Finding the Original Point Set Hidden among Chaff

Ee-Chien Chang Ren Shen Francis Weijian Teo

School of Computing
National University of Singapore

{changec,shenren,teoweiji}@comp.nus.edu.sg

ABSTRACT

In biometric identification, a fingerprint is typically represented as a set of minutiae which are 2D points. A method [4] to protect the fingerprint template hides the minutiae by adding random points (known as chaff) into the original point set. The chaff points are added one-by-one, constrained by the requirement that no two points are close to each other, until it is impossible to add more points or sufficient number of points have been added. Therefore, if the original template consists of s points, and the total number of chaff points and the original points is m, then a brute-force attacker is expected to examine half of m chooses s possibilities to find the original. The chaff generated seem to be "random", especially if the minutiae are also randomly generated in the same manner. Indeed, the number of searches required by the brute-force attacker has been used to measure the security of the method. In this paper, we give an observation which leads to a way to distinguish the minutiae from the chaff. Extensive simulations show that our attacker can find the original better than brute-force search. For e.g. when $s = 1$ and the number of chaff points is expected to be about 313, our attacker on average takes about 100 searches. Our results highlight the need to adopt a more rigorous notion of security for template protection. We also give an empirical lower bound of the entropy loss due to the sketch.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Authentication; E.3 [**Data**]: Coding and Information Theory

General Terms

Security, Algorithms

Keywords

Secure Sketch, Fingerprint template, Biometric privacy protection, Online parking.

ASIACCS'06 March 21-24, 2006, Taipei, Taiwan.

1. INTRODUCTION

Biometric data is usually noisy due to noise introduced during data capturing. For instance, two scanned images of a same finger are probably different. The inevitable noise poses challenges in applying classical cryptographic techniques on biometric templates. Recently, a few schemes have been proposed to handle the noise, for example fuzzy commitment[8], shielding function[9], and secure sketch[5, 2]. These schemes use a short piece of public information to recover the original data from the noisy version. More specifically, the public data P_X is constructed from the original biometric template X. The data P_X has the property that, from P_X and another biometric template Y, the original X can be recovered, provided that Y is close to X. In other words, P_X can be used to remove the noise from Y, if Y is close to X. We follow the definitions by Dodis et al.[5] and call the public data P_X a sketch.

The following example illustrates an application of a sketch. In this application, the fingerprint is to be used as the secret key in encrypting a file. Before encryption, the fingerprint of the owner is scanned, and the fingerprint features are extracted. The extracted features X, represented as a string, serves as the secret key. The file is then encrypted using the secret key. Next, a sketch P_X is computed from X, and P_X is stored in the header of the encrypted file in clear. To decrypt the file, a fingerprint is obtained and its features Y are extracted. From the secure sketch P_X and Y, by property of the sketch, the original X can be reconstructed if Y is closes to X. Now, from the reconstructed X, the secret key can be recovered and the file can be decrypted. Since P_X is published in clear, it is important that it does not reveal too much information about X.

The design of sketch is dependent on the underlying metric in measuring the distance between two templates. Fingerprints are widely used in biometric identification. Typically, a fingerprint is represented as a set of 2D points, known as minutiae[10]. Under noise introduced during scanning and processing, each minutia may be perturbed by a small distance. Let us call this type of noise the *white noise*. In addition, a small number of minutiae may be missing and some new minutiae may be introduced. Let us call this type of noise the *replacement noise*. Figure 1 shows an example of minutiae and the noise[1].

Although extensive studies have been conducted for finger-

[1]Fingerprint raw image is obtained from [1]

(a) (b)

Figure 1: (a) The original fingerprint. The dots are the extracted minutiae. (b) The dots are the original minutiae. The "+" are minutiae extracted from another scan of the same finger.

prints, there are few secure sketch constructions on fingerprints. Perhaps the earliest construction is by Clancy et al. [4], which is the focus of this paper. Yang et al. [13] followed the approach of adding chaff and proposed an alternative feature representation that reduces (but not eliminates) the effect of white noise. Chang et al. [3] gave a provably secure (under the notion of entropy loss) construction using a combination of rounding and a different way of generating chaff points. However, it is not clear whether the construction by Chang et al. loses less entropy compared to the construction by Clancy.

