Random Permutation Maxout (RPM) transform

Andrew Beng Jin Teoh*, Sejung Cho, Jihyeon Kim

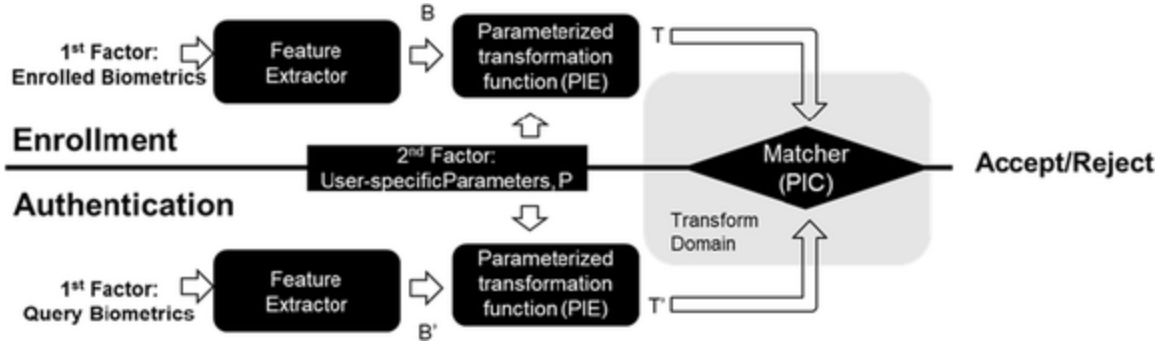
Random permutation Maxout transform for cancellable facial template protection. Multimedia Tools Appl. 77(21): 27733-27759 (2018)

Motivation

- 生物特征是不变的，表达主体丰富的生物信息
- 如果主体的生物特征被盗，可能会发生冒名顶替和隐私入侵的情况

Cancellable biometrics construct

- Pseudonymous Identifier Encoder (PIE)
  - Generate the corresponding protected template $T = PIE(B, P)$
  - Parameter: a biometric signal or template B, an user-specific parameter P
- Pseudonymous Identifier Comparator (PIC)
  - Compare pairs of Ts during authentication



Methodology description (Overview)

Suppose $\boldsymbol{x} \in \mathbb{R}^d$ be a biometric feature vector with length d, RPM transform is carried out in the following steps:

1. Generate p user-specific token-seeded stacked m permutation array $P_i \in \{0,1\}^{d\times d\times m}, i = 1, \dots, p$ where m is the desired length of resulting RPM transformed vector. Note that a permutation matrix is a square binary matrix that has exactly one entry of 1 in each row and each column and 0s elsewhere.
2. Multiplying $\boldsymbol{x}$ to P_i, yield a matrix, $\boldsymbol{V}_i = \boldsymbol{x}P_i \in \mathbb{R}^{d\times m}, i = 1, \dots, p$.
3. Perform Hadamard product (element-wise product) yields $\boldsymbol{Y} = \boldsymbol{V}_1 \circ \cdots \circ \boldsymbol{V}_p = \prod_{i=1}^{p} \boldsymbol{V}_i \in^{d\times m}$ where p is known as Hadamard product order.
4. Discard last $d - k$ column vectors from $\boldsymbol{Y}$, yield $\boldsymbol{Y}' \in \mathbb{R}^{k\times m}$, where we named k as truncation size.
5. For each row vectors of $\boldsymbol{Y}'$, the indices of the largest magnitude entry are recorded and form a discrete RPM transformed vector, $\boldsymbol{z} \in [1\ k]^m$.

Methodology description (Step 0)

Suppose $\boldsymbol{x} \in \mathbb{R}^d$ be a biometric feature vector with length d.

$d = 5$

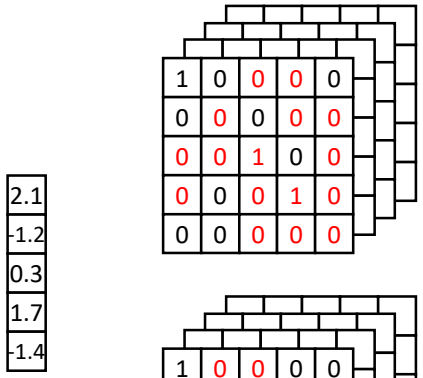
2.1
-1.2
0.3
1.7
-1.4

$\boldsymbol{x} \in \mathbb{R}^5$

Methodology description (Step 1)

