A Study on Vulnerability in On-line Writer Verification System

Yasushi Yamazaki Akane Nakashima
Dept. of Information and Media Sciences
The University of Kitakyushu
yamazaki@env.kitakyu-u.ac.jp
nakashima.akane@is.env.kitakyu-u.ac.jp

Kazunobu Tasaka Naohisa Komatsu
Dept. of Information and Computer Science
Waseda University
{tasaka, komatsu}@kom.comm.waseda.ac.jp

Abstract

The analysis of vulnerabilities and threats in biometrics-based personal authentication systems is indispensable for the development and promotion of wide spread utilization of biometric technologies in accordance with international standardization of information security technologies. In this paper, we discuss the vulnerabilities and threats in a writer verification system which is an application of biometric technologies using handwriting information. We propose some attack methods that use verification results or user templates of a writer verification system. From some simulation results, it is clear that an attacker can estimate an authorized user's biometric information (handwriting information) and produce some forged handwritings which are accepted by the verification system.

1. Introduction

Biometrics-based personal authentication systems (biometric systems) that use physiological or behavioral characteristics are becoming increasingly popular, compared to traditional token-based or knowledge-based authentication methods [1]. However, it is pointed out that the biometric systems are vulnerable to attacks that can decrease their security [2]. Therefore, the analysis of vulnerabilities and threats in biometric systems is indispensable for the development and promotion of wide spread utilization of biometric technologies in accordance with international standardization of information security technologies. In [3], for example, these vulnerabilities and threats are identified and grouped into some classes. Among those, we focus on the vulnerability called 'estimation'. This vulnerability means that an attacker can estimate an authorized user's biometric information by using his user template stored in a biometric system or using verification results (scores) that the system returns. In the previous work, some attack methods that use this vulnerability of estimation are proposed in regard to face recognition [4] and fingerprint recognition [2]. However, in the realm of writer recognition [5,6], which is an application of biometric systems that use handwriting information as biometrics, the vulnerability of estimation is not discussed sufficiently. Therefore, in this paper, we propose some attack methods based on the vulnerability of estimation and discuss the threats in writer verification systems.

2. Vulnerability of estimation

As described in the previous section, we focus on the vulnerability of estimation in writer verification systems. It is important to clarify what kind of threats might be possible by using this vulnerability. At present, impersonation by using the estimated biometric information is known as a threat of using this vulnerability and the method of the impersonation is categorized as follows.

(1) Hill-climbing attack: An attacker repeats the input of synthetic biometric information by gradually modifying it so that the verification result (score) may approach the score that corresponds to the target (genuine) user.
(2) Template-based attack: An attacker estimates the target user's biometric information from his user template stored in a biometric system, and pretends him by inputting the estimated information to the biometric system.

In this paper, we discuss the vulnerability of estimation in writer verification systems based on the above two attack methods. In section 3, we propose an attack method based on the hill-climbing attack focusing on an on-line signature verification system. Moreover, in section 4, we propose another attack method based on the template-based attack focusing on a text-independent writer verification system.

3. Hill-climbing attack against signature verification system

Figure 1 shows a procedure of hill-climbing attack against an on-line signature verification system. When the system returns the value of distance or similarity between the inputted signature and the stored user template as a verification result, there exists a threat of impersonation in which an attacker repeats the input of synthetic signature until he is accepted as a target (genuine) writer by gradually modifying it so that the verification result may approach the verification result of the target writer. In this section, we describe the method of signature estimation by using forged handwritings (forgeries) which are produced artificially and automatically on a computer.

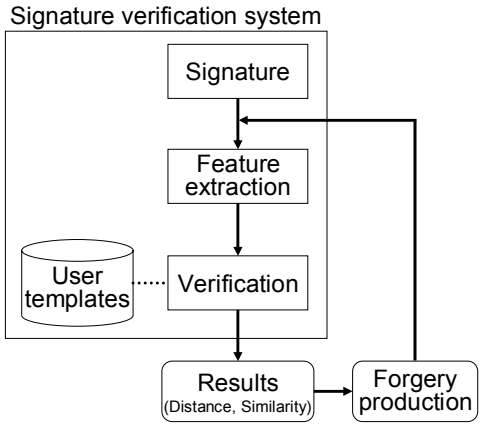


Figure 1. Overview of making forgeries by hill-climbing attack

3.1. Attack algorithm

In this subsection, we describe a procedure of making forgeries based on the above hill-climbing attack (see Figure 2).

