Secure Iris Recognition

Christine Lu, Mentored By: Avinash L. Varna, Min Wu

***Abstract*—Biometrics is a major class of approaches for user authentication. This project uses iris recognition as a demonstrating example to explore techniques for building secure biometrics systems. After segmenting and identifying the region of the iris, unique and robust features of each person's iris are extracted and used to match with iris patterns in a database. In order to provide privacy protection and deter unauthorized use of sensitive biometric data, iris features should be properly encrypted by jointly employing cryptography, signal processing, and coding. This project examines two encryption techniques suitable for secure and robust biometric matching, and compares their performance.**

***Index Terms*—Cryptography, iris recognition, secure biometrics**

I. Introduction

BIOMETRICS is an approach to recognizing and identifying individuals based on intrinsic and unique physical traits. They are used today in numerous authentication scenarios, such as verifying passports and travel parties, controlling access to government buildings, and identifying perpetrators in criminal justice systems. The US Department of Defense has issued ID cards with digitized photographs and holograms with biometrics to its military personnel. Facial and fingerprint analysis and recognition are used internationally for a plethora of purposes, and can be found in many high security authentication scenarios. Due to the ubiquity of such technology, protection and security of an individuals' biometric data becomes valuable and attractive.

Though biometrics has certain benefits over classical user authentication systems, it does have disadvantages. When biometric data is compromised, the system cannot reissue a key, nor can the user easily cancel and change their physical traits. Therefore, a desirable feature for biometrics is to have matching performed in a secure way – one technique is to hash the sensitive information and authenticate users with their representative ciphertext within this encrypted realm, and to do so without saving their biometric data directly into a database or providing a separate decryption function. With no critical or meaningful information revealed to the system, and no way to recover the original information, there is additional layer of protection from malicious activity. As the biometric data is never saved, the information is hidden from untrustworthy system administrators.

Conventional encryption techniques use hashing – a one way encryption method where small changes in the input drastically alter the results of the output. For example, a single bit difference in the input can lead to hundreds of bits difference in the output. However, with the varying nature of biometrics, the recognition scheme needs to be able to tolerate small changes from one iris code to another to account for minute variations from the image or data capture source. Some examples of such differences in iris recognition would be head tilt, cyclovergence, image quality and resolution.

Currently, matches between two irises are determined by the hamming distance of the representative iris codes. However, this requires that the iris codes themselves be saved to be used for comparison. While this method is effective, it does not address the issue of information security. It is desirable to find a method of including hashing and encryption into the iris recognition system – modified to tolerate the variable nature of biometrics. In this paper, we explore two methods for achieving this so that secured iris information can be stored and compared to determine a match.

Prior work in this area focused on the ability to use the quantized and condensed representation of the iris information to match and identify different people, and explored a method that we will refer to as the XOR-ECC design. Techniques for searching secured images in a database, based on certain image properties were proposed in [2].

II. Securing the Iris

Small changes between different iris images reflect in small changes between iris codes – with conventional hashing, these differences would directly interfere with the ability to analyze the results. Since the imperfect nature of biometrics would be heavily reflected in the hashed results, the resultant bit string no longer precisely represents reproducible iris information, and thus loses its uniqueness. In the remainder of this section, we describe techniques to secure the iris information while allowing for robust authentication.

A. XOR-ECC Method

This method uses the idea that certain logical operations can be used to obfuscate the iris information with a randomly generated key. Using XOR, we can wrap and unwrap the key with the iris code, and because of the random nature of the

Manuscript received August 3, 2009. This material is based upon work supported by the National Science Foundation under Grant No. 0755224.

Christine Lu is with the Department of Computer Science and Engineering, Pennsylvania State University, College Park, PA 16802 USA (e-mail: cyl5041@ psu.edu).

Avinash Varna is with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: varna@ umd.edu).

Dr. Min Wu is with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: minwu@ umd.edu).

key, the security of the system is then ensured. However, since dissimilarities exist even between different pictures of the same iris, varying methods of error correction were used to recalculate the original key, which can be used for hashing and identification.

This resultant key, if not identical to the original key, will result in a different hashed result, and should be rejected. Since raw iris codes that are dramatically different indicate a rejection, the ability of the error correction code to overcompensate for mistakes can lead to false accepts, and if the error correction code is too weak, there may be other problems with correctly identifying the users.

