CERIAS Tech Report 2004-99
Private Fingerprint Verification without Local Storage
by Mikhail J. Atallah
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Private Fingerprint Verification without Local Storage

Florian Kerschbaum[1], Mikhail J. Atallah[2], David M'Raïhi[1], and John R. Rice[2]

[1] Arxan Technologies
720 Market Street
San Francisco, CA 94102
fkerschbaum,dmraihi@arxan.com
[2] Dept. of Computer Science
Purdue University
West Lafayette, IN 47906
mja,jrr@cs.purdue.edu

Abstract. We describe a protocol to solve the problem of comparing fingerprints without actually exchanging them. We extend this to private verification where the verifier does not obtain fingerprint information. We show that no substantial information is leaked to the other party even during multiple protocol runs and present evaluation results from a prototype implementation.

1 Introduction

There are specific security concerns regarding fingerprint systems [11], but fingerprints have distinct advantages, because they cannot be lost or forgotten and they can be measured cheaply and non-intrusively. The disadvantages of fingerprints are that they cannot be changed or re-issued. Their disclosure is permanent. Large databases linking fingerprints to private information exist. It is therefore beneficial to keep fingerprints private.

We present a protocol where two parties each have a fingerprint image. They want to compare them for a match, i.e. that both images are from the same finger, but they do not want to reveal them, unless they are a match. We extend this a protocol where one party does not learn anything about the fingerprint, but can use it to verify the other's party identity.

Fingerprints are approximate, i.e. small changes happen every time one measures it. Cryptographic methods, such as encryption or hashing are not distance-preserving. Small input changes lead to outputs that cannot be compared for proximity any longer.

2 Related Work

We present work from three related areas of research. First, we review literature combining cryptography and biometrics, second, fingerprint matching algorithms and third, secure multi-party protocols.

D. Zhang and A.K. Jain (Eds.): ICBA 2004, LNCS 3072, pp. 387–394, 2004.

2.1 Private Biometrics

[4] presents an algorithm combining error correcting codes and hash functions such that biometric templates can be compared using Hamming distance. [10] extends that algorithm, such that the biometric template is no longer treated as only the information part of the code, but as a corrupted code word itself. Our protocol provides fingerprint verification using hamming distance and homomorphic encryption.

[2] describes an algorithm to make biometric templates non-transferable. The biometric template is XOR-ed with a random number, or simply permutated before enrolling. Although this achieves non-transferability, it does not protect the template during comparison.

[1] introduces a very nice system combining fingerprints with cryptography. A one-way function is applied to each fingerprint image's Fourier transform that results in a bit pattern that is mapped into a cryptographic key. The comparison itself reveals the fingerprint, so it has to be done in a secure device.

[16] presents another technique to achieve the ability to re-issue biometrics. It disturbs the biometric image using a random image transformations generating a large number of templates, but it does not guarantee the user's biometric privacy, because it could be undone using external knowledge or important parts might not have been modified.

2.2 Fingerprint Matching

Many algorithms for electronic fingerprint matching have been developed, to list a few [9, 14, 17]. A common type of algorithm matches fingerprints based on minutiae. Several clever methods have been developed to extract minutiae from the fingerprint image, e.g. [15].

The set of minutiae is the fingerprint template. It contains all information necessary to participate in a fingerprint comparison protocol, but less than a fingerprint image.

Our implementation follows the extraction algorithms from [9]. We distinguish two types of minutiae – forks and endings – and extract the ridge orientation with basic texture extraction algorithm. We do not improve upon those algorithms, but use them in our protocol.

2.3 Secure Multi-Party Computation

General secure multi-party computation [7, 18] can construct protocols for any joint computation where the parties do not want to reveal their input. The resulting protocols are not optimized to efficiently solve specific problems.

Many specific protocols have been developed. We will mention only some closely related ones. [5] studies approximate matching for databases access. [6] examines the security of privately computing approximations.

In the next section we will give an overview of the building blocks of our protocol, then we will present the fingerprint comparison protocol, private fingerprint verification, its security and practical evaluation.

3 Assumptions and Model of Computation

In the Fingerprint Verification Protocol both parties, Alice and Bob, have a fingerprint image. We compose the Fingerprint Comparison Protocol from two building-block protocols. Then we show how to modify these protocols, such that Bob will not obtain fingerprint information, nevertheless can still verify Alice's identity using an enrolled template.

