On the Guessability of Binary Biometric Templates: A Practical Guessing Entropy based Approach

Guangcan Mai* Meng-Hui Lim† Pong C. Yuen*

Abstract

A security index for biometric systems is essential because biometrics have been widely adopted as a secure authentication component in critical systems. Most of biometric systems secured by template protection schemes are based on binary templates. To adopt popular template protection schemes such as fuzzy commitment and fuzzy extractor that can be applied on binary templates only, non-binary templates (e.g., real-valued, point-set based) need to be converted to binary. However, existing security measurements for binary template based biometric systems either cannot reflect the actual attack difficulties or are too computationally expensive to be practical. This paper presents an acceleration of the guessing entropy which reflects the expected number of guessing trials in attacking the binary template based biometric systems. The acceleration benefits from computation reuse and pruning. Experimental results on two datasets show that the acceleration has more than 6x, 20x, and 200x speed up without losing the estimation accuracy in different system settings.

1. Introduction

1.1. Background

Biometric systems are being widely deployed in critical applications (*e.g.*, border security[1] and banking[2]) and often served as a secure authentication subsystem. It is critical to have a security index for biometric systems [16, 24]. A biometric security index can be used as a criterion to select systems for applications with various security requirements. Moreover, it could help biometric vendors to improve the security of their systems in design phases.

One of the popular biometric system models is client-server based, where the sensor and feature extraction modules are on client, the comparison and storage modules are on server (i.e., model A in ISO/IEC 24745:2011 [9]). It is reasonable to assume that both client and server can be regarded as a black-box, where their inner operations are difficult to be hacked and thus be more secure. Consequently, a biometric system is more vulnerable to attacks on the input of client (a.k.a., presentation or spoofing attack [17, 25]) and the interconnection between client and server (a.k.a., template tapping attack).

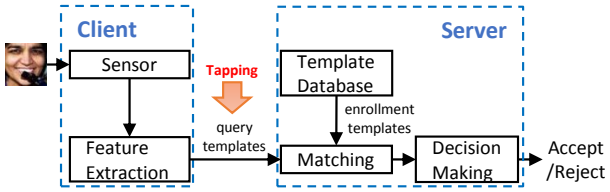


Figure 1. Template tapping attack on a biometric system (comparison and storage on server, model A in ISO/IEC 24745:2011 [9]). Transmission of query templates are assumed to be tapped and attackers gain system access by guessing and submitting query templates as the target user.

In this study, we aim to measure the security of biometric systems under template tapping attack (Fig. 1), where the interconnection between client and server is assumed to be tapped. In case that users in the target biometric system are anonymous, there is no way to obtain the target users' raw biometric data from outsource such as social network (*e.g.*, Facebook and Twitter). To access the target system, attackers are required to either guess target users' raw biometric data for spoofing attack or a query template for template tapping attack. In general, the entropy of a query template is smaller than the entropy of raw biometric data from which the template was extracted. Since the guessing of a query template for target user is easier than raw biometric data, the security of target biometric system is determined by the security under template tapping attack.

Since typical biometric cryptosystems (*e.g.*, fuzzy commitment [7, 10] and fuzzy extractor [4]) accept binary input only, we target at binary template based biometric systems [6, 15, 19]. The scenario of our security study can be described with the adversarial machine learning framework

*Guangcan Mai and Pong C. Yuen are with the Department of Computer Science, Hong Kong Baptist University, Kowloon, Hong Kong. Email:{csgcmai,pcyuen}@Comp.Hkbu.Edu.Hk

†Meng-Hui Lim is with SmartPeep, Masai, Malaysia. Email: menghui.lim@gmail.com

[1]www.theverge.com/2017/4/18/15332742/us-border-biometric-exit-facial-recognition-scanning-homeland-security

[2]www.citibank.com.hk/english/ways-to-bank/voice-biometrics.htm

[2, 18] from the following four perspectives:

- *Adversary's goal:* The attacker aims to impersonate the target user to access target biometric systems.
- *Adversary's knowledge:* The attacker is assumed to know the following information: (a) a *black-box* feature extractor for generating binary templates from raw biometric data, and (b) false accept rate (FAR). Both of knowledges (a) and (b) can be obtained from the vendors of target biometric system. Note that the impostor distance distribution $\mathcal{P}_i$ and the decision threshold τ can be estimated using public domain biometric databases and knowledges (a) and (b). Here, to simplify, we also assume the $\mathcal{P}_i$ and τ are known.
- *Adversary's capability:* With the assumption that the interconnection between client and server is tapped, the attacker is able to insert binary templates as if the templates were transmitted from client. The corresponding matching decision[3] (i.e., either accept or reject) can then be observed. In addition, we assume that the target system does not lock a user after a few failed trials. This is known as rate-limiting policy and could be improperly implemented [22].
- *Adversary's strategy:* Under the above assumptions, the attacker can iteratively guess the most probably binary template for the target user and submit the guessed template to access the system.

