Supplementary Material:
Analysis of Master Vein Attacks on Finger Vein Recognition Systems

Huy H. Nguyen¹, Trung-Nghia Le¹,³,⁴, Junichi Yamagishi¹, and Isao Echizen¹,²
¹National Institute of Informatics, Tokyo, Japan ²The University of Tokyo, Tokyo, Japan
³University of Science, VNU-HCM, Vietnam ⁴Vietnam National University, Ho Chi Minh City, Vietnam
{nhhuy,jyamagis,iechizen}@nii.ac.jp

This Supplementary Material is organized as follows. First, we provide the detail of the latent variable algorithm (LVE) used to craft master veins in section 1. Next, we include some additional visualizations of real and generated finger veins in different settings in section 2. Last, we present an ablation study on top-k AdvML attack in section 3.

1. Latent Variable Evolution Algorithm

The LVE algorithm, which is used by the LVE-based methods to generate master veins (visualized in Fig. 3 in the main paper), is described in detail by Alg. 1. Regarding the implementation of the CMA-ES, we used the pycma library¹. For simplicity, we used its default parameters.

Algorithm 1 Latent variable evolution.

\[ m \leftarrow 18 \]  \hspace{1cm} \triangleright \text{Population size, default value by the pycma library.}

\[ \text{procedure } \text{RUNLVE}(m, n) \]
\[ \text{MasterVeins } = \emptyset \]
\[ \text{Scores } = \emptyset \]
\[ z \leftarrow \text{rand()} \]
\[ \text{for } n \text{ loops do} \]
\[ V \leftarrow p_\theta(z) \]
\[ s \leftarrow 0 \]
\[ \text{for each vein image } V_i \text{ in } V \text{ do} \]
\[ \text{for each vein image } D_j \text{ in database } D \text{ do} \]
\[ s_i \leftarrow s_i + \text{Matcher}(V_i, D_j) \]
\[ s_i \leftarrow \frac{s_i}{|D|} \]
\[ V_b, s_b \leftarrow \text{GetBestVeinImage}(V, s) \]
\[ \text{MasterVeins } \leftarrow \text{MasterVeins } \cup \{V_b\} \]
\[ \text{Scores } \leftarrow \text{Scores } \cup \{s_b\} \]
\[ z \leftarrow \text{CMA-ES}(s) \]
\[ V_{gb}, s_{gb} \leftarrow \text{GetBestVeinImage}(\text{MasterVeins, Scores}) \]
\[ \text{return } V_{gb}, s_{gb} \]  \hspace{1cm} \triangleright \text{Identify global best master vein image}

2. Additional Visualizations of Generated Finger Veins

Additional samples of real finger veins and those generated by the LVE-based methods are shown in Fig. 1. Effects of the number of iterations and filters used by the AdvML method on the quality of adversarial master veins are visualized in Figs. 2 and 3, respectively.

¹https://github.com/CMA-ES/pycma
Figure 1. Examples of real finger veins (first row) and those generated by our proposed method (LVE\(^3\), second row), \(\beta\)-VAE (LVE\(^2\), third row), and WGAN-GP (LVE\(^1\), last row). Since latent codes are sampling from noise, generated images have the randomness property. Among synthetic finger veins, those generated by our method had the best quality, while those generated by WGAN-GP do not look like finger veins.

Figure 2. Effect of the number of iterations on the quality of adversarial master veins.
3. Ablation Study on Top-\(k\) AdvML Attack

Examples of adversarial master veins generated with different \(k\) values in the AdvML attack with top-\(k\) labels are shown in Fig. 4. There are no significant differences in the amounts of perturbations between them. The relationship between \(k\) and FARs is visualized in Fig. 5. There are no significant differences in the FARs, especially when \(k\) is in the [5\%, 60\%] range. In practice, it is better to avoid the extreme values of \(k\). If \(k\) is too small, its attack ability is limited. If \(k\) is too large, it makes the optimization hard to converge.

![Figure 5. Relationship between \(k\) and FARs in top-\(k\)-label targeted attack. FARs were calculated using the ResNeXt-50-based recognition system on the training set of the SDUMLA-HMT database.](image-url)