Remote authentication using
Vaulted Fingerprint Verification

Hamdan Alzahrani, a Terrance E. Boult
University of Colorado, Colorado Springs, CO, USA

ABSTRACT

This paper presents a novel approach to remotely authenticating a user by applying the Vaulted Fingerprint Verification (VFV) protocol. It proposes an adaptation of the Vaulted Verification (VV) concept with fingerprint minutia triangle representation. Over the past decade, triangle features have been used in multiple fingerprint algorithms. Triangles are constructed from three fingerprint minutiae and result in a feature vector that is translation and rotation invariant. In VFV, the user’s minutia triangles are arranged into blocks; each block of triangles is paired with a chaff block. In turn, each real/chaff block is encrypted with a key that is only known to the users. These encrypted block pairs can be used to generate a “challenge” by swapping blocks according to a random bitstring and requiring the remote user to reproduce that exact string. For identity verification, the user creates a new triangle feature vector from his or her fingerprint. This feature vector is matched against each block, which allows the user to identify the “real” block in each pair and recover the bitstring. In this process, individual triangle matching rates are improved by approximate matching on the feature vectors, grouping several feature vectors together, and correcting errors on the final bitstring. This paper presents data on an optimal threshold for approximate matching, the accuracy of triangle matching, the distinguishability between a user’s triangle and a chaff triangle, and the accuracy of the VFV system.

Keywords: Vaulted Verification, biometrics, fingerprint, privacy.

1. INTRODUCTION

Currently, there is strong demand for identity assurance systems. Existing authentication systems determine a user’s identity through the possession of a token, generally a password or smartcard. Token-based identity transactions are relatively easy to repudiate since unauthorized persons may possess the token. A system that can guarantee a user’s presence during authentication would greatly enhance the non-reputability of these transactions. Biometrics can provide this strong link between users and their identities.

Different fingerprint-based remote user authentication schemes have been proposed in literature. Most of the proposed schemes depend heavily on the usage of smartcards. Fan et al. proposed an authentication scheme that preserves privacy. They encrypt the random strings using a biometric template. Other researchers apply a one-way hash function to the exchanged message. They also include the nonce in the exchanged message to ensure the integrity of the message.

The use of biometrics is becoming increasingly prevalent. With the advancement of automated authentication technology, privacy and security concerns about users’ biometric data templates are also becoming more apparent.

Biometric features are not practically changeable; a compromised template no longer uniquely identifies the target person. Therefore, protecting and securing the user’s biometric template is imperative. Various techniques have been proposed for biometric template protection. The survey by Jain et al. categorized them into transformation, biometric cryptosystems, and hybrid.

a Email Hamdan Alzahrani: halzahra@uccs.edu
Pre-print of a paper that will be presented at SPIE 2014
With this in mind, merely protecting a biometric template is not sufficient. One must ensure a user’s privacy by both preventing impersonation and tracking the user through multiple systems. While preserving privacy, the server needs to affirm the identity of the user and reject imposters.

Vaulted Verification (VV) is an algorithm that is capable of ensuring security and remotely authenticating a user while preserving privacy. The original VV was for face other research has extended VV to iris verification and to voice verification. In our research, we adapt VV to use fingerprint data.

In order to successfully match and authenticate the user, a template needs a clear, concise representation of the fingerprint. Fingerprints are composed of many structures. One type of structure is fingerprint minutiae, which are the points where ridges end or bifurcate. Minutiae are useful due to their ability to extract most of the relevant information from a fingerprint, but they can be vulnerable to translation and rotation. Local structures can be formed from these points, which are invariant to rigid transformations.

Many techniques have been used to represent fingerprint minutiae; in our own protocol, we have chosen a triangle representation. A minutia triangle representation is constructed from three fingerprint minutiae, which results in a feature vector that is translation and rotation invariant. Information extracted from this triangle can include the distance between the points, the angle between the sides, and the (invariant) orientation of the minutiae.

Triangle-feature-matching algorithms have been adopted widely in biometric systems. M. A. Medina-Pérez et al. proposed M3gl. The proposed approach arranges the minutiae in a clockwise direction, which eliminates the sensitivity to minutiae order. R. Germain et al. designed a system to index fingerprints using minutia triplets, which are then used for identification searches in a large database. P. Li et al. demonstrated the utility of minutia triangle representation and key generation by constructing a key directly from fingerprint minutiae.

Other techniques have been proposed in the literature for including more features in minutia triangles. A. C. Chau et al. incorporated texture quality in their matching algorithm. Other features, such as ridge count and having the same ridge, are included in the unprotected matching algorithm.

