Dynamic Random Projection for Biometric Template Protection

Bian Yang, Daniel Hartung, Koen Simoens, and Christoph Busch

Abstract—Random projection provides a good diversification effect for biometric template protection but is drawing increasing security concerns under the token-stolen (or public parameter) case. We propose a dynamic random projection method to alleviate these security concerns due to the stolen token by increasing the computational complexity to search for the unprotected biometric features. This is achieved by a projection process which dynamically assembles a random projection matrix from a set of candidate projection vectors. The selection of projection vectors is decided by the biometric feature vector itself and thus forms a nonlinear projection process. The proposed method permits the public and secure storage of all the candidate random vectors without the need for external secret keys. Experiments on the 800 samples in the database FVC2002DB2_A demonstrate the well-kept biometric performance of the proposed method.

I. INTRODUCTION

To address the security and privacy requirements of a biometric template, renewability [1,2] was proposed for a biometric template protection mechanism. Renewable biometric templates, diversified from the same biometric features and able to be compared directly for verification, are required to be irreversibly to their original biometric features and unlinkable among each other. Standard encryption mechanism, such as DES, AES, …, etc., can be an option for biometric template protection: using different secret keys to diversify the original biometric features to achieve unlinkability, and keeping the secret keys well protected to achieve irreversibility. However in this case, decryption is always required before comparison to tolerate the sample acquisition distortion of the fuzzy biometric signals. This demands an additional secure hardware environment for decryption and comparison, such as a secure access module used in [14]. Besides, standard encryption mechanisms become reversible when the secret key is available to the deciphering entity. So far many software based schemes [3-13,18] other than standard encryption mechanisms have been proposed to address the requirements of irreversibility and unlinkability. Unlike the standard encryption mechanisms which always assumes safe storage of the secret keys, software based template protection schemes also considers the security under the token-stolen case in which all the transformation parameters are assumed public. Because storing these parameters for all protected templates has the same security concern as storing the plain-text biometric features, it is desirable that these parameters are safe enough to be public.

Random projection [11,12] provides good distinguishability (thus desirable for generating unlinkable templates) by projecting a biometric feature vector onto a set of orthonormal random vectors. The security can be guaranteed by keeping secret the projection matrix in a token. However, security concerns were recently raised [15-18] on the token-stolen case, which is subject to information leakage caused by the linear projection operation.

We analyze these security concerns in Section II, and propose a dynamic random projection mechanism in Section III to enhance the security by increasing the searching complexity for a genuine unprotected template. Extension of length of the projection matrix was also tested in biometric performance. Section IV presents experimental results on the biometric performance tested on the full database FVC2002DB2_A [19]. Section V analyzes the security of the proposed method. Section VI concludes this paper.

The proposed method is to achieve the following two properties:

1. Enhanced security under the token-stolen case (i.e., computational complexity based irreversibility with public random projection matrices);
2. Tunable biometric performance and template sizes;

II. SECURITY CONCERNS OVER RANDOM PROJECTION

A. Pseudo Inverse Based Estimation Attack

For a $p$-dimensional biometric feature vector $b = (b_1, b_2, \ldots, b_p)$ and an orthonormal random matrix $R = (r_1, r_2, \ldots, r_q) \in \mathbb{R}^{q\times p}$ ($q \leq p$, $r_i \in \mathbb{R}^{p\times q}$ for $i = 1, 2, \ldots, q$) is a $p$-dimensional random vector), the random projection result is a $q$-dimensional real-value vector $v$:

$$v = R^\dagger b$$

(1)

By thesholding, a $q$-dimensional binary vector $t$ can be obtained as the protected template:

$$t = \text{Sign}(v)$$

(2)
where $\text{Sign}(\cdot)$ is the sign bit extraction to output a $q$-bit binary vector $t = (b_1, b_2, \ldots, b_q)$ as the protected template. The randomly-generated orthonormal matrix $R$ can approximately preserve two feature vectors $b_1$ and $b_2$: Euclidean distance in the projection space, and maintains the biometric performance of the protected templates in the token-stolen case [11,12]. On the other hand, this distance preservation property makes the estimation towards the genuine inverse $b$ from the protected template $t$ easier than brute-force searching.