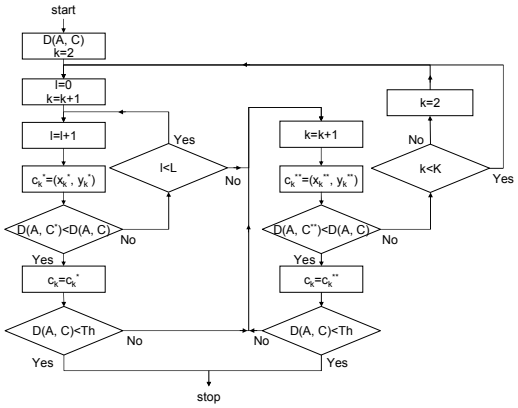


Figure 2. Flowchart of the attack algorithm

[step 1] Making a look-up table

By quantizing time-series hand-written data regarding writing direction in various handwritten characters, we calculate the observation probability of the writing direction and store it in a look-up table. The time-series data are expressed as $A = a_1, a_2, \cdots, a_I$, where I denotes the number of the sampled points and $a_i = (x_i, y_i)$, a set of two-dimensional pen-position data on a tablet. Next, the adjacent sampled points are connected with a straight line and the direction of the line is quantized into L values by referring to Figure 3. Here, the quantized data sequence is defined as $Q = q_1, q_2, \cdots, q_{I-1} \in \{1,2,\cdots,L\}$. Next, the conditional probability $P_m(n)$ which denotes the probability of $q_{i+1} = n$ under $q_i = m$ is calculated. In the look-up table, the values of $P_m(n)$, $\overline{x_{mn}}$, $\overline{y_{mn}}$ are stored, where $\overline{x_{mn}}$, $\overline{y_{mn}}$ denote the mean distance between a_{i+1} and a_{i+2} that satisfy $q_{i+1} = n$ under $q_i = m$ and defined as follows,

$$\overline{x_{mn}} = \frac{1}{s}\sum(x_{i+2} - x_{i+1}), \quad \overline{y_{mn}} = \frac{1}{s}\sum(y_{i+2} - y_{i+1})$$

where, s is the number of the sampled points that satisfy $q_{i+1} = n$ under $q_i = m$.

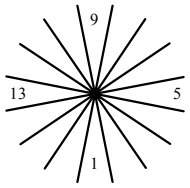


Figure 3. Quantized writing direction (*L*=16)

[step 2] Selection of modifying point

For a set of forgery data $C = c_1, c_2, \cdots, c_k, \cdots, c_K$, the modification point k is selected sequentially from 3 to K, where K denotes the number of the sampled points in C.

[step 3-1] Data modification (I)

By referring to the look-up table, c_k is modified based on the values of m and n such that $P_m(n)$ is the l-th largest under $m = q_{k-2}$. Here, $c_k = (x_k, y_k)$ is modified into $c_k^* = (x_k^*, y_k^*)$ so as to satisfy $x_k^* = x_{k-1} + \overline{x_{mn}}$ and $y_k^* = y_{k-1} + \overline{y_{mn}}$.

[step 3-2] Data update (I)

If the value of $D(A, C^*)$, the distance between the genuine signature and the modified forgery is less than the value of $D(A, C)$, the distance between the genuine signature and the original forgery, the

modified c_k^* is saved as c_k. Otherwise, c_k^* is discarded. In the former case, if the distance satisfies the threshold condition, the attack process is terminated. Otherwise, we proceed to [step 4-1] after incrementing the value of k as $k = k + 1$. In the latter case, if the value of l satisfies $l < L$, we go back to [step3-1] after incrementing the value of l as $l = l + 1$. Otherwise, we proceed to [step 4-1] after incrementing the value of k as $k = k + 1$.