Clancy et al. [4] proposed the following method of generating a sketch which comprises of 2 parts. The first part is an unordered set of points $R = (X \cup C)$, where X is the original minutiae, and the points in C are randomly selected and are called the *chaff*. The set R is δ-separated in the sense that no two points are within a distance of δ to each other. The chaff points are selected one-by-one in the following manner: First, uniformly and randomly pick a 2D point. If this point is within a distance δ to any point in X, or any selected chaff, then discard it. If not, select it as a chaff point. The process is repeated until it is impossible to add any more chaff points or sufficient number of chaff points have been selected. Now, the description of R will be the first part of the sketch. Suppose Y is a noisy version of X corrupted by white noise, from Y together with R, we can recover X. The role of the second part of the sketch is to recover X from replacement error.

We are interested in the first part of the sketch $R = (X \cup C)$. Intuitively, the minutiae X are hidden among the random chaff and it seems impossible to distinguish them. Suppose $|X| = s$ and $|X \cup C| = m$, then on average, a brute-force search examines $\frac{1}{2}\binom{m}{s}$ possible combinations in order to find X. In fact, the amount of searches required by the brute-force search has been used to measure the security of the sketch. Based on the typical number of minutiae, noise parameters, and the assumption that the attackers employ brute-force search, it was estimated the attacker has to invest 2^{69} more time to find X compared to a user who has a noisy version of the minutiae [4]. Now, this leads to the following interesting questions: *Is it possible to distinguish X from C? Is there an attacker that can perform better than the brute-force-search*?

We give a method that on average, can find the original X among $(X \cup C)$ better than the brute force search. Based on simulation results, when $|X| = 1$ and the average size $|R| = 312.6$, on average we can find the sole minutia using about 100 searches, whereas a brute-force search on average requires 156.5 searches. When $|X| = 38$, then the average speedup factor compare to the brute-force search is 2192.7.

The speedup provided by our attacker does not sufficiently imply that the sketch $R = (X \cup R)$ reveals too much information and hence is insecure. Instead, it highlights the need to adopt a more rigorous formulation of security in analyzing sketches. For example, using the notion of entropy loss proposed by Dodis et al. [5]. On the other hand, the stochastic process in generating the chaff is intriguing and difficult to analyze. A corresponding process known as online parking has attracted much attention[12, 11, 6]. Many fundamental questions remain open, for example the Palasti's conjecture[11]. This is especially so in 2D due to the involvement of geometry. Hence, establishing a bound of the entropy loss analytically would not be easy. Nevertheless, based on our simulation and some approximations, we are able to give an empirical lower bound of the entropy loss.

Main Idea. Our method is based on the following observation. Recall that the chaff points are generated one-by-one. We observe that a chaff point that is generated late in the process, tends to have smaller *free area*. We will define free area later in Section 2. Informally, a point with smaller free area has more neighboring points in $(X \cup C)$. In other words, we observe that for different local arrangements, the likelihood of a point being the minutia can be different. This observation is formulated as the inequality (3). However, we are unable to prove it analytically. Nevertheless, it is verified experimentally. This observation leads to an attacker who gives higher priority to points with large free area during the search.

2. MODELS AND ASSUMPTIONS

Attacker Model. Given the sketch P_X of the original X where $|X| = s$, the goal of an attacker is to find X. The attacker can query a blackbox. On input of a set Q of s points, the blackbox will return `YES` iff $Q = X$. The effectiveness of an attacker is measured by the number of queries he sent. In the application given in introduction, the blackbox is the decryption of the file using the key Q. The output of `YES` corresponds to the situation where the file is successfully decrypted. Note that we only count the number of calls to the blackbox. Other computations carried out by the attackers, for example, in deciding which query to be sent, are not counted. It is appropriate and convenient to count only the blackbox calls. Typically, the blackbox operation is computationally intensive, for instance, file decryption in the above application. In addition, in some applications, the blackbox operations are carried out by a remote server. The attackers have limited access to the server, but have ample computing resources.

Online Parking. Let us call the following process *online parking*. This process selects a set of points one-by-one. Each point is uniformly and randomly chosen from the do-

main $[0, n] \times [0, n]$. If it is within unit distance from any previously selected points, then it is discarded. If not, it is selected. The process is repeated until the stopping condition is met. Here are two possible stopping conditions. We can repeat the process until it is impossible to add more points. Note that if we employ this condition, the total number of points selected is not deterministic. Alternatively, we can repeat the process until a predetermined number of points have been selected. In this paper, we employ the first condition to generate the sketch. We also conduct preliminary investigation for the second condition (Section 4.6).

For each selected point, if it is the k-th point selected, then we say that its *arrival order* is k.

Distribution of minutiae. The minutiae X is a set of s points from the bounded domain $[0, n] \times [0, n]$. The set X is separated in the sense that for any two different points $x, y \in X$, the Euclidean distance $\|x - y\|_2 > 1$.

