FFR: Fragmented Face Recognition-Security Purpose

Lijimol K
Computer Science and Engineering (M.Tech)
HKBK College of Engineering
Bangalore, India,
lijikarun@gmail.com

***Abstract*: Secure facial identification systems compare an input face to a protected list of subjects. If this list were to be made public, there would be a severe privacy/confidentiality breach. A common approach to protect these lists of faces is to store a representation (descriptor or vector) of the face that is not directly map able to its original form. In this, we consider a recently developed secure identification system, Secure Computation of Face Identification (SCiFI) [1], A facial descriptor of this system does not allow for a complete reverse mapping. However, we show that if a malicious user is able to obtain a facial descriptor, it is possible that he/she can reconstruct an identifiable human face. In particular, we present 1) a cryptographic attack that allows a dishonest user to undetectably obtain a coded representation of faces, and 2) a visualization approach that exploits this breach, Whereas prior work considered security in the setting of honest inputs and protocol execution, the success of our approach underscores the risk posed by malicious adversaries to to-days automatic face recognition systems**

Keywords- Face recognition, Image reconstruction, cryptographic protocols, Data visualization

1. INTRODUCTION

Face recognition research has tremendous implications for surveillance and security, and in recent years the field has seen much progress in terms of representations, learning algorithms, and challenging new datasets [25, 16]. At the same time, automatic systems to recognize faces (and other biometrics) naturally raise privacy concerns. Not only do individuals captured in surveillance images sacrifice some privacy about their activities, but system implementation choices can also jeopardize privacy—for example, if the list of persons of interest on a face recognition system ought to remain confidential, but the system stores image exemplars. Recent work in security and computer vision explores how to simultaneously meet the privacy, efficiency, and robustness requirements in such problems [22]. While secure facial matching is theoretically feasible by combining any recognition algorithm with general techniques for secure computation [24, 8], these methods are typically too slow to be deployed in real-time. Thus, researchers have investigated ways to embed secure multiparty computation protocols into specific face recognition [7, 17, 15] and detection [1, 2] algorithms, noise resistant one-way hashes for biometric data [21, 5], revocable biometrics [4], and obscuring sensitive content in video [19, 3]. On the security side, much effort has also been put into improving the efficiency of general, secure two-party protocols [14, 10].

Figure 1. I present an attack on a secure face identification sys-tem using both cryptographic and computer vision tools. While the system ought to maintain the privacy of both the suspect list and passengers, our attack recovers coded versions of their faces and sketches human-understandable images from those codes.

In this work, we take on the role of a malicious adversary who intends to break the privacy of a secure face identification system. In doing so, we demonstrate how computer vision techniques can actually accentuate the impact of a successful attack. In particular, we examine the recently introduced Secure Computation of Face Identification (SCiFI) [15] approach. SCiFI is an elegant system that allows two mutually untrusting parties to securely compute whether their two respective input face images match. One compelling application of the system is for a client surveillance camera to test images against a set of images on a server [15]. The parties will want to learn if there are any matches, but nothing more. For example, imagine a watch list of suspected terrorists for an airport security system: the airport authorities should be able to submit face images of passengers as queries, and learn only if they are on the list or not. However, no one should be able to find out which individuals are on the list, nor should the database authority be able to create travel profiles of innocent parties. The SCiFI protocol meets the desired properties under the "honest-but-curious" model of security [15], where security is guaranteed if each party follows the protocol.

We investigate the consequences of a dishonest user that uses malformed inputs to attack the SCiFI protocol.1 Our work consists of two phases: a cryptographic attack phase and

a visualization phase. For the first phase, we show that by submitting an ill-formed input, an attacker can learn if a particular feature is present in a target image. By repeating this attack multiple times, an entire vector encoding the facial parts' appearance and layout of a target person can be recovered. While recovering the facial vector alone constitutes an attack, it is not necessarily usable by a human observer, since the result is a sparse set of patches with coarse layout. Thus, in the second phase, we show how to reconstruct an image of the underlying face via computer vision techniques. Specifically, we draw on ideas in subspace analysis [6, 11, 18, 9, 23] to infer parts of the face not explicitly available in the recovered facial encoding. The resulting image is roughly comparable to a police sketch of a suspect, visualizing the identity our attack discovered..