The straightforward idea [15] to estimate $b$ is to constrain the vector direction of $b$ into the intersection of the $q$ half-hyperspace regions defined by the $q$ column vectors $r_i$ in $R$ via checking the $t$ (i.e., sign bits of $v$). A larger $q$ gives a more accurate estimation of $b$’s direction in the $p$-dimensional hyperspace. This vector direction estimation gives a constraint to the components to constitute $b$. Here we define all possible solutions for $b$ satisfying the constraint as the set of pseudo inverses $b^*$. By virtue of these pseudo inverses, two types of estimation attacks can be launched:

Type I. Reference template estimation. This scenario [16] assumes the attacker has available two random projection matrices (an obsolete matrix $R_1$ and an updated matrix $R_2$) and one corresponding obsolete template $t_0$ and wants to increase the probability to match the unknown updated template $t_1$. The estimation can be formulated as

$$t_1^* = \text{Sign}(R_1^T b_0^*) \quad (3)$$

where the pseudo-inverse $b_0^*$ satisfies

$$t_0 = \text{Sign}(R_1^T b_0^*) \quad (4)$$

The set of estimated $t_1^*$ constitute a narrower searching space for the genuine updated template $t_1$ than brute-force searching. More details on this attack can be found in [16].

Type II. Genuine inverse estimation. This scenario assumes the attacker has available both the random matrix $R$ and the protected template $t_1$ and wants to estimate the genuine inverse $b$. An effective estimation method proposed in [17] exploits the Euclidean distance preservation property (the Hamming distance after binarization is approximately preserved as well) as a condition to find a set of likely neighbors to the genuine inverse $b$ from other independent biometric feature vectors. Then the same number of closest pseudo inverses to these likely neighbors can be found by an optimization algorithm. From these selected pseudo inverses, an accurate estimation of the genuine inverse $b$ can be linearly composed in different weights. An effective countermeasure to this genuine inverse estimation is a multi-level binarization process which scatters the solution regions to a wider area in the hyperspace. More details can be found in [17].

B. Full-Rank Based Inversion Attack

The underlying principle of the irreversibility achieved by random projection [11,12] is the fact that there are infinite solutions for a non-full-rank linear equation system constructed by the random matrix $R$. Therefore, this irreversibility can be easily compromised if an attacker knows the linkage among $l$ protected templates $t_i$ ($i=1,2,\ldots,l$) diversified from the same $b$ by projection onto $l$ different $R_i$ ($i=1,2,\ldots,l$) with the following full-rank condition satisfied:

$$\text{Rank}([R_1, R_2, \ldots, R_l]) = p \quad (5)$$

where $[R_1, R_2, \ldots, R_l]$ is the matrix horizontally concatenated by the $l$ different $R_i$. When the projection matrix becomes a full-rank one, the projection is completely reversible. The condition of Eq.(5) is easy to satisfy when $q$ and $l$ increase and $R_i$ are random enough, with the necessary condition $ql \geq p$. In practice, this linkage can happen in the case that different protected templates are generated for different applications or the case that protected templates are updated at different times. Despite the later binarization operation, the attacker can exploit the leaked information to reverse the genuine biometric feature vector $b$ or to narrow the searching space for $b$.

III. PROPOSED METHOD

We are constructing a nonlinear dynamic random projection mechanism and thus increase the computational complexity to reverse the genuine biometric feature vector $b$ from the protected template $t$ and a set of public random matrices. This irreversibility is achieved without needing secret keys.

A. Dynamic Random Projection

Two facts make the conventional random projection in [11,12] vulnerable to the attacks introduced in Section II.: (1) the assumption that the random matrix $R$ should be public for security analysis; and (2) the projection is a linear operation which preserves the distance very well. These two facts make possible the estimation towards the updated protected template (Type I attack) and the genuine inverse (Type II attack). Our idea to circumvent these two facts is to design a mechanism to dynamically construct a random matrix instead of using a fixed random matrix for projection as illustrated in Fig.1, where $q$ random vector slots are set to be public with each slot $i$ containing $L$, $p$-dimensional random real-value vectors $r_{ij}$ ($1 \leq i \leq q$, $1 \leq j \leq L$). From each slot, one of the $L$ random vectors is selected for projecting the biometric feature vector $b$ to obtain one dimension of the projected result vector $v$. The selection of $r_{ij}$ will be decided by an index $id$, which is the $j$th dimension of an index vector $ID = (id_1, id_2, \ldots, id_q)$ ($id_j \in \mathbb{Z}$ and $1 \leq id_j \leq L$) such that with all the public slots there is no clue to the attacker which random vector $r_{ij}$ out of the slot $i$ was selected as $r_{i,\text{id}}$ to project $b$ into the $i$th dimension of $v$:

$$v_i = r_{i,\text{id}} b \quad (6)$$

An index vector generator function (as shown in Fig.1) can be designed to take the biometric feature vector $b$ as the input and output the $q$-dimensional integer index vector $ID$. This new mechanism makes the creation of the projection matrix dependent on the to-be-projected biometric feature vector. Thus it is difficult for an attacker to reverse to $b$ via an inverse linear transformation with a feasible searching complexity from the protected template $t$ and the public random matrix.
slots.

Now the task is to design an index vector generator function which is a mapping

\[ f : R^p \rightarrow [1, L]^q \]  

We present two simple examples as follows.

**Function I. Amplitude quantization of the biometric feature vector.** Supposing \([A_{\min}, A_{\max}]\) to be the dynamic range of the biometric feature vector \(b\)'s components' amplitude values, this range can be divided into \(L\) non-overlapped segments \(S_i\) \((1 \leq i \leq L)\) by \((L-1)\) thresholds \(T_1, T_2, ..., T_{L-1}\). In practice in order to avoid the bias of the quantization results, the middle threshold \(T_{L/2}\) (assuming \(L\) is an even number) can be set equal to the expectation value of the components' amplitude values of all \(b\) in the biometric training set, e.g., the enrolment set. Since in the random projection method \([11,12]\) \(q \leq p\), an arbitrary set of \(q\) components out of the \(p\) components of \(b\) can be selected for such a quantization and output a \(q\)-dimensional index vector \(ID = (id_1, id_2, ..., id_q)\) with an index value \(id_i(1 \leq i \leq q)\) from 1 to \(L\) for each dimension:

\[ id_i = \arg \min_j Q(b_j, \{S_i\})(1 \leq i \leq L) \]  

where \(Q\) is a quantizer with \((L-1)\) thresholds \(T_1, T_2, ..., T_{L-1}\). The index \(id_i\) indicates the \((id_i)\)th random vector \(r_{id_i}\) in the \(j\)th slot to be selected for projection.

**Function II. Amplitude quantization of a fixed-matrix based random projection result vector.** A more complex design is to use an orthonormal random matrix \(R_{id} = (r_{1id}, r_{2id}, ..., r_{qid})\) \(\in \mathbb{R}^{q \times p}\) to project \(b\) and output a \(q\)-dimensional real-value vector \(v_{id}\). Then the same operations as in the Function I can be applied to \(v_{id}\) to generate a \(q\)-dimensional index vector \(ID\) with each dimension as:

\[ id_i = \arg \min_j Q(v_{id}^{(b)}, \{S_i\})(1 \leq i \leq L) \]  

Particularly, the random column vectors \(r_{id}\) in \(R_{id}\) have to be normalized with the range [-1,1] to constrain the amplitudes of \(v_{id}\) within a similar dynamic range as \(b\)'s components such that the threshold setting in the Function I still works.

Now with the \(q\) indices \(id_i(1 \leq i \leq q)\) obtained, \(q\) random vectors can be selected from the corresponding \(q\) slots and assemble a random matrix \(R = [r_{1id}, r_{2id}, ..., r_{qid}]\) for random projection expressed by the Eq.(1). In this proposed random projection mechanism, all the random vectors in the \(q\) slots can be stored publicly together with the protected template \(t\).

**B. Random Matrix Extension**

In light of the information loss caused by the binarization following the random projection, the biometric performance of the protected templates should have a potential to improve by reducing this information loss. A simple idea is to increase \(q\) (the number of column random vectors inside \(R\)). An increased number \(q\) larger than \(p\) can easily make the \(q\) random vectors redundant to be a basis to span the \(p\)-dimensional hyper-space. This redundancy can help reduce the information loss from the binarization operation and thus we expect higher biometric performance by this projection matrix extension.