$p = 2$ $m = 4$

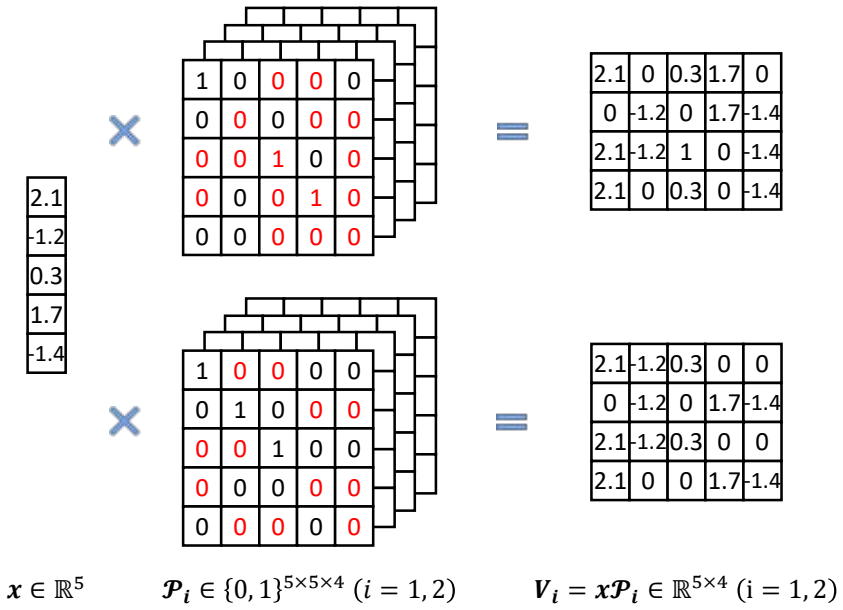
Generate p user-specific token-seeded stacked m permutation array $P_i \in \{0,1\}^{d \times d \times m}, i = 1, \ldots, p$ where m is the desired length of resulting RPM transformed vector. Note that a permutation matrix is a square binary matrix that has exactly one entry of 1 in each row and each column and 0s elsewhere.



$\boldsymbol{x} \in \mathbb{R}^5$ $\boldsymbol{\mathcal{P}_i} \in \{0,1\}^{5\times5\times4}$ $(i = 1, 2)$

Methodology description (Step 2)

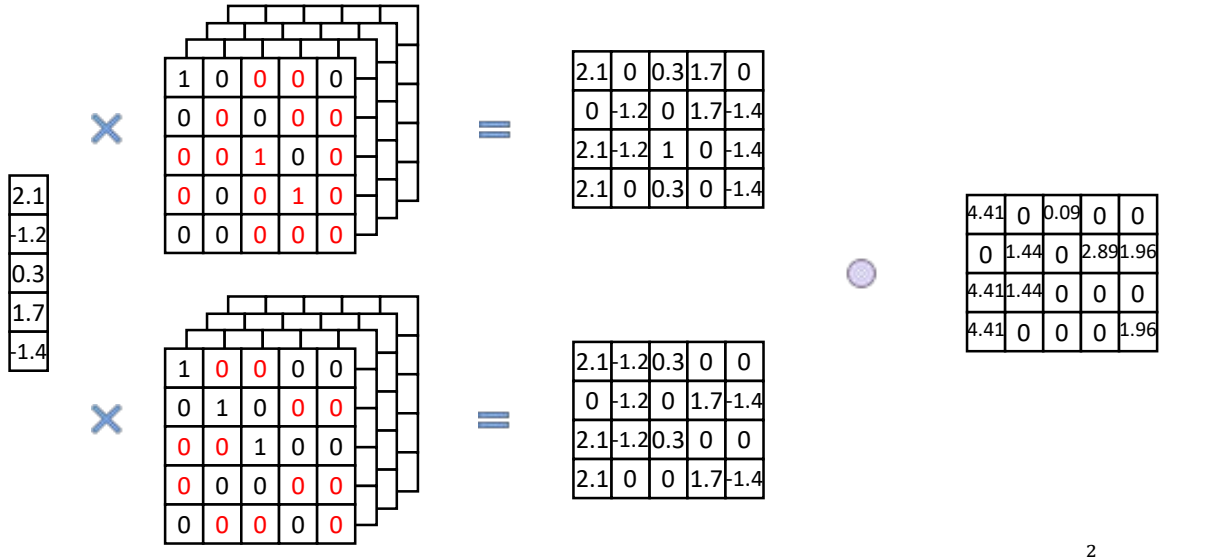
Multiplying $\boldsymbol{x}$ to P_i, yield a matrix, $\boldsymbol{V}_i = \boldsymbol{x}P_i \in \mathbb{R}^{d\times m}, i = 1, \ldots, p$.



Methodology description (Step 3)

Perform Hadamard product (element-wise product) yields $\boldsymbol{Y} = \boldsymbol{V}_1 \circ \cdots \circ \boldsymbol{V}_p = \prod_{i=1}^{p} \boldsymbol{V}_i \in^{d \times m}$ where p is known as Hadamard product order.