2. BACKGROUND: THE SCIFI SYSTEM

The SCiFI (Secure Computation of Face Identification) system [20] (which matches images taken by a client camera to a list of images (of potential suspects) which are held by a server.) is comprised of two parties, a client and a server. The server stores a list of faces and the client inputs a single face into the system. The goal of the system is to securely test whether the face input by the client is present in the server's list. The typical setting has the server's list comprised of faces of suspects or criminals, while the client inputs a face of a passer-by from a surveillance camera. The face acquired by the client might be from a person in the database; however, in general these faces will not match exactly. Thus, the SCiFI identification algorithm must be robust enough to match different photographs of the same person's face. In addition, SCiFI aims to do the matching computation while preserving the privacy of both the client and the server. This requires that neither the server nor the client learn any information. The only exception to this is that the server will learn if the client's input matches a face in the Server's list.

Face Representation Given a public database Y of face images, a standard set of p facial parts is extracted from each image (e.g., corners of the nose, mouth, eyes). For the i-th part, the system quantizes the associated image patches in Y to establish an appearance vocabulary $V^i = \{V_1^i, \ldots, V_N^i\}$ comprised of N prototypical examples ("visual words") for that part. Note there are p such vocabularies. In addition, each part has a corresponding spatial vocabulary $D^i = \{D_1^i, \ldots D_Q^i\}$ consisting of Q quantized distances of the feature from the center of the face.

Comparing Faces To compare two faces, SCiFI uses the symmetric difference between their two respective sets— that is, the number of elements which are in either of the sets and not in their intersection. The distance is computed separately for the appearance and spatial components, and then summed. If the total distance is under a given thresh-old, the two faces are considered a match.

Then, the full representation for a given face is the concatenation of all these vectors: $w = [w_1^a, \ldots, w_p^a, w_1^s, \ldots, w_p^s]$. In the following we refer to such a vector as a "face vector" or "facial code". This conversion is valuable because the Hamming distance can be computed securely using cryptographic algorithms, as we briefly review next.

Secure Protocol The input to the SCiFI (Secure Computation of face Identification)protocol is a single face vector w from the client and a list of M face vectors $w_1, \ldots, w_M$ and thresholds $t_1, \ldots, t_M$ from the server. Let H denote the Hamming distance. The output of the protocol is "match", if $H(w_i, w) < t_i$ for some i, and "no match" otherwise.

The client shares the public key with the server and keeps the private key to itself. Encryption is done over Z_m for some m = rq, where r and q are primes, while exploiting an exclusive-or implementation of the Hamming distance. Once the client has decrypted the server's message, an oblivious transfer protocol [13] is initiated. In short, both the client and server learn only if the Hamming distance between any pair of their vectors exceeds a threshold. See [15] for details, including novel optimizations that improve the efficiency.

3. CRYPTOGRAPHIC MALFORMED INPUT ATTACK

Cryptographic Malformed Input Attack The proposed attack on SCiFI allows the attacker to obtain a face code (w) that was meant to remain private. The attack relies on the fact that a dishonest adversary is able to input vectors of any form, not just vectors that are properly formatted.

Suppose the client's vector is w. A dishonest server can add any vector wm to its suspect list, and choose each corresponding threshold value, tm, arbitrarily. First, the server inputs the vector wm = [1, 0, . . . , 0], with a 1 in the first position and zero everywhere else. Next, the protocol comparing w and wm is run as usual. By learning whether a match was detected, the server actually learns information about the first bit, w1, of the client's input. We know that the nonzero entries of the input client vector must sum to exactly p(n+z). This creates two distinct possibilities in the outcome of the protocol:

- w1 = 1: In this case, the two input vectors will not differ in the first position. Therefore, they will only differ in the remaining p(n + z) − 1 positions where w is nonzero. Hence, we know that the Hamming distance between the two vectors is H(w,wm) = p(n+z)−1.
- w1 = 0: In this case, the two input vectors will differ in the first position. In addition, they will differ in all of the p(n + z) remaining places where w is nonzero. Hence, we know the H(w,wm) = p(n + z) + 1. Taking advantage of these two possible outcomes, the dishonest server can fix the threshold tm = p(n+z). Then, if a match is found, it must be the case that H(w,wm) = p(n + z) − 1 □ p(n + z), so w1 = 1. If a match is not found, then H(w,wm) = p(n + z) + 1 > p(n + z), so w1 = 0. Thus, the dishonest server can learn the first bit of the client's input.

Consequently, the attacker can learn the client's entire vector by creating l vectors wim , 1 □ i □ l, where the i-th bit is set to 1. We have portrayed the attack from the perspective of the server, where the server recovers facial codes for the

client. However, we can also adapt this attack for the client, in which case the client learns the confidential faces on the server.