2. VAULTED VERIFICATION PROTOCOL

The VV protocol enables users to be verified remotely and preserves their privacy. This protocol requires enrollment and verification. It can also be used for a key exchange. The basic concept of VV is as follows:

2.1 Enrollment process

The user enrolls in the system by submitting the appropriate biometric modality to the client. The client then constructs feature vectors and groups them into multiple blocks called “real blocks.” Each real block is paired with a randomized chaff block. Each block is independently encrypted by the user’s encryption key. The entire template is then encrypted by the server’s encryption key.

The template becomes a group of encrypted real and chaff blocks. This allows the server to swap blocks and create a challenge response, which is later used to authenticate a potential user.

2.2 Verification process

The user needs to prove his identity in order to access the server. The server creates a random binary bitstring of length $N$, where $N$ is the number of block pairs in the template. The server permutes the blocks according to the bitstring created. If the bit is 1, then the blocks are swapped. If the bit is 0, no swapping is conducted. The blocks are encrypted and sent to the client. The client decrypts the received blocks and compares them against live input to identify the real blocks and ultimately recover the correct bitstring.
3. VAULTED FINGERPRINT VERIFICATION

Our approach uses Fuzzy Features Representation (FFR). As mentioned in section 1, the FFR system is based on minutia triplets. The triangles are extracted to vectors that can be compared for a match.

We have modified the FRR approach by extending it into a voting system that helps the client distinguish real blocks from chaff, which is necessary to recover the VFV bitstring. An overview of the enrollment process is illustrated in Figure 1.

3.1 Enrollment process

Figure 1. Enrollment process.

Fingerprint minutiae are the building blocks of VFV. We use Mindtct from the NIST toolset, which outputs the x and y coordinates for the minutia, the minutia orientation, and the quality value as illustrated in 1:

\[ M_i = \{x, y, \theta, q\} \]  

We eliminate low-quality minutiae and compute all possible combinations of three minutia points to construct a set of triangles. We compute the distance, interior angle, and invariant orientation of each triangle. The distances are ordered from largest to smallest. Invariant orientation is the difference between the minutia orientation and the bisector of the triangle through that point. However, not all of the triangles we construct are useful. We only accept triangles whose distances are between two threshold values. Overly long triangles are likely to have distorted distances.

The filtered triangles are then permuted randomly. This removes any relationship between minutia locations in the triangle set. These filtered, randomized triangles are grouped into blocks, which give us more control over accuracy at the expense of template size. Increasing block size increases accuracy but consumes more triangles per bit.
Chaff triangles are constructed in the same manner; the only difference is that the chaff is drawn from a separate pool of subjects.

Each real block is then paired with a chaff block. Individual blocks are encrypted with the user’s encryption key. Next, the entire block is encrypted with the server key. Then, the encrypted template is sent to the server for use in the verification process.

### 3.2 Verification process

A user initiates the identity verification process by requesting authentication from the server. The server decrypts the template using its key and creates a challenge bitstring of length $N$, where $N$ is the number of blocks in the saved template. The server then swaps pairs according to the generated bitstring. If the bitstring is 1, the pair is swapped; if it is 0, there is no change. The permuted template becomes encrypted again with the user’s public key, and it is sent to the client.

The client decrypts the template and matches it against a live input. Our matching algorithm consists of three parts: score generation, vote summation, and group comparison. A score function takes each triangle from the gallery set and probe set and outputs a score. We use Equation 2 to compare all triangles from gallery $T_g$ to all triangles of probe $T_{pr}$:

$$
\text{score} = \begin{cases} 
1 - |T_g - T_{pr}|/\text{Threshold}, & \text{if } |T_g - T_{pr}| < \text{Threshold} \\
0, & \text{otherwise}
\end{cases}
$$

The score is summed along the probe set dimension, resulting in a triangle vote vector. The triangle vote vector is summed along each block, which results in a Block Vote Vector (BVV).

The BVV allows us to compare the two gallery sets. The gallery set with the larger BVV is coded as a “real” block, and the other is coded as a “chaff” block. If the BVV is equal for both sets, we code the block as undetermined. When combined with the error correction parity bits, this allows us to recover the permutation bitstring. An overview of the verification process is illustrated in Figure 2.

![Figure 2. Verification process.](image-url)
4. EVALUATION

The proposed VFV protocol has been evaluated using data from the DB1-A of Fingerprint Verification Competition 2002 (FVC2002), which contains 8 images per person for 100 people. An implementation of VFV requires setting a quality threshold, maximum and minimum distance thresholds, block size, and template length, which are listed in Table 1.

Table 1. Triangle construction parameters.

