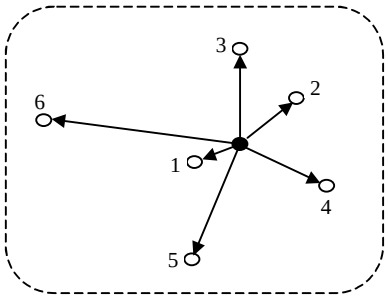


Figure 2. Neighborhood of a minutia.

When two minutiae are compared, the relative position and angles of their 6 nearest neighbor minutiae are investigated. We can rewrite equation (1) to obtain another form of the local feature vector. Assume one minutia M in the input fingerprint:

$$L_M = \{\{d_1, \varphi_1, \vartheta_1\}, \{d_2, \varphi_2, \vartheta_2\}, ..., \{d_N, \varphi_N, \vartheta_N\}, \Psi\} \quad (3)$$

And one minutia M' in the stored template:

$$L_{M'} = \left\{\left\{d_1', \varphi_1', \vartheta_1'\right\}, \left\{d_2', \varphi_2', \vartheta_2'\right\}, ..., \left\{d_N', \varphi_N', \vartheta_N'\right\}, \Psi'\right\} \quad (4)$$

To decide whether or not M and M' are a matched minutiae pair, a small four-dimension range box is set for $(d, \varphi, \vartheta, \Psi)$ respectively: $\{\Delta_d, \Delta_\varphi, \Delta_\vartheta, \Delta_\Psi\}$. The first step is checking the local ridge directions of the two minutiae. If $|\Psi - \Psi'| > \Delta_\Psi$, M and M' are not matched. Therefore the matcher searches for another minutiae pair. Otherwise, the matcher continues to investigate the neighbor minutiae according to the neighborhood condition described in (5):

$$\begin{cases} \left|d_i - d_j'\right| <= \Delta_d \\ \left|\varphi_i - \varphi_j'\right| <= \Delta_\varphi \\ \left|\vartheta_i - \vartheta_j'\right| <= \Delta_\vartheta \end{cases} \quad (5)$$

If the conditions in (5) are all satisfied, the i^{th} neighbor of the input minutia M and the j^{th} neighbor of the template minutia M' are considered "marked". After a thorough check of all the neighbor minutiae of M and M', the number of marked neighbor pairs is accumulated as A. If this number is above some specific threshold TH_A, the minutiae M and M' are considered as a matched minutiae pair. The threshold is set according to experimental results, which we will discuss later. Following this procedure, a comparison of all the minutiae in the input and template fingerprints results in the total number of matched minutiae pairs, B. Assume that the number of the minutiae of input and template fingerprints are NUM_{input} and NUM_{temp}, respectively. Then the final matching score is calculated as:

$$Score = \frac{B}{\max(NUM_{input}, NUM_{temp})} \quad (6)$$

Two fingerprints will be verified as from the same finger if their matching score is higher than a fix-set threshold.

3.2 Secure Partitioning

The above section describes the overall algorithm for our secure fingerprint matching technique. Recall that, even if the encrypted template is stored in the embedded device, due to the leakage of side channel information, there are still security holes. Any handling of the template will result in a specific power consumption pattern, which can be detected and analyzed with a Differential Power Analysis.

To address this problem, we introduce Sense Amplifier Based Logic into our system to provide the secure storage and handling of the biometrics template. A design in SABL has a constant power consumption and does not emit side channel information [14].

Before implementation of the DPA-proof technique, the whole system needs to be partitioned into secure part and non-secure part. The secure part is defined as the part of algorithm, which is

related to the sensitive biometric template. It includes the accessing of the template data as well as the computation related to those values. Figure 3 shows the overall flowchart of the whole system. From a security aspect, the biometric template is the sensitive data part of this algorithm. Therefore, the comparators with one or more sensitive operands also need to be protected. Thus, In Figure 3, the area in shade needs to be implemented into the DPA-proof logic.

