Mitigating Iris-Based Replay Attacks

Joseph Shelton, Kaushik Roy, Brian O'Connor, and Gerry V. Dozier

***Abstract*—In this paper, we present an iris-based access control protocol that is resistant to iris-based replay attacks. This new style of biometric-based access control protocol is similar to the so called, 'one time password' approach used by some conventional username/password access protocols. Our results show that not only is this new type of iris-based access protocol effective, it can also distinguish between when a user attempting to gain access has supplied a poor sample of their iris and when the access control system is experiencing an iris-based replay attack.**

***Index Terms*—Biometrics, cyber security, genetic and evolutionary computation, iris recognition.**

I. Introduction

September 20, 2013 will be remembered as the day when biometric-based access control was introduced to the masses via the Apple iPhone 5s [1]-[3]. The iPhone 5s comes complete with two forms of user authentication: traditional and biometric-based [1]-[3]. For the traditional form of authentication, the user is asked to supply a password in order to unlock the phone. For the biometric-based form of authentication, the user is asked to supply a fingerprint [4], [5].

Within 48 hours of the release of the iPhone 5s, a number of techniques were envisaged that would be able to defeat the fingerprint sensor and the fingerprint identification algorithm [6]-[8]. The Electronic Frontier Foundation lists a number of additional techniques that claim to be able to defeat fingerprint sensors and the algorithms used for fingerprint identification [9], [10].

The common characteristic of many of the attacks on fingerprint authentication is that the underlying mechanism used to extract the salient features of ones fingerprint is deterministic [11], [12]. This means that if your biometric information is further being used to gain access to confidential information across the internet then you can fall victim to a biometric-based replay attack [13], [14]. In a biometric-based replay attack, a hacker may intercept packets sent across the internet that contain the biometric information belonging to someone else. The hacker can then resend these same packets at a later time to gain access to a confidential information disguised as the individual associated with the stolen biometric information [13], [14]. This is because most feature extractors used for biometric recognition are deterministic -- they always extract exactly the same features from a given image every time. The net effect of this is that once someone steals your biometric information you cannot change it without changing the physical you.

Non-deterministic feature extractor generators [13]-[15], on the other hand, can develop a number of different feature extractors that all have relatively the same recognition accuracy. A set of feature extractors can then be used to develop a biometric-based version of the so called 'One Time Password' Approach [14]-[16]. In this paper, we show how a system referred to as GEFE (Genetic & Evolutionary Feature Extraction) can be used to develop a number of different feature extractors in an effort to develop iris-based access control protocols. This technique can be extended to all forms of biometrics including fingerprint, face, ocular, etc.

The remainder of the paper is as follows. Section II provides a background of Genetic and Evolutionary Feature Extraction and its application towards mitigating replay attacks. In Section III we explain our experiments, in Section IV, we show our results and in Section V, we provide conclusions and future work.

II. Disposable Local Binary Pattern Feature Extraction

A. Local Binary Patterns

The Local Binary Pattern (LBP) feature extraction method is a technique proposed by Ojala *et al.* [17], [18]. This technique can be used to classify textures patterns in images and uses these textures to create feature vectors (FVs) with images. For facial recognition, the LBP technique works by segmenting the image into uniform sized, non-overlapping regions, as shown in Fig. 1. Each region has a histogram associated with it, where the bins in the histogram correspond to the texture patterns found in each region. A FV is created by concatenating the histograms from all regions of a segmented image.

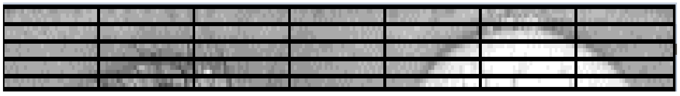

Fig. 1. Image partitioned into patches.

Texture patterns are created by comparing center pixels, a pixel that is surrounded by i number of neighboring pixels on all sides, with the i neighboring pixels. A texture pattern can be represented as a binary string, and that string can be decoded into a decimal value, denoted as LBP(N_i, c), where c

Manuscript received October 30, 2013; revised January 4, 2014. This work was supported in part by the National Science Foundation, the Army Research Laboratory, the Center for Advanced Studies in Identity Sciences, and the Science & Technology Center: Bio/Computational Evolution in Action Consortium.

Joseph Shelton, Kaushik Roy, Brian O'Connor, and Gerry V. Dozier are with the Computer Science Department, North Carolina Agricultural and Technical State University, Greensboro, NC 27405 USA (e-mail: jashelt1@aggies.ncat.edu, kroy@ncat.edu, bpoconno@aggies.ncat.edu, gvdozier@ncat.edu).

DOI: 10.7763/IJMLC.2014.V4.413

is the pixel intensity value of a center pixel, N is a set of neighboring pixel intensity values and i is the i^{th} neighboring pixel of c. LBP(N_i, c) is computed in (1) and (2), where the difference is taken between each neighboring pixel and a center pixel.

$$LBP(N_i,c)=\sum_{i=0}^{i-1} s\left(N_i,c\right)2^i \tag{1}$$

$$s\left(N_i,c\right)=\begin{cases}0, \text{if } N_i-c\le 0\\ 1, \text{if } N_i-c>0\end{cases} \tag{2}$$

Fig. 2 shows an example of a binary string being formed to be converted into an LBP texture pattern. The first matrix, Pixel Values, shows the intensity values of a center pixel and its neighboring pixels. The second matrix, Differences, shows the differences of neighboring pixels and the center pixel, and the third matrix shows the resulting binary string. The order in which the binary string starts to be decoded is user specified, so for Fig. 2, the pattern can start from the top-left corner, resulting in a texture pattern of 10011111, or a LBP value or 159.

120	90	111
119	113	113
224	198	201

Pixel Values

7	-23	-2
6	--	0
111	85	88

Differences

1	0	0
1	--	1
1	1	1

Pattern

Fig. 2. Center pixel resulting in 10111000.

The total number of texture patterns that can exist depend on the number of neighboring pixels, i, where the number of possible patterns are 2i. However, the common way to create FVs with the LBP technique is to use mostly uniform patterns for bins in the histograms. A uniform pattern is one where the bit transitions in a texture pattern changes two or fewer times when traversing the texture pattern circularly. For example, the bit string 10111000 has a total of four changes and is considered a non-uniform pattern. The changes are from the first to second bit; second to third; fifth to sixth; and eight back to the first. An example of a uniform bit pattern would be 10011111 with two changes from the first to second bit and the third to fourth bit. The number of uniform patterns would be $i\times(i-1)+2$, or 58 patterns if i = 8. In addition to the uniform patterns, one pattern is designated as a bin for all the non-uniform patterns found within a patch, giving a total of 59 bins for a histogram. In this case, the length of a FV would be $(i\times(i-1)+3)\times n)$, where n is the number of regions an image has been segmented into.

B. Local Binary Pattern Variations

The common variation of the LBP technique as presented above is popular; however, there are three types of modifications that can be applied to create different variations.

The first modification is to simply consider all of the possible patterns as opposed to just uniform patterns. In this case, histograms would be 2_i.

The second type of modification is to compare each neighboring pixel to the average of the neighboring intensity values as opposed to the center pixel. In this case, the LBP value would be computed using (3) and (4).

$$LBP(N_i)=\sum_{i=0}^{i-1} s\left(N_i\right)2^i \tag{3}$$

$$s\left(N_i\right)=\begin{cases}0, \text{if } N_i-\dfrac{\sum_{x=0}^{x=i-1}N_x}{i}\le 0\\ 1, \text{if } N_i-\dfrac{\sum_{x=0}^{x=i-1}N_x}{i}>0\end{cases} \tag{4}$$

Using these two modifications and looking at neighboring areas of size 8, we construct four LBP variations denoted below:

- LBP-59: This is the traditional LBP technique with 59 bins per region.
- LBP-a59: This is the traditional LBP technique that uses the second modifier of comparing neighboring pixels to the average of the neighborhood, represented by ‘a’.
- LBP-256: This is the LBP technique using the first modifier, or all 256 possible patterns/256 bins per histogram.
- LBP-a256: Similar to iii, with the second modifier.