The Function II as described in Section III A seems more convenient for projection matrix extension via extending the random matrix \(R_{id}\)'s size to the case \(q > p\) with an arbitrary integer value \(q\). Correspondingly, the \(R\)'s size should be extended to the same size as \(R_{id}\)'s. With an increasing dimension \(p\) of the biometric feature vector \(b\), the column random vectors in \(R_{id}\) can be approximately orthogonal to each other \([11,12]\). This fact can easily result in a full-rank \(R_{id}\) and \(R\) in the case \(q > p\), which leads to reversibility from the projection result vectors \(v_{id}\) and \(v\) to the biometric feature vector \(b\). Fortunately, in the dynamic random projection mechanism, \(v_{id}\) is always thresholded after being fed to the index vector generator and \(R\) is always dynamically assembled dependent on the \(L\)-leveled amplitudes of \(v_{id}\). For an attacker, the unknown \(v_{id}\) (and thus unknown projection matrix) constitutes the searching complexity for \(b\). The searching complexity increases with \(L\) and \(p\). Detailed analysis will be given in Section V.

**C. Case Study: Fingerprint Minutiae Template Protection**

As in our previous work \([9,13]\), we apply the proposed dynamic random projection method to the fingerprint minutiae template protection case. Extraction of the biometric feature vector \(b\) is based on local minutiae quadruplets (called minutiae vicinity). Different from the global fixed-length feature extraction in \([10]\), we extract local fixed-length feature vectors as \(b\). For each minutia \(m_i(1 \leq i \leq m)\) in the original template consisting of \(M\) minutiae, 3 closest (in terms of Euclidean distance) neighboring minutiae are found around \(m_i\) and the quadruplets (including \(m_i\)) are defined as a minutia vicinity \(V_i\). Denote the 4 minutiae in one vicinity as \(J_i\) \((j = 1, 2, 3, 4)\) indexed in an ascending order of the Euclidean distance from \(m_i\) \((d(m_i,J_j))\) as shown in Fig.2), thus \(J_i = \{m_i,J_j\}\) and the 6 orientations \(O_l(l = 1, 2, ..., 6)\) can be defined between the minutiae pairs (e.g., pair \(m_i\) and \(c_2\) defining \(O_2\) in Fig. 2)
and along each orientation the remaining minutiae pair \( (c_{i1} \text{ and } c_{i3} \text{ in the example in Fig. 2) can be geometrically-aligned (resulting \( J_{a1} \text{ and } J_{a2} \text{ respectively). Mathematically, if the orientation } O_i \text{ is formed by two points } J_p \text{ and } J_q (1 \leq p,q \leq 4, p \neq q) \text{ in the old coordinate system, the remaining 2 minutiae } J_j \text{ (1\leq j \leq 4, p \neq j) will be aligned as:}

\[
\begin{align*}
J_{a1}(x) &= \cos(\theta_1) J_j(x) - \sin(\theta_1) GMP_j(x) \\
J_{a1}(y) &= -\sin(\theta_1) J_j(x) + \cos(\theta_1) GMP_j(y)
\end{align*}
\]

(10)

where \( GMP_j \) (corresponding to the \( o' \) in Fig. 2) is the geometric middle point of \( J_j \) and \( J_p \):

\[
\begin{align*}
GMP_j(x) &= (J_p(x) + J_q(x)) / 2 \\
GMP_j(y) &= (J_p(y) + J_q(y)) / 2
\end{align*}
\]

(11)

and \( \theta_1 \) is the angle formed by the orientation \( O_i \) and the old \( x \)-axis calculated as:

\[
\theta_1 = \tan^{-1}\left( \frac{J_q(y) - GMP_j(y)}{J_q(x) - GMP_j(x)} \right)
\]

(12)