<table>
<thead>
<tr>
<th>Minutia quality</th>
<th>Min. distance</th>
<th>Max. distance</th>
<th>Block size</th>
<th>Template length</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>10</td>
<td>150</td>
<td>8</td>
<td>255</td>
</tr>
</tbody>
</table>

The thresholds for the scoring function (2) were determined empirically. We performed a grid search over a training set of the threshold space to locate the values that maximized the distinguishability between the real and chaff triangles. The mean optimal thresholds and standard deviation are shown in Table 2.

Table 2. Optimal threshold.

<table>
<thead>
<tr>
<th>Mean</th>
<th>Standard deviation</th>
<th>Sample size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Distance</td>
<td>Interior angle</td>
</tr>
<tr>
<td>31.1</td>
<td>14.7</td>
<td>17.3</td>
</tr>
</tbody>
</table>

The accuracy of the VFV system is determined by all of the parameters listed in tables 1 and 2. Blocking plays a critical role in accuracy. Larger block sizes improve accuracy, which reduces the need for error correction but reduces template length. On the other hand, a smaller block size reduces accuracy but increases template length allowing larger keys. Larger block size increase uniqueness for block matching, suggesting directions for future improvements.

The bitstring produced by the client is corrected by Reed-Solomon ECC with n= 255 and k=65. This allows us to correct 95 errors in any bitstring, leaving 65 bits of security. With these parameters, VFV has an ERR of about 7.5% (Figure 3). While below the state-of-the-art privacy-enhanced algorithms on this dataset; e.g., the 2.1%EER reported by Bolt et al. back in 2007, these initial experiments do suggest that VFV has potential in viable remote authentication method. The section on future work discusses methods for improving the system.

Figure 3: FAR and FRR curves
5. SECURITY

The first layer of security is encryption. Blocks are encrypted using user’s personal (symmetric) key. Personal encryption prevents server and attacker from reading the contents of the template. Within the template, groups are randomly permuted by an initialization vector so they cannot be identified. The encrypted template is digitally signed by the user’s private key, which ensures the integrity of the template. The template is then encrypted using the server’s encryption key. The additional layer of encryption maintains the confidentiality of the transmitted template.

At the verification stage, the server digitally signs the permuted template with its private key and encrypts the permuted template using the user’s public key. This ensures a unique challenge to prevent replay attacks. The encrypted template is then sent to the user. This ensures the integrity and confidentiality of the challenge.

We can consider different scenarios of attack upon the system. If an attacker gains access to a server’s private key at the enrollment stage, the attacker still cannot modify the template due to the signature.

If the user’s private key is compromised, the attacker can inject his template at the enrollment stage. At verification, the attacker can authenticate himself as a legitimate user while denying access to the legitimate user. Since the user can no longer authenticate, this can be detected through verification after enrollment.

If both the server and user private keys are compromised, the communication channel is no longer secure. However, the template is still protected by the user’s personal key and by the chaff.

If the user’s personal key is compromised at the enrollment stage, the template is protected through the server’s public key. At the verification stage, the template is protected through the user’s private key.

If the user’s personal and private keys are compromised, the template is still protected at the enrollment stage through the server’s public key. However, the attacker can recover the template at the verification stage.

If all keys are compromised, the attacker has the full control over the system and can recover the template at any stage, leaving the chaff as the final layer of protection. Without the biometric, if the chaff is good, the attacker still has a random chance to recover the actual key. If the attacker has also broken into the server, they don’t need the random string to impersonate the user, but with both templates and the all keys, the attacker can recover the triangles. We currently know of no algorithm to recover minutiae from the triangle data, but cannot rule out that it might be possible.

6. FUTURE WORK AND CONCLUSION

Our current implementation proves the viability of VFV. We will further improve our implementation by modifying the following parts.

The triangle attributes used in our implementation are distance, angle, and invariant orientation. The most important improvement is improving triangle match accuracy. The matcher depends on the stability and uniqueness of the triangle attributes. Further study of triangle attributes will determine which are both stable and unique. This information will allow us to determine an optimal attribute set.

The triangle matcher forms the basis of the VFV system’s accuracy. The current matcher demonstrates the proof of concept of VFV. However, as the experiments show, improving accuracy, larger block sizes, and error correction will improve the usability of the system. Added features may allow increased accuracy without larger block sizes.

If an attacker can defeat the encryption, then VFV requires chaff that should be, to an attacker, indistinguishable from the user. Chaff chosen from a large pool may be slightly distinguishable from the real user. An improved chaff generator may eliminate this weakness.
In this paper, we designed a novel system that uses VV to verify users using fingerprint modality. We also discussed how the accuracy of triangle matching could be improved through our different parameters. Lastly, we examined how the block size can affect the accuracy of the system. This first mixing of VV and fingerprints has potential, if the blocks - matching accuracy can be improved.

REFERENCES