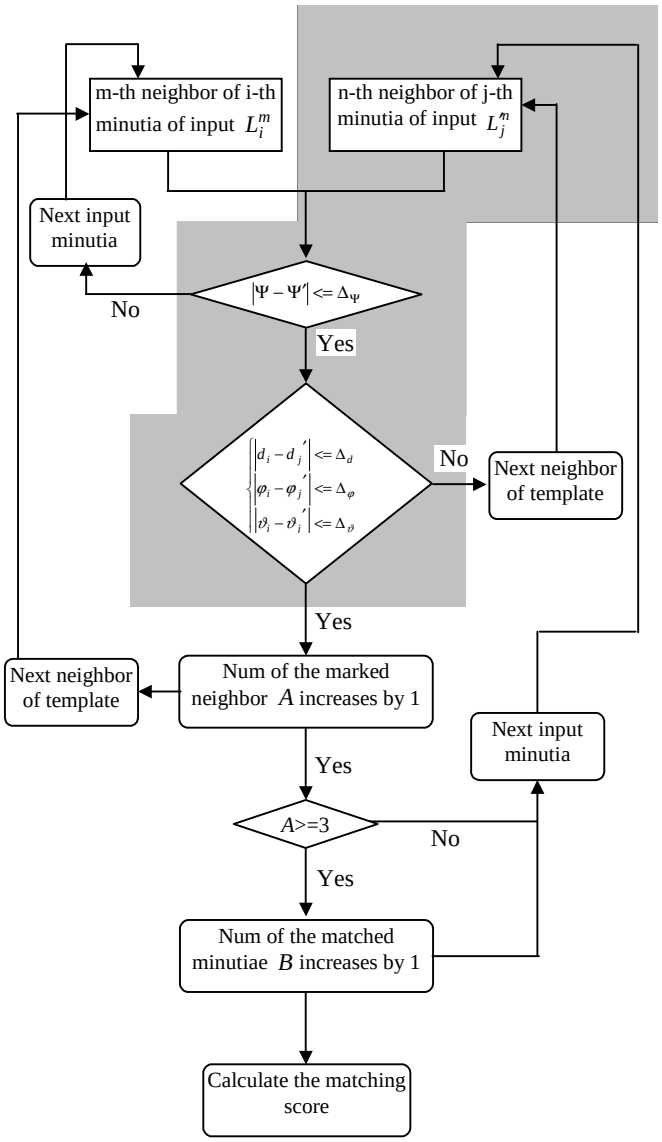


Figure 3. Flowchart of the matching system.

In this paper, we build a DPA-proof block, an Oracle, into which the secure part is put. Of course, the easiest and safest way to implement the matching algorithm is porting the whole system in the Oracle. However, the DPA-proof logic costs around two times more area and power than a regular implementation. Therefore, a careful partitioning of the matching algorithm is important to minimize the cost of the system.

Also, we need to decide the format of the template stored in the Oracle. In traditional minutiae-based fingerprint recognition systems, the stored information for each minutia is the x,y-coordinate, the local ridge direction, etc. When the matching engine starts, this information is converted to a more useful format, for example, the distance between the minutia and its neighbor, the related angle between two neighboring minutiae, and so on. However, in the proposed system, data stored inside of the Oracle cannot be accessed from outside, which means that all the calculations related to the stored data need to performed within the Oracle. A further study of the algorithm shows that some complex mathematic calculations, such as *atan, sqrt,* etc., are needed in the template convertion. This will increase the area as well as the design complexity of the Oracle. Therefore a preprocessing step is performed before the template storage. According to equation (2), the preprocessed minutiae information is calculated and stored as template in our system, shown in figure 4.

d_1	φ_1	ϑ_1	Ψ
d_2	φ_2	ϑ_2	
d_3	φ_3	ϑ_3	
. . .	. . .	. . .	
d_N	φ_N	ϑ_N	

Figure 4. Stored template for fingerprint.

4. ALGORITHM ANALYSIS

In this section, we will first discuss the experimental results of the proposed matching algorithm. Then a security analysis is presented.

4.1 Experimental Results

An Authentec AF-2 CMOS imaging sensor is used to collect fingerprint samples. The sensor has an accuracy of 8bits/sample. However, to save energy and time of the embedded device, we adopt a 3bits/sample rate in our system. This will result in relatively low quality input image. Moreover, porting the minutiae extraction processing to the embedded device introduces some extra error due to the time constraints and finite word length limitations [17]. All these embedded device constraints require a robust matching algorithm.

Our proposed method is based on the local structure of the minutiae. One of the important parameters is the number of neighbors taken as the local structure for each minutia. If the number is too small, which means that the matching condition is quite loose, some un-matched minutiae pairs are very likely to satisfy the matching condition, which may lead to a high FAR (False Accept Rate). On the other hand, if the number is too large, the matching condition becomes too strict. Many matched pairs may fail because the fingerprint image is sometimes uncompleted and the minutiae detection is not very precise. This may result in a high FRR (False Reject Rate). To select the proper number of neighbors taken as local feature, experiments are done for a local structure of 4,5,6 and 7 neighbors. For each case, different marked neighbor pair thresholds are investigated. Since the baseline

accuracy needed for modern biometric systems is 1% FRR and 0.01% FAR [1], table 1 presents whether or not the selection can reach this standard. The image set consists of 10 fingerprints per finger from 10 different fingers for a total 100 fingerprint images.

Table 1. Possibility to achieve baseline accuracy for different local structure definition and threshold.

Neighbor Num / TH_A	4	5	6	7
2	No	No	No	No
3	No	Yes	Yes	No
4	—	No	No	No
5	—	—	No	No
6	—	—	—	No

Figure 5 shows the result from two different local structure definitions, which both can achieve the baseline accuracy. The X-axis is the FRR and the Y-axis shows the FAR.

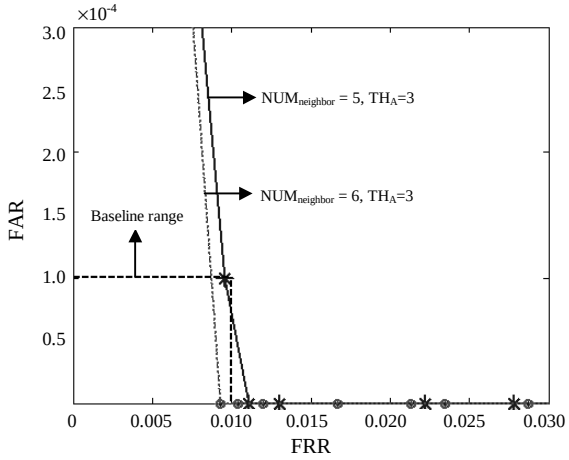


Figure 5. FRR and FAR for different selection of local structure.

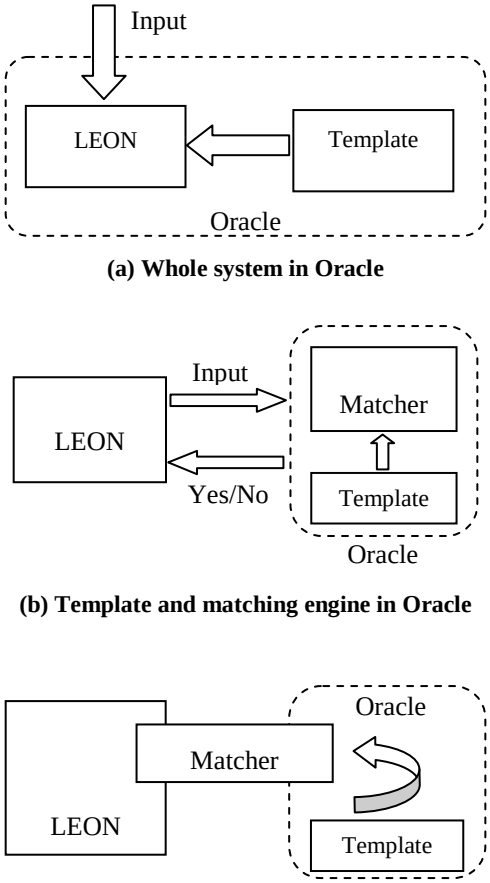
After analyzing the above result, we define the number of neighbors as 6 and the marked neighbor pair threshold TH_A is set to 3. By doing this, we obtain a FRR of 1% and a FAR of less than 0.1 FAR*.

4.2 Security Partitioning Analysis

As mentioned in section 3, due to the high area cost of the DPA-proof technique, a careful partitioning needs to be done for the matching algorithm. From the system point of view, we have several ways to separate the algorithm into the LEON and the Oracle. The most straightforward way is to port the whole system, including the LEON processor into the Oracle (see Figure 6(a)). This will make the whole recognition system protected. However, obviously, the disadvantage of this implementation is that the area cost of the whole system increases by at least a factor of 2.

Another solution for this problem is to store the biometric template in the Oracle. The input signal from the sensor is converted into the template-like feature vector and sent to the Oracle (see Figure 6(b)). The matching engine runs inside the Oracle and gives out the final matching result. This division is quite clear but the matching algorithm in the Oracle requires a large computation capability. This also would make the size of the Oracle too big.

To lower the system cost, the Oracle block needs to be made as small as possible. In Figure 6(c), the template is stored in the Oracle and the matching engine is divided into two parts. Only the sensitive part is running on the Oracle.



(a) Whole system in Oracle

(b) Template and matching engine in Oracle

(c) Matcher is partitioned into two parts

Figure 6. System level partitioning of fingerprint recognition system.

In this paper, we adopt the last partitioning of the system. As shown in the flowchart in Figure 3, the shaded part of the algorithm only consists of comparators and adders. The rest of computations are performed on the LEON embedded processor. By doing this, only the sensitive part of the algorithm is protected. Therefore, only a relative small part of algorithm needs to be implemented in special hardware. This will lower the overall system design complexity and make the Oracle block smaller.

