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.

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.](image-url)

Texture patterns are created by comparing center pixels, a pixel that is surrounded by i number of neighboring pixels on all sides, with the j 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, e), where c
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 \( c \). \( \text{LBP}(N_i, c) \) is computed in (1) and (2), where the difference is taken between each neighboring pixel and a center pixel.

\[
\text{LBP}(N_i, c) = \sum_{i=0}^{i-1} s(N_i, c) 2^i
\]  

(1)

\[
s(N_i, c) = \begin{cases} 
0, & \text{if } N_i - c \leq 0 \\
1, & \text{if } N_i - c > 0 
\end{cases}
\]  

(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.

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 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, \( f_e \), can be represented as a six-tuple, \( \langle X, Y, W, H, M, d \rangle \), where \( X = \{ x_{i,0}, x_{i,1}, \ldots, x_{i,p-1} \} \) and \( Y = \{ y_{i,0}, y_{i,1}, \ldots, y_{i,p-1} \} \) 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 \( f_i \), 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 (GEFEmL) 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

GEFEmL 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 GEFEmL 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 GEFEmL, 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 of 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 permuted 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, \( \gamma \).

\[
\text{NMD}(f_i, f_j) = \frac{\sum_{z=0}^{l-1} | f_{i,z} - f_{j,z} |}{\max(\sum_{z=0}^{l-1} f_{i,z}, \sum_{z=0}^{l-1} f_{j,z})}
\]

### IV. RESULTS

GEFEmL was run 30 times, resulting in 30 optimized FEs on the training set and 30 best performing FEs on the validation set. GEFEmL 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 GEFEmL 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 GEFEmL-256, GEFEmL-a256 and GEFEmL-a59 in regards to recognition accuracy. In regards to patches activated, there was a statistical significance among GEFEmL-256, GEFEmL-a256 and GEFEmL-a59 in regards to recognition accuracy. In regards to patches activated, there was a statistical significance among GEFEmL-256, GEFEmL-a256 and GEFEmL-a59 in regards to recognition accuracy. In regards to patches activated, there was a statistical significance among GEFEmL-256, GEFEmL-a256 and GEFEmL-a59 in regards to recognition accuracy.

The results 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 GEFE\textsubscript{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\textsubscript{ML}-256 compared to LBP-256. There is a distinction between performances when looking at the ROC curves, where GEFE\textsubscript{ML}-256 is also superior.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>PERFORMANCE ON THE TEST SET</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBP-59</td>
<td>94.5% 78</td>
</tr>
<tr>
<td>LBP-a59</td>
<td>93.5% 78</td>
</tr>
<tr>
<td>LBP-256</td>
<td>96.0% 78</td>
</tr>
<tr>
<td>LBP-a256</td>
<td>95.2% 78</td>
</tr>
<tr>
<td>mLBP</td>
<td>96.0% 78</td>
</tr>
<tr>
<td>&lt;trn&gt;GEFE\textsubscript{ML}-59</td>
<td>98.77% (95.42%) 32.29</td>
</tr>
<tr>
<td>&lt;trn&gt;GEFE\textsubscript{ML}-a59</td>
<td>98.77% (95.27%) 34.79</td>
</tr>
<tr>
<td>&lt;trn&gt;GEFE\textsubscript{ML}-256</td>
<td>99.64% (96.42%) 45.32</td>
</tr>
<tr>
<td>&lt;trn&gt;GEFE\textsubscript{ML}-a256</td>
<td>99.12% (95.34%) 46.71</td>
</tr>
<tr>
<td>&lt;trn&gt;GEFE\textsubscript{ML}-mLBP</td>
<td>99.64% (96.52%) 45.32</td>
</tr>
<tr>
<td>&lt;val&gt;GEFE\textsubscript{ML}-59</td>
<td>98.77% (95.74%) 41.3</td>
</tr>
<tr>
<td>&lt;val&gt;GEFE\textsubscript{ML}-a59</td>
<td>98.77% (95.64%) 41.3</td>
</tr>
<tr>
<td>&lt;val&gt;GEFE\textsubscript{ML}-256</td>
<td>99.56% (96.49%) 52.99</td>
</tr>
<tr>
<td>&lt;val&gt;GEFE\textsubscript{ML}-a256</td>
<td>99.56% (96.14%) 51.22</td>
</tr>
<tr>
<td>&lt;val&gt;GEFE\textsubscript{ML}-mLBP</td>
<td>99.56% (95.26%) 52.63</td>
</tr>
</tbody>
</table>

V. CONCLUSION AND FUTURE WORK

In this paper, we show not only that GEFE\textsubscript{ML} 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


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.

Joseph Shelton 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, 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.