We use the honest-but-curious model for the communicating parties, i.e. we assume that the participating parties will follow the protocol, but compute information about the other party's secret data. Our security analysis shows that no substantial information is leaked to the other party in any part of the protocol even if it is run multiple times between the same parties.

Our performance results conclude that it is possible to match fingerprints using our protocol by evaluating the matching results from the prototype implementation on a generated fingerprint database.

4 Fingerprint Comparison Protocol

Each party starts with a fingerprint image. Minutiae extraction can be done without the other's party input. We denote the fingerprint template as $S_A = (a_1, ..., a_n)$ and $S_B = (b_1, ..., b_m)$ for Alice and Bob, respectively. Each minutia is a four tuple (x, y, t, ϕ), where x and y are x- and y- coordinates, t is the type and ϕ the angle of the ridge orientation at coordinate (x, y).

Fingerprint Alignment translates and rotates the two templates such that matching minutiae have approximately the same absolute location. Fingerprint Comparison counts the number of those minutiae with the approximate same absolute location as matches. The final score of our protocol is the number of matching minutiae.

4.1 Fingerprint Alignment

Our alignment protocol uses two minutiae P and Q to align the templates. These two minutiae will be rotated and translated into two common locations P' and Q'. The templates are aligned, if Alice and Bob have picked matching minutiae. Then most matching minutiae will have approximately the same absolute location.

Fingerprint Alignment Protocol

Input: Alice has a fingerprint template S_A and Bob has a fingerprint template S_B.
Output: Alice has an aligned fingerprint template S'_A, Bob has an aligned fingerprint template S'_B, such that if $a_i = (x_a, y_a, \phi_a, t_a)$ matches $b_j = (x_b, y_b, \phi_b, t_b)$ then $x_a \approx x_b, y_a \approx y_b, \phi_a \approx \phi_b$.

1. Alice chooses a random value r as a cryptographic hash of her fingerprint image I: $r = H(I)$,
2. Alice chooses a random permutation Γ_r based on r and permutes her n minutiae $S_A = (a_1, \ldots, a_n)$: $\boldsymbol{a'} = (a_{\Gamma_r(1)}, \ldots, a_{\Gamma_r(n)})$.
3. Alice forms $\lfloor \frac{n}{2} \rfloor$ element-distinct pairs $\boldsymbol{p} = (p_1, ..., p_{\lfloor \frac{n}{2} \rfloor})$ where $p_j = (a'_{2j-1}, a'_{2j})$ for $j = 1, \ldots, \lfloor \frac{n}{2} \rfloor$.
4. Alice randomly selects a pair $A_{r,r'} = (a_r, a_{r'})$ from $\boldsymbol{p}$.
5. Alice rotates and translates S_A into S'_A, such that:
   - the location of a_r is at the origin: $a_r = (0, 0, t_a, \phi_a)$;
   - the location of $a_{r'}$ is on the positive x-axis: $a_{r'} = (x'_a, 0, t'_a, \phi'_a), x'_a > 0$.
6. Alice sends the pair $A_{r,r'}$ to Bob.
7. For each $i \in \{1, \ldots m\}$ and each $j \in \{1 \ldots m\}$ Bob performs the following operations:
   (a) Bob forms the pair $B_{i,j} = (b_i, b_j)$.
   (b) Bob rotates and translates $B_{i,j}$, such that $b_i = (0, 0, t_b, \phi_b)$ and $b_j = (x'_b, 0, t'_b, \phi'_b)$.
   (c) If $t_a = t_b$ and $t'_a = t'_b$, Bob computes a score $s_{i,j}$ for the minutiae pair as $s_{i,j} = (x'_a - x'_b)^4 + (\phi_a - \phi_b)^2 + (\phi'_a - \phi'_b)^2$.
8. Bob picks the pair B_{min} with the minimum score and accordingly rotates and translates S_B into S'_B.

4.2 Fingerprint Comparison

The number of minutiae per aligned template varies between the two parties and the minutiae are not ordered, such that their indices restrict the number of possible matches. Alice and Bob perform the following steps before they proceed to compare their fingerprint templates.

Fingerprint Comparison Preparation

Input: An aligned template S'.
Output: A rasterized template S''.