$p = 2$

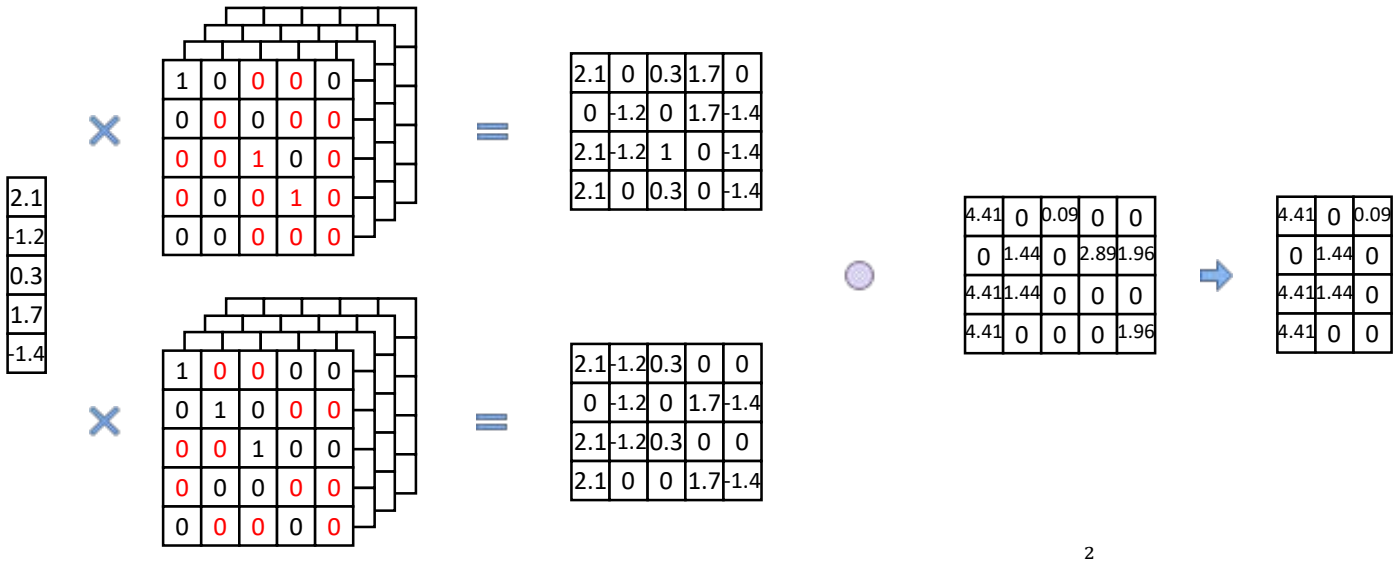


$\boldsymbol{x} \in \mathbb{R}^5$ $\quad \boldsymbol{\mathcal{P}_i} \in \{0,1\}^{5\times5\times4}\ (i = 1,2)$ $\quad \boldsymbol{V_i} = \boldsymbol{x}\boldsymbol{\mathcal{P}_i} \in \mathbb{R}^{5\times4}\ (\mathrm{i} = 1,2)$ $\quad \boldsymbol{Y}^T = V_1 \circ V_2 = \prod_{i=1}^{2} V_i \in \mathbb{R}^{4\times5}$

Methodology description (Step 4)

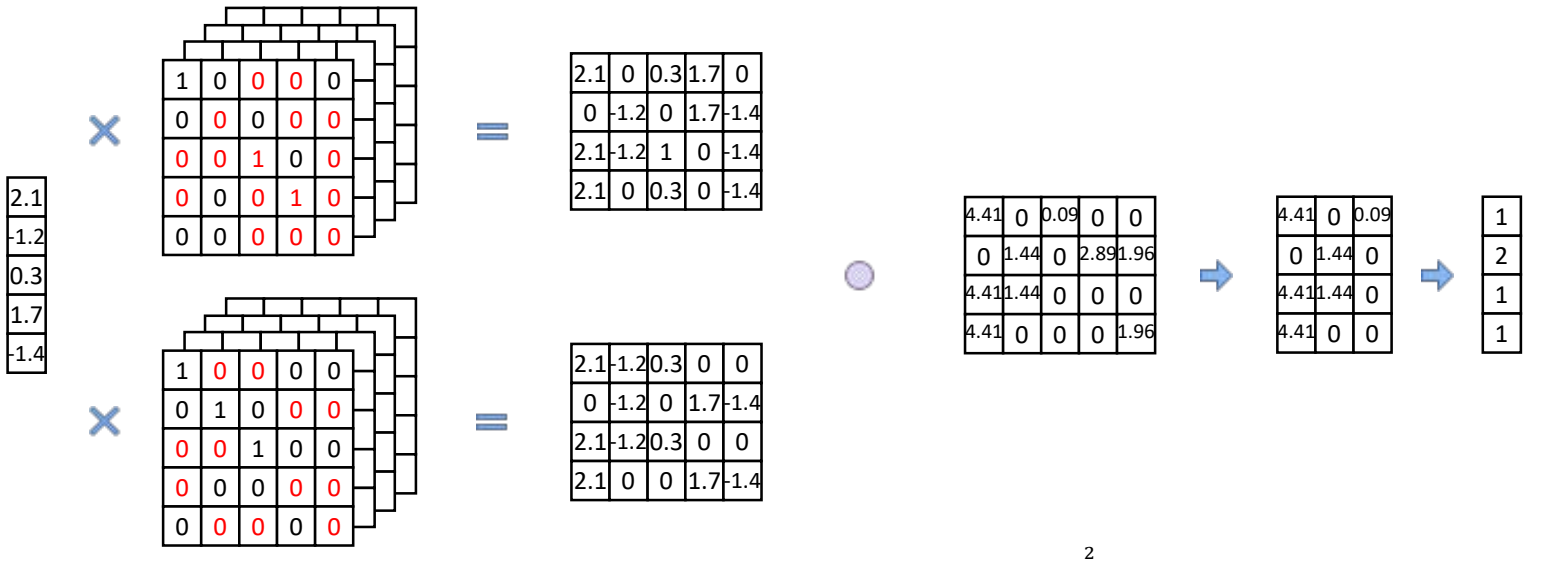
$d - k = 5 - 3 = 2$ $k = 3$

Discard last $d - k$ column vectors from $\boldsymbol{Y}$, yield $\boldsymbol{Y'} \in \mathbb{R}^{k \times m}$, where we named k as truncation size.



Methodology description (Step 5)

For each row vectors of $\boldsymbol{Y}'$, the indices of the largest magnitude entry are recorded and form a discrete RPM transformed vector, $\boldsymbol{z} \in [1\ k]^m$.