1.2. Related Works

State-of-the-art security measurement methods for biometric systems under template tapping attacks can be broadly categorized into either single-acceptable-input based [3, 4, 8, 11] or multiple-acceptable-input based [16, 23, 28]. Single-acceptable-input based methods assume that the target system can be accessed by a query template whose distance to the enrollment template of target users is zero. The degree of freedom (DOF) [3] and the Renyi entropy [8] measure the average difficulties for guessing a template that is identical to the target enrollment template. The min entropy [4, 11] measures the difficulties for guessing the most probable template, not for the target user, among all users in the target system. Note that almost all biometric systems are made of accepting multiple query templates for a target user to tolerate the intra-user variations. Therefore, the single-acceptable-input based methods are inappropriate because they cannot reflect the multiple-accept characteristic of most biometric systems.

Multi-acceptable-based methods assume that the target system can be accessed by multiple query templates. The minimum decoding complexity [19, 23] measures the fractional difficulties under their decoding strategy for finding a template whose distance to the enrollment template of target user is less than the system specific threshold, where the fractional feature (bit) length and threshold are used to address the non-uniform distribution of binary templates. However, it could be inaccurate because the calculation based on fractional feature length and threshold does not reflect the actual attack. The relative entropy [28] measures the average difficulties for guessing templates derived from samples of a target user using templates of different users. It is assumed that all templates from the target user are allowed to access the system. This assumption is again inappropriate for reflecting the actual systems. In addition, the templates with very large intra-user variations lead to an under estimation of system security. The guessing entropy [16, 21] measures the expected number of trials for guessing a template whose distance to the enrollment template of target user is less than the system decision threshold, where a guessing strategy is proposed. The estimation of expected number of guessing trials requires determining both the guessing template and the corresponding success probability at every trial. This is computationally expensive and hence limits the application of guessing entropy.

[3] Different from hill-climbing attacks [5, 20] that the matching score of every attempt is assumed to be observed by attackers, only the matching decision is assumed to be observed in template tapping attack in this study.

1.3. Contributions

To make the guessing entropy [16] practical for real applications, we propose to accelerate the guessing entropy. In summary, this study makes the following contributions:

- We accelerate the guessing entropy [16] with computation reuse and pruning. Large numbers of computations at successive guessing trials are reused with our new formulation for the probability of adversarial success. We prune the guessing codeword generation by deriving an error bound which guarantees an accurate guessing entropy without computational expensively generating all possible code-words.
- We empirically show that our numerical results are consistent with original guessing entropy [16] and demonstrate that our accelerated algorithm works on more-practical systems whose bit length is almost four times the maximum bit length reported in [16]. Note that the computational cost increases exponentially respective to bit length.

1.4. Paper Organization

The basic idea and the guessing strategy of guessing entropy proposed in [16] is briefly described in section 2. Section 3 illustrates our acceleration. The speed up and the accuracy of our acceleration are evaluated in section 4, where our results on more-practical systems are also reported. Finally, we draw some concluding remarks in section 5.

2. Guessing Entropy

2.1. Basic Idea

The guessing entropy [16, 21] is determined by the expected number of guessing trials $E(T)$ for accessing a biometric system as a target user. Mathematically, $E(T)$ can be expressed as [16]:

$$E(T) = \sum_{T=1}^{T_{max}} T \cdot P(X_{trial} = T) \tag{1}$$

where T_{max} denotes the maximum number of guessing trials and $P(X_{trial} = T)$ denotes the probability of first success guessing after taking T guessing trials. In our study, the T_{max} is only determined by the guessing strategy, since we assume that the rate-limiting policy is not/improperly implemented [22] and target system allow arbitrary number of guessing trials. The $P(X_{trial} = T)$ is jointly determined by both guessing strategy and the target system.

2.2. Guessing Strategy

One of the straightforward methods to guess binary templates is to first estimate the occurrence probability for all binary templates and then guess the templates in the descend order of their probabilities. However, due to the large feature space (*e.g.*, 2^n possible templates for bit length n), there are often insufficient collected samples for estimating the template probabilities and insufficient memory for storing these estimated probabilities. Alternatively, Lim and Yuen [16] proposed an impostor distance distribution $\mathcal{P}_i$ based guessing strategy.

In the guessing strategy of Lim and Yuen [16], start with an impostor (adversarial) template X_{adv}, they iteratively find a modification vector $X_{mod}^{(T)}$ at T-th trial and submit the modified template $X_{adv}^{(T)} = X_{adv} \oplus X_{mod}^{(T)}$ to the target system. They use k_1 and k_3 to denote the hamming distance of the enrollment template X_E to the adversarial template X_{adv} and the modified template $X_{adv}^{(T)}$ at T-th trial, resp., and use k_2 to denote the hamming weight of the modification vector $X_{mod}^{(T)}$ at T-th trial. Note that k_1, the hamming distance between the adversarial template X_{adv} and the enrollment template X_E follows the impostor distance distribution $\mathcal{P}_i$. The objective of their guessing strategy in T-th trial is to find an optimal k_2 resulting to the highest probability $P(X_{trial} = T)$ for accessing the target system, i.e, the corresponding k_3 less than the system decision threshold τ.