We also consider the so called Modified Local Binary Pattern (mLBP) algorithm [19]. This algorithm is similar to LBP, except there is an addition. Not only is there a single bit being assigned after comparing a neighboring pixel to a center pixel, but an additional bit is assigned depending on whether the difference of the two pixels is less than or greater than/equal to the average of the difference of the neighborhood pixels and the center pixel. A ‘0’ bit is assigned if it’s less than, or a ‘1’ bit otherwise. So if there are i pixels in a neighborhood, the resulting bit string would have 2i bits. The resulting pattern is split evenly, and the two patterns are used to build two histograms for each patch. All histograms are ten concatenated together.

C. $GEFE_{ML}$

The Genetic and Evolutionary Feature Extraction (GEFE) technique is an instance of a Genetic and Evolutionary Computation (GEC) [20]-[22]. GEFE evolves a set of feature extractors (FEs) in an effort to increase recognition accuracy and reduce the number of required features. Initially, a population of randomly generated FEs is created. Each FE is then evaluated and assigned a fitness based on its accuracy on the training set. Next, parents are chosen from the population based on their fitness and are allowed to create offspring FEs. The offspring are each assigned a fitness and typically replace weaker members of the previous population. This evolutionary process of selecting parents, allowing them to procreate, and replacing weaker members of the population with the newly formed offspring is repeated until a user-specified stopping condition is reached.

An FE, fe_i, can be represented as a six-tuple, $<X_i,Y_i,W_i,H_i,M_i,f_i>$, where $X_i=\{x_{i,0}, x_{i,1},\ldots, x_{i,p-1}\}$ and $Y_i=\{y_{i,0}, y_{i,1}, \ldots, y_{i,p-1}\}$ represents the x-coordinates and the y-coordinates of the center of the *n* possible patches. The widths and heights of the *n* patches are represented by W_i and H_i. Each patch could have its own dimensions, W_i and H_i, but previous experimental results have shown that uniform sized patches are superior to non-uniform patches [23], [24]. Each patch has a masking value, $M_i = \{m_{i,0}, m_{i,1}, \ldots, m_{i,n-1}\}$, that determines whether the features extracted from a patch will be used in matching. The variable f_i represents the fitness of fe_i, which is determined by the number of incorrect matches that an FE obtained on a dataset as well as the percentage of patches that are not masked out.

Originally, GEFE was designed to optimize FEs on a training set. However, Genetic and Evolutionary Feature Extraction – Machine Learning (GEFE_{ML}) is designed to evolve FEs that generalize to unseen subjects: subjects not contained in the training set. To prevent overfitting FEs on a training set during the evolutionary process, cross-validation is used. While offspring are applied to the training set to be evaluated, they were also applied to a mutually exclusive validation set which does not affect the evolutionary process. The offspring with the best performance on the validation dataset is recorded regardless of its performance on the training set.

D. Mitigating Replay Attacks

GEFE_{ML} is used to create a set of FEs to be used in an access control system, in place of a deterministic feature extractor [14]-[16]. When a user wants to gain access to a system, the user will provide their biometric sample and the system will select a FE to extract from the information to create a feature vector (FV). The FV is then transmitted across a network and is compared with the user's previous enrolled FVs. After the FV comparisons, the FE used by the system is disposed, which is why we refer to the set of FEs as disposable FEs.

There are two protocols that we are focused on: Protocol II assumes that a disposable FE will create a unique FV so only a FV needs to be passed along a network [14]. Protocol III permutes the order of histograms in a FV, allowing for a disposable FE can have a set of unique FVs associated with it by permuting the order of histograms [15]. Protocol III can be described as follows. If a FV consists of *m* histograms, then there are *m*! possible permutations of the histograms, thus the set of FVs that can be created has a cardinality of *m*!. In order for the FVs in the set to be unique, all histograms within a FV must be significantly different from each other.