Note that the minutiae pair \( J_p \) and \( J_q \) which defines the new coordinate system will not be included in the final geometric alignment result. With the 6 ordered orientations, 6 ordered minutiae pairs \( J_{a1}, J_{a2}, J_{a3}, J_{a4}, J_{a5}, J_{a6} \) \( (1 \leq j \leq 6) \) can be obtained as the aligned and ordered vicinity \( V_{ai} = (J_{a1}, J_{a2}, ..., J_{a6}) \), where inside each ordered minutiae pair the two aligned minutiae are ranked in ascending order by their original index \( j \). Associated with each \( J_{ai} \) \((k=1,2,...,12) \)'s geometrically-aligned coordinates \( (x,y) \), the \( J_{ai} \)'s minutiae angle \( J_{ai}(\omega) \) is also updated by subtraction of the rotation angle \( \theta_1 \) from its original value \( J_{ai}(\omega) \):

\[
J_{ai}(\omega) = J_{ai}(\omega) - \theta_1
\]

(13)

Finally, a 36-dimensional feature vector can be formed as \( b \) by concatenating the coordinates \( x \), \( y \), and the angle \( \omega \) of all the 12 aligned vicinities. Then the dynamic random projection and the random matrix extension described in Section III. A and B can be performed to generate \( t \) as the protected template. Both the matrix \( R_{ID} \) and the \( q \) random vector slots can be stored publicly with \( t \) in the biometric system’s database, or in the subject’s token. In the case the quantization level number \( L \) is large but the storage resource is constrained, the seed for a pseudo-random number generator to generate \( R_{ID} \) and the \( q \) slots can be stored instead.
Correspondingly, the final protected vicinity binary vector identity which the token is linked to. and the subject only probes his/her finger to verify the corresponding projection matrices are assumed to be public; both the protected vicinity binary vectors and their token-stolen scenario [11,12] was tested. In this scenario, 40 correspondingly. The biometric performance in the vectors can match or not are manually set to be 3, 10, 20 and thresholds used to judge if two protected vicinity binary lengths are 36, 72, 144, and 288 bits. The Hamming distance T shows the biometric performance in terms of EER achieved reasonable. To check the stability of the expectation value distribution in Fig. 3 seems close to the middle point of the 4th and the 5th regions, indicating that the assumption of $E = 0$ is reasonable. To check the stability of the expectation value $E$, 100 tests were done with each test using a different set of 36×36 sized random matrices. The mean value of the 100 expectation values in the experiment was found perfectly equal to 0.00 and the standard deviation was 0.0055, indicating a good stability of the assumption $E = 0$. Thus throughout all our experiments, the 7 thresholds are set as $[T_1, T_2, \ldots, T_7] = [-75, -50, -25, 0, 25, 50, 75]$.

C. Biometric Performance

To test the biometric performance of the proposed method, the number of random vectors inside each projection matrix is set to be $q = 36, 72, 144$ and 288, respectively. Correspondingly, the final protected vicinity binary vector $t$’s lengths are 36, 72, 144, and 288 bits. The Hamming distance thresholds used to judge if two protected vicinity binary vectors can match or not are manually set to be 3, 10, 20 and 40 correspondingly. The biometric performance in the token-stolen scenario [11,12] was tested. In this scenario, both the protected vicinity binary vectors and their corresponding projection matrices are assumed to be public; and the subject only probes his/her finger to verify the identity which the token is linked to.

Out of all the settings of $q$, $q = 36$ is the case which has similar template size as the ISO standard conformed template. Fig. 4 shows the False Match Rate (FMR) and False Non-Match Rate (FNMR) curve in the case $q = 36$, with an Equal Error Rate (EER) around 0.0453 and a performance point FNMR = 0.0991 at the FMR ≈ 0.001. The enrolment set and the verification set from which this curve was generated are the 1st 100 samples and the remaining totally 700 samples, respectively, from the FVC2002DB2_A database.

Note that in the FVC2002DB2_A database, the 700 samples for verification are generally degrading in sample quality from the 2nd 100 samples to the 8th 100 samples. Table I shows the biometric performance in terms of EER achieved by the proposed method (Dynamic Random Projection Function II) using the 2nd~8th 100 samples (denoted as the test sets 1~7 respectively in Table I) separately as 7 sets of probes, in four cases with various number of random projection vectors $q = 36, 72, 144$ and 288 (i.e. various protected vicinity binary vectors’ lengths). All the EER results are the average values of 10 tests with each test using a different set of random vectors slots. It can be observed that in general extension of the random vectors’ number results in improved biometric performance in our experiments, which supports the analysis in Section III B. But the improvements become marginal when $q$ increases. In addition, biometric performance comparison between the proposed method and our previous work [13] can be found in the Table 1, where distinct improvement in performance is observed.