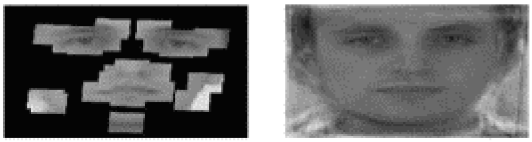


Figure 2. We first reconstruct the quantized patches based on the binary encoding (left), and then expand the reconstruction to hallucinate the full face given those patches (right).

4. FACIAL RECONSTRUCTION APPROACH

The cryptographic attack yields a binary vector encoding the appearance of some individual. However, the code itself is lossy (lost data and quality from the original version.) compared to the original image, and spatially it covers only about 40% of the face. Thus, we next propose an approach to form a human-interpretable visualization from the recovered binary encoding. The main idea is to first use the recovered indices of the most similar prototypical patches and spatial information for each facial part to render patches from the public vocabulary, placing each one according to the recovered approximate relative distance.

This yields a "patch face" that focuses on the key facial features. Given this patch face, we then estimate a full face image using a subspace reconstruction approach. This "hallucinated face" integrates both the back projected patches obtained from the attack as well as the learned statistics of faces in general. the two forms of reconstruction.

4.1. Offline Vocabulary and Subspace Learning

Before reconstructing any face, we must first perform two offline steps :(1) prepare the facial fragment "vocabularies", and (2) construct a generic face subspace. As in the original SCiFI system, the face images used to create the vocabularies come from an external (possibly public) database Y , which can be completely unrelated to the people enrolled in the recognition system. All faces are normalized to a canonical scale, and the positions of key landmark features (i.e., corners of the eyes) are aligned. Given these face images; we use an unsupervised clustering algorithm (k-means) to quantize image patches and distances to form the appearance and spatial vocabularies.

We also use Y to construct a generic face subspace. As has been long known in the face recognition community [20, 12], the space of all face images occupies a lower-dimensional subspace within the space of all images. This fact can be exploited to compute low-dimensional image representations. While often used to perform nearest-neighbor face recognition (e.g., the Eigen face (a large set of images depicting different human faces) approach [20]), we instead aim to exploit a face subspace in order to "hallucinate" the portions of a reconstructed face not covered by any of the p patches.

4.2. Patch Face Reconstruction

Now we can define the "patch face" reconstruction process. The cryptographic attack defined above yields the n selected appearance vocabulary words and z selected distance words, for each of the p facial parts. This encoding specifies the indices into the public vocabularies, revealing which prototypical appearances (and distances) were most similar to those that occurred in the original face.

Thus, we retrieve the corresponding quantized patches and distance values for each part, and map them into an image buffer. To reconstruct the appearance of a part i, we take the n quantized patches and simply average them, since the code does not reveal which among the n was the closest. We place the resulting average into the buffer relative to its center, displaced according to the direction and the amount given by the recovered quantized distance bin.

4.3. Full Face Reconstruction

The second stage of our approach estimates the remainder of the face image based on the constraints given by the initial patch face. While these regions are outside of the original SCiFI representation, we can exploit the structure in the generic face subspace to hypothesize values for the remaining pixels. Related uses of subspace methods have been explored for dealing with partially occluded images in face recognition—for example, to recognize a person wearing sunglasses, a hood, or some other strong occlusion [6, 11, 18, 9, 23]. In contrast, in our case, we specifically want to reconstruct portions of the face we know to be missing, with the end goal of better visualization for a human observer. We adapt a recursive PCA technique previously shown to compensate for an occluded eye region within an otherwise complete facial image [23]. The main idea is to initialize the result with our patch face, and then iteratively project into and reconstruct from the public face subspace, each time adjusting the face with our known patches. Relative to experiments in [23], our scenario makes substantially greater demands on the hallucination, since about 60% of the total face area has no initial information. Given a novel face x, we project it onto the top K eigenvectors to obtain its lower-dimensional coordinates in face space. Specifically, the i-th projection coordinate is:

Figure 3. Illustration of iterative PCA reconstruction. After initializing with the patch face reconstruction (leftmost image), we iteratively refine the estimate using successive projections onto the face subspace. Iterations shown are t = 0, 5, 100, 500, and 1000

5. RESULTS

The underlying goal of the experiments is to show that our reconstructed faces are recognizable and therefore compromise confidentiality. We test four aspects:

1. What do the reconstructed face images look like?
2. Quantitatively, how well do they approximate the appearance of the true (hidden) faces?
3. How easily can a machine vision system recognize the faces

we reconstruct?

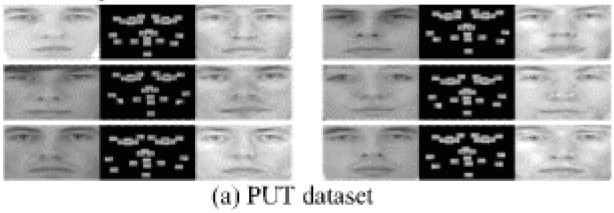
4. How well can a human viewer recognize the faces we reconstruct?

Experimental Setup We use two public datasets: the PUT Faces [14], which has p = 30 annotated landmarks, and a subset of Face Tracer [15], which consists of a highly diverse set of people and p = 10 landmarks (6 provided, 4 estimated by us). For both, we use only cropped frontal faces in order to be consistent with SCiFI. This left us with 83 total individuals and 205 images for PUT, and ⇠ 600 individuals and 701 images for Face Tracer. The PUT dataset is less diverse, but provides well aligned high-quality images that are good for building the face subspaces. In contrast, Face Tracer's diversity yields richer vocabularies, but is more challenging.

We use K = 194 eigenvectors based on analyzing the eigenvalues (An eigenvector of a square matrix is a non-zero vector that, when the matrix is multiplied by, yields a constant multiple of, the multiplier being commonly denoted by . That is: The number is called the eigenvalue of corresponding to) to capture 95% of the variance. Finally, we run the iterative PCA algorithm with ✏ = .0001 and a maximum of 2000 iterations. (We did not tune these values.) On average, it takes about 5 seconds to converge on a full reconstruction..

Qualitative Results: Example Reconstructions Figure 4 displays example reconstructions. We see that the reconstructed faces do form fairly representative sketches of the true underlying faces. We emphasize that the reconstructed image is computed directly from the encoding recovered with our cryptographic attack; our approach has no access to the original face images shown on the far left of each triplet. The fact that the full face reconstructions differ from instance to instance in the regions outside of the patch locations demonstrates that we are able to exploit the structure in the face subspace effectively; that is, the surrounding **content** depends on the appearance of the retrieved quantized patches.

We noticed that quality is poorer for the female faces in PUT. This is well-explained by that dataset's gender imbalance, where only 8 of the 83 individuals are female. This biases the face subspace to account more for the masculine



(a) PUT dataset

(b) Face Tracer dataset

Figure 4. Reconstruction examples from each dataset. Each triplet is comprised of the ground truth face, patch face, and our reconstructed face. Our reconstructed faces resemble the ground truth, and are much more easily interpretable than the sparse patch faces

Variations, and as a result, the reconstructed faces for a female's facial encoding tend to look more masculine. Nevertheless, we can see that the general structure of the internal features is reasonably preserved. Of course, in a real application one could easily ensure that the public set Y is more balanced by gender.

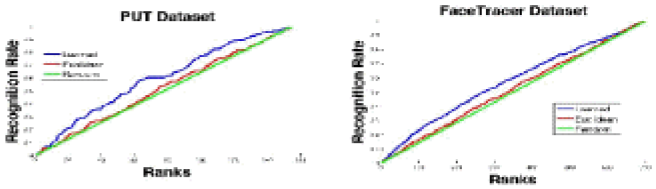


Figure 5. Machine recognition results. Curves show the recognition accuracy for a vision system that predicts the identity of our method's reconstructed faces

Machine Face Identification Experiment Next we test to what extent the reconstructions are machine-recognizable. In our setting, this corresponds to how well a computer vision system would be able to exploit the security breach to identify the individuals who were meant to remain private.

We input into the recognition system a reconstructed face and a database, T , of original face images. The original face associated with the reconstructed example is also in T (though unavailable to our algorithm). We have the system rank each database face from 1 to |T | according to its belief that the reconstructed image represents that person.

Figure 5 shows the results, comparing the learned distance approach to both a simpler Euclidean distance base-line as well as a random ranking. We plot the recognition rate as a function of rank—a standard metric in face identification. We see that the learned distance outperforms the baselines, showing the system benefits from learning how to associate the sketches with "real" images. More importantly, we see that the vision system can indeed automatically pinpoint the identity of the reconstructed facial codes with moderate accuracy.

Figure 6 Human subject experiment interface. The top row shows the reconstructed face (repeated 4 times). The task for the subject is to rank from 1 to 4 (1 being the best match) how close each face in the second row is to the first row.

Figure 7. Human subject test results. Boxplots show accuracy for 30 subjects on 30 test cases, compared to chance performance (green dashes) and machine recognition (purple dashes).