[step 4-1] Data modification (II)

$c_k = (x_k, y_k)$ is modified into $c_k^{**} = (x_k^{**}, y_k^{**})$ so as to satisfy $x_k^{**} = 2x_{k-1} - x_{k-2}$ and $y_k^{**} = 2y_{k-1} - y_{k-2}$.

[step 4-2] Data update (II)

If the value of $D(A, C^{**})$, the distance between the genuine signature and the modified forgery in [step 4-1] is less than the value of $D(A, C)$, the modified c_k^{**} is saved as c_k. Otherwise, c_k^{**} is discarded. In the former case, if the distance satisfies the threshold condition, the attack process is terminated. Otherwise, we go back to [step 4-1] after incrementing the value of k as $k = k + 1$. In the latter case, if the value of k satisfies $k < K$, we go back to [step 2]. Otherwise, we go back to [step 2] after setting the value of k as $k = 2$.

3.2. Simulation experiments

(1) Signature verification algorithm

As an example of signature verification algorithms, we use a DTW (Dynamic Time Warping)-based on-line signature verification algorithm by referring to [7]. For preprocessing, we eliminate duplicate points in the handwritten data where there is no hand movement and normalize the data in regard to the number of the sampled points, position and size, respectively.

(2) Data description

We use Kanji (Chinese-Japanese characters) signatures in the experiment. All handwritten data are gathered using a standard digitizing tablet with a spatial resolution of 0.01mm and a sampling rate of 200 samples/s. We collect 20 signatures from each of 5 writers (A~E), from which 3 signatures are used for making a template and the remaining 17 signatures are used for verification. On the other hand, it is considered that the difficulty of impersonation depends on the elaborateness of initial forgery which is given at the beginning of the hill-climbing attack. Therefore, we prepare the following different forgeries and give them to the verification system as the initial data of the hill-climbing attack.

1. Simple forgery (1): An attacker makes the forgery by his own handwriting manner.
2. Simple forgery (2): An attacker makes the forgery by referring to the target's handwritings.
3. Simulated forgery (1): An attacker makes the forgery by tracing the target's handwritings.
4. Simulated forgery (2): An attacker makes the forgery by tracing the target's handwritings and also following the writing order of them.

The larger the number, the more elaborate the forgery becomes. For every writer, we prepare 16 forgery data for each of the above four-type forgeries.

(3) Experimental results

By considering no acceptance of the forgery at the beginning of the hill-climbing attack, we set the verification threshold for each writer so that the value of FAR (False Acceptance Rate) is 0 and the value of FRR (False Rejection Rate) is the minimum for each of four-type forgeries. Table 1 shows the number of iteration needed for the forgery to be accepted as the genuine signature. Figure 4 shows a sample of the produced forgery of writer A by the proposed algorithm when the data of simple forgery (2) are given as the initial forgery. Moreover, Figure 5 shows the relationship between the number of iteration and the value of verification distance for the same data as Figure 4.

Table 1. Number of iteration needed for the acceptance of forgeries

1. Simple forgery (1)

writer	mean	minimum	maximum
A	1302	707	2750
B	1859	694	3093
C	994	454	3853
D	1718	702	3711
E	1900	948	5876

2. Simple forgery (2)

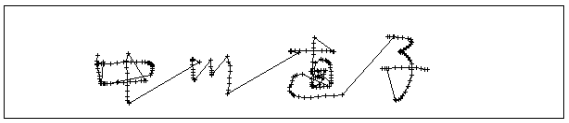
writer	mean	minimum	maximum
A	587	169	1830
B	1715	980	3967
C	793	213	1774
D	987	332	2566
E	925	25	2022

3. Simulated forgery (1)

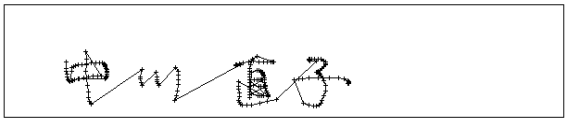
writer	mean	minimum	maximum
A	577	6	917
B	2071	641	6649
C	410	32	625
D	445	145	678
E	690	115	1354