1. Each aligned template S' will be divided into equal-sized squares $\boldsymbol{Squ} = (squ_1, \ldots, squ_\sigma)$.
2. Each square squ_i is mapped to a bit of S":

$$S''_i = \begin{cases} 0 \text{ if } squ_i \text{ does not contain any minutiae} \\ 1 \text{ if } squ_i \text{ contains at least one minutia} \end{cases}$$

Alice computes her rasterized template S''_A from S'_A and Bob computes S''_B from S'_B. The number of matching minutiae n_{match} can be computed from the Hamming distance d_{Ham} between S''_A and S''_B:

$$n_{match} = \frac{|S_A| + |S_B| - d_{Ham}(S''_A, S''_B)}{2}$$

The Hamming distance can computed privately using the semantically secure, homomorphic, public-key encryption scheme by [8]. For brevity of discussion details are left to the reader.

5 Private Fingerprint Verification

We present an extension of the protocol where Bob does not obtain fingerprint information. He stores a hidden template and uses it to verify Alice's identity. Alice only needs to remember a short PIN.

Enrollment Phase

Input: Alice has her fingerprint template S_A, the minutiae pairs $\boldsymbol{p}$ and PIN pin.
Output: Bob has an enrolled template for Alice.

1. Alice generates $r_1, \ldots, r_n$ where $r_i = H(pin, i)$.
2. For each $i \in 1, \ldots, n$ Alice performs the following steps:
   (a) Alice aligns S_A to p_i as in step 5 of the Fingerprint Alignment Protocol obtaining aligned template S'_i.
   (b) Alice prepares $\boldsymbol{v}_i$ from S'_i using the Fingerprint Comparison Preparation algorithm.
   (c) Alice computes the XOR of r_i and v_i: $h_i = r_i \oplus v_i$.
3. Alice sends $p_1, \ldots, p_n$ and $h_1, \ldots, h_n$ to Bob.

During verification Alice enters her PIN and has her fingerprint captured by a sensor. Bob sends the stored minutiae pair information to Alice and Alice aligns her fingerprint before she applies the hiding procedure. Then Bob and Alice engage in the Hamming distance protocol using the hidden bit vectors.

6 Evaluation

6.1 Privacy and Security

In this section we will show that for an attacker who is honestly participating in the protocol, it is not possible to reconstruct enough fingerprint information from his view of the protocol to successfully forge a matching template, even if the protocol is run multiple times.

In the Fingerprint Alignment Protocol one party sends pairs of minutiae to the other party. These pairs have been randomized by rotating and translating them. Therefore the information leaked is only their distance, the difference of their ridge orientation and their type. The protocol ensures that no two pairs have any minutiae in common. The Fingerprint Verification Protocol reveals the same pairs of minutiae to the identifying party each time preventing the combination of pairs from multiple runs. An attacker needs to guess the absolute position and ridge orientation of half of the minutiae.

The Fingerprint Comparison reveals the Hamming distance between two rasterized templates and the number of minutiae in the other party's template. An attacker can try to obtain a rasterized template by guessing the bit vector and

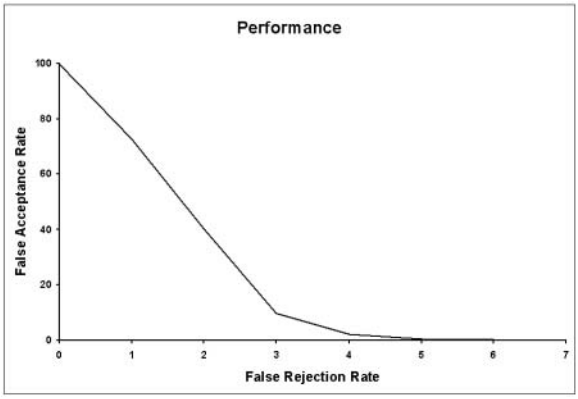

Fig. 1. Matching Performance

in the Fingerprint Verification Protocol also the user's PIN. Let l be the number of bits in the rasterized template, δ the maximum Hamming distance for a successful match and k the number of digits of the PIN, then the number of bits $b_{verification}$ he needs to guess is:

$$b_{verification} = \log_2 \frac{\binom{l}{n} \cdot 10^k}{\sum_{i=0}^{\delta} \binom{l}{i}}$$

For 36 minutiae, a 512-bit rasterized template, a maximum Hamming distance of 31 and a 5 digit PIN, this equals to about 46 bits.