The Oracle is a black box to the LEON. Whenever there is a need of a comparison against template values, the LEON processor sends the input values and the address of the requested template

* In the simulation of our database, there is no False Accept Error.

minutia to the Oracle. The result of the comparison is sent back to the LEON processor in a simply Yes/No format. Based on this result, LEON makes a decision what to do next and what is the final matching score.

5. CONCLUSION

In this paper, we present a novel secure fingerprint recognition system, in which the minutiae-based matching algorithm is robust against relative low quality of input fingerprint images and minutiae detection. Secure partitioning is performed to guarantee low system cost as well as the safety of the precious biometric template by storing them into a DPA-proof block, an Oracle. By properly defining the local structure of the minutiae, we achieve 1% FRR and less than 0.01% FAR*.

6. ACKNOWLEDGMENT

The authors would like to acknowledge the funding of NSF account no CCR-0098361 and would like to thank the Thumbpod teammates [6].

7. REFERENCES

[1] Anderson, R.J., Security Engineering, A Guide to Building Dependable Distributed Systems**,** John Wiley & Sons, 2001.

[2] Chari, S, Jutla, C.S, Rao, J.R. and Rohatgi, P., Towards sound approaches to counteract power-analysis attacks. Advances in Cryptology - CRYPTO'99. 19th Annual International Cryptology Conference. Proceedings. Springer-Verlag. 1999, pp.398-412. Berlin, Germany.

[3] Clavier, C., Coron J. and Dabbous, N., Differential power analysis in the presence of hardware countermeasures, Cryptographic Hardwareand Embedded Systems – CHES 2000. Second International Workshop. Proceedings (Lecture Notes in Computer Science Vol.1965). Springer-Verlag. 2000, pp.252-63. Berlin, Germany.

[4] Daemen, J. and Rijmen, V., "Resistance Against Implementation Attacks: A Comparative Study of the AES Proposals", Proceedings of the 2nd AES Candidate Conference, 1999, pp.122-132.

[5] http://www.gaisler.com.

[6] http://www.ThumbPod.com.

[7] Hrechak, AK and McHugh, JA. Automated fingerprint recognition using structural matching, Pattern Recognition, vol.23, no.8, 1990, pp.893-904. UK.

[8] Jain, A., Lin, H. and Bolle, R., On-line fingerprint verification, IEEE Transactions on Pattern Analysis & Machine Intelligence, vol.19, no.4, April 1997, pp.302-14. Publisher: IEEE Comput. Soc, USA.

[9] Jiang, X., Yau, W., Fingerprint minutiae matching based on the local and global structures, Proceedings 15th International Conference on Pattern Recognition. ICPR-2000. IEEE Comput. Soc. Part, vol.2, 2000, pp.1038-41 vol.2. Los Alamitos, CA, USA.

[10] Kocher, P., Jaffe, J., and Jun, B., Differential power analysis, Proceeding of Advances in Cryptology – Crypto'99. 19th Annual International Cryptology Conference. 1999, pp.388-97. Berlin, Germany.

[11] Messerges, T.S., Using second-order power analysis to attack DPA resistant software, Cryptographic Hardware and Embedded Systems - CHES 2000. Second International Workshop. Proceedings (Lecture Notes in Computer Science Vol.1965). Springer-Verlag. 2000, pp.238-51. Berlin, Germany.

[12] Prabhakar, S., Pankanti, S., and Jain, A. K., Biometric Recognition: Security and Privacy Concerns, IEEE Security and Privacy Magazine, Vol. 1, No. 2, pp. 33-42, March-April 2003.

[13] Tiri, K., Akmal, M. and Verbauwhede, I., A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards. ESSCIRC 2002. Proceedings of the 28th European Solid-State Circuit Conference. Univ. Bologna. 2002, pp.403-6. Bologna, Italy.

[14] Tiri, K. and Verbauwhede, I., Security encryption algorithms against DPA at the logic level: next generation smart card technology, Workshop on Cryptographic Hardware and Embedded Systems (Lecture Notes Computer Science Vol.2779), Sept. 2003, pp 125-136, Cologne, Germany.

[15] User's Guide to NIST Fingerprint Image Software (NFIS). NISTIR 6813, National Institute of Standards and Technology.

[16] Wahab A., Chin, S.H., Tan, E.C., Novel approach to automated fingerprint recognition. IEE Proceedings: Vision, Image & Signal Processing, vol.145, no.3, June 1998, pp.160-6. Publisher: IEE, UK.

[17] Yang, S., Sakiyama, K. and Verbauwhede, I., A Secure and Efficient Fingerprint Verification System for Embedded Systems, 37th Asilomar Conference on Signal, Systems, and Computers, Nov. 2003, Pacific Grove, CA.