4. Simulated forgery (2)

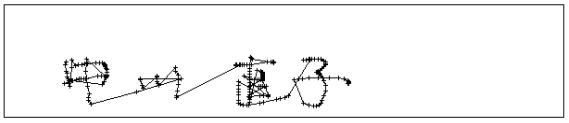
writer	mean	Minimum	maximum
B	689	34	3201
C	460	37	808



(a) An example of genuine signature of writer A



(b) An example of initial forgery of writer A



(c) Produced forgery by the proposed algorithm

Figure 4. Example of forgeries

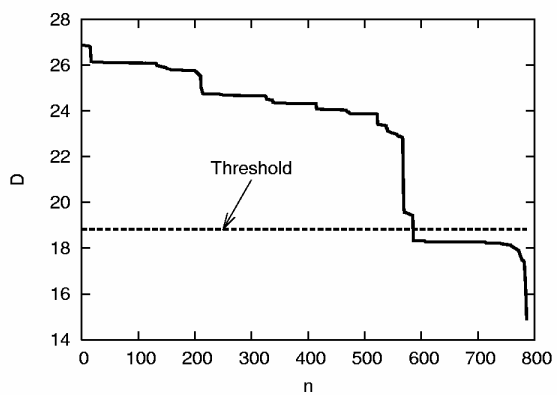


Figure 5. Relationship between the number of iteration and the value of verification distance

Although the number of iteration needed for the forgery to be accepted as the genuine signature depends on the distance between a user template and an initial forgery, it can be seen in Table 1 that the forgery which is accepted after a finite number of iteration can be produced for every type of initial forgeries. Moreover, when the initial forgery is elaborate enough, the distance between a user template and an initial forgery is relatively small, which enables the production of acceptable forgery within several dozen of times of iteration as is shown in the case of 3. and 4. in Table 1. From these simulation results, it is clarified that the falsely accepted forgeries can be produced artificially by using the hill-climbing attack algorithm even in the signature verification system that is difficult to deceive by using manually produced forgeries.

4. Template-based attack against a text-independent writer verification system

Figure 6 shows a procedure of the template-based attack against a text-independent writer verification system. When the user templates are leaked from the verification system by an illegal operation for example, there exists a threat of impersonation in which an attacker makes forgeries from the templates and attacks against the system by using the produced forgeries. Also, we should consider that the attacker might acquire a user's on-line handwritten data for making the forgeries, especially when the writer verification system is operated in a network environment. In this section, we describe another method of handwriting estimation by using the forgeries.

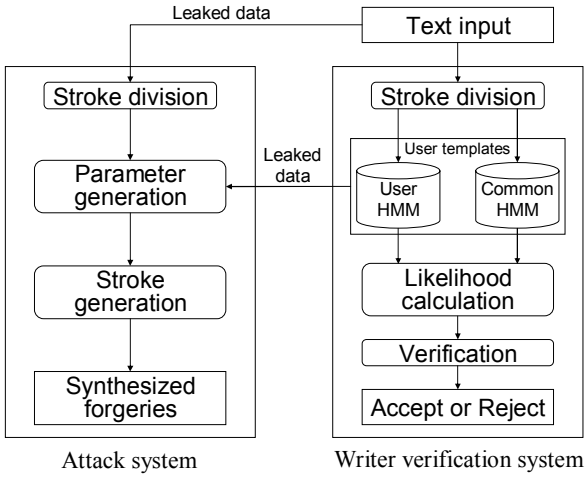


Figure 6. Overview of making forgeries by template-based attack

4.1. Attack algorithm

As an example of text-independent writer verification algorithms, we use a HMM (Hidden Markov Models)-based on-line writer verification algorithm based on our previous work [8]. In the proposed method, enrollment is processed as follows. First, the handwritten data are divided into a set of strokes and classified into several categories. Second, a 'Common HMM' and 'User HMM' are produced by using the categorized data. Here, the former represents the typical shapes of handwritten strokes that are common to many writers, and the latter represents each writer's personal features in handwriting. In the verification process, the likelihood for both HMMs for the given handwritten data is calculated and the decision is made whether the writer is accepted or not.

Against this writer verification algorithm, we propose the following attack methods.

1. Forgery (a): An attacker makes the forgery by using the leaked templates from the Common HMM and the User HMM.
2. Forgery (b): An attacker makes the forgery by using the target's on-line handwritten data.

4.2. Simulation experiments

(1) Data description

We use on-line handwriting database [9] in the experiment. In the experiment, the Common HMM is produced by using 1000 characters (50 characters x 20 writers), and the User HMM is produced by using 500 characters for each writer. Here, the Common HMM and the User HMM are made from different writers and characters. Moreover, we use another set of data for making Forgery (b).

(2) Experimental results

Figure 7 shows the verification results under the condition that each of 10 writers writes 5 characters which are selected randomly. As is shown in this figure, both Forgery (a) and Forgery (b) are more likely to be accepted falsely than the genuine data. From these simulation results, it is clear that an attacker can make the forgery that is accepted by a text-independent writer verification system by using the leaked user template or on-line handwritten data.

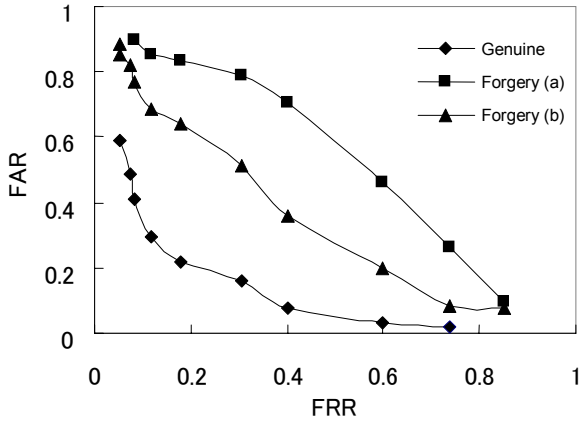


Figure 7. Verification results

5. Conclusion

In this paper, we discuss the vulnerabilities and threats in writer verification systems and propose some attack methods that use verification results or user templates of the verification systems. From some simulation results, it is clear that an attacker can estimate an authorized user's handwriting information and produce some forgeries that are accepted by the verification system. Our future work may involve the further research on the vulnerabilities in writer recognition systems and the proposal of effective countermeasures against possible threats.

References

[1] A.Jain, R.Bolle, and S.Pankanti, BIOMETRICS - Personal Identification in Networked Society, Kluwer, 1999.

[2] U.Uludag and A.Jain, Attacks on biometric systems: a case study in fingerprints, Proc. SPIE-EI 2004, Security, Steganography and Watermarking of Multimedia Contents VI, pp. 622-633, San Jose, CA, January 18-22, 2004.

[3] Japan Automatic Identification Systems Association (Eds.), Standardization of the development of social infrastructure utilizing personal identification technologies based on biometric information (in Japanese), 2004.

[4] A.Adler, Sample images can be independently restored from face recognition templates, http://www.site.uottawa.ca/~adler/publications/2003/adler-2003-fr-templates.pdf, 2003.

[5] R.Plamondon and G.Lorette, Automatic signature verification and writer identification - The state of the art, Pattern Recognition, Vol.22, No.2, pp.107-131, 1989.

[6] F.Leclerc and R.Plamondon, Automatic signature verification : The state of the art - 1989-1993, Int. J. of Pattern Recognition and Artificial Intelligence (IJPRAI), Vol.8, No.3, pp.643-660, 1993.

[7] M.Yoshimura, Y.Kato, S.Matsuda and I.Yoshimura, On-line signature verification incorporating the direction of pen movement, IEICE Trans. E74, 7, pp.2083-2092, 1991.

[8] Y.Yamazaki, T.Nagao, and N.Komatsu, Text-indicated writer verification using Hidden Markov Models, ICDAR2003, pp.329-332, 2003.

[9] TUAT Nakagawa Lab., On-line Handwriting Database, HANDS-nakayosi_t-98-09.