In the first step of the process, we generate a random key and add redundancy through error correction code to provide tolerance to errors. Because we design this code to be the same length as the iris code, we can then XOR the redundant |key with the iris code computed from the iris scan, iris1, and store this XORed result as *securedIrisCode*. Given just the *securedIrisCode*, an attacker would not be able to gain any information about the original iris code or the original key, and the security of biometric data is ensured. To prevent an attacker from obtaining information regarding the original key, a cryptographic hash is applied to the key and the key is then securely erased. The hashed key and *securedIrisCode* are then stored on a physical device like a smart card.

When authenticating a user, the iris is scanned to obtain the iris code iris2. This is then XORed with the *securedIrisCode*. If the iris is from the same person, the resultant bit string would ideally return the original key error correction decoding. Since iris1 and iris2 are not identical, there would be errors in this result where the codes differ, which can be resolved with the error correction code. The hashed value of this resultant key is then compared to that of the original key to determine a match.

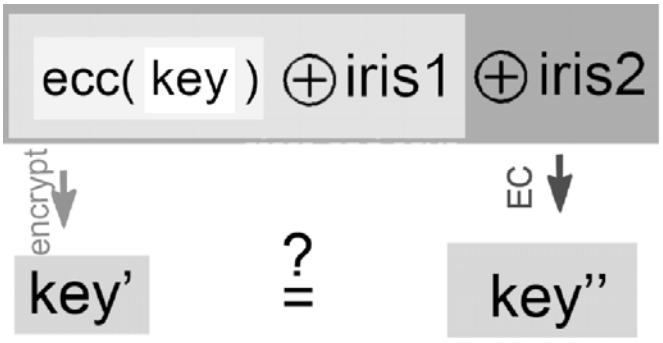

Fig. 1: Scheme for XOR-ECC method

Since the key in the XOR-ECC method will be hashed and used for comparison, the recomputed key after error correction has to be identical to the original key. Thus, the error correction used in this process has a heavy influence on the robustness of the system. If it is too strong, the error correction can overcompensate for mistakes, and will result in more false accepts. If it is too weak, it will be unable to deduce the original key, and will result in fewer correct accepts. Since error correction code can have such a dramatic effect on the determination of the outcome, we examined the influence of different error correction codes and their effectiveness in this method.

The first method we used was repetition, where the key was copied multiple times to fit the size of the iris code. Since the errors would not typically occur in bursts, the rounded average of these bits could then be effectively used to represent the original bit.

The second method used was a combination of Hadamard and Reed Solomon error correction. Hadamard error correction works well for random errors. To do this, we create an orthogonal Hadamard code H_c with 2^{k-1} columns and 2^k rows, which is constructed as shown.

$$H_1 = 1 \tag{1}$$

$$H_k = \begin{bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{bmatrix} \tag{2}$$

$$H(k) = \begin{bmatrix} H_k \\ -H_k \end{bmatrix} \tag{3}$$

The iris code is then split into groups of k bits – each group of k bits is then converted from binary to decimal and used to indicate a specific row of the Hadamard matrix. The -1s are then treated as 0s to allow for XOR. This new matrix, represented by various rows of the Hadamard matrix, is then used to XORed with the iris code.

When decoding, we find the dot product of each string of 2^{k-1} bits and every row of the Hadamard matrix. Since the Hadamard matrix is orthogonal, the row with the max value after the dot product should be the same row that was used to represent the code. Using this row, we can convert the decimal value back to the binary to recover the original key, k bits at a time.

Since Hadamard error correction code wasn't strong enough to completely recalculate the key, additional Reed Solomon codes were used since it is robust against burst errors that occur with obstruction of the iris by eyelashes, eyelids, etc.

B. Random projections

For this method, we used random vectors to hide the original values of the iris using random projection. Reducing the dimensionality of the iris information will still approximately preserve their distances, with which can be used to identify equivalent irises. This idea has been used before in secure image retrieval through feature protection [2] to search for encrypted images in a database. Because this method has been seen to be highly efficient at searching

large databases, we extended the same techniques to secure iris recognition.

For random projection, we used the iris information after the filters and transforms, representing this data as a vector **f** of size n. A random matrix is generated with dimensions sizeR = m x n, and used to hide the original iris information. To encrypt the iris information, we then store the product of this random matrix with the feature vector. Since these vectors are randomly generated, a malicious user would not be easily able to retrieve the original iris information.

III. Performance Analysis

Using an iris recognition program as a basis, we explored different options for secure biometrics. To do this, we used code by Libor Masek [3] that was inspired by Daugman's original research and implementation. In [4], iris information is extracted from a segmented picture of the eye (fig. 2), whereupon certain spatial frequency filters and transforms are applied to capture the texture and variations in the iris, then quantized to generate a sequence of bits known as the 'iris code'.

To test our algorithms, we used the CASIA Iris Image Database (ver 1.0) which has 756 iris images from 108 different people, taken on two separate days. After using Masek's code, we generated two objects: the iris code, a 20 x 480 = 9600 bit iris representation, and a mask of a similar size, which denotes regions of the iris where there are possible obstructions – like eyelids, eyelashes.

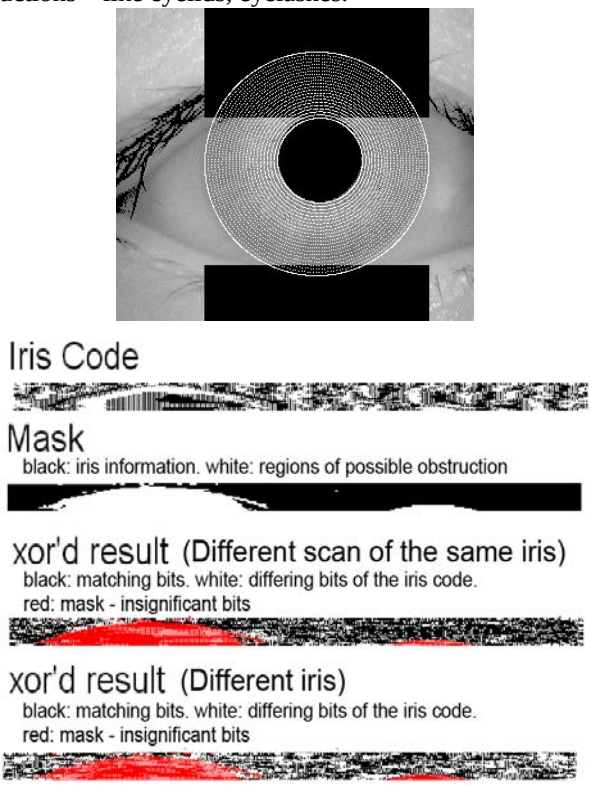

Fig. 2 shows a representative example of a segmented iris. Fig 2: Example of a segmented iris and mask

After filtering and quantizing this image, we get the following iris code and mask. The first XORed result shows a typical result when comparing two different pictures of the same iris, while the second shows what would be seen comparing two different irises. The increased whitespace in the second result indicates that there are more bits different.

Since the mask's bits are often quite similar from one to the next, we ultimately decided not to include it in the secure hashing, and work from the iris template alone. This meant that our percentage of error between iris codes rose dramatically. Another source of error comes from the matching of the iris code: in Masek's original implementation, head tilt and cyclovergence of the eyes are accounted for in the comparison stage, where the hamming distance between the two are measured. This is dealt with by shifting the bits on one template and having the bits wrap around. This template is shifted 8 bits in either original direction, and each one is compared to the constant template. The result is then taken to be the best match between these shifts.

Even with these errors, we found an average hamming distance between the same eyes to be 24.6%, with a range from 22-30.4%, while the average for two different eyes was 48%, ranging from 40-55%. With these errors, we decided to use k=10 for our Hadamard code, and a length of 12 for the Reed Solomon code. As the size of k increases, we had to use smaller and smaller keys – but each key could be recovered with greater accuracy. However with a smaller size, the complexity of the key has been compromised. As this size shrinks, less bits represent each individual iris, which leads to a tradeoff exists between the correct acceptance rate and the correct rejection rate.

Fig. 3: Comparison of different error correction techniques used for XOR-ECC

To evaluate our secure encryption techniques, we compared each iris picture to the other six images of the same iris, then one of those iris images to every other iris in the database.

Fig 3 shows a comparison of the two techniques using fixed key sizes of 100 for repetition, and 200 for Hadamard and Reed Solomon. We found that a 60% accuracy rate for

a single correct accept between two different pictures of the same iris could be achieved using our implementation of Hadamard and Reed Solomon error correction codes. If multiple images for a single iris were to be stored for matching and comparison, the percentage of correct acceptance could rise to 93.6% for correct acceptance. Additionally, since very few false acceptances were registered, these multiple entries stored will not have a significant effect on the overall results of the secure recognition program.

To test random projection, we had to consider several other variables: a correct acceptance is determined based on the distance of one iris vector to the other. Unlike the XOR-ECC method, where an exact match of the key is required for the system to work, using random projection, a limit can be dictated by the system so that all iris vectors with a distance greater than some number will be rejected, while all iris vectors with a distance less than that number will be accepted.

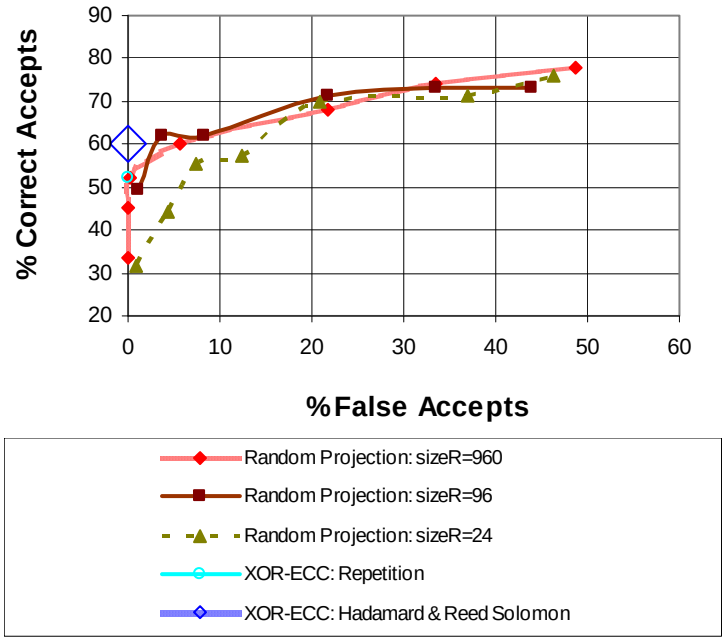

Fig. 4: Comparison of tested techniques

Fig 4 shows a comparison of the results with random projection and the XOR-ECC method. We found that as the size of the encryption matrix for random projection increased, we would get better results. For example, with an encryption matrix size of 960, a 0% false acceptance rate is possible, while this is not true for an encryption matrix size of 24. The problem with this is that as the size of the matrix increases, more information about the iris is revealed. A larger matrix, while leading to better results, would not necessarily be the best solution.

From these results, we determined that the XOR-ECC method had the best overall accuracy rate if Hadamard and Reed Solomon error correction codes are both used. However, while this method yields a greater accuracy rate, we found that our particular method of implementation of XOR-ECC with Hadamard and Reed Solomon ran an average of 31 times slower than that of random projection. This may mean that random projection would be better to use to search through a large database of irises, where a 10% correct acceptance margin is sacrificed for a faster search algorithm.

IV. Conclusion

This paper explores methods of secure biometrics that utilize secure encryption techniques to use for iris recognition. We examined the influence of error correction code on the XOR-ECC method of secure hashing, and further investigate the possibility of using random projection to obfuscate the iris information.

From our research, we found that using our methods of implementation, the XOR-ECC method using Hadamard and Reed Solomon error correction codes has the highest rate of correct accept (60%) and reject (100%).

However, we also found random projection to run faster, which may make it better for searches through a database.

Acknowledgment

Christine Lu thanks the National Science Foundation for funding this research, University of Maryland and the MERIT-BIEN program for the opportunities, and finally, Dr Min Wu and Avinash Varna for their tutelage, mentorship, and support.

References

[1] Feng Hao, Ross Anderson, John Daugman. (2005). “Combining cryptography with biometrics effectively,” Technical report, University of Cambridge.

[2] Wenjun Lu, Avinash L. Varna, Ashwin Swaminathan, Min Wu. “Secure Image Retrieval Through Feature Protection,” IEEE International Conference on Acoustics Speech and Signal Processing. 2009.

[3] Libor Masek, Peter Kovesi. *MATLAB Source Code for a Biometric Identification System Based on Iris Patterns*. The School of Computer Science and Software Engineering, University of Western Australia. 2003. [Online]. Available: http://www.csse.uwa.edu.au/~pk/studentprojects/libor/sourcecode.html

[4] Daugman, J. "High confidence visual recognition of persons by a test of statistical independence." ***IEEE Transactions on Pattern Analysis and Machine*** *Intelligence*, vol. 15(11), pp. 1148-1161, 1993.