$\boldsymbol{x} \in \mathbb{R}^5$ $\quad \boldsymbol{\mathcal{P}_i} \in \{0,1\}^{5\times5\times4}\ (i = 1,2)$ $\quad \boldsymbol{V_i} = \boldsymbol{x}\boldsymbol{\mathcal{P}_i} \in \mathbb{R}^{5\times4}\ (\mathrm{i} = 1,2)$ $\quad \boldsymbol{Y}^T = V_1 \circ V_2 = \prod_{i=1}^{2} V_i \in \mathbb{R}^{4\times5}$ $\quad Y' \in^{4\times3}$ $\quad \boldsymbol{z} \in [1,3]^4$

Experimental setup (Benchmark)

- AR dataset
  - 99 subjects
    - 6 samples from expression and illumination groups
    - 12 samples from occlusion group
    - Size 128×128
- FERET
  - 1196 individuals
    - 5 images
    - Size 128×128

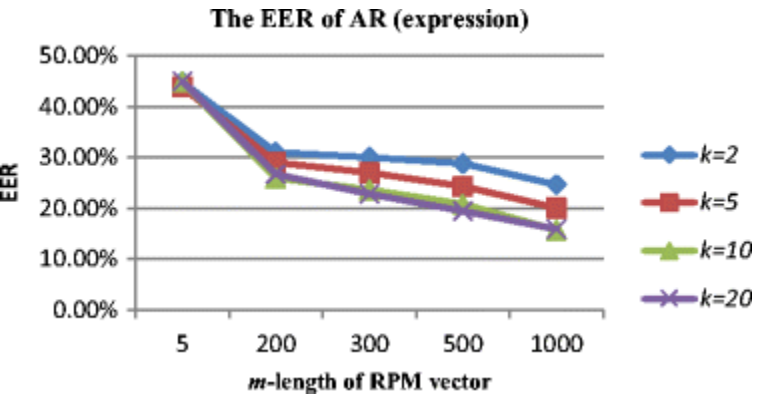
Experimental setup (Indicator)

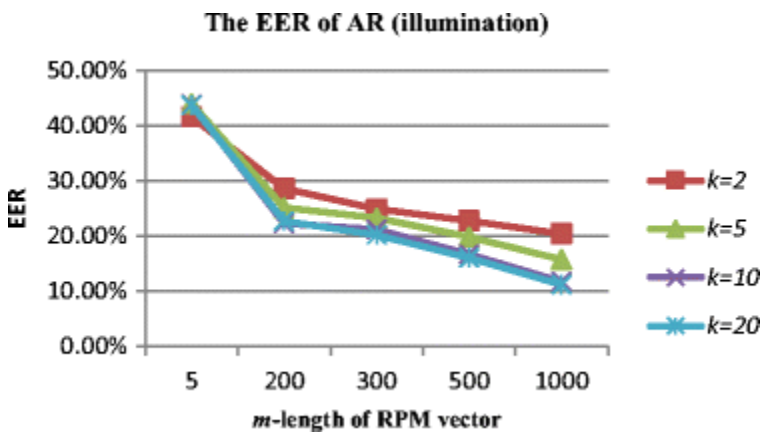
- Equal Error Rate (EER)
  - Estimate False Acceptance Rate (FAR) and False Rejection Rate (FRR)

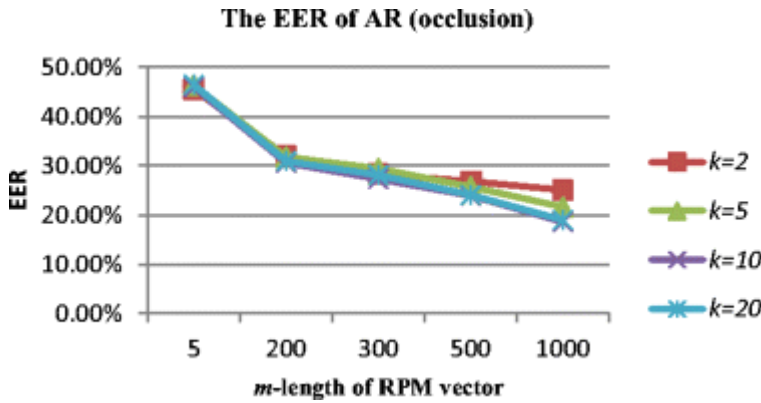
Experimental setup (Parameter)

- Discrete Cosine Transform based Region Log-Tied Rank Covariance Matrices (DCT-RLTCM)
  - Filter size $k = 11$
  - # of filter $T = 30$
  - Region size $r = 16$
  - WPCA dimension $d = 300$

Accuracy performance (m and k fixing $p = 2$)