III. Experiment

For our experiment, we took images from the CASIA Version 3 interval dataset [25]. We applied a Variational Level Set (VLS)-based localization algorithm, reported in [26], to find inner/outer boundary of the iris. We first evolved FEs using GEFE_{ML} and performed cross validation during the evolutionary process to record FEs that generalized well on the validation set. At the end of each run of GEFE_{ML}, the FE with the best performance on the training set, and the FE with the best performance on the validation set are recorded. From the CASIA dataset, a total of 249 subjects were used to create three datasets: the training set, consisting 149 subjects, the validation set, consisting of 50 subjects, and the test set, which consists of 50 subjects. All subjects had at least three images.

To measure the effectiveness of Protocol II, we recorded the similarity distances of FVs using Equation (5). The Normalized Manhattan Distance (NMD) is calculated by taking two FVs, f_i and f_j, and calculating the Manhattan distance of both. The Manhattan distance is then divided by the sum of the *l* max values of the features of each position, *z*. We record both the NMDs of FVs created by matching FEs and non-matching FEs. To measure the effectiveness of Protocol III, we measure the uniqueness of different permutated FVs created by FEs. We do this by measuring the collision of permuted FVs; a collision occurs when the NMD of two permuted FVs is greater than a user specified threshold value, γ.

$$NMD(f_i, f_j) = \frac{\sum_{z=0}^{l-1} | f_{i,z} - f_{j,z} |}{\sum_{z=0}^{l-1} \max(f_{i,z}, f_{j,z})} \tag{5}$$

IV. Results

GEFE_{ML} was run 30 times, resulting in 30 optimized FEs on the training set and 30 best performing FEs on the validation set. GEFE_{ML} was an instance of an Estimation of Distribution Algorithm (EDA) [27], with a population of 20 and the single best performing FE from a previous generation always survives to the subsequent generation. Table I shows the performance of the traditional LBP variations as well as the hybridized FEs to the test set. The column 'Method' denotes the LBP variation used and the GEFE variation used. For the GEFE variations, those with <trn> represent FEs optimized on the training set whereas those with <val> represent FEs that had the best performance on the validation set. The Test Set column has two sub-columns, 'Accuracy' and 'P' (Patches). Both of these columns denote the accuracy and number of patches for a LBP and GEFE_{ML} variation for their respective dataset. The 'Accuracy' column shows the recognition accuracy of the best FE from 30 runs (shown outside the parenthesis) and the average accuracy of all 30 runs (shown inside the parenthesis).

The results suggest that the LBP-256 and GEFE-256 variants have a better performance than the other GEFE variants. The results also show that all GEFE variants outperform all traditional LBP variants, in regards to accuracy and features. A statistical ANOVA test was run on the results of the GEFE variants and GEFE_{ML}-256, GEFE_{ML}-a256 and GEFE_{ML}-mLBP outperformed GEFE_{ML}-59 and GEFE_{ML}-a59 in regards to recognition accuracy. In regards to patches activated, there was a statistical significance among GEFE_{ML}-256 GEFE_{ML}-a256 and GEFE_{ML}-mLBP. The <trn>variants used fewer patches than their <val>variants, however there was no statistical significance among the <trn> variants. It appears as though <val>GEFE_{ML}-256 had the highest average recognition rate but <trn>GEFE_{ML}-256 had the single best performing FE. Shown in Fig. 3 are the Cumulative Match Characteristic (CMC) curves and shown in

Fig. 4 are the Receiver Operator Characteristic (ROC) curves for LBP-256 and the best FE from <trn> GEFE$_{ML}$-256. The CMC curve plots the rank accuracies of the methods on CASIA, while the ROC curve plots the true accept rate of subjects and the false accept rate of subjects. The CMC curves show a superior performance of GEFE$_{ML}$-256 compared to LBP-256. There is a distinction between performances when looking at the ROC curves, where GEFE$_{ML}$-256 is also superior.