6.2 Performance

We have implemented a prototype version of the comparison protocol in 2001. Since then results and data of performance competitions have been published [12, 13]. This section summarizes our performance evaluation showing the practicality of our matching algorithm.

We generated 108 fingerprint images – 27 "fingers" with 4 prints each – using an automated tool [3]. We used the minimum distance of 10 protocol runs. 1000 non-matching and 100 matching randomly selected pairs were privately compared.

Figure 1 shows our results. The false rejection rate (FRR) is the percentage of fingerprint pairs that have been detected as not matching although they were from the same finger. The false acceptance rate (FAR) is the percentage of pairs that have been incorrectly detected as a match. We varied the maximum Hamming distance to gather the plotted measurements. Our prototype performed closest to the equal error rate at nine matching minutiae for a match.

7 Conclusion

We have presented a Private Fingerprint Verification protocol analyzed its security and experimentally evaluated its matching performance. We plan to modify and extend this protocol and its implementation to address problems of active attacks, live fingerprint capture and comparing other biometrics.

References

[1] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. K. V. Kumar "Biometric Encryption" In ICSA Guide to Cryptography, eds. R. K. Nichols, McGraw-Hill Publishers, 1999 388

[2] J. L. Cambier, U. C. von Seelen, M. Braithwaite, R. Moore, R. Glass and I. Scott "Application-Dependent Biometric Templates" BC14, The Biometric Consortium Conference, 2002 388

[3] R. Cappelli, A. Erol, D. Maio and D. Maltoni "Synthetic Fingerprint-image Generation" ICPR 2000, Proceedings International Conference on Pattern Recognition, 2000 392

[4] G. I. Davida, B. J. Matt, R. Peralta and Y. Frankel "On the relation of error correction and cryptography to an off line biometric based identification scheme" WCC99, Workshop on Coding and Cryptography, 1999 388

[5] W. Du and M. J. Atallah "Protocols for Secure Remote Database Access with Approximate Matching" ACMCCS 2000, 7th ACM Conference on Computer and Communications Security, The First Workshop on Security and Privacy in E-Commerce, 2000 388

[6] J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. J. Strauss, and R. N. Wright "Secure Multiparty Computation of Approximations" (Extended Abstract) ICALP 2001, 28th International Colloquium on Automata, Languages and Programming, pp. 927-938, 2001 388

[7] O. Goldreich "Secure Multi-Party Computation" Manuscript, http://www.wisdom.weizmann.ac.il/ oded/pp.html, 2002 388

[8] S. Goldwasser and S. Micali "Probabilistic Encryption" Journal of Computer and System Sciences, Vol. 28(2), 1984 390

[9] A. K. Jain, L. Hong and R. Bolle "On-line Fingerprint Verification" IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19(4), pp. 302-314, 1997 388

[10] A. Juels and M. Wattenberg "A Fuzzy Commitment Scheme" CCS'99, Proc. of the 6th ACM Conference on Computer and Communications Security, 1999 388

[11] T. Matsumoto, H. Matsumoto, K. Yamada and S. Hoshino "Impact of Artificial "Gummy" Fingers on Fingerprint Systems" SPIE Vol #4677, Optical Security and Counterfeit Deterrence Techniques IV, 2002 387

[12] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman and A. K. Jain "FVC2000: Fingerprint Verification Competition" IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24(3), pp. 402-412, 2002 392

[13] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman and A. K. Jain "FVC2002: Second Fingerprint Verification Competition" ICPR 2002, Proc. of International Conference on Pattern Recognition, 2002 392

[14] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar "Handbook of Fingerprint Recognition" Springer Verlag, June 2003 388

[15] S. Prabhakar, A. K. Jain, and S. Pankanti "Learning Fingerprint Minutiae Location and Type" Pattern Recognition, Vol. 36, No. 8, pp. 1847-1857, 2003 388
[16] N. Ratha, J. Connell and R. Bolle "Enhancing security and privacy in biometrics-based authentication systems" IBM Systems Journal, Vol. 40(3), pp. 614-634, 2001 388
[17] A. Ross, A. K. Jain and J. Reisman "A Hybrid Fingerprint Matcher" Proc. of International Conference on Pattern Recognition (ICPR), Vol.3, pp. 795-798, 2002 388
[18] A. Yao "Protocols for Secure Computations" FOCS '82, Proc. of the Twenty-third IEEE Symposium on Foundations of Computer Science, pp. 160-164, 1982 388