We assume that the distribution of the set of s minutiae is same as the distribution of the first s points generated by the online parking process. In practice, the minutiae might follow another distribution. Knowledge of such distribution may further help to identify the minutiae.

Sketch generation. Recall that the sketch P_X consists of two parts. Let us call the first part the *white noise sketch*, since its role is to recover from white noise, and the second part the *replacement sketch*. As mentioned in the introduction, the white noise sketch is the description of the unordered set $R = (X \cup C)$, where X is the original and C is generated by the online parking process. The role of replacement sketch is to correct t replacement errors (that is, t points are replaced by t random points), where t is a predetermined parameter. The actual value of t is not crucial in our analysis. There are a number of known sketch schemes for replacement noise [7, 5, 3]. For instance, Juels et al. [7] proposed using a polynomial of degree $(s - 2t + 1)$, and employed BCH in decoding.

The sketch P_X reveals some information of X. For instance, the white noise sketch, which is the point set $R = (X \cup C)$, reveals that a minutia must be one of the points in R. The replacement sketch further reveals information on X, and imposes more restrictions on the point sets that can generate P_X. Let us say that a point set X' is a *candidate* consistent with a sketch, if the sketch can be generated from X'.

Brute-force attacker. A brute-force attacker enumerates all candidates consistent with the given sketch, and sends the candidates to the blackbox one-by-one until a `YES` is obtained. The white noise sketch reveals possible locations of the minutiae, and the number of candidates consistent with white noise sketch is $\binom{m}{s}$ when $|X \cup C| = m$. The number of candidates can be further reduced by considering the replacement sketch. If the replacement sketch can correct up to t errors, and the set-difference scheme [7] is employed, then the average number of candidate consistent with both white noise and replacement sketch is approximately

$$\binom{m}{s} m^{-(s-2t)}.$$

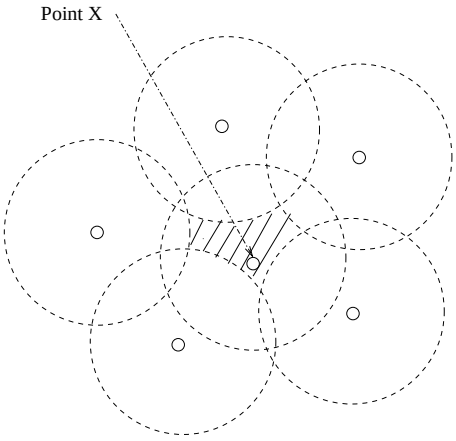

Figure 2: Illustration of free area. $\mathcal{F}_R(x)$ is the area of the shadow. Each circle is a unit disk.

We assume that the brute-force attacker is randomized. The order of sending the candidates to the black box is randomly permuted. Hence, for any sketch, the expected number of calls required is half of the total number of candidates.

Free area. Given a set of points W, define $\mathcal{A}(W)$, the *available region*, to be the set

$$\mathcal{A}(W) = \{x \in [0, n] \times [0, n] : \text{ for all } w \in W, \|x - w\|_2 > 1\}.$$

A point in the available region can be added into W and yet W remains separated. For a point set $\widetilde{R}$ and a point $x \in \widetilde{R}$, define the *free area* of x with respect to $\widetilde{R}$ as,

$$\mathcal{F}_{\widetilde{R}}(x) = |\mathcal{A}(\widetilde{R} - \{x\}) - \mathcal{A}(\widetilde{R})|, \tag{1}$$

where "$-$" is the set difference operator, and $|\cdot|$ gives the area of the region. Figure 2 illustrates the free area. When it is clear in the context, we omit $\widetilde{R}$ and write the free area as $\mathcal{F}(x)$.

Consider the online parking process. Let A_x be the random variable on the arrival order of x, given that x is selected. Let us write $\mathcal{F}(x) = f$ as the event that x is selected and contained in some point set, and its free area in that point set is f. We are interested in the conditional probability

$$Pr(A_x \leq s \mid \quad \mathcal{F}(x) = f). \tag{2}$$

Since X and C follow the distribution of online parking, we can treat R as the output of an online parking process. Hence, if the arrival order of x is not more than s, then x is a minutia. Although the attackers know the point set R, the conditional probability (2) does not exploit full knowledge of R. Instead, only the free area of x is used. Nevertheless, such partial information is sufficient in distinguishing the minutiae.

2.1 Summary of Notations

X, Y	: X is the set of the original minutiae. Y is a noisy version of X.
s	: Number of minutiae. $s = \|X\|$.
C, R, m	: C is the set of chaff. $R = X \cup C$ and $m = \|R\|$.
n	: Width of the domain. All points are in $[0, n] \times [0, n]$.
P_X	: P_X is the sketch generated from X.
$\mathcal{A}(W)$	: Available region of a point set W.
$\mathcal{F}_{\widetilde{R}}(x), \mathcal{F}(x)$	: Free area of x in the point set $\widetilde{R}$.
`LookUp`	: Look up table used by the attackers.
A_x	: The arrival order of x, given that x is selected.

2.2 Differences from Clancy et al. method.

For clarity, we now describe two subtle differences of our model from the model proposed by Clancy et al. [4]. Firstly, we measure the effectiveness of an attacker by the number of calls to the black box. In contrast, Clancy et al. considered the number of computational steps required by the attacker, and the authentic user during decoding (with respect to a specific decoding algorithm). The effectiveness of an attacker is the ratio of the steps taken by the attacker over the authentic user. Secondly, Clancy et al. proposed to generate a predetermined number of chaff points so that the authentic user can decode efficiently. We do not consider the decoding complexity and hence generate as many chaff points as possible. Although there are differences in measuring effectiveness of attackers, our attacker can be adopted and successful in the scenario studied by Clancy et al.

3. ATTACKER

Our main observation is that, for $f_0 > f_1$ and any s,

$$\Pr(\ A_x \le s \mid\ \mathcal{F}(x) = f_0) > \Pr(\ A_x \le s \mid\ \mathcal{F}(x) = f_1). \quad (3)$$

That is, for a R randomly generated by online parking, if a point $x \in R$ has larger free area compared to another point y in R, then it is more likely that x arrived earlier than y. Unfortunately, we are unable to analytically prove the observation. Nevertheless, extensive simulations give strong evidence to support the claim. Figure 4 shows an estimation of the likelihood function. Note that each function is increasing with respect to the free area.

3.1 Identifying one point, $s = 1$

To illustrate the attacker's algorithm, we first give an attacker for $s = 1$, that is, when there is only one minutia in X. In this scenario, the goal of the attacker is to identify the very first point that arrive in the online parking process. If R has m points, then a brute force attacker, on average, takes $m/2$ calls to the black box.

Given the white noise sketch, which is a description of R, the consistent candidates are all the singleton subsets. Our attacker carries out the following steps:

1. The attacker computes $\mathcal{F}(x)$ for all $x \in R$.
2. Next, it enumerates points in R in decreasing order with respect to $\mathcal{F}(x)$. The enumerated points are sent to the black box. The attacker stops when the black box outputs `YES`.

3.2 Likelihood of the first s points

Online parking is not a memoryless process and thus there is dependency between two points. Thus, A_x is dependent on A_y. Nevertheless, if x and y are not close to each other, the effect of one point on the other should not be significant. Hence, we employ the following approximation:

$$\Pr(A_x \le s, A_y \le s \mid\ \mathcal{F}(x) = f_1, \mathcal{F}(y) = f_2) \approx \Pr(A_x \le s \mid\ \mathcal{F}(x) = f_1) \cdot \Pr(A_y \le s \mid\ \mathcal{F}(y) = f_2). \quad (4)$$

Using (4), we can obtain an approximation of the likelihood for each candidate (which is a set of s points) consistent to P_X. This leads to the following attacker:

1. Computes the likelihood of each candidate consistent to P_X.
2. Enumerates the candidates in decreasing order with respect to their likelihood. Next, send the enumerated candidates to the blackbox until the blackbox outputs `YES`.

Consider the situation where an attacker has sent a few candidates to the blackbox and they are all not the correct original. Intuitively, the above attacker does not fully exploit the fact that the previously sent candidates are not correct. Hence, the attacker can be improved. In our simulation, we experiment with various ways to estimate the likelihood. We replace step 1 above by the following.

1(a). For each candidate $\{x_1, x_2, \ldots, x_k\}$, assigns it the value $\prod_{i=1}^{s} \texttt{LookUp}(\mathcal{F}(x_i))$, where `LookUp` is a predetermined lookup table.

In our experiments, it turns out that by using the identity function as lookup, that is, $\texttt{LookUp}(i) = i$, we already can achieve noticeable speedup over the brute-force attacker.

3.3 Min-entropy retained by publishing the sketch

A way to analyze the security of a sketch is by investigating the remaining entropy of the biometric data, given that the sketch is made public. Dodis et al. [5] proposed using the average min-entropy of A given B, which is,

$$\widetilde{H}_\infty(A|B) = -\log(\mathbf{E}_{b \leftarrow B}[\max_a \Pr(A = a|B = b)]). \quad (5)$$

By treating X and P_X as the random variables for the minutiae and the sketch respectively, the min-entropy loss due to the sketch is defined as,

$$H_\infty(X) - \widetilde{H}_\infty(X|P_X),$$

where the min-entropy $H_\infty(X) = -\log\ (\max_a \Pr(X = a))$.

As mentioned in the introduction, online parking is not easy to analyze, and hence bound on the entropy loss may be

difficult to obtain. However, from simulation and the approximation (4), we can obtain an estimation of

$$\max_{\{x_1,x_2,..,x_s\}} \Pr(X = \{x_1, \ldots, x_s\} | \mathcal{F}_R(x_1), \ldots, \mathcal{F}_R(x_s)). \quad (6)$$

Note that for random variables A and B, and a deterministic function f,

$$\max_a \Pr(A = a | B = b) \geq \max_a \Pr(A = a | f(B) = f(b)).$$

Therefore, $\max_a \Pr(X = a | P_X = R)$ is greater or equal to (6). This suggests an empirical method to estimate an upper bound on the min-entropy, which in turn gives us an lower bound on entropy loss. In the next section, we give more details on the simulation results.

4. SIMULATION AND COMPARISON

4.1 Experiment Settings.

For convenience, to simulate the online parking process, we discretized the domain $[0, n] \times [0, n]$. Therefore, minutiae and chaff points are selected from a set of discrete points. Each unit interval is discretized into 100 points. Hence, there are 100^2 points in $[0, 1) \times [0, 1)$. For each experiment, we collected 10000 samples. Each sample is a point set R obtained through online parking. For each point in R, its arrival order is recorded and its free area is approximated by counting the discrete points in the region.

The experiments are conducted in $[0, n] \times [0, n]$ for $n = 50$ and $n = 22$. The average $|R|$ is 1668.4 and 312.6 for $n = 50$ and $n = 22$ respectively. By treating each point as a disk of diameter 1, the average packing density (for both $n = 50$ and $n = 22$) is about 0.525. This is slightly less than the Palasti's conjecture[11] of 0.559, probably due to the different treatment of the domain boundary.

We also conducted experiments in 1D. That is, the domain is the interval $[0, n]$, and for any two selected points x and y, $|x - y| \geq 1$. The free area of a 1D point x can be defined similarly as in (1). In 1D, the free area of x is simply $(x_r - x_l - 2)$ where x_r and x_l is the right and left neighbors of x respectively. For $n = 1340$, the average $|R|$ is 994.8, which gives packing density of 0.743.

4.2 Likelihood

From the 10000 samples, we can estimate the conditional probability $\Pr(A_x \leq s \mid \mathcal{F}(x) = f)$ by first estimating $\Pr((A_x \leq s) \cap (\mathcal{F}(x) = f))$ and $\Pr(\mathcal{F}(x) = f)$. Figure 3 plots $\Pr((A_s \leq s) \cap (\mathcal{F}(x) = f))$ against the free area f for different s. A function in Figure 4 shows $\Pr(F(x) = f)$ with respect to f. Observe in Figure 4 that a large proportion of points have small free space. In contrast, as illustrated in Figure 3, for points that arrive early, relatively small proportion of them have small free space. This implies that, given that a point has large free area, it is more likely to have arrived early, and hence more likely to be a minutia.

Figure 4 show the likelihood function for different s. Note that $\Pr(A_x \leq s \mid \mathcal{F}(x) = f)$ is increasing as a function of f. Since the average $|R|$ is 1668.4, the probability that a randomly chosen point from R to arrive not later than 165 is about 0.1. From the graph, the likelihood $\Pr(A_x \leq 165 \mid \mathcal{F}(x) = 50)$ is more than 0.15. Hence a point with free area more than 50 is 1.5 times as likely to arrive not later than 165, compared to a randomly chosen point.

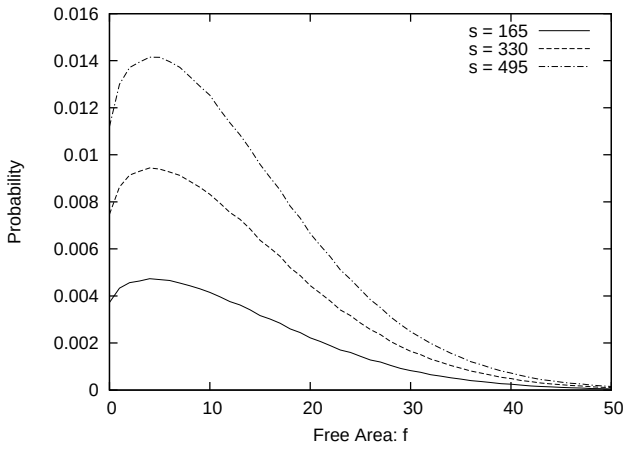

Figure 3: Probability density function, $Pr(A_x \leq s \cap F(x) = f)$, for $s = 165, 300$ and 495. The domain is $[0, n] \times [0, n]$ where $n = 50$.

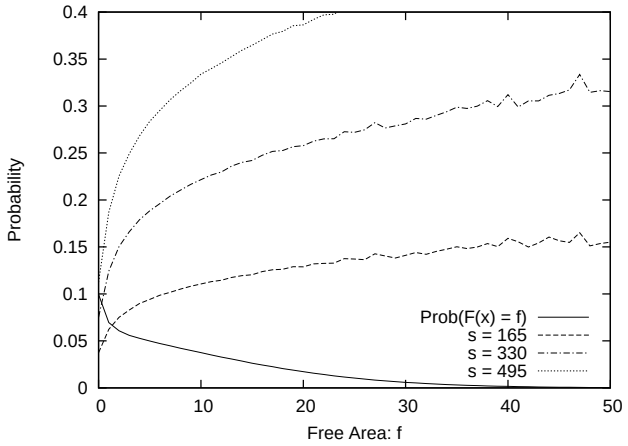

Figure 4: Distribution of free area, $\Pr(F(x) = f)$, and conditional probability of arrival order given free area, $\Pr(A_x < s | F(x) = f)$, for different $s = 165, 330, 495$. The domain is $[0, n] \times [0, n]$ where $n = 50$. The average $|R| = 1668.4$.

Similar observations can be made for experiments in the 1D domain, illustrated in Figure 5.

For different n and s, it seems that the conditional probabilities are almost the same as long as the ratio (s/n^d) is the same, where d is the dimension of the domain. This is illustrated in Figure 6 where $d = 2$.

4.3 Brute-force attacker for $s = 1$

When $|X| = 1$, our attacker simply computes `LookUp`(x) for all $x \in R$, and sends them to the blackbox according to the looked-up values in descending order. For each sample R, the number of blackbox calls required is the number of points

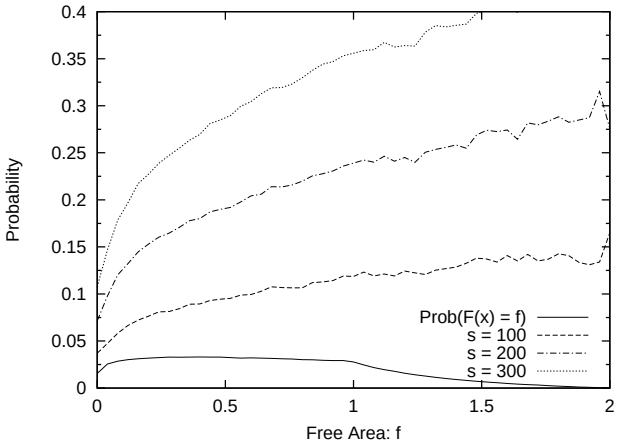

Figure 5: Distribution of free area, $\Pr(F(x) = f)$, and conditional probability of arrival order given free area, $\Pr(A_x < s|F(x) = f)$, for different $s = 100, 200, 300$. The domain is $[0, n]$ where $n = 1340$. The average $|R| = 994.8$.

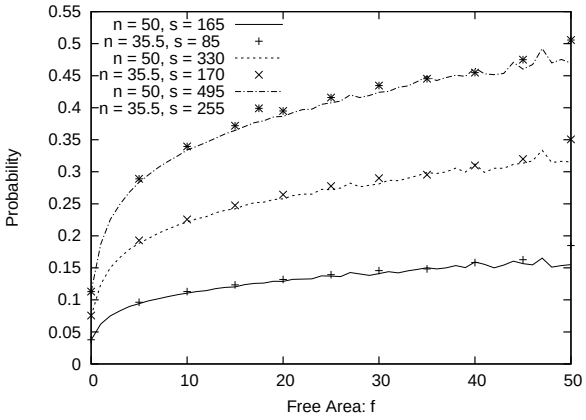

Figure 6: Comparison of $\Pr(A_x < s|F(x) = f)$ with different n and s. There are 3 pairs of functions. For each pair, the ratio (s/n^d) is approximately the same. For example, $(165/50^2) \approx (85/35.5^2)$.

in R whose looked-up value is larger or equal to the looked-up value of the sole minutia. Figure 7 gives the histogram of the number of calls, and the average is 100. Note that the width of domain $n = 20.7$, and the brute-force attacker on average takes 156.5 calls.

4.4 Brute-force attacker for $s > 1$

The average speedup factor compare to the brute-force attacker can be estimated in the following ways. Consider a sample point set $R = \{x_1, x_2, \ldots, x_m\}$. We can compute the likelihood of the actual minutiae X. Recall that we estimate the likelihood of the set X by $\prod_{x \in X} \texttt{LookUp}(x)$. Also recall that our attacker sends the candidates to the blackbox in the order of decreasing likelihood. Thus, the number of blackbox calls required to hit X is same as the number of

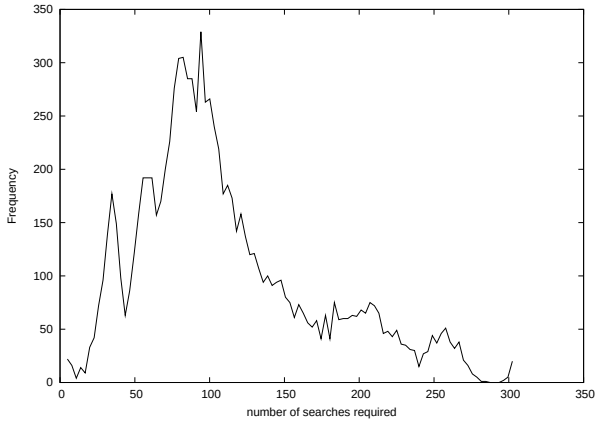

Figure 7: Histogram of number of calls made by our attacker. Number of Bins= 100, Bin Size= 2.97

candidates whose likelihood is larger than that of X.

Now, consider the set $L = \{\log(\texttt{LookUp}(x_1)), \ldots, \log(\texttt{LookUp}(x_m))\}$. By Central Limit Theorem, the distribution of the sum of s randomly chosen numbers from L can be approximated by a normal distribution, whose mean and variance can be derived from the mean and variance of L. From this normal distribution, we can estimate the proportion of subsets of R whose likelihood is larger than that of X. Assuming that the candidates are randomly located among all subsets of R, we can obtain the speedup factor provided by the attacker.

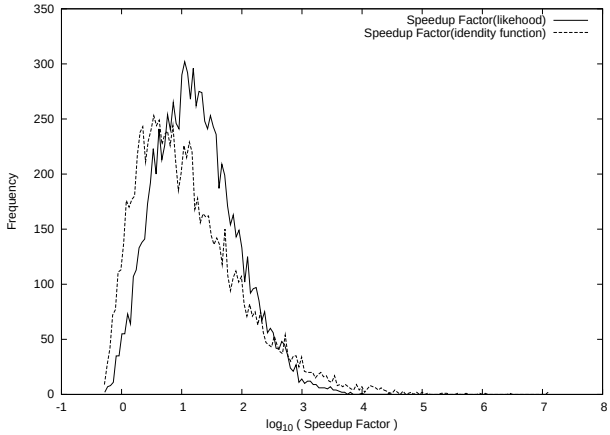

Figure 8: Comparison on speedup factor, $n = 20.7$.

Figure 8 shows the histogram of the speedup factor, when $s = 38$ and expected number of points in R is 312.6. The average speedup for the 10000 samples is 77.5, and the geometric mean is 18.0. It also shows the result for the same s, but using a different `LookUp`. Here, we simply choose the identity function as the lookup function. Interestingly, the average improves to 2192.7, and the geometric mean reduces to 13.88. Observe that for some samples, the speedup factor reaches 10^5. This is undesirable in security applications because it indicates that, with some probability, small but

noticeable, the attack can be very successful.

4.5 Entropy loss

We want to estimate the min-entropy of the minutiae X given the white noise sketch R, that is, $\widetilde{H}_\infty(X|R)$. Note that each sample R is a randomly chosen white noise sketch. Using the approximation in (4), the set $\{x_1, x_2, \ldots, x_s\}$ that maximizes the conditional probability $\Pr(X = \{x_1, \ldots, x_s\}|R)$ is the set with s largest looked-up value. Hence, for a sample R, we can obtain $\max_a \Pr(X = a|R)$. By averaging over all samples, we have a lower bound of min-entropy of X given the white noise sketch. When $s = 38$ and $n = 20.7$, the min-entropy is at most 61.2 bits. In other words, by making the white noise sketch public, the min-entropy of the minutiae is reduced to at most 61.2.

The above estimate does not consider the replacement sketch. The entropy loss of many set-difference schemes is known. For example, if we employ the scheme by Juels et al. [7], and the replacement noise is $t = 3$, then the entropy loss due to the replacement sketch is at least $2t \log_2 |R| < 49.72$. As an approximation, let us assume that the replacement sketch is generated independently from the white noise sketch. Then, the min-entropy given the sketch P_X is at most $61.2 - 49.72 = 11.48$.

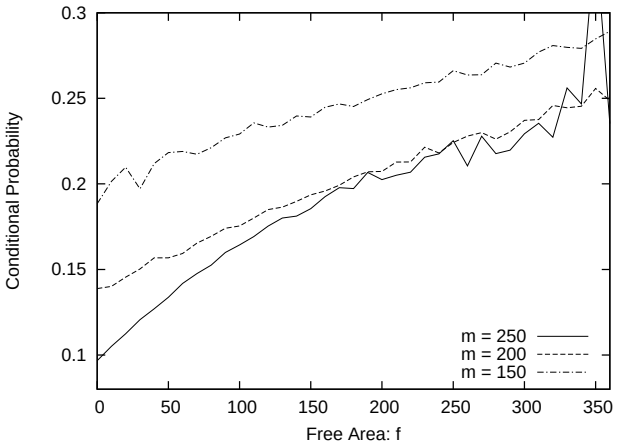

Figure 9: Conditional probability whereby the online parking generates fixed number of chaff points. The figure shows $\Pr(A_x < s|F(x) = f)$ where m, the number of chaff points, is 250, 200,or 150, and $s = 38$. The domain is $[0, n] \times [0, n]$, where n is 20.7.

4.6 Online Parking with Fixed Number of Chaff Points

In previous sections, we employ the first stopping condition for online parking process (that is, the process stops when it is impossible to add any more chaff points). It would be interesting to investigate the second stopping condition that stops at a fixed number of chaff points. We conduct experiment with $m = 250$, 200, and 150, where m is the total number of chaff points. The result is illustrated in Figure 9. Observe that the conditional probability is still increasing although for smaller m, it increases more gradually.

5. CONCLUSION

A known sketch scheme for fingerprint templates hides the minutiae by adding random chaff points. The chaff generation is essentially the online parking process where random points are selected one-by-one. The chaff points are randomly selected and thus seems impossible to be distinguished from the minutiae. However, since the selection of a new point depends on the location of the previously selected points, the online process is not memoryless. Hence, statistical properties of the points that arrive early may be different from the latecomers. We observed that the latecomers tend to have more nearby points. The observation is formulated using free area, and we conjecture that the latecomers are more likely to have smaller free area (inequality (3)). This leads to the use of free area in distinguishing the minutiae from the chaff points.

6. REFERENCES

[1] Fvc2004 databases. *http://biometrics.cse.msu.edu/fvc04db/index.html*.

[2] Boyen, X. Reusable cryptographic fuzzy extractors. In *11th ACM conf. on Computer and Communications Security* (2004), pp. 82–91.

[3] Chang, E.-C., and Li, Q. Small secure sketch for point-set difference. *Cryptology ePrint Archive, Report 2005/145* (2005).

[4] Clancy, T. C., Kiyavash, N., and Lin, D. J. Secure smartcardbased fingerprint authentication. In *ACM SIGMM workshop on Biometrics methods and applications* (2003), pp. 45–52.

[5] Dodis, Y., Reyzin, L., and Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Eurocrypt'04* (2004), pp. 523–540.

[6] Jr., E. C., Flatto, L., and Jelenković, P. Interval packing: the vacant interval distribution. *The Annals of Applied Probability 10*, 1 (2000), 240–257.

[7] Juels, A., and Sudan, M. A fuzzy vault scheme. In *IEEE Intl. Symp. on Information Theory* (2002).

[8] Juels, A., and Wattenberg, M. A fuzzy commitment scheme. In *ACM Conf. on Computer and Communications Security* (1999), pp. 28–36.

[9] Linnartz, J.-P. M. G., and Tuyls, P. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *AVBPA 2003* (2003), pp. 393–402.

[10] Maltoni, D., Maio, D., Jain, A. K., and Prabhakar, S. *Handbook of Fingerprint Recognition.* Springer-Verlag, 2003.

[11] Palasti, I. On some random space filling problems. *Publ. Math. Inst. Hung. Acad. Sci. 5* (1960), 353–359.

[12] Rnyi, A. On a one-dimensional problem concerning random space-filling. *Publ. Math. Inst. Hung. Acad. Sci. 3* (1958), 109–127.

[13] S. Yang, I. V. Secure fuzzy vault based fingerprint verification system. In *38th Asilomar Conf. on Signals, Systems, and Computers* (2004), vol. 1, pp. 577– 581.