TABLE I: PERFORMANCE ON TEST SET

METHOD	PERFORMANCE ON THE TEST SET	
	ACCURACY	P
LBP-59	94.57%	78
LBP-a59	93.51%	78
LBP-256	96.00%	78
LBP-a256	95.21%	78
mLBP	96.00%	78
<trn>GEFE$_{ML}$-59	98.77% (95.42%)	32.29
<trn>GEFE$_{ML}$-a59	98.77% (95.27%)	34.79
<trn>GEFE$_{ML}$-256	***99.64%*** (96.42%)	45.32
<trn>GEFE$_{ML}$-a256	99.12% (95.34%)	46.71
<trn>GEFE$_{ML}$-mLBP	99.64% (96.52%)	45.32
<val>GEFE$_{ML}$-59	98.77% (95.74%)	41.3
<val>GEFE$_{ML}$-a59	98.77% (95.64%)	41.3
<val>GEFE$_{ML}$-256	99.56% (96.49%)	52.99
<val>GEFE$_{ML}$-a256	99.56% (96.14%)	51.22
<val>GEFE$_{ML}$-mLBP	99.56% (95.26%)	52.63

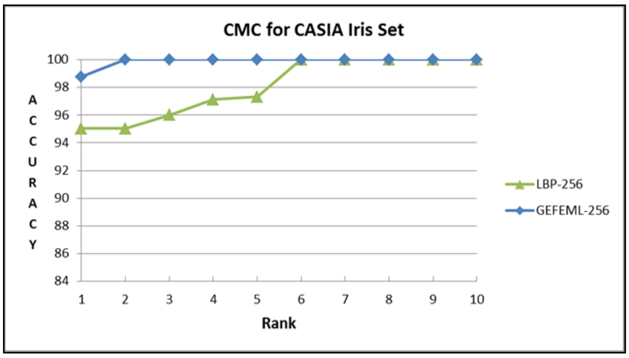

Fig. 3. Cumulative match characteristic (CMC) curve.

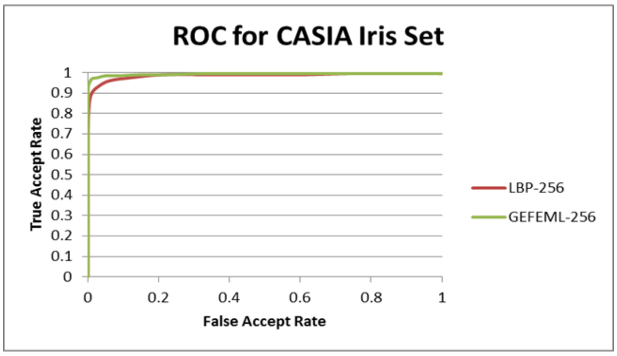

Fig. 4. Receiver operation characteristic (ROC) curve.

The disposable FEs were created with <trn> GEFE$_{ML}$-256. We choose this GEFE variant because it had the best performance. For each set of FEs, two sets of NMDs were recorded; the NMDs of FVs created by the same FE (labeled Same FEs), and the NMDs of FVs by different FEs (labeled Different FEs). Fig. 5 shows the distances of FVs of subjects from CASIA. The x axis represents the distance while the y axis represents the frequency that a distance occurs. In Fig. 5, the NMDs grouped to the left are the distances of FVs using the same FEs while the NMDs grouped to the right are those from different FEs. The figure show a clear separation between distances, meaning that a replay attack can be easily detected if a system has a threshold that falls between the groups of NMDs.

Fig. 6 shows the uniqueness of histograms within FVs of subjects from CASIA created by FEs from GEFE$_{ML}$-256 and LBP-256. The figure plots the collision rate of GEFE$_{ML}$ along a threshold value. Results show that collisions for disposable FEs occur later than for LBP-256, meaning the permutations of histograms within FVs created by the FEs from GEFE$_{ML}$-256 are more unique than LBP-256.

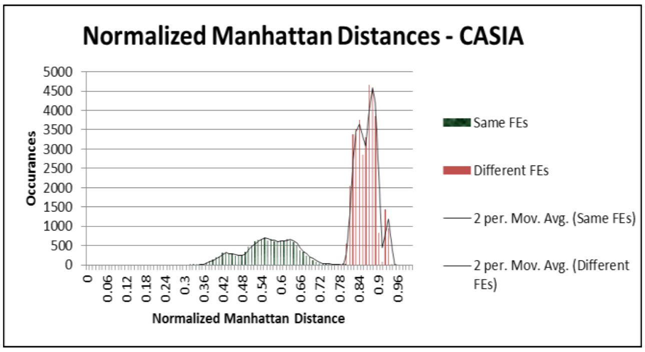

Fig. 5. Similarity scores of disposable FVs.

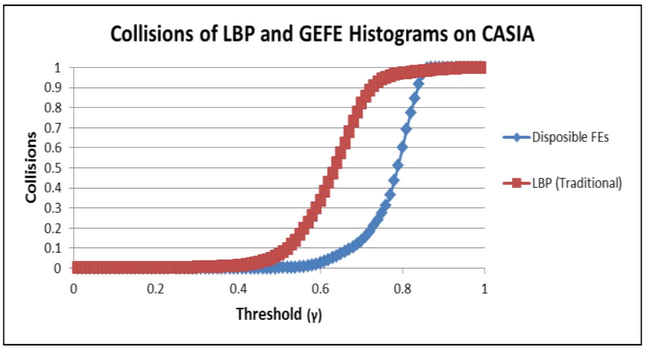

Fig. 6. Collision rate of disposable FVs.

V. CONCLUSION AND FUTURE WORK

In this paper, we show not only that GEFEML can evolve feature extractors that have improved recognition accuracy, but that disposable FEs can also be used to mitigate replay attacks on an iris based access control system. Evolved FEs use far fewer patches than traditional LBP yet the collision rates for traditional LBP rises faster than the rate for disposable FEs.

In the future, we seek to hybridize genetic and evolutionary computation with other forms of FEs traditionally used for iris recognition. We also seek to apply the concept of disposable FEs to other biometric modalities in an effort to make access control systems based on those modalities resistant to replay attacks.

ACKNOWLEDGMENT

The authors would like to thank the ARL and the NSF for their support of this research.

REFERENCES

[1] T. Brewer and A. Bessette. (Sept. 17, 2013). iPhone 5s & iPhone 5c Arrive on Friday, September 20. *Apple Press Info. N.p.* Web. 23 Oct. 2013. [Online]. Available: http://www.apple.com/pr/library/2013/09/16iPhone-5s-iPhone-5c-Arrive-on-Friday-September-20.html

[2] A. Barr. (Sept. 10, 2013). New iPhone 5S will read your fingerprints. *USA TODAY N.p.* Web. 23 Oct. 2013. [Online]. Available: http://www.usatoday.com/story/tech/2013/09/07/iphone-finger-print/2777849/

[3] B. Molina. (Sept. 11, 2013). 10 questions about iPhone's fingerprint sensor. *USA TODAY. N.p.* Web. 23 Oct. 2013. [Online]. Available: http://www.usatoday.com/story/tech/personal/2013/09/11/questions-iphone-fingerprint-sensor/2797471/

[4] C. Mallenbaum. (Sept. 19, 2013). Giving the new iPhone more than the finger. *USA TODAY. N.p*. Web. 23 Oct. 2013. http://www.usatoday.com/story/news/usanow/2013/09/19/iphone-5s-fingerprint-scanner-touch-id/2837299/

[5] E. C. Baig. (Sept. 11, 2013). Baig: Fresh iPhone colors, fingerprint tech are cool; is it enough. *USA TODAY. N.p.* Web. 23 Oct. 2013. [Online]. Available: http://www.usatoday.com/story/tech/columnist/baig/2013/09/10/apple-iphone-ios7-first-impressions/2793065/

[6] J. Kopstein. (Sept. 23, 2013). Hackers Defeat iPhone Fingerprint Reader—but Should You Really Worry? *The Slate Group. N.p.* Web. 23 Oct. 2013. [Online]. Available: http://www.slate.com/blogs/future_tense/2013/09/23/chaos_computer_club_breaks_iphone_defeats_iphone_fingerprint_reader.html

[7] C. Authur. (Sept. 23, 2013). iPhone 5S fingerprint sensor hacked by Germany's Chaos Computer Club. *Guardian News. N.p.* Web. 23 Oct. 2013. [Online]. Available: http://www.theguardian.com/technology/2013/sep/22/apple-iphone-fingerprint-scanner-hacked

[8] W. Ross. (Sept. 13, 2013). Experts Say iPhone 5S Fingerprint Security Feature Can Be Hacked. *Designtechnica Corporation. N.p.* Web. 23 Oct. 2013. [Online]. Available: http://www.thedailybeast.com/articles/2013/09/13/experts-say-iphone-5s-fingerprint-security-feature-can-be-hacked.html

[9] R. Bowe. (July 12, 2012). Red Flag On Biometrics: Iris Scanners Can Be Tricked. *Electronic Frontier Foudation. N.p.* Web. 23 Oct. 2013. https://www.eff.org/deeplinks/2012/07/red-flag-biometrics-iris-scanner-vulnerability-revealed

[10] R. Bowe. (July 27, 2012). Growing Mistrust of India’s Biometric ID Scheme. *Electronic Frontier Foudation. N.p.* Web. 23 Oct. 2013. https://www.eff.org/deeplinks/2012/07/red-flag-biometrics-iris-scanner-vulnerability-revealed

[11] A. K. Jain and A. Kumar, "Biometrics of next generation: an overview,” in *Second Generation Biometrics*, E. Mordini and D. Tzovaras, Eds. Springer, 2011.

[12] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, 2nd ed., Springer Publishing Company, Incorporated. 2009.

[13] C. Roberts, “Biometric attack vectors and defences,” *Computers Security*, vol. 26, no. 1, pp. 14-25, 2007.

[14] J. Shelton, K. Bryant, S. Abrams *et al.*, “Genetic & evolutionary biometric security: disposable feature extractors for mitigating biometric replay attacks,” in *Proc. The 10th Annual Conference on Systems Engineering Research (CSER)*, 2012 .

[15] J. Shelton, G. Dozier, J. Adams, and A. Alford, “Permutation-based biometric authentication protocols for mitigating replay attacks,” in *Proc. the 2012 IEEE World Congress on Computational Intelligence*, 2012.

[16] J. Shelton, J. Adams, A. Alford *et al.*, “Mitigating Replay Attacks Using Darwinian-Based Feature Extraction,” in *Proc. the IEEE Symposium on Computational Intelligence for Security and Defence Applications (CISDA)*, 2012.

[17] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987.

[18] T. Ahonen, A. Hadid, and M. Pietikainen, “Face description with local binary patterns: application to face recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037-2041.

[19] F. Ahmed, E. Hossain, A. S. M. H. Bari, and Md. S. Hossen, “Compound local binary pattern (clbp) for rotation invariant texture classification,” *International Journal of Computer Applications (IJCA)*, vol. 33, no. 6, pp. 5-10, 2011.

[20] A. Abraham, N. Nedjah, and L. Mourelle, “Evolutionary computation: from genetic algorithms to genetic programming,” *Studies in Computational Intelligence*, Springer, 2006, pp. 1-20.

[21] W. Ding and G. Marchionini, 1997, “A Study on Video Browsing Strategies,” Technical Report, University of Maryland at College Park.

[22] D. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, 2000.

[23] J. Shelton, G. Dozier, K. Bryant, L. Smalls, J. Adams, K. Popplewell, and K. Ricanek, “Comparison of genetic-based feature extraction methods for facial recognition,” in *Proc. Midwest Artificial Intelligence and Cognitive Science Conference*, April 2011, pp. 216.

[24] L. Davis, Handbook *of Genetic Algorithms*, New York: Van Nostrand Reinhold, 1991.

[25] CASIA-Iris Version 3 interval dataset. [Online]. Available: http://www.cbsr.ia.ac.cn/IrisDatabase.htm.

[26] K. Roy, P. Bhattacharya, and C. Y. Suen, “Iris segmentation using variational level set method,” *Optics and Lasers in Engg.*, vol. 49, no. 4, pp. 578–588, 2011.

[27] P. Larranaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Kluwer Academic Publishers, 2002.

Joseph Shelton is a doctoral student at North Carolina Agricultural and Technical State University, Greensboro, NC, USA, in the Computer Science Department. Joseph obtained both his bachelor’s degree and master’s degree in computer science at North Carolina A&T State University in 2010 and 2012.

He is currently working as a research assistant at North Carolina A&T and has done so for the last three years. He has helped publish a book chapter in ‘New Trends and Developments in Biometrics’ titled ‘Genetic and Evolutionary Biometrics’ and has published over 20 articles in the field of biometrics and genetic & evolutionary computations. Joseph’s research interests include biometrics, cyber security and evolutionary computation.

Mr. Joseph Shelton has received an award for 1st runner up for best student paper at the Conference on Systems Engineering Research (CSER 2012).

Kaushik Roy received his PhD from Concordia University, Montreal, QC, Canada in 2011 in computer science. He also completed his MS degree in computer science from the Concordia University in 2006 and B.Sc. degree in computer science from University of Rajshahi, Bangladesh in 2001.

Kaushik Roy is currently an assistant professor at the Department of Computer Science, and an assistant director of the Center for Advanced Studies in Identity Sciences (CASIS), North Carolina A&T State University, USA. Previously, he worked as a postdoctoral fellow in the Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada during 2011-2012. He also taught at Rajshahi University of Engineering and Technology (RUET) as a lecturer of the Department of Computer Science and Engineering during 2001-2004.

Dr. Kaushik Roy is also the recipient of several fellowships and awards including the prestigious NSERC Visiting Fellowship, FQRNT B3 (Postdoctoral), NSERC (Doctoral) and FQRNT B2 (Doctoral). His research interests include Biometrics, cyber identity, game theory, information fusion, computer vision, machine learning, and pattern recognition. He has published 1 book, 2 book chapters, 11 journal articles and 38 conference articles.

Brian O’Connor is an undergraduate student at North Carolina Agricultural and Technical State University, Greensboro, NC, USA, in the Computer Science Department. Brian is scheduled to graduate with his bachelor’s degree in computer science in spring 2014.

He is currently working as a research assistant at North Carolina A&T and has been in his position for a year. His research interest includes Cyber Identity, Biometrics and Machine Learning.

Gerry Vernon Dozier is a professor and the chair of the Computer Science Department at North Carolina A&T State University. He earned his Ph.D. from North Carolina State University. He is the director of the Center for Advanced Studies in Identity Sciences (CASIS), as well as the PI for the Center for Cyber Defense (recognized by the National Security Agency and the Department of Homeland Security as a Center for Academic Excellence in Information Assurance Education).

During Gerry's tenure as chair, the department has seen an increase in extramural funding and research publications as well as the establishment of a Ph.D. program. He has also lead in the development of an undergraduate research program where approximately 20% of the undergraduate students are active participants in funded research projects. Under Gerry's leadership, the NSF Alliance for the Advancement of African American Researchers in Computing (A4RC, www.a4rc.org) experienced a threefold increase (from 6 to 20) in the number of participating universities. A4RC was effective in increasing the number of African-American recipients of advanced degrees in Computer Science.

Dr. Gerry Vernon Dozier has published over 130 conference and journal publications. He has served as an Associate Editor of the IEEE Transactions on Evolutionary Computation and the International Journal of Automation & Soft Computing. Gerry is also a member of the Editorial Board for the International Journal of Intelligent Computing & Cybernetics. His research interests include: Artificial & Computational Intelligence, Genetic, Evolutionary, and Neural Computing, Biometrics, Identity Sciences, Cyber Identity, Distributed Constraint Reasoning, Artificial Immune Systems, Machine Learning and Network Intrusion Detection.