3. Accelerate Guessing Entropy

The guessing entropy [16] reflects the actual hardness for accessing a binary template based biometric systems in the template attacking scenario we studied. The expensive computational cost of the guessing entropy limits its application in practical systems. In this section, we describe our acceleration of the guessing entropy by (a) reformulating the probability for success adversarial guess at different trials, where computations at successive trials can be reused to reduce the computational cost (section 3.1); (b) giving an error bound of $E(T)$ to reduce guessing trials without influencing the accuracy of estimating $E(T)$ (section 3.2); and (c) implementation details in section 3.3. In this section, unless otherwise stated, we use the notation of k_1, k_2, and k_3, where k_1 denotes the hamming distance between X_{adv} and X_E, k_2 denotes the hamming weight of the modification vector $X_{mod}^{(T)}$, and k_3 denotes the hamming distance between $X_{adv}^{(T)}$ and X_E.

3.1. Reformulate Success Probability

According to the objective of guessing strategy [16] that is to find a k_2 with maximum probability $P(X_{trial} = T)$ for accessing the target system at T-th trial, we have

$$\begin{aligned} P(X_{trial} = T) &= \max_{k_2} P(X_{trial} = T|k_2) \\ &= \max_{k_2} \left(\sum_{k_1=0}^{n} \mathcal{P}_i(k_1) \cdot P(X_{trial} = T|k_1, k_2) \right) \end{aligned} \tag{2}$$

where $P(X_{trial} = T|k_2)$ and $P(X_{trial} = T|k_1, k_2)$ denotes the probability of first success guessing after taking T guessing trials conditioned on k_2, and both k_1 and k_2, respectively. The n denotes the bit length of the binary templates in the target system.

The guessing of the binary template for the target user can be analogized to a sampling without replacement problem, where the submission of previously guessed templates can be avoided by storing them. By definition, the adversary continues to submit a new template until a guess is accepted as the target user in the system. The success of T-th guess implies that the previous $T-1$ guesses are failed. Mathematically,

$$\begin{aligned} &P(X_{trial} = T|k_1, k_2) = \\ &P\left(\text{succ}|k_1, k_2, \mathcal{T}_{k_2}^{(T)}\right) \prod_{t=1}^{T-1} \left(1 - P\left(\text{succ}|k_1, k_2^{(t)}, \mathcal{T}_{k_2^{(t)}}^{(t)}\right)\right) \end{aligned} \tag{3}$$

where $P\left(\text{succ}|k_1, k_2, \mathcal{T}_{k_2}^{(T)}\right)$ denotes the probability for success guessing at T-th trial conditioned on both k_1 and k_2. $\mathcal{T}_{k_2}^{(T)}$ denotes the number of modification vectors with hamming weight k_2 which were guessed before T-th trials. $k_2^{(t)}$ denotes the hamming weight of modification vector guessed in the t-th trial.

For T-th trial, let $m(k_1, k_2)$ denote number of k_2-hamming-weight modification vectors which results in a

system accept, and $\mathcal{N}\left(k_1, k_2, \mathcal{T}_{k_2}^{(T)}\right)$ denote number of non-guessed k_2-hamming-weight modification vectors, we have

$$P\left(\text{succ}|k_1, k_2, \mathcal{T}_{k_2}^{(T)}\right) = \frac{m\left(k_1, k_2\right)}{\mathcal{N}\left(k_1, k_2, \mathcal{T}_{k_2}^{(T)}\right)} \tag{4}$$

A bit in modification vector with value '1' could either modify bits in the X_{adv} to increase or decrease the resultant hamming distance, k_3. We assume that there are p and k_2-p modification bit to decrease and increase the resultant hamming distance, resp., while applying a modification vector $X_{mod}^{(T)}$ to the adversarial template X_{adv}. The corresponding resultant hamming distance k_3 can be expressed as

$$k_3 = k_1 - p + (k_2 - p) = k_1 + k_2 - 2p \tag{5}$$

The number of modification vectors that introduce resultant hamming distance to k_3 is

$$\binom{k_1}{p}\binom{n-k_1}{k_2-p} = \binom{k_1}{\frac{k_1+k_2-k_3}{2}}\binom{n-k_1}{\frac{k_2+k_3-k_1}{2}} \tag{6}$$

Consequently,

$$m\left(k_1, k_2\right) = \sum_{k_3=k_{3l}}^{k_{3u}} \binom{k_1}{\frac{k_1+k_2-k_3}{2}}\binom{n-k_1}{\frac{k_2+k_3-k_1}{2}} \tag{7}$$

where $k_{3l} = |k_1 - k_2|$ denote the minimum achievable value of k_3 and $k_{3u} = \min\{k_1 + k_2, 2n - k_1 - k_2, \tau\}$ denotes the maximum k_3 that will be accepted by the system. Both k_{3u} and k_{3l} are derived conditions:

$$\begin{cases} 0 \leq p \leq k_1 \\ 0 \leq k2 - p \leq n - k_1 \\ 0 \leq k_3 \leq \tau \end{cases} \tag{8}$$

Similarity,

$$\mathcal{N}\left(k_1, k_2, \mathcal{T}_{k_2}^{(T)}\right) = \binom{n}{k_2} - \mathcal{T}_{k_2}^{(T)} \tag{9}$$

Note that $P\left(X_{trial} = T|k_1, k_2\right)$ in Eq. (3) can be computed from $P\left(X_{trial} = T-1|k_1, k_2\right)$, where the details are described in section 3.3. Therefore, large numbers of computations can be reduced to accelerate the algorithm.

3.2. Error Bound

The basic idea of the error bound base on that the entropy of uniformly distributed templates higher than non-uniformly distributed templates. Mathematically,

$$E(T) \leq \hat{E}(T) \tag{10}$$

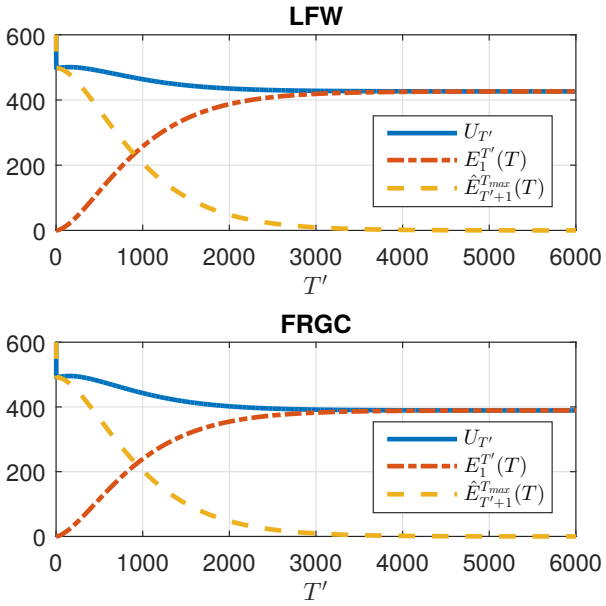


Figure 2. The error bound $\hat{E}_{T'+1}^{T_{max}}(T)$ and the error bound of $E(T)$ determined at T'-th trial, where $U_{T'} = E_1^{T'}(T) + \hat{E}_{T'+1}^{T_{max}}(T)$. These results are based on 36-bit binary templates which generated from LBP+PCA+LSSC [14] on both LFW [12] and FRGC v2.0 [26] datasets, where the matching threshold is '7' and the impostor distance distribution $\mathcal{P}_i$ are generated with the first trial (out of ten cross-validations) of BLUFR verification protocol [13].

where $\hat{E}(T)$ denotes minimum expected number of trials in guessing uniformly distributed templates. Let $\hat{P}\left(X_{trial} = T\right)$ denote the corresponding probability of first success after taking T guessing trials, for $T' \geq 1$, Eq. (10) can alternatively be expressed as

$$E_1^{T'}(T) + E_{T'+1}^{T_{max}}(T) \leq \hat{E}_1^{T'}(T) + \hat{E}_{T'+1}^{T_{max}}(T) \tag{11}$$

where

$$\begin{aligned} E_i^j(T) &= \sum_{T=i}^{j} T \cdot P\left(X_{trial} = T\right) \\ \hat{E}_i^j(T) &= \sum_{T=i}^{j} T \cdot \hat{P}\left(X_{trial} = T\right) \end{aligned} \tag{12}$$

The error of $E(T)$ after T'-th trial is $E_{T'+1}^{T_{max}}(T)$, which can be shown that

$$ERR_{T'} = E_{T'+1}^{T_{max}}(T) \leq \hat{E}_{T'+1}^{T_{max}}(T) \tag{13}$$

Let $m_{k_1} = \sum_{k_2=0}^{n} m\left(k_1, k_2\right)$ and $\mathcal{N}_{k_1} = 2^n$, it can also be shown that

$$\hat{E}_{T'+1}^{T_{max}}(T) = \sum_{k_1=0}^{n} \mathcal{P}_i(k_1) \cdot \sigma_{k_1}^{T'}\left(\frac{\mathcal{N}_{k_1} - T' + 1}{m_{k_1} + 1} + T'\right) \tag{14}$$

where

$$\sigma_{k_1}^{T'} = \prod_{T=1}^{T'} \left(1 - \frac{m_{k_1}}{\mathcal{N}_{k_1} - T + 1}\right) \tag{15}$$

Moreover, the error bound $\hat{E}_{T'+1}^{T_{max}}(T)$ is decreasing while the number of trials T' increases (Fig. 2). It is observed that the upper bound of $E(T)$ determined at T'-th trial, $U_{T'}$ converges to the $E(T)$ as T' approaches to infinite.

3.3. Implementation Details

Our implementation of the accelerated guessing entropy is described in Algorithm 1. The $\mathcal{P} = \{p_{k_1k_2}\}$ records the $P(X_{trial} = T|k_1, k_2)$ as expressed in Eq. (3). The $\mathcal{N} = \{n_{k_1k_2}\}$ records the number of remaining modification vector of hamming weight k_2 as expressed in Eq. (9). The basic idea of the algorithm is, in each trial (iteration) T, determining hamming weight k_2chose (line[4] 7) of the modification vector resulting to the highest guessing probability $P(X_{trial} = T) = \mathcal{P}_{k_2}(k_2chose)$. The expected number of guessing trials E_T is initialized to zero and increased by $T \cdot P(X_{trial} = T)$ in each trial (line 9).

In our algorithm, as long as k_1 is a possible hamming distance between the enrollment and the adversarial templates, $\mathcal{P} = \{p_{k_1k_2}\}$ is first updated as (line 17):

$$p_{k_1k_2} = \begin{cases} p_{k_1k_2} \cdot \dfrac{n_{k_1k_2chose} - m_{k_1k_2chose}}{n_{k_1k_2chose}}, & k_2 \neq k_2chose; \\ p_{k_1k_2} \cdot \dfrac{n_{k_1k_2chose} - m_{k_1k_2chose}}{n_{k_1k_2chose} - 1}, & k_2 = k_2chose. \end{cases} \tag{16}$$

and $\mathcal{N} = \{n_{k_1k_2}\}$ is then updated as (line 17):

$$n_{k_1k_2} = \begin{cases} n_{k_1k_2}, & k_2 \neq k_2chose; \\ n_{k_1k_2} - 1, & k_2 = k_2chose. \end{cases} \tag{17}$$

Note that update $\mathcal{P} = \{p_{k_1k_2}\}$ and $\mathcal{N} = \{n_{k_1k_2}\}$ in this way reuses the computations at previous trials. In addition, the pruning implemented at lines 11 and 12 could greatly reduce the number of guessing trials T with a pre-specific error tolerance tol. It could be empirically shown that even with the pruning, the generated guessing series S_{k_2} provides sufficient k_2 for guessing most of the enrollment templates of target users. Both the computation reuse and the pruning mentioned above largely reduce the computational efforts.

Algorithm 1: Guessing sequence generation

Input: bit length n, decision threshold τ, impostor distance distribution $\mathcal{P}_i$, and error tolerance tol
Output: a guessing sequence S_{k_2}, and the expected number of guessing trials E_T

```
// τ is used in m(k1, k2) (7)
```
1 $\mathcal{M} \leftarrow \{m_{k_1k_2} = m(k_1, k_2)\}, k_1, k_2 = \{0, \cdots, n\}$
2 $\mathcal{N} \leftarrow \{n_{k_1k_2} = 2^{k_2}\}, k_1, k_2 = \{0, \cdots, n\}$
3 $\mathcal{P} \leftarrow \{p_{k_1k_2} = \frac{m_{k_1k_2}}{n_{k_1k_2}}\}, m_{k_1k_2} \in \mathcal{M}, n_{k_1k_2} \in \mathcal{N}$
4 $T \leftarrow 1$; $E_T \leftarrow 0$; $flagk_1 \leftarrow \mathbf{1}^{n+1}$
5 **while** $(\sum_{k_1} \mathcal{P}_i(k_1) \geq 0)$ **do**
6 $\quad \mathcal{P}_{k_2} \leftarrow \text{transpose}(\mathcal{P}) \times \mathcal{P}_i$; `// dot product`
7 $\quad k_2chose \leftarrow \arg_{k_2} \max \mathcal{P}_{k_2}(k_2)$;
8 $\quad$ Append k_2chose to S_{k_2};
9 $\quad E_T \leftarrow E_T + T \cdot \mathcal{P}_{k_2}(k_2chose)$;
10 $\quad errBound \leftarrow \hat{E}_{T+1}^{T_{max}}(T)$ (14);
11 $\quad$ **if** $errBound \leq tol$ **then**
12 $\quad\quad$ return S_{k_2}, and E_T;
13 $\quad T \leftarrow T + 1$;
14 $\quad$ **for** $(k_1 = 0; k_1 \leq n; k_1 \leftarrow k_1 + 1)$ **do**
15 $\quad\quad$ **if** $flagk_1(k_1) == 1$ **then**
16 $\quad\quad\quad$ **if** $\mathcal{N}(k_1, k_2chose) > \mathcal{M}(k_1, k_2chose)$ **then**
17 $\quad\quad\quad\quad$ Update $\mathcal{P}$, and $\mathcal{N}$;
18 $\quad\quad\quad$ **else**
19 $\quad\quad\quad\quad flagk_1(k_1) \leftarrow 0$;
20 $\quad\quad\quad\quad \mathcal{P}_i(k_1) \leftarrow 0$;
21 $\quad\quad\quad\quad \forall k_2, \mathcal{N}_i(k_1, k_2) \leftarrow 0$;
22 return S_{k_2}, and E_T;

4. Experiments

Datasets: Our acceleration of the guessing entropy [16] has been evaluated on two face benchmarking datasets, i.e., LFW [12] and Face Recognition Grand Challenge (FRGC) v2.0 [26]. LFW [12] consists of 13,233 images of 5,749 subjects downloaded from the web. For the FRGC v2.0 [26], we use 16,028 images of 466 subjects (as specified in the target set of Experiment 1 [26]). The verification protocol BLUFR [13] is used.

Feature extraction: We use three different kinds of features from the face images, where the preprocess for these three features are

- PCA + LSSC [14]: The gray face images are used and first aligned using two eyes and nose. We then crop the aligned images to 96×96 pixels and vectorized them.
- LBP+PCA+LSSC: The LBP feature we use are provided by [13] [5].
- FaceNet [27]+PCA+LSSC: The FaceNet feature are extracted using an open source implementation [1] [6].

[4] All lines in this subsection are in Algorithm 1, unless otherwise stated.

[5] http://www.cbsr.ia.ac.cn/users/scliao/projects/blufr/

[6] https://github.com/cmusatyalab/openface

Table 1. Expected number of trials $E(T)$ and the corresponding computational time (s) for PCA+LSSC on LFW

	$n(\tau)$	8(0)	12(1)	16(2)
$E(T)$	Ours	108.48	264.12	432.27
	Lim & Yuen [16]	108.48	264.12	432.27
Time	Ours	0.05	0.57	3.93
	Lim & Yuen [16]	0.36	12.57	901.46

Table 2. Expected number of trials $E(T)$ and the corresponding computational time (s) for LBP+PCA+LSSC on LFW

	$n(\tau)$	8(0)	12(1)	16(2)
$E(T)$	Ours	108.58	264.17	433.92
	Lim & Yuen [16]	108.58	264.17	433.92
Time	Ours	0.06	0.57	4.11
	Lim & Yuen [16]	0.49	12.52	895.19

Table 3. Expected number of trials $E(T)$ and the corresponding computational time (s) for PCA+LSSC on FRGC

	$n(\tau)$	8(0)	12(1)	16(2)
$E(T)$	Ours	106.21	263.57	428.64
	Lim & Yuen [16]	106.21	263.57	428.64
Time	Ours	0.05	0.56	3.91
	Lim & Yuen [16]	0.34	12.82	926.35

Table 4. Expected number of trials $E(T)$ and the corresponding computational time (s) for LBP+PCA+LSSC on FRGC

	$n(\tau)$	8(0)	12(1)	16(2)
$E(T)$	Ours	104.29	260.26	420.58
	Lim & Yuen [16]	104.29	260.26	420.58
Time	Ours	0.06	0.64	4.25
	Lim & Yuen [16]	0.36	12.92	958.32

The PCA is then performed on the preprocessed data with number of principle components $n/2$, where n denotes final bit length. We encode each feature component to a 2-bit binary vector using LSSC [14] and obtain templates with bit length n.

Parameters: The impostor hamming distance $\mathcal{P}_i$ is assumed to be known in the attack and is obtained according to the BLUFR verification protocol [13]. The decision threshold τ is set to either $(\frac{n}{4} - 2)$ or $\frac{n}{4}$, given by most of error-correcting-code based template protection scheme (*e.g.*, [4, 10]) can tolerate up to $\sim 25\%$ bit errors. The error tolerance of $E(T)$ is set as $tol = 10^{-5}$. Note that there are ten cross-validations in the BLUFR protocol [13] and we report the average results.

4.1. Speed Up and Accuracy

In this section, we evaluate the accuracy and the corresponding computational time of our accelerated and the original guessing entropy [16]. Our accelerated guessing entropy is implemented with Python 2.7, and the original guessing entropy[7] [16] is implemented with MATLAB R2015a. Both of our accelerated and the original guessing entropy are run with machines of same configurations, i.e., dual Intel Xeon X5650 @ 2.67GHz (CPU) with 32GB memory.

The expected number of trials for guessing templates of PCA+LSSC and LBP+PCA+LSSC on LFW dataset are shown in tables 1 and 2, respectively. Tables 3 and 4 show the corresponding results on FRGC v2.0 dataset. Due to the long running time of the original guessing entropy algorithm, the results of bit length less than 16 are shown. It is observed that the expected number of trials given by both our accelerated and the original guessing entropy are the same. However, our accelerated guessing entropy has more than 6x, 20x, and 200x speed up compared with the original guessing entropy while bit lengths (thresholds) are 8(0), 12(1), and 16(2), respectively.

4.2. More-practical Applications

After evaluating the accuracy of accelerated guessing entropy, we present the expected number of guessing trials with more practical system settings. Note that the original guessing entropy [16] reports results for systems with bit length up to 18 bits, due to the expensive computational efforts that increase exponentially respective to bit length. A practical system typically has feature length much larger than 18 bits, in this section, we investigate the number of guessing trials for more-practical systems whose template length ranges from 32 to 64.

The expected number of trials for guessing templates of PCA+LSSC, LBP+PCA+LSSC, and FaceNet+PCA+LSSC on LFW and FRGC v2.0 datasets are shown in Fig. 3 and Fig. 4, respectively. Fig. 3(a) and Fig. 4(a) show the results of threshold $\tau = \frac{n}{4} - 2$. Fig. 3(b) and Fig. 4(b) show the results of threshold $\tau = \frac{n}{4}$. The results of upper bound denotes the theoretical expected number of trials in guessing uniformly distributed templates [16], which are

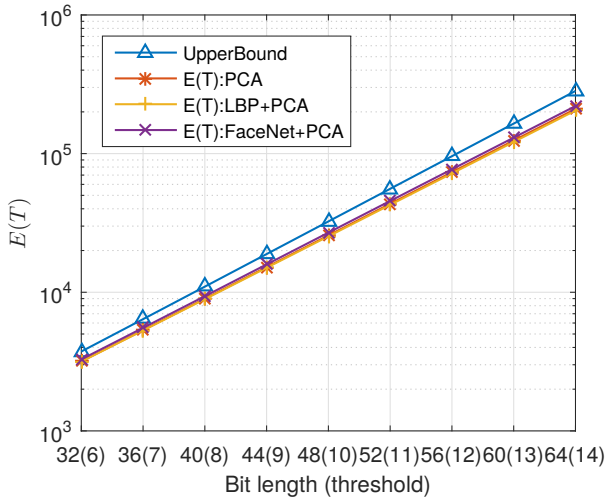
$$\hat{E}(T) = \frac{1 + 2^n}{1 + \sum_{i=0}^{\tau} \binom{n}{i}} \tag{18}$$

It is observed that the templates of FaceNet+PCA+LSSC is most difficult to be guessed, and then the templates of PCA+LSSC and LBP+PCA+LSSC.

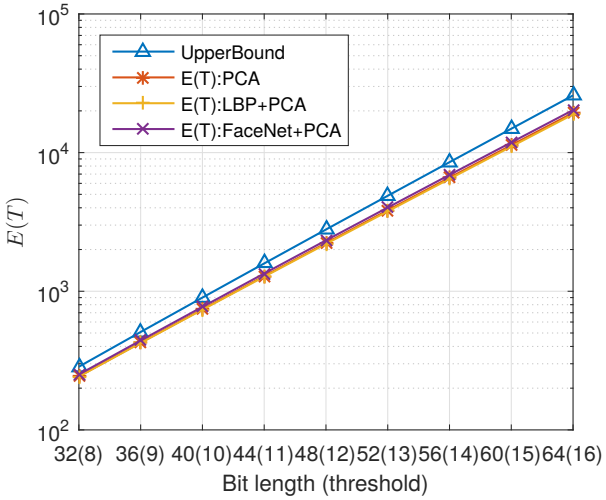
5. Conclusions

We have accelerated the guessing entropy [16] for measuring the security of binary templates based biometric systems. The acceleration based on both computation reuse and pruning. We reuse computations in successive iterations/trials by the newly formulated probability for adversarial success. The pruning is done by our proposed error

[7] The codes are provided by the authors

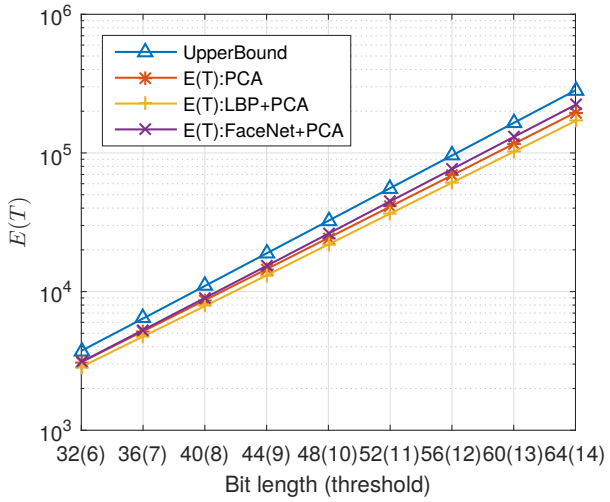


(a) $\tau = \frac{n}{4} - 2$

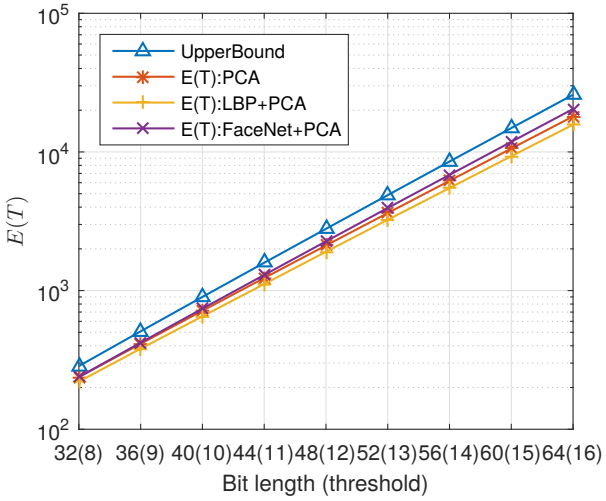


(b) $\tau = \frac{n}{4}$

Figure 3. Expected number of guessing trials $E(T)$ on LFW



(a) $\tau = \frac{n}{4} - 2$



(b) $\tau = \frac{n}{4}$

Figure 4. Expected number of guessing trials $E(T)$ on FRGC

bound for the expected number of guessing trials, where the error can be guaranteed within the tolerance. The experiments conducted on two face benchmarking datasets justify that our acceleration does not lose the accuracy and has more than 6x, 20x, and 200x speed up on systems with bit lengths (thresholds) 8(0), 12(1), and 16(2), respectively. Note that the computational efforts for the guessing entropy increases exponentially respect to the feature length. We demonstrate the application of the accelerated guessing entropy on more-practical systems, whose binary templates with length ranges from 32 to 64 bits.

We have already push the guessing entropy [16] forward to be feasible for binary templates with length 64 bits within a reasonable computation time. However, template size of practical systems could be thousands. The guessing entropy remains unfeasible for these systems even with our acceleration. The future works along this direction is to further accelerate the guessing entropy by (a) searching a tighter error bound, and (b) using emerging hardware such as GPU and FPGA.

Acknowledgments

This study was partially supported by a Hong Kong RGC grant (HKBU 12201414) and the Madam Kwok Chung Bo Fun Graduate School Development Fund, HKBU. The authors would like to thank Mr. Jiawei Li for his helpful suggestions.

References

[1] B. Amos, B. Ludwiczuk, and M. Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, 2016.

[2] B. Biggio, P. Russu, L. Didaci, and F. Roli. Adversarial biometric recognition: A review on biometric system security from the adversarial machine-learning perspective. *IEEE Signal Processing Magazine*, 32(5):31–41, 2015.

[3] J. Daugman. The importance of being random: statistical principles of iris recognition. *Pattern recognition*, 36(2):279–291, 2003.

[4] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM journal on computing*, 38(1):97–139, 2008.

[5] Y. C. Feng, M.-H. Lim, and P. C. Yuen. Masquerade attack on transform-based binary-template protection based on perceptron learning. *Pattern Recognition*, 47(9):3019–3033, 2014.

[6] Y. C. Feng and P. C. Yuen. Binary discriminant analysis for generating binary face template. *Information Forensics and Security, IEEE Transactions on*, 7(2):613–624, 2012.

[7] Y. C. Feng, P. C. Yuen, and A. K. Jain. A hybrid approach for generating secure and discriminating face template. *Information Forensics and Security, IEEE Transactions on*, 5(1):103–117, 2010.

[8] S. Hidano, T. Ohki, and K. Takahashi. Evaluation of security for biometric guessing attacks in biometric cryptosystem using fuzzy commitment scheme. In *Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6. IEEE, 2012.

[9] ISO/IEC 24745:2011 Information technology – Security techniques – Biometric information protection. Standard, International Organization for Standardization, June 2011.

[10] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security*, pages 28–36, 1999.

[11] N. Karimian, Z. Guo, M. Tehranipoor, and D. Forte. Highly reliable key generation from electrocardiogram (ecg). *IEEE Transactions on Biomedical Engineering*, 2016.

[12] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua. Labeled faces in the wild: A survey. In *Advances in Face Detection and Facial Image Analysis*, pages 189–248. Springer, 2016.

[13] S. Liao, Z. Lei, D. Yi, and S. Z. Li. A benchmark study of large-scale unconstrained face recognition. In *IEEE International Joint Conference on Biometrics*, pages 1–8, 2014.

[14] M.-H. Lim and A. B. J. Teoh. A novel encoding scheme for effective biometric discretization: Linearly separable subcode. *IEEE Transactions on pattern analysis and machine intelligence*, 35(2):300–313, 2013.

[15] M.-H. Lim, S. Verma, G. Mai, and P. C. Yuen. Learning discriminability-preserving histogram representation from unordered features for multibiometric feature-fused-template protection. *Pattern Recognition*, 60:706–719, 2016.

[16] M.-H. Lim and P. C. Yuen. Entropy measurement for biometric verification systems. *IEEE Transactions on Cybernetics*, 46(5):1065–1077, 2016.

[17] S. Liu, P. C. Yuen, S. Zhang, and G. Zhao. 3d mask face anti-spoofing with remote photoplethysmography. In *European Conference on Computer Vision*, pages 85–100. Springer, 2016.

[18] G. Mai, K. Cao, P. C. Yuen, and A. K. Jain. Face image reconstruction from deep templates. *preprint arXiv:1703.00832*, 2017.

[19] G. Mai, M.-H. Lim, and P. C. Yuen. Binary feature fusion for discriminative and secure multi-biometric cryptosystems. *Image and Vision Computing*, 58:254–265, 2017.

[20] E. Maiorana, G. E. Hine, and P. Campisi. Hill-climbing attacks on multibiometrics recognition systems. *IEEE Transactions on Information Forensics and Security*, 10(5):900–915, 2015.

[21] J. L. Massey. Guessing and entropy. In *IEEE International Symposium on Information Theory*, page 204, 1994.

[22] W. Melicher, B. Ur, S. M. Segreti, S. Komanduri, L. Bauer, N. Christin, and L. F. Cranor. Fast, lean and accurate: Modeling password guessability using neural networks. In *Proceedings of USENIX Security*, 2016.

[23] A. Nagar, K. Nandakumar, and A. K. Jain. Multibiometric cryptosystems based on feature-level fusion. *Information Forensics and Security, IEEE Transactions on*, 7(1):255–268, 2012.

[24] K. Nandakumar and A. K. Jain. Biometric template protection: Bridging the performance gap between theory and practice. *IEEE Signal Processing Magazine*, 32(5):88–100, 2015.

[25] K. Patel, H. Han, and A. K. Jain. Secure face unlock: Spoof detection on smartphones. *IEEE Transactions on Information Forensics and Security*, 2016.

[26] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 947–954, 2005.

[27] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015.

[28] K. Takahashi and T. Murakami. A measure of information gained through biometric systems. *Image and Vision Computing*, 32(12):1194–1203, 2014.