Manifold Learning Towards Masking Implementations: A First Study

Changhai Ou, Degang Sun, Zhu Wang, Xinping Zhou and Wei Cheng

[1] Institute of Information Engineering, Chinese Academy of Sciences
[2] School of Cyber Security, University of Chinese Academy of Sciences
ouchanghai@iie.ac.cn

Abstract. Linear dimensionality reduction plays a very important role in side channel attacks, but it is helpless when meeting the non-linear leakage of masking implementations. Increasing the order of masking makes the attack complexity grow exponentially, which makes the research of nonlinear dimensionality reduction very meaningful. However, the related work is seldom studied. A kernel function was firstly introduced into Kernel Discriminant Analysis (KDA) in CARDIS 2016 to realize nonlinear dimensionality reduction. This is a milestone for attacking masked implementations. However, KDA is supervised and noise-sensitive. Moreover, several parameters and a specialized kernel function are needed to be set and customized. Different kernel functions, parameters and the training results, have great influence on the attack efficiency. In this paper, the high dimensional non-linear leakage of masking implementation is considered as high dimensional manifold, and manifold learning is firstly introduced into side channel attacks to realize nonlinear dimensionality reduction. Several classical and practical manifold learning solutions such as ISOMAP, Locally Linear Embedding (LLE) and Laplacian Eigenmaps (LE) are given. The experiments are performed on the simulated unprotected, first-order and second-order masking implementations. Compared with supervised KDA, manifold learning schemes introduced here are unsupervised and fewer parameters need to be set. This makes manifold learning based nonlinear dimensionality reduction very simple and efficient for attacking masked implementations.

Keywords: machine learning · manifold learning · dimensionality reduction · ISOMAP · LLE · Laplacian Eigenmaps · masking · side channel attack

1 Introduction

Cryptographic devices may leak secret information through side channels such as electromagnetic [1], acoustic [12] and power consumption [15] during the hardware and software implementation of cryptographic algorithms. These leakages are usually unconscious and difficult to be discovered. By taking advantage of statistical correlation between assumed power consumption of intermediate values (e.g. the outputs of Sbox in the first round of AES-128) and these side channel leakages, an attacker can recover the key used in the target devices. Side channel attacks, such as Differential Power Analysis (DPA) [15], Correlation Power Analysis (CPA)[4], Template Attacks (TA) [6] and Collision Attacks (CA) [25, 23], pose serious threats to the security of cryptographic implementation. Countermeasures, such as masking [24, 8], must be taken to prevent against them, which make side channel attacks hard to perform. However, these countermeasures also make it hard to evaluate the security of implementation.

Higher-order attacks against masking implementations are widely researched [7, 19]. For a d-th order masking implementation, the masking includes $d+1$ shares, of which the first d ones are random, and the last one is fixed and calculated from the first d shares and the intermediate values (see Section 2.2). For all shares parallel performed masking implementations, the attacker can simply traverse the time samples and perform his attacks. However, for serial software implementations, different shares are performed at different times. That's to say, secret information leaks at different time samples. In order to perform higher order attacks, the attacker needs to find at least a time sample combination of $d + 1$ shares. However, most of current papers perform their attacks under the hypothesis that the locations of all $d + 1$ shares are known. This brings higher-order attacks back to the unprotected first order attacks case and ignores the detection of Points-of-Interest (POI) [18].

As far as we know, there are only several papers talking about POI selection for masking implementation. Oswald et al. used educated guess that performing an exhaustive search over all possible $(d + 1)$-tuples of time samples in a selected window [17]. Reparaz et al. proposed an alternative solution based on Mutual Information Analysis (MIA) [13] in [20]. However, both of these two solutions are very time-consuming, since the power traces acquired are usually very long. This makes the higher-order attacks very difficult to perform. Linear dimensionality reduction methods, such as PCA [11], LDA [27], project the high dimensional power traces into a lower dimensional subspace by optimizing objective functions. However, these methods are limited to the dimensionality reduction in linear space. They become powerless when meeting high-dimensional nonlinear leakage for higher-order masking implementations.