Human Subject Identification Experiment Finally, we examine how well human subjects can identify the people sketched by our method. We recruited 30 subjects—a mix of students and non-students, and none involved with this project. We generated a series of 30 test questions, each considering a different reconstruction result, and all using females from Face Tracer.

Figure 6shows a screenshot for an example question. We display the reconstructed face 4 times, and below it we display 4 real face images—one of which is the true under-lying face for that reconstruction. The subject must rank these choices according to their perceived nearness to the reconstructed face.

Figure 7 shows the results, in terms of the accuracy based on the first (left) or first two (right) guesses. The results are quite promising: while chance performance would be 25% and 50% for one and two attempts, respectively, the subjects have median accuracies of 41% and 62%. This plot also records the machine recognition accuracy on the same 30 tests using the learned metric defined above.

6. CONCLUSION

In this project we presented a novel attack on a secure face identification system that leverages insight from both security as well as computer vision techniques. While the SCiFI system appropriately claims security only under the honest but- curious model (and thus has no flaws in its claims), it limits of subspace-based reconstruction algorithms for visualization of severely occluded faces, face recognition accuracy can boosts using metric learning with synthetic sketch images and it analyze the performance of our system with two challenging datasets

References

[1] S. Avidan and M. Butman. Blind vision. In ECCV, 2006.
[2] S. Avidan and M. Butman. Efficient methods for privacy preserving face detection. In NIPS, 2006.
[3] T. Boult. PICO: Privacy through invertible cryptographic obscura-tion. In Wksp Comp Vis for Interactive and Intelligent Env, 2005.
[4] T. Boult. Robust distance measures for face recognition supporting revocable biometric tokens. In Face and Gesture, 2006.
[5] C. Chen, R. Veldhuis, T. Kevenaar, and A. Akkermans. Biometric binary string generation with detection rate optimized bit allocation. In CVPR Workshop on Biometrics, 2008.
[6] C. Du and G. Su. Eyeglasses removal from facial images. Pattern Recognition Letters, 2005.
[7] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft. Privacy preserving face recognition. In PETS, 2009.
[8] O. Goldreich, S. Micali, and A. Wigderson. How to prove all np-statements in zero-knowledge, and a methodology of crpytographic protocol design. In CRYPTO, 1986.
[9] B.-W. Hwang and S.-W. Lee. Reconstruction of partially damaged face images based on morphable face model. PAMI, 25(3), 2003.
[10] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. Extending oblivious trasnfer efficiently. In CRYPTO, 2003.
[11] A. Lanitis. Person identification from heavily occluded face images. In ACM Symposium on Applied Computing, 2004.
[12] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. PAMI, 24(6):780–788, June 2002.
[13] M. Naor and B. Pinkas. Oblivious transfer and polynomial evalua-tion. In STOC, 1999.
[14] M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In SODA, 2001.
[15] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich. SCiFI - a system for secure face identification. In IEEE Symp on Security and Privacy, 2010.
[16] P. Phillips, P. Flynn, W. Scruggs, K. Bowyer, J. Chang, K. Hoffman,J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In CVPR, 2005.
[17] A. Sadeghi, T. Schneider, and I. Wehrenberg. Efficient privacy-preserving face recognition. In Intl Conf on Information Security and Cryptology, 2009.
[18] Y. Saito, Y. Kenmochi, and K. Kotani. Estimation of eyeglassless facial images using principal component analysis. In ICIP, 1999.
[19] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Y.-L. Tian, A. Ekin, J. Connell, C. Shu, and M. Lu. Enabling video privacy through com-puter vision. IEEE Security and Privacy, 3(3):50–57, 2005.
[20] M. Turk and A. Pentland. Face recognition using eigenfaces. In CVPR, 1992.
[21] P. Tuyls and J. Goseling. Capacity and examples of template-protecting biometric authentication systems. In ECCV Workshop on BioAW, 2004.
[22] U. Uludag, S. Pankanti, S. Prabhakar, and A. Jain. Biometric cryptosystems: Issues and challenges. Proceedings of the IEEE, 92(6):948–960, 2004.
[23] Z. Wang and J. Tao. Reconstruction of partially occluded face by fast recursive PCA. In Intl Conf on Comp Intell and Security, 2007.
[24] A. Yao. Protocols for secure computations. In FOCS, 1982.
[25] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. Face recogni-tion: A literature survey. ACM Comp Surveys, 35(4):399–458, 2003.