Table II shows the biometric performance in terms of FNMR at the FMR around 0.001 achieved by the proposed method (Dynamic Random Projection Function II) corresponding to the cases $q = 36, 72, 144$ and 288.
To evaluate the influence of the amplitude quantization operations in the dynamic random projection mechanism on the biometric performance, we compare the biometric performance of the three cases – RP (fixed-matrix random projection used in [11,12]), DRP I (dynamic random projection Function I) and DRP II (dynamic random projection Function II) in Fig. 5. In all the three cases, \( q = 36 \) in the experiments, and all the performance results were averages obtained from 10 tests with a different set of random vector slots used in each test. The Hamming distance thresholds used to judge if two protected vicinity binary vectors can match or not are manually set to be 1, 3, and 3 for RP, DRP I, and DRP II, respectively. It is observed that in general the dynamic random projection does not degrade the biometric performance compared with the fixed-matrix based random projection when dealing with the biometric feature vectors \( b \) obtained by the method described in Section III C.

V. Security Analysis

We give a brief security analysis over the proposed method from the two security requirements defined in [1,2] – irreversibility from the protected vicinity binary vector \( t \) to the geometrical-aligned vicinity feature vector \( b \), and unlinkability among the diversified protected vicinity binary vector \( t \) derived from the same \( b \). Note that unlike standard encryption mechanism using secret keys under well protection to achieve the irreversibility and unlinkability, there is no secret keys or any other secret information in our proposed method used to generate a protected template, due to the security requirement in the token-stolen case. The security analysis is done towards the protected vicinity instead of the whole template because piece-wise comparison is used and the basic comparison unit is the protected vicinity.

A. Irreversibility

Similar to a one-way function which is easy to compute on an input, but hard to reverse given the image of the input, the irreversibility requires the proposed template protection method infeasible to find the genuine biometric feature \( b \) from the protected template \( t \) and the public random vector slots. The irreversibility of the protected vicinity binary vectors is achieved by the dynamic matrix selection mechanism in which the attacker has no clue which subset of \( q \) random vectors out of the total \( qL \), random vectors were actually used for projection, which constitutes a searching complexity of \( L^q \) possibilities to form a projection matrix \( R \). In both Function I and II, the index vectors’ generation are completely decided by the original biometric feature vector \( b \). Considering one possible projection matrix \( R \) is full-rank (which is easy to attain when \( q \geq p \)), for any real-value projection result vector \( v \), there are only two cases – none solution or a single solution to inverting \( v \) to \( b \) via a candidate \( R \). Note that if \( v \) is known by the attacker (in practice only \( t \) can be known), whether there existing a valid single solution or none solution to \( b \) depends on the knowledge of at least \( p \) components out of the \( q \)-dimensional \( v \). For each component of \( v \), the attacker faces \( L \) different random vectors for projection. Therefore from \( v \), an attacker needs at least searching \( L^q = 2^{36} \) possibilities to check if the known \( v \) can be obtained from a tested \( b \). In the case using the Function I, this directly indicates a searching complexity of \( 2^{108} \) to find the correct biometric feature vector \( b \) and in the case using the Function II, this indicates a searching complexity of \( 2^{288} \) to find the correct real-value vector \( v_{ID} \) and the same complexity also applies to the case of inversion to \( b \) if the random matrix \( R_{ID} \) is full-ranked (thus the single solution to \( b \) can be found). This searching complexity ensures the security of public storage of all the random vectors in all the \( q \) slots.

Since practically the attacker can only know \( t \) instead of \( v \), it is possible that more than one solution out of the \( 2^{108} \) possible combinations to invert from \( t \) to \( b \), and thus the searching complexity to find the genuine solution to \( b \) can be even higher than \( 2^{108} \). Note that a larger (>36) \( q \) will not either increase the searching complexity more than \( 2^{108} \) to find a possible single solution to \( b \), nor decrease the searching complexity. However, increased \( q \) will leak more information on \( b \) and \( v_{ID} \), i.e., lead to a more precise estimation of \( b \) after the \( 2^{108} \) searches.

It is not hard to follow from the above analysis that the irreversibility is enhanced by the proposed method against the “Type I. Reference template estimation” introduced in Section II A and the “full-rank based inversion attack” introduced in Section II B. Although the proposed method does not directly address the “Type II. Genuine inverse estimation” attack proposed in [17] and discussed in Section II A, it is easy to use multi-level binarization [17] to replace the sign extraction in Eq.(2) to further enhance the irreversibility. Interestingly, we noticed in our additional experiments that the biometric performance was in general not degraded by doing so when we used the same thresholds \([T_1, T_2, \ldots, T_q] = [-75, -50, -25, 0, 5, 20, 50, 75]\) as the binarization levels and the interleaved quantization output – \([0, 1, 0, 1, 0, 1, 0, 1]\) assigned to the \( L = 8 \) amplitude regions \( S_1 \sim S_8 \).

Note that all above analysis are based on the token-stolen scenario in which both the protected template and the random vector slots are assumed to be public.

B. Unlinkability

Due to the good diversification ability of random projection [11,12], unlinkability can be well achieved among the protected templates \( t \) derived from the same biometric feature \( b \). Thanks to the different independent random vector slots used on the same \( b \) for diversification, it is not obvious to find the linkage among the protected vicinity binary

<table>
<thead>
<tr>
<th>TABLE III PAIR-WISE INDEPENDENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hamming distance</td>
</tr>
<tr>
<td>( q = 36 )</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>( q = 288 )</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
vectors. We analyzed the distribution of Hamming distance between two protected vicinity binary vectors to test the pair-wise independence. Such tests were done across

I. different protected templates with different sets of random vector slots;
II. different protected vicinities in the same template with the same set of random vector slots; and
III. different diversified templates derived from the same finger with different random vector slots.

In the Test I, the 100 templates using different sets of \( q \) random vector slots were employed for projection, with randomly selected 1000 pairs for test. In the Test II., the 100 templates using the same set of random vectors were employed, with all vicinity binary vectors inside each protected template cross-paired for test. In the Test III., 100 diversified templates were generated from each of the 100 original templates (thus totally 10000 templates generated) by 100 different sets of random vector slots, and out of the diversified 100 templates from each same original template, every two vicinity binary vectors generated from the same original minutia were paired and then 1000 pairs were randomly selected for test. Two cases \( q = 36 \) and \( 288 \) were tested as examples. Assuming the proposed algorithm generates iid bits, the Hamming distance will have a binomial distribution \((n,p)\). In our case \( n = 36, 288 \) and \( p = 0.5 \), the mean value \( \mu = np = 18 \) and standard deviation \( \sigma = (np(1-p))^0.5 = 3 \) for the case \( n = 36 \); and the mean value \( \mu = np = 144 \) and standard deviation \( \sigma = (np(1-p))^0.5 \approx 8.49 \) for the case \( n = 288 \). Table III presents the mean values of the actual expectations and standard deviations for the two cases in the three tests. Test I. (the inter-template test) shows nearly perfect pair-wise independence, indicating good unlinkability among the protected templates with different random vector slots. Test III. (the template-diversification test) shows slightly biased expectations and increased standard deviations compared to the theoretical values, indicating that some small correlations still exist among the diversified protected templates. But in general the linkage in terms of this pair-wise dependence among diversified templates is weak, which indicates a fair protection of the user’s privacy in the biometric template diversification case.

VI. CONCLUSIONS

We summarize in this paper security concerns over the fixed-matrix based random projection [11,12] for biometric template protection, and then proposed a new dynamic random projection mechanism which forms a non-linear projection process by relating the random matrix’s assembly to the biometric feature vector itself. This dynamic random projection immensely increases the computational complexity to launch inversion attacks in the token-stolen scenario. In addition, random matrix extension improves the biometric performance. Experiments on the public fingerprint database FVC2002DB2_A demonstrate the desirable biometric performance of the proposed method. Besides, tunable-sized protected templates can be generated by adjusting the number of column random vectors for projection.

REFERENCES