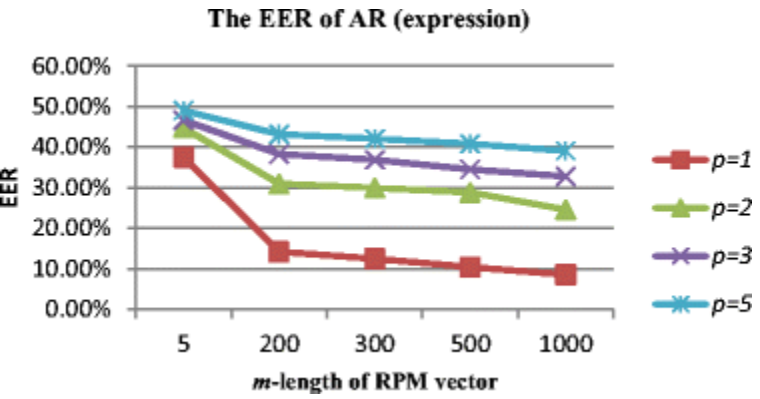


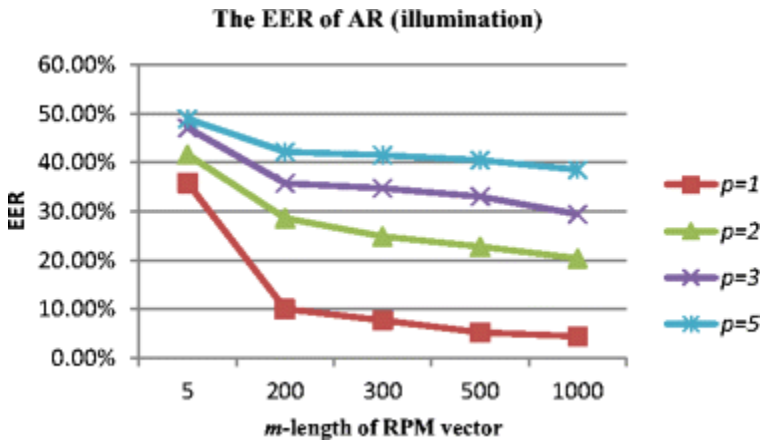


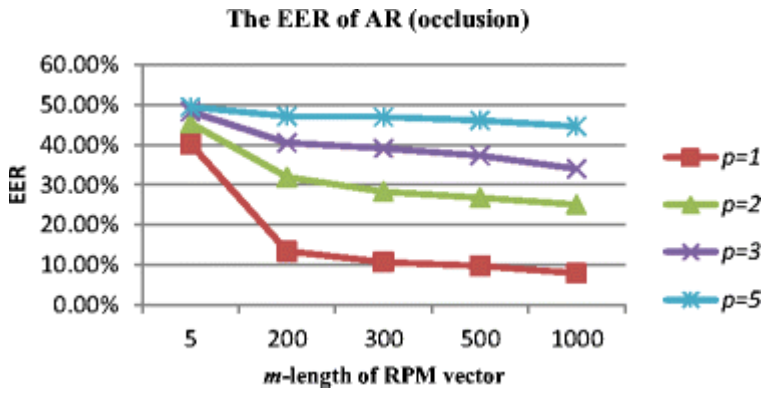


k 越小， max ranked feature 的信息越少

Accuracy performance (p and m fixing $k = 2$)

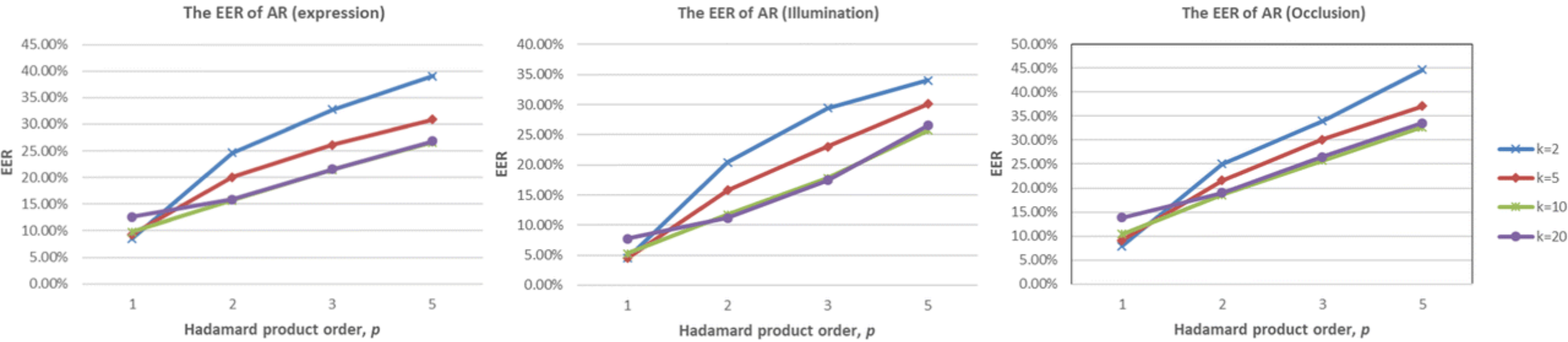






p 越大，Hadamard product 的次数越多，引入了噪声，引起失真

Accuracy performance (p and k fixing $m = 1000$)



最优参数 $\{p = 2, m = 1000, k = 20\}$

Justification

Suppose $\boldsymbol{x} \in \mathbb{R}^d$ be a biometric feature vector with length d, RPM transform is carried out in the following steps:

用户改变用户特定的种子

1. Generate p user-specific token-seeded stacked m permutation array $P_i \in \{0,1\}^{d\times d\times m}, i = 1, \dots, p$ where m is the desired length of resulting RPM transformed vector. Note that a permutation matrix is a square binary matrix that has exactly one entry of 1 in each row and each column and 0s elsewhere.

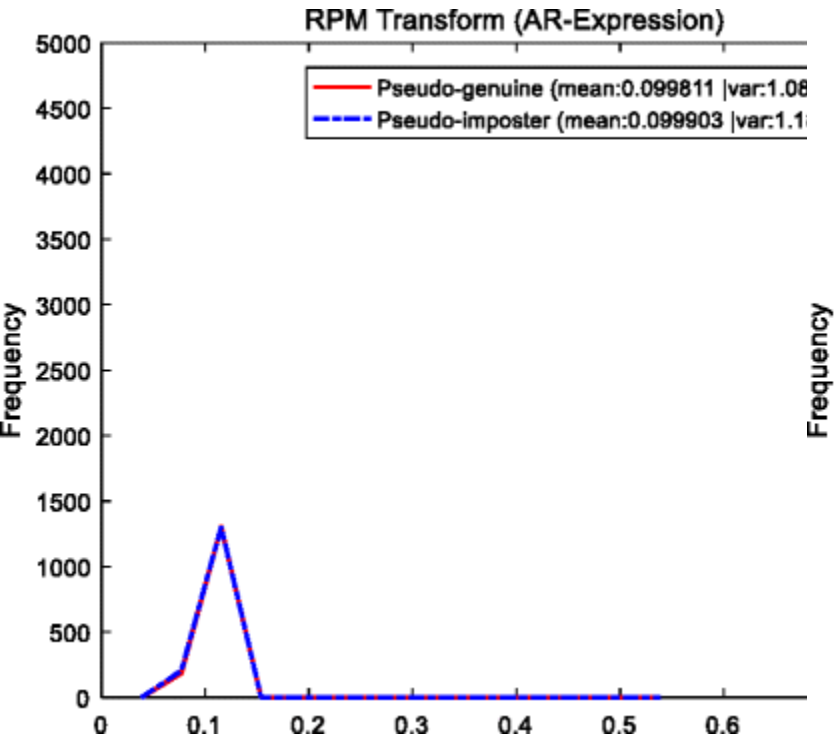
salting

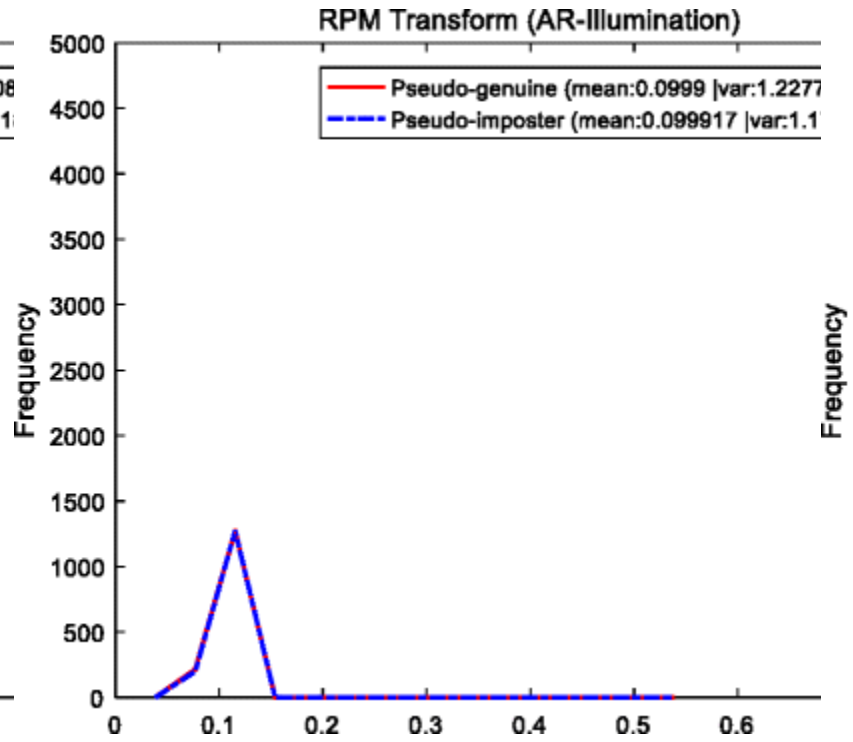
2. Multiplying $\boldsymbol{x}$ to P_i, yield a matrix, $\boldsymbol{V}_i = \boldsymbol{x}P_i \in \mathbb{R}^{d\times m}, i = 1, \dots, p$.

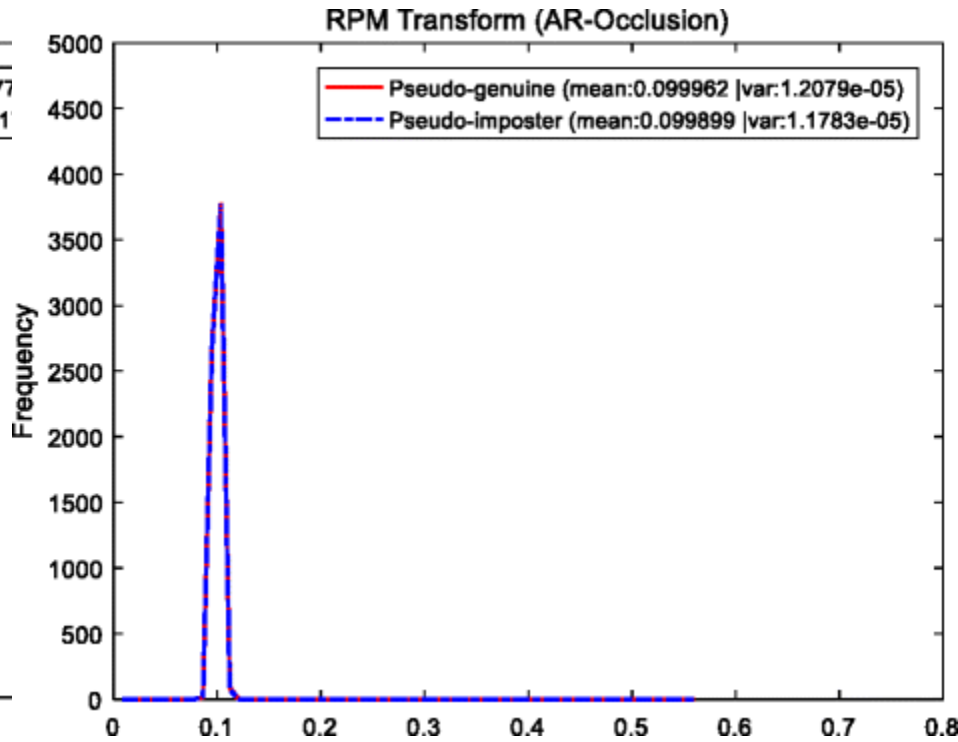
revocation & unlikability

3. Perform Hadamard product (element-wise product) yields $\boldsymbol{Y} = \boldsymbol{V}_1 \circ \cdots \circ \boldsymbol{V}_p = \prod_{i=1}^{p} \boldsymbol{V}_i \in^{d\times m}$ where p is known as Hadamard product order.
4. Discard last $d - k$ column vectors from $\boldsymbol{Y}$, yield $\boldsymbol{Y}' \in \mathbb{R}^{k\times m}$, where we named k as truncation size.
5. For each row vectors of $\boldsymbol{Y}'$, the indices of the largest magnitude entry are recorded and form a discrete RPM transformed vector, $\boldsymbol{z} \in [1\ k]^m$.

Non-linkability analysis

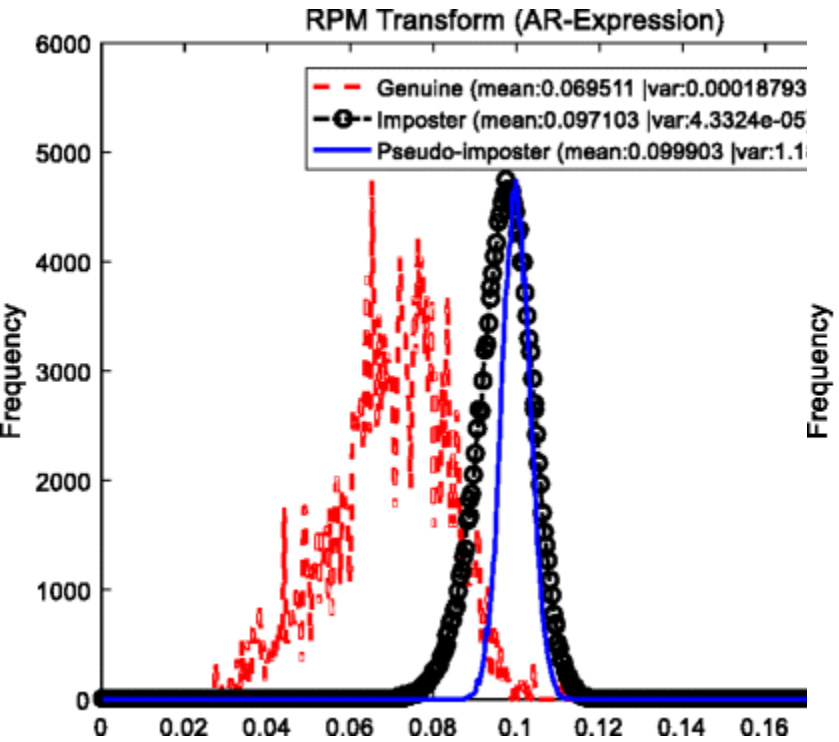


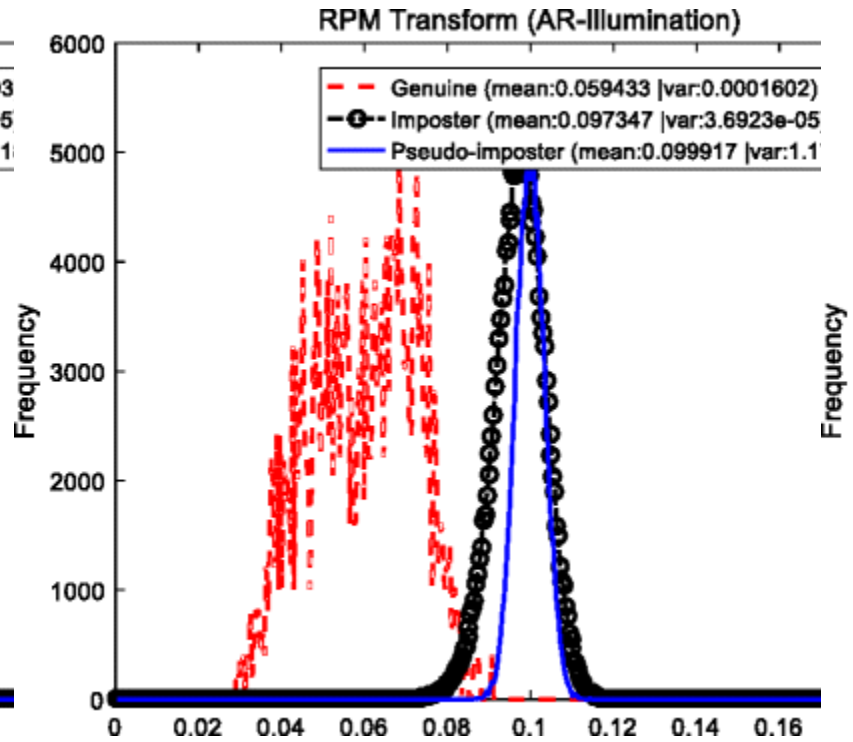


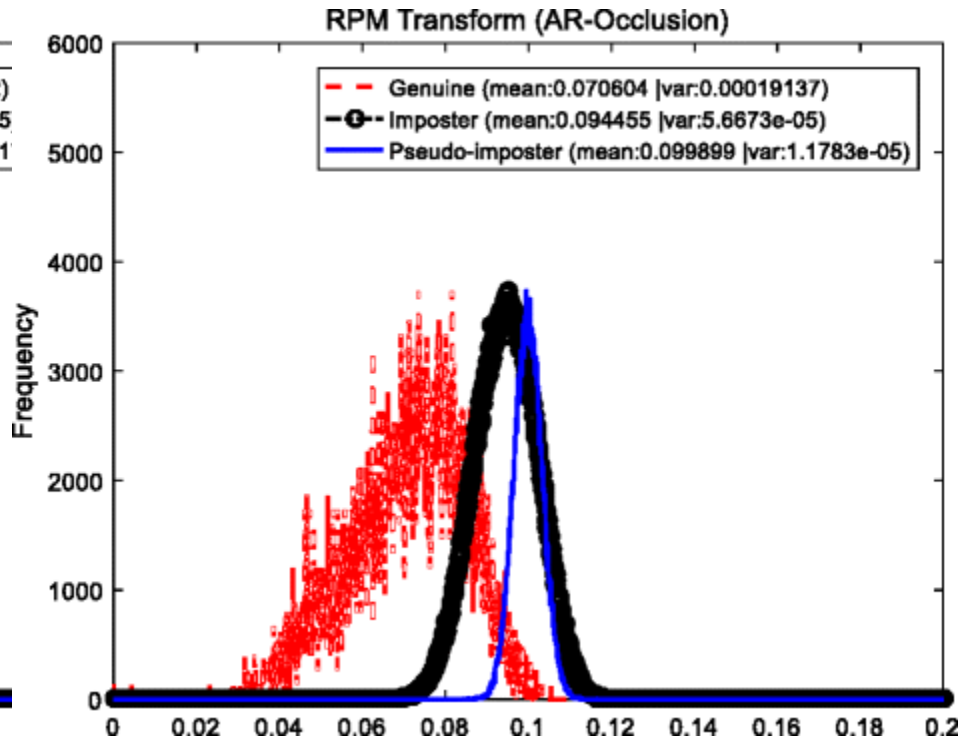


如果假阳率和假阴率的分布不同，敌手能够区分两个转换后的向量是否属于相同的主体

Revocability analysis







给定 50 个随机扰动种子，相同的源 DCT-RLTCM 向量，生成的 RPM 向量是不同的

Justification

Suppose $\boldsymbol{x} \in \mathbb{R}^d$ be a biometric feature vector with length d, RPM transform is carried out in the following steps:

1. Generate p user-specific token-seeded stacked m permutation array $P_i \in \{0,1\}^{d \times d \times m}, i = 1, \ldots, p$ where m is the desired length of resulting RPM transformed vector. Note that a permutation matrix is a square binary matrix that has exactly one entry of 1 in each row and each column and 0s elsewhere.
2. Multiplying $\boldsymbol{x}$ to P_i, yield a matrix, $\boldsymbol{V}_i = \boldsymbol{x}P_i \in \mathbb{R}^{d \times m}, i = 1, \ldots, p$.
3. Perform Hadamard product (element-wise product) yields $\boldsymbol{Y} = \boldsymbol{V}_1 \circ \cdots \circ \boldsymbol{V}_p = \prod_{i=1}^{p} \boldsymbol{V}_i \in^{d \times m}$ where p is known as Hadamard product order.
4. Discard last $d - k$ column vectors from $\boldsymbol{Y}$, yield $\boldsymbol{Y}' \in \mathbb{R}^{k \times m}$, where we named k as truncation size.
5. For each row vectors of $\boldsymbol{Y}'$, the indices of the largest magnitude entry are recorded and form a discrete RPM transformed vector, $\boldsymbol{z} \in [1\ k]^m$.

Non-invertibility

Privacy attack (Non-invertibility analysis)

- 假设敌手获取转换后的向量、令牌，RPM 算法及其参数
- 在 min-max 范围内枚举生成伪 RPM 向量

AR	Min value with 4 decimal precision	Max value with 4 decimal precision	(Max-Min)1000	Trial numbers
Expression	−0.1571	0.1561	$3132(\approx 2^{12})$	$\approx(\mathbf{2^{12}})^{\mathbf{300}}=\mathbf{2^{3600}}$
Illumination	−0.1156	0.1374	$2530(\approx 2^{11})$	$\approx(\mathbf{2^{11}})^{\mathbf{300}}=\mathbf{2^{3300}}$
Occlusion	−0.1116	0.1157	$2273(\approx 2^{11})$	$\approx(\mathbf{2^{11}})^{\mathbf{300}}=\mathbf{2^{3300}}$