User Identification and Key Regeneration System Employing Rotated Reference Images of the Iris

Guilherme Nunes Melo[1], Valdemar C. da Rocha Jr.[1] e José S. Lemos-Neto[1]

Abstract— **A new test called *rotation search* is proposed for user identification and cryptographic key regeneration in systems employing a digital representation of the iris (iris code). When applied to the BIOSECURE, CASIA and NIST-ICE data bases the rotation search shows, on average, a two fold reduction in *false rejection ratio* (FRR) with a *false acceptance ratio* (FAR) equal to zero, in comparison with the standard search employed in other systems. The highest improvement reached in FRR by the rotation search against the standard search is 100 times, and in many cases the measured FRR is equal to zero.**

Keywords— **Iris code, biometry, coding, authentication.**

I. Introduction

User identification employing biometric data is now a reality in many computer based systems including banking, voting, access to vaults, etc. Applications using biometrics will certainly grow and will reach more users as soon as they become more reliable, by granting access to genuine users and denying access to impostors with high probability. Many identification systems using biometrics operate on user fingerprints, the palm, the face or the iris. In this paper we restrict attention to systems employing digital iris data (iris code). The reason for this choice is due to the fact that the iris code presents the highest entropy in comparison to other biometric data currently in use [1]. Our idea is to take advantage of the higher iris code entropy in order to achieve higher security levels against impostors. We refer to the test proposed in this paper as *rotation search*, which performs a search to identify a user by employing both, rotated reference images and rotated test images. As described in the sequel, rotation search applied to the BIOSECURE, CASIA and NIST-ICE data bases shows, on average, a two fold reduction in *false rejection ratio* (FRR) with a *false acceptance ratio* (FAR) equal to zero, in comparison with the standard search employed in other systems [2], [3], [4]. The highest improvement reached in FRR by the rotation search against the standard search is 100 times, and in many cases the measured FRR is equal to zero.

The rest of this paper is organized as follows. In Section II we describe the main relevant aspects of the data bases employed. In Section III we present the key regeneration system proposed in this paper which employing error-correcting codes. In Section IV we describe the new proposed test. In Section V the experiments using the proposed system and the proposed test are presented and the results are compared to the results of the system in [4]. Finally, in Section VI, we present our conclusions.

II. Data Bases

Typically, data bases contain sets of images for each user, and that includes both reference images (I_{ref}) and test images (I_{sam}). A reference image is understood to be generated under ideal conditions while a test image is understood to be generated by a user identification equipment, i.e., in less than ideal conditions. Following [2], for each image in a data base, a binary string of length $1,188$ bits is derived from an infrared image of an iris. These binary strings are denominated *iris codes*. The iris codes obtained from reference images and test images are respectively denoted by θ_{ref} and θ_{sam}. In this paper, the iris codes used in the tests are derived from the following data bases: BIOSECURE [5], CASIA [5] and NIST-ICE [6].

The BIOSECURE and CASIA data bases are formed by $1,200$ images each, originating from 60 distinct users, each user having 20 images, where 10 images are reference images and the remaining 10 images are test images. Another possible interpretation is to consider 30 distinct users, being 10 reference images for the right eye, 10 test images for the right eye, 10 reference images for the left eye, and 10 test images for the left eye. Using these two data bases, $6,000$ tests for genuine users are possible in each data base by using one iris image at a time, or $3,000$ tests are possible by using iris images for both eyes at the same time. The NIST-ICE data base is formed by $2,953$ images, which are divided in two tests namely, ICE-exp1 and ICE-exp2. The ICE-exp1 test is formed by 124 users having a total of $1,425$ images and refers to the right eye, while the ICE-exp2 test is formed by 120 users having a total of $1,528$ images and refers to the left eye. The ICE-exp1 data base allows $12,214$ tests for genuine users while the ICE-exp2 test allows $14,653$ tests. In the NIST-ICE data base the number of images per user is not fix, being possible to find users having a number of images ranging from 1 to a maximum of 31 images. Consequently a control file is required in order to keep a list of which images will be used in each test. We call data bases which have the same number of images per user as a *regular data bases* (BIOSECURE and CASIA), while for data bases which the number of images per user varies as an *irregular data base* (NIST-ICE).

III. Key Regeneration System Proposed

Figure 1 presents a block diagram for the key regeneration system proposed in this paper. We use this system for testing images available from the iris data bases of Section II. The proposed system is essentially that proposed in [2] except for the insertion of random numbers [3] instead of zeros. The main features of the proposed system are described in the sequel.

[1] Communications Research Group, Department of Electronics and Systems, Federal University of Pernambuco, 50740-550, Recife, PE, Brazil, E-mails: {guilherme.nmelo, vcr, jose.lemosnt}@ufpe.br.

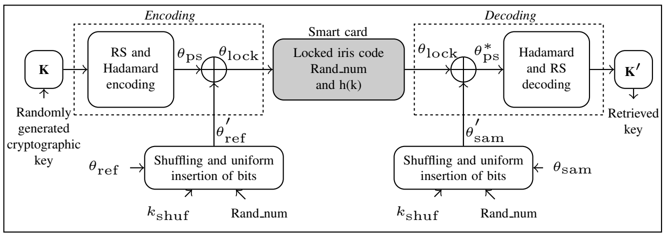

Fig. 1. Key regeneration unibiometric system, employing smart card, iris and password.

As illustrated in Figure 1, encoding of the cryptographic key $\mathbf{K}$ is performed sequentially by employing first a Reed-Solomon (RS) code [7, p.294] and then a Hadamard code [7, p.44]. Following [2], we consider shortened RS codes of block length 61 over $\mathrm{GF}(2^6)$ and the Hadamard $(32, 6, 16)$ code. The error-correcting capability t_{RS} of the RS code is adjusted for the range $1 \leq t_{\mathrm{RS}} \leq 22$ satisfying the relation $k = 61 - 2t_{\mathrm{RS}}$, where k denotes the number of information symbols of the RS code. Values of t_{RS} greater than 22 are avoided because they increase the FAR, i.e., for $t_{\mathrm{RS}} > 22$ the scheme begins to erroneously consider some impostors as genuine users.

The shuffling operation, according to [2], consists in segmenting the $1,188$ bits iris code into 198 blocks of 6 bits each, and these blocks are then reordered using a randomly generated binary shuffling key (k_{shuf}) of length 198 bits as follows. The block reordering (shuffling operation) can be more clearly described by making an analogy with the situation which occurs when passengers are going to board a plane. Originally the 198 blocks (passengers) form a single queue and sequentially each block (passenger) in the queue is assigned a token which is either a 1 or a 0, obtained by sequentially reading the bits in the shuffling key. Then, respecting the original order of the blocks in the queue, two new queues are formed. One queue receives those blocks for which their token is 1 (premium class passengers) and the other queue receives those blocks for which their token is 0 (standard class passengers). Obeying their arrival order, blocks with token 1 board first, followed by blocks with token 0, i.e., the shuffled sequence contains the blocks with token 1 followed by the blocks with token 0.

The uniform insertion of bits consists of concatenating a cascade of blocks formed by three bits from the iris code followed by two bits from a random or pseudo random sequence. Since the iris code consists of a binary sequence of length $1,188$ bits, it follows that after bit insertions a sequence of length $1,980$ bits results. However, since the block length of θ_{ps} is $61{\times}32 = 1,952$, it is necessary to delete 28 bits from the sequence of length $1,980$ and thus achieve a length of $1,952$ for θ'_{ref}. Among the 28 bits deleted there are 18 bits from the iris code which are lost. The lost iris code bits correspond to approximately 1.52% of the total, and it was verified that the error correction performance is not significantly affected by this loss.

Decoding starts with the Hadamard code, which means that for each one of its codewords a 6 bit byte is delivered to form one symbol of a codeword for the RS code. A little trick [2] which speeds up the computer simulation is then employed. Since we know the RS codeword that has been generated, we compare this generated RS codeword with the word coming out of the Hadamard decoder and count the number of symbol errors. The number of symbol errors t is compared against the number of errors t_{RS} that the RS code can correct. If $t \leq t_{\mathrm{RS}}$ then we recover the cryptographic key $\mathbf{K}$, thus avoiding the actual decoding of the RS code and reducing processing time.

IV. New Proposed Test

In this section we propose a new user identification test called *rotation search*. Before we present the proposed test, we briefly describe the test used in [2]-[4], that we call *standard search*, in order to compare it to the rotation search test. We emphasize that the tests use the iris codes θ_{ref} and θ_{sam} instead of their respective I_{ref} and I_{sam}, as indicated in the block diagram in Figure 1.

A. Standard search

In order to allow testing a given key regeneration system based on iris both I_{ref} and I_{sam} are used for each user. Furthermore, for each stored image, either a reference image or a test image, the data base stores 20 rotated versions of that image, i.e., a total of 21 images. We denote by $I_{\mathrm{ref}}(r, i, u)$ the r^{th} rotated version of reference image number i, belonging to user u, for $1 \leq r \leq 21$, $1 \leq i \leq N$, and $1 \leq u \leq U$. Similarly, we write $I_{\mathrm{sam}}(r, j, u)$, $1 \leq j \leq M$ to number test images. We refer to each comparison of images as a test, and call the set of tests for all users an experiment.

For the systems in [2], [3], [4], the tests are performed for each user u by picking one reference image for $r = 11$, i.e. $I_{\mathrm{ref}}(11, i, u)$, and comparing it with up to 21 versions of a corresponding test image $I_{\mathrm{sam}}(r, j, u)$, $1 \leq r \leq 21$. If a positive identification occurs when testing image $I_{\mathrm{sam}}(r, j, u)$, then the test with image j stops with match acceptance. However, if no positive identification is reached for $1 \leq r \leq 21$, then an identification error for test image j is computed, and if $j < M$ then the test image $I_{\mathrm{sam}}(r, j + 1, u)$ is the next one to be compared with $I_{\mathrm{ref}}(11, i, u)$. When the value $j = M$ is reached, the tests with image $I_{\mathrm{ref}}(11, i, u)$ are concluded, reference image $I_{\mathrm{ref}}(11, i + 1, u)$ is then selected in order to continue the tests, which then proceed in a manner similar to what was done for reference image i. When all reference images for user u have been selected, i.e., when $i = N$ and there are no more test images for user u, i.e., when $j = M$, tests with user u are concluded, user $u + 1$ is then selected, together with the corresponding reference and test images in order to continue the tests. The experiment finishes when the tests with user $u = U$ are completed. Hereafter, we refer to this procedure as *standard search* which is illustrated in Figure 2.

We remark that in the standard search up to 21 rotations of each test image $I_{\mathrm{sam}}(r, j, u)$, $1 \leq r \leq 21, 1 \leq j \leq M$, are performed per reference image $I_{\mathrm{ref}}(11, i, u)$, $1 \leq i \leq N$, and no rotation of reference images are employed. Summarizing, in the standard search 21 rotations are performed for each one of M test images, for N reference images and for U users.

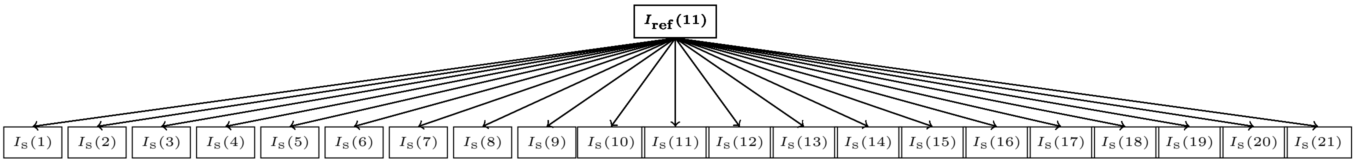

Fig. 2. Standard search technique, where $I_{\text{s}}(r) = I_{\text{sam}}(r)$, $1 \leq r \leq 21$.

B. Rotation search

The *rotation search* performs a search to identify a user by employing both, rotated reference images $I_{\text{ref}}(r, i, u)$ and rotated test images $I_{\text{sam}}(r, j, u)$. A sketch of the test performed by the rotation search is described next. Figure 3 illustrates, for a randomly selected user u^*, a few tests employing reference image i, $1 \leq r \leq 21$, i.e., $I_{\text{ref}}(r, i, u^*)$, for $r \in \{1, 6, 11, 16, 21\}$. In general, up to 441 tests for each test image can be performed to verify authenticity, by employing up to 21 distinct rotated versions of each reference image and up to 21 distinct rotated versions of each test image. Clearly many more situations are considered in a rotation search in comparison with a standard search and as a consequence there is an increase in the required processing time. We remark that each one of the data bases considered already in Section II contains all the rotated images required by the rotation search to perform the tests, i.e., no new data was necessary or was required by the proposed system.

The rotation search consists in systematically comparing a pair of images where one of them is either a reference image or one of its rotated versions $I_{\text{ref}}(r, i, u)$, and the other image in the pair is either a test image or one of its rotated versions $I_{\text{sam}}(r, j, u)$, where $1 \leq r \leq 21$, $1 \leq i \leq N$, $1 \leq j \leq M$ and $1 \leq u \leq U$. For example, for the BIOSECURE or CASIA data bases, an experiment employs $N = 10$ reference images per user, $M = 10$ test images per user and $U = 60$ distinct users. Thus a total of $10 \times 10 \times 60 = 6,000$ tests are performed, and for each test a worst case maximum of $r \times r = 21 \times 21 = 441$ verifications are performed when all rotated versions of both reference images and test images are required. In this manner the simulation time for rotation search is increased when rotated versions of a reference image need to be used.

V. Experiments and Results

In this section we compare the standard search and the rotation search, Section IV, in terms of their implementation efficiency. Furthermore, we analyze the performance of the proposed key regeneration system of Section III in terms of the values of FAR and FRR, and compare them to the results in [4].

A. Implementation efficiency

This paper is a follow up of the research reported in [2], [3], [4] with new contributions. The software employed in [2], [3] was developed using the proprietary programming language MATLAB, which comes with a series of pre-programmed mathematical operations and routines. For the development and implementation of the tests proposed here we noticed that the use of MATLAB was not the best choice as its data processing speed could not meet the speeds required. Consequently, the C++ programming language was employed because of the faster responses and reduction in the time to perform each test. The data in the available data bases is in the form of MATLAB compact tables, which do not allow a fast access by a software developed outside of MATLAB. For this reason, the iris codes in the data bases had to be converted to a new format which allows a more efficient access using C++. In this manner, 21 files were converted for each data base plus three control files for the NIST-ICE data base. The control files define which images belong to a certain user and who these users are.

The implementation efficiency of the rotation search is assessed by counting the number of comparisons per second and comparing it against the standard search method. We notice that the processing time is directly proportional to the number of decoding failures, which in turn grows with increasingly poor quality of user iris images. For any reference image or rotated reference image, $I_{\text{ref}}(r_1, i, u)$, the search procedure is halted as soon as a positive identification is found with a rotated test image $I_{\text{ref}}(r_2, j, u)$, or if all possible comparisons have been tried without success. The rotation search then selects another rotated version of the same reference image $I_{\text{ref}}(r_1', i, u)$ in order to perform comparison with rotated test images. After all rotated versions of a given reference image have been tested and there was no positive identification then an error is declared. The rotation system then selects another reference image to continue the search. When performing tests with the rotation search it, was observed that the least number of identification errors occurred for the data base NIST-ICE-exp1, followed by the data bases CASIA, BIOSECURE and NIST-ICE-exp2. The BIOSECURE is a regular data base which presents the worst results for the rotation search among regular data bases. For this reason, we have chosen the BIOSECURE data base for measuring implementation efficiency. The reason for not choosing the data base NIST-ICE-exp2 is because it is an irregular data base.

The results obtained for implementation efficiency using the standard search are presented in Table I, which shows the time in seconds necessary to perform the tests as well as the number of comparisons for 60 users, with 10 reference images and 10 test images, for $1 \leq t_{\text{RS}} \leq 22$. We considered the average time taken over three repetitions of each test. Observing the data in Table I we notice that the standard search runs roughly 20 times faster when implemented in C++ than its MATLAB implementation.

The way we implemented the software for rotation search allows access to all data in intermediary stages of a test. In

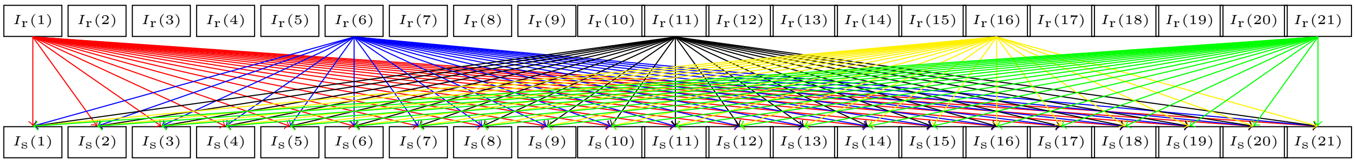

Fig. 3. Example, where $I_r(r) = I_{ref}(r)$, $1 \leq r \leq 21$, $I_s(r) = I_{sam}(r)$, $1 \leq r \leq 21$, for $I_{ref}(r) : r = 1; r = 6; r = 11; r = 16; r = 21$, in our proposed rotation search.

TABLE I
STANDARD SEARCH SIMULATION RESULTS FOR THE BIOSECURE DATA BASE USING MATLAB AND C++.

	MATLAB	C++
Time (s)	12,295	598
Comparisons	132,000	132,000
Comparisons/s	11	220

this manner we can identify which user images show more errors and subsequently identify which users were harder to be identified. We observed in the tests performed that most of the positive identifications occurred when we used $I_{ref}(r, i, u) = I_{ref}(11, i, u)$ reference images and compared them with *more central* test images like $I_{sam}(10, j, u)$, $I_{sam}(11, j, u)$ and $I_{sam}(12, j, u)$, not necessarily in this order. We took into consideration this observation in our implementation of the rotation search, by testing central images first and, if necessary, continue the test using images farther from the center ($r = 11$). As a result we produced Table II, where it is seen that the standard search became about 47 times faster then its MATLAB implementation.

TABLE II
COMPARATIVE RESULTS USING THE BIOSECURE DATA BASE, (*) DENOTES SIMULATION EMPLOYING CENTRALIZED SEARCH.

	Standard (MATLAB)	Standard* (C++)	Rotation* (C++)
Time (s)	12,295	253	1,936
Comparisons	132,000	132,000	132,000
Comparisons/s	11	521	68

Still referring to Table II, we observe that rotation search using C++ is about six times faster than the standard search using MATLAB. By analyzing test data we estimate that the rotation search would take approximately $4,500$ seconds if a centralized search is not employed, still 2.7 times faster than the standard search using MATLAB.

The *centralized search*, i.e., a rotation search which moves from central images to more peripheral images, obeys the following search sequence order: $I_{sam}(11, j, u)$, $I_{sam}(11-l, j, u)$, $I_{sam}(11+l, j, u)$, $1 \leq l \leq 10$. The same search sequence order employed in $I_{sam}(r, j, u)$ is employed in reference images $I_{ref}(r, i, u)$, $1 \leq r \leq 21$. This change in the search order reduced the image identification processing time by a factor greater than two. On the other hand, the standard search implemented in MATLAB selects test images $I_{sam}(r, j, u)$, $1 \leq r \leq 21$, in increasing order, beginning with $I_{sam}(1, j, u)$ and ending with $I_{sam}(21, j, u)$.

B. *Performance of the proposed system*

Table III presents three columns for each data base. The first column of each data base contains the result obtained with our C++ software, running the standard search as implemented in [3], and serves as a reference for comparison between our results and those obtained in [4]. The second column of each data base contains the results obtained in [4], and the third column of each data base contains the results obtained with our new proposed rotation search.

Table III shows that in all tests performed, for various values of t_{RS} and for all data bases considered, the rotation search consistently shows better results then the current search procedure. We emphasize that for all values of t_{RS}, FAR is always zero using the rotation search test while for the system in [4], FAR is greater than zero for $t_{RS} \geq 10$.

Figure 4 illustrates the situation where no user separation technique is employed and Figure 5 illustrates the user separation resulting from the application of shuffling key (k_{shuf}), random numbers and rotation search. We notice that Figure 5 is very similar to the one obtained in [3], which means that the inclusion of a shuffling key in our scheme did not significantly contribute for user separation. It should be emphasized that the significant reduction in FRR obtained here, in comparison with the system in [4], is due essentially to the rotation search in combination with error-correction.

VI. CONCLUSIONS

By employing rotated reference images of the iris we developed a rotation search that is far more efficient then the approach in [4] using a single iris for positive user identification and cryptographic key reconstruction. It is now possible to recover cryptographic keys with 198 bits with measured FRR of 0.24% and FAR of 0.00% for NIST-ICE-exp1. The increase in processing time for implementing rotation search remains smaller than the processing time for standard test employing MATLAB. Our ongoing research on identification systems employing the iris of both eyes of a user already indicates substantial improvements.

We observe that the rotation search can be implemented as an upgrade in most iris identification systems with some minor changes. Clearly, one can first perform a standard search, which can be followed by a rotation search if necessary. In other words, simulation results using rotation search are always at least as good as those based on a standard search. In the experiments performed with the data bases indicated earlier, rotation search always achieved the best results.

TABLE III

PERCENT FRR FOR THE STANDARD, KANADE ET AL. [4] AND ROTATION SEARCH TEST. FAR IS ALWAYS ZERO FOR THE ROTATION SEARCH TEST.

	BIOSECURE V1			CASIA V2			ICE-exp1			ICE-exp2		
t_{RS}	Standard	Kanade	**Rotation**	Standard	Kanade	**Rotation**	Standard	Kanade	**Rotation**	Standard	Kanade	**Rotation**
1	30.79	30.53	**15.15**	50.19	49.70	**23.11**	48.79	49.39	**21.37**	52.01	52.99	**24.34**
2	22.15	22.12	**11.04**	36.10	35.78	**15.09**	34.30	33.26	**13.75**	37.37	37.74	**16.17**
3	16.52	16.37	**8.63**	26.08	26.27	**10.82**	23.97	24.26	**9.53**	27.12	25.78	**11.44**
4	13.12	12.88	**7.49**	18.99	19.25	**7.81**	17.12	16.50	**6.97**	19.95	20.10	**8.73**
5	10.75	10.65	**6.63**	14.55	14.82	**5.85**	12.60	12.67	**5.23**	15.10	16.25	**6.59**
6	9.32	8.98	**5.99**	11.46	11.70	**4.42**	9.34	10.31	**3.83**	11.65	11.81	**4.89**
7	8.34	8.35	**5.39**	9.13	9.52	**3.29**	7.13	7.29	**2.79**	9.27	9.42	**3.69**
8	7.46	7.27	**4.74**	7.38	7.32	**2.49**	5.50	5.93	**2.15**	7.19	7.77	**2.69**
9	6.71	6.60	**4.06**	5.79	5.97	**1.69**	4.27	4.61	**1.56**	5.73	6.26	**1.93**
10	6.08	5.87	**3.40**	4.65	4.85	**1.09**	3.32	3.63	**1.16**	4.42	4.54	**1.38**
11	5.35	5.28	**2.74**	3.75	3.77	**0.64**	2.48	2.48	**0.83**	3.40	3.49	**0.91**
12	4.68	4.57	**2.00**	2.83	3.13	**0.28**	1.82	2.13	**0.62**	2.57	3.05	**0.63**
13	4.04	3.97	**1.59**	2.13	2.12	**0.14**	1.38	1.46	**0.39**	1.88	2.12	**0.46**
14	3.25	3.25	**0.95**	1.49	1.57	**0.03**	1.06	1.04	**0.24**	1.41	1.41	**0.35**
15	2.54	2.67	**0.60**	1.00	1.07	**0.01**	0.80	0.76	**0.16**	1.08	1.09	**0.26**
16	1.98	2.00	**0.36**	0.60	0.63	**0.00**	0.58	0.69	**0.12**	0.77	0.94	**0.18**
17	1.41	1.43	**0.19**	0.39	0.30	**0.00**	0.46	0.47	**0.08**	0.55	0.61	**0.10**
18	0.97	1.00	**0.06**	0.20	0.25	**0.00**	0.29	0.38	**0.05**	0.39	0.46	**0.08**
19	0.61	0.63	**0.04**	0.12	0.15	**0.00**	0.22	0.26	**0.04**	0.31	0.39	**0.03**
20	0.27	0.42	**0.00**	0.05	0.05	**0.00**	0.16	0.15	**0.03**	0.25	0.29	**0.01**
21	0.17	0.23	**0.00**	0.03	0.03	**0.00**	0.11	0.13	**0.02**	0.19	0.20	**0.01**
22	0.07	0.13	**0.00**	0.01	0.00	**0.00**	0.08	0.11	**0.01**	0.13	0.13	**0.01**

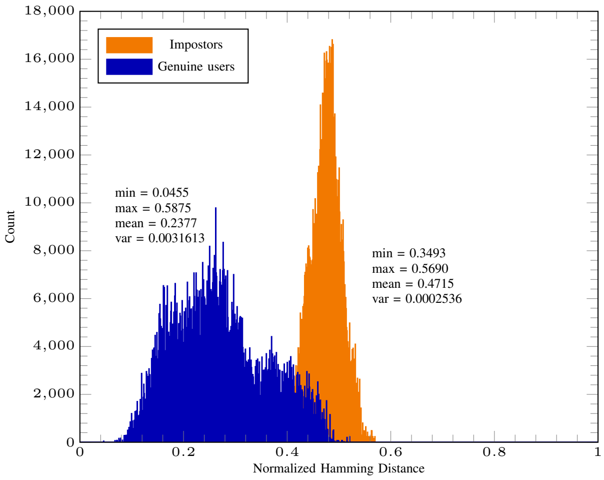

Fig. 4. Normalized Hamming distance for genuine users and impostors for Biosecure database using a single iris.

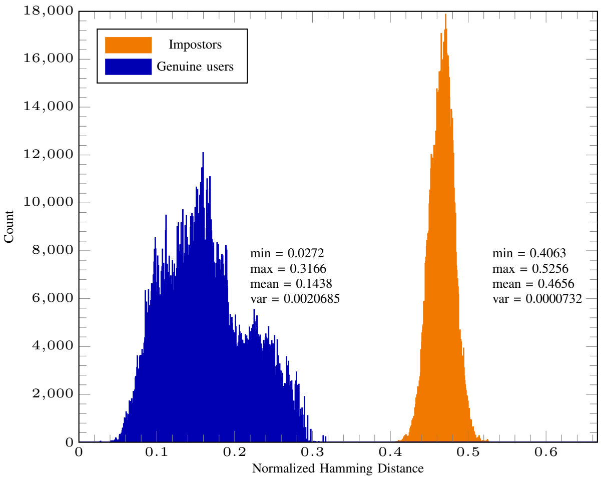

Fig. 5. Normalized Hamming distance for genuine users and impostors for Biosecure database using a single iris and employing shuffling key (k_{shuf}), random numbers and rotation search.

ACKNOWLEDGEMENTS

The first author is grateful to Dr. Danielle Camara, for his introduction to the field of biometrics, for supplying MATLAB simulation software and for explaining details of the iris data bases. The second author acknowledges partial support of this work by the Brazilian National Council for Scientific and Technological Development (CNPq), Project No. 304696/2010-2.

REFERENCES

[1] P. Tuyls, B. Skoric and T. Kevenaar, *Security with Noisy Data: On Private Biometrics, Secure Key Storage and Anti-Counterfeiting*, Springer, 2007.

[2] S. Kanade, D. P. B. A. Camara, E. Krichen, D. Petrovska-Delacrétaz and B. Dorizzi, "Three factor scheme for biometric-based cryptographic key regeneration using iris", *The 6th Biometrics Symposium 2008 (BSYM2008)*, pp. 59 - 64, Tampa, Florida, USA, 2008.

[3] D. P. B. A. Camara, J. S. Lemos-Neto, and V. C. da Rocha Jr., "Multi-instance Based Cryptographic Key Regeneration System", *JCIS*, vol. 29, No. 1, May 2014.

[4] S. Kanade, D. P. B. A. Camara, D. Petrovska-Delacrétaz and B. Dorizzi, "Application of biometrics to obtain high entropy cryptographic keys", *Proceedings of World Academy of Science, Engineering and Technology*, Vol. 27, pp. 251 - 255, Hong Kong, China, March 2009. http://www.waset.org/journals/waset/v27/v27-43.pdf

[5] "BioSecure Network of Excellence", www.biosecure.info.

[6] National Institute of Science and Technology (NIST), "Iris Challenge Evaluation", 2005, http://iris.nist.gov/ice.

[7] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North Holland, 1988.