Durvaux et al. proposed a new solution to select POIs based on Projection Pursuits (PP) [10]. This solution returns the locations of time samples for shares. It needs to set several parameters. As stated in [5], it works very well when attacking second order masking implementations. However, it becomes powerless and fails when $d > 2$ in the experimental results of Cagli et al. [5]. In this case, Kernel Discriminant Analysis (KDA) [5] still works in third- and fourth- order attacks. It projects high dimensional traces leaked by higher-order masking implementation to low dimensional space through a nonlinear kernel function (see KDA introduced in Section 3). KDA is currently the only feasible solution for nonlinear dimensionality reduction towards masked implementations. It makes the complex and time-consuming higher-order attacks simple and feasible. This is a milestone for higher order attacks. However, for non-linear dimensionality reduction, there is still a lot of work to do. Since the correlation between the intermediate values and combined power consumption becomes very weak after performing KDA, which makes the attack require a lot of traces (see the experimental results detailed in Section 5). Moreover, for higher-order masking implementations, a power trace usually includes hundred thousand or even millions time samples, which makes the storage and computing time very huge, and greatly increases the difficulty of the attacks.

Nonlinear dimensionality reduction methods can be divided into two classes: kernel based ones and eigenvalues based ones. The former class, such as KPCA [22], KICA [31] and KDA introduced in this paper, introduce kernel function into linear dimensionality reduction schemes to realize nonlinear dimensionality reduction. The latter class, such as ISOMAP [29], Locally Linear Embedding (LLE) [21], Laplacian Eigenmaps (LE) [2] and LTSA [32], uses nonlinear mapping to map the high-dimensional data into low dimensional space and maintains the local structure of the original data set. In order to solve the problems detailed in the above paragraph, we firstly introduce manifold learning into side channel attacks and use them for nonlinear dimensionality reduction on traces acquired when performing higher-order masking implementations. Solutions such as ISOMAP, LLE and LE, are used. Compared with KDA, manifold learning based dimensionality reduction does not require custom kernel functions, nor does it need to set complex parameters. Moreover, these three manifold learning solutions are unsupervised. These make manifold learning based dimensionality reduction very simple and efficient for attacking masking implementations.

The rest of the paper is organized as follows. Notations, non-linear leakage of masking implementation and intrinsic dimensionality estimation in manifold learning are introduced in Section 2. KDA is introduced in Section 3. Manifold learning schemes including ISOMAP, LLE and LE are detailed in Section 4. Second- and third-order attacks are performed on simulated traces to highlight the superiority of our manifold learning schemes in comparison with KDA in Section 5. Finally, Section 6 draws general conclusions.

2 Backgrounds

2.1 Notations

The principle of dimensionality reduction is to map data samples from high dimensional input space through linear or nonlinear mapping to a low dimensional space, so as to find meaningful low dimensional structures hidden in high dimensional observation data. Suppose that the attacker encrypts n plaintexts $P = (p_1, p_2, \ldots, p_n)$, acquires n power traces $x_1, x_2, \ldots, x_n$ and saves them in matrix X. Thus, $X = \{x_1, x_2, \ldots, x_n\} \subset R^D$. Here R denotes the real number space, D denotes the number of samples on each acquired power trace. If the high-dimensional samples in X can be generated from the data set Y in the low dimensional space through an unknown mapping function f:

$$x_i = f(y_i) + \theta_i, \tag{1}$$

then $f : R^d \to R^D$ is an embedded mapping. Here θ_i is noise and $d \ll D$. If f is a linear function, then we say it's a linear mapping. Otherwise, it's a nonlinear mapping. The main purpose of manifold learning is to obtain the low dimensional samples Y according to the given high dimensional observation samples X, and construct a non-linear mapping f^{-1} from high-dimension to low dimension.

2.2 Non-linear Leakage of Masking Implementation

The principle of masking is secret sharing. Take a d-th order masking of AES-128 for example, the sensitive intermediate values of S-box outputs in the first round are divided into $S_1, S_2, \ldots, S_d$ and S_{d+1} pieces:

$$I = S_1 \circ S_2 \circ \cdots \circ S_d \circ S_{d+1}, \tag{2}$$

$\circ$ here denotes a group operation, such as XOR. The shares $S_1, S_2, \cdots, S_d$ are independent with each other and random. The last share S_{d+1} is fixed and satisfies Equation 1. Thus, The masking implementations eliminate the correlation between the side channel leakage $X = \{x_1, x_2, \ldots, x_n\}$ and intermediate values I. To perform a successful attack, the attacker needs to find at least a time sample combination of these $d + 1$ shares. If at most d shares are found, the combined power consumption is still independent of the assumed leakage of intermediate values. The introduction of masking into encryption and decryption slows down the implementation, makes the power trace long. If the sampling rate is set very high, this problem is very obvious. This makes it hard for the attacker to find a time sample combination of shares.

Not only the group operation XOR in Equation 2, the intermediate values I can be generalized as the non-linear function $f^{'}$ of $d + 1$ shares:

$$I = f^{'}(S_1, S_2, \cdots, S_{d+1}), \tag{3}$$

the side channel leakage can be simply regarded as the observation of this manifold structure. The intrinsic dimension of the side channel leakage is always much smaller than the corresponding power trace set. Intrinsic dimension here is defined as the actual dimension of low-dimensional manifold structure corresponding to the high-dimensional leakage samples. In this way, the problem of side channel attacks on masking implementation can be changed to the problem of non-linear dimensionality reduction. The dimension d of Y after dimensionality reduction is much smaller than the dimension D of original power trace set X. Moreover, the information of shares are mapped into the low dimensional manifold. We can simply consider that the D-dimension samples are projected onto a vector for each dimension after mapping.

Manifold learning can map high-dimensional samples to low-dimensional space and preserve the original structure of data. If the first-order CPA performed on a vector have a high correlation coefficient, the classification of samples on this dimension after projection is good. Thus, the attacker can directly perform first order side channel attacks on the data set Y. Therefore, the non-linear dimensionality reductions based on manifold learning can significantly improve the attack efficiency. If template attack (TA) is used here, the attacker has to take several mapped dimensions into consideration.

2.3 Intrinsic Dimensionality Estimation

As mentioned in Section 2.2, the intrinsic dimension of side channel leakage is always much smaller than the corresponding dimension of power trace set. Dimensionality reduction can not only obtain computational advantages, but also greatly improve the comprehensibility of data set. Although the intrinsic dimension of the high-dimensional leakage samples for a certain masking implementation is determined, it is difficult to estimate it from the leakage samples since the presence of other operations and noise. An attacker can usually describe the intrinsic dimension by using the minimum number of independent variables required by model the observed samples.

Schemes such as Geodesic Minimal Spanning Tree (GMST) [9], k-Nearest Neighbor (k-NN) [26], Maximum Likelihood Estimator (MLE) [16], can be used to estimate the intrinsic dimension in manifold learning. Actually, it's enough to keep several projection directions, since the high correlation usually occurs on them.

3 Kernel Discriminant Analysis

Let $(x_i^{z_i})_{i=1,\ldots,n}$ denote the class $z = z_i$ including all traces corresponding to class z_i. For example, the power traces are divided into 9 classes according to their intermediate values. Let $\mathsf{K} = \left(x_j^{z_j}, x_i^{z_i}\right)_{i=1,\ldots,n;j=1,\ldots,n}$ store only columns indexed by the indices i such that $z_i = z$, I denote a $n_z \times n_z$ identify matrix, and I_{n_z} denote a $n_z \times n_z$ matrix with all entries equal to $\frac{1}{n_z}$. KDA introduced in [5] uses a kernel function

$$\mathcal{K}(x_i, x_j) = (x_i \cdot x_j)^d \tag{4}$$

to turn linear distinguisher LDA [27] into nonlinear distinguisher, where · denotes the dot product. Then, it uses the labeled set $(x_i^{z_i})_{i=1,\ldots,n}$ and the kernel function to construct a between-class scatter matrix M:

$$\mathsf{M} = \sum_{z \in \mathcal{Z}} n_z \left(\mathsf{M}_z - \mathsf{M}_T\right) \left(\mathsf{M}_z - \mathsf{M}_T\right)^{\mathsf{T}}, \tag{5}$$

where M_z and M_T are two n-size column vectors whose entries are given as:

$$\mathsf{M}_z[j] = \frac{1}{n_z} \sum_{i:z_i=z}^{n_z} \mathcal{K}(x_j^{z_j}, x_i^{z_i}), \tag{6}$$

and

$$\mathsf{M}_T[j] = \frac{1}{n_z}\sum_{i=1}^{n}\mathcal{K}(x_j^{z_j}, x_i^{z_i}). \tag{7}$$

T here represents a matrix transpose. Cagli et al. then constructed a within-class scatter matrix N:

$$\mathsf{N}_z[j] = \sum_{z\in\mathcal{Z}} \mathsf{K}_z(\mathsf{I} - \mathsf{I}_{n_z})\mathsf{K}_z^{\mathsf{T}}. \tag{8}$$

Cagli et al. then regularized the matrix N:

$$\mathsf{N} = \mathsf{N} + \mu\mathsf{I}, \tag{9}$$

and found Q non-zero eigenvalues $\lambda_1, \ldots, \lambda_Q$ and the corresponding eigenvectors $\nu_1, \ldots, \nu_Q$ of $\mathsf{N}^{-1}\mathsf{M}$. This step aims at minimizing the within-class variance and maximizing between-class variance. $(\nu_j)_{j=1,\ldots,Q}$ here denotes different projection directions. If a new trace x is projected onto the l-th nonlinear d-th order discriminant component, then the projection is

$$\varepsilon_l^{\mathrm{KDA}}(x) = \sum_{i=1}^{n}\nu_\iota[i]\mathcal{K}\left(x_i^{z_i}, x\right). \tag{10}$$

The most time-consuming step in KDA is to find the eigenvalues of $\mathsf{N}^{-1}\mathsf{M}$, of which the complexity is $O(n^3)$. When the number of traces used is very large, KDA works very slowly. Moreover, KDA is a supervised nonlinear dimensionality reduction, the training result has direct impact on the dimensionality reduction results.

4 Manifold Learning

Since the proposal of ISOMAP [29], Manifold learning schemes such as LLE [21], LE [2], LTSA [32], LPP [14] and MFA [30], have been widely studied. ISOMAP, LLE and LE is three most classical schemes, which are introduced in Sections 4.1, 4.2 and 4.3. It is worth noting that different manifold learning schemes are based on different mathematical principles. So, it is all depend that whether the scheme can be used to reduce the dimension of time samples acquired when performing masking implementation. The reason why we use these three methods is that they are able to attack our simulated masking implementations in addition to their classic.

4.1 Isomap

ISOMAP is one representative of isometric mapping methods and maintains the global structure of the original data. Euclidean distance used in MDS [3] can't be exactly represent the distance relation between points on manifold structure. For example, a worm creeps from a point A to a point B on the curved surface. Since it can only creep on the surface. So, directly using the Euclidean distance between these two points to present the creeping distance of the worm is clearly not appropriate. Here if we use geodesic distance to represent the distance, then it can correctly reflect the shortest distance that the insect has creep. In this sense, ISOMAP improves MDS by using geodesic distance instead of Euclidean distance, the procedures of which are introduced as follows:

Neighbors Selection. The attacker defines the adjacency graph G containing all points (here each power trace is regarded as a point). If two nodes i and j are adjacent, then there is a side between them, of which the length is $d_x(i,j)$. Euclidean distance is used here. It is worth noting that Euclidean distance is only used to represent the distance between two adjacent points, and the geodetic distance is used to find the shortest path between any two sample points.

Finding Shortest Paths. The attacker calculates the approximate geodesic distance matrix $J = \{d_G(i,j)\}_{n \times n}$. If nodes i and j are connected, the shortest path $d_G(i,j) = d_x\left(i,j\right)$. Otherwise, $d_G(i,j) = \infty$. The attacker performs Floyd or Dijkstra algorithm on adjacency graph G to find the shortest paths. The geodesic distance is defined as:

$$d_G(i,j) = \begin{cases} d_G(i,j), & \text{if nodes i and j are adjacent} \\ min\left\{d_G(i,j), d_G(i,k) + d_G(k,j)\right\}, & \text{otherwise} \end{cases} \tag{11}$$

The shortest path matrix $J_G = \left(d_G\left(i,j\right)\right)$ among sample vectors finally converges after many iterations. The complexity is $O(n^2)$ when using Dijkstra algorithm to find the shortest path between two power traces. If the shortest paths between any two sample points are calculated, the complexity is $O(kn^2logn)$. k here is the number of nearest neighbors.

Computing Low Dimensional Embedding. Isomap uses MDS to obtain the low dimensional embedding, a d-dimensional space that maintains the essential geometry of power traces. The procedures of MDS are as follows:

Step 1. The attacker computes $S = (d_{ij}^2)$.

Step 2. He then sets $h_{ij} = \delta_{ij} - \frac{1}{n}$. The centralization matrix $H = (H_{ij})$. δ_{ij} satisfies that

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & j \neq j \end{cases} \tag{12}$$

Step 3. The attacker then doubles the centering and computes

$$\tau(J) = -\frac{HSH}{2}. \tag{13}$$

Let J_Y denote the Euclidean distance between samples after dimensionality reduction. ISOMAP aims at minimizing the objective function:

$$E = \left\|\tau\left(J_G\right) - \tau\left(J_Y\right)\right\|_{L^2}, \tag{14}$$

where $\|A\|_{L^2} = \sqrt{\sum_{ij} A_{ij}^2}$. The low dimensional embedding $y_1, y_2, \ldots, y_n$ are the d eigenvectors corresponding to the d minimum eigenvalues. ISOMAP uses MDS to perform eigendecomposition, the complexity is $O(n^3)$. Compared to KDA, ISOMAP is unsupervised. The attacker only needs to set two parameters k and d.

4.2 Locally Linear Embedding

Unlike ISOMAP, Locally Linear Embedding (LLE) [21] maintains local properties of data, and it's a local optimization solution. LLE considers the structure of data as linearity in local sense. Therefore, a data point x_i can be approximated by the linear combination $\sum_{j=1}^{k} w_{ij}x_j$ of its neighbor points x_j-s. That is, $x_i \approx \sum_{j=1}^{k} w_{ij}x_j$. In this way, reconstruction weights are constructed between each data point and its neighboring points. The weight vectors constructed by reconstruction weights keep the local linear structure of high-dimensional data. The local linear structure is also maintained in the low dimensional space. That is, $y_i \approx \sum_{j=1}^{k} w_{ij}y_j$.

The procedures of LLE are as follows:

Neighbors Selection. Similar to ISOMAP, the first step of LLE is also to find k nearest neighbors for each point. k is a predetermined value given by the attacker. It is a very important parameter and has great impact on the results of side channel attacks. Another important parameter here is dimension d after dimensionality reduction introduced in Section 4.1. The complexity of neighbors selection is $O(Dn^2)$.

It must be noted here that both the ISOMAP and LLE have two parameters k and d, so as to the LE introduced in Section 4.3. k-NN is also used in their first step. However, the mathematical principles of them are different. So, we can't simply think that these two parameters are same. Actually, they are tested separately in our experiments.

Computing Reconstruction Weight Matrix. Since we use $\sum_{j=1}^{k} w_{ij} x_j$ to approximate x_i, an error of reconstruction

$$min\, \varepsilon(W) = \sum_{i=1}^{n} \left\| x_i - \sum_{j=1}^{k} w_{ij} x_j \right\|^2 \tag{15}$$

is between point x_j and its neighbors. w_{ij} here denotes the reconstruction weight of data point x_i and its neighborhood x_j. If x_j is not a neighborhood of x_i, then $w_{ij} = 0$. Otherwise, $w_{ij} \neq 0$ and $\sum_{j=1}^{k} w_{ij} = 1$. In order to minimize the reconstruction error, W satisfies symmetry. The complexity of computing reconstruction weight matrix is $O((D + k)k^2 n)$.

Computing Low Dimensional Embedding. In this step, the linear relationship in local areas is established by using the weights obtained in Step 2. The low dimensional space Y is obtained by minimizing the cost function:

$$min\, \varphi(Y) = \sum_{i=1}^{n} \left\| y_i - \sum_{j=1}^{k} w_{ij} y_j \right\|^2 = \left\| Y(I - W^T) \right\|^2 = tr(YMY^T), \tag{16}$$

where y_i is the low dimensional mapping vector of x_i, and y_j is the mapping vector of x_j (the neighborhood of x_i). Y satisfies $\frac{1}{n} YY^T = I$ (I is the unit matrix), and $\sum_{i=1}^{n} y_i = 0$. The above objective function can be transformed into calculating the eigenvectors of matrix

$$M = (I - W)^T (I - W). \tag{17}$$

If the eigenvalues of M are sorted in ascending order, the first one is about 0 and discarded. Finally, the eigenvectors corresponding to the minimum d non-zero eigenvalues of matrix M are chosen as low dimensional coordinate. The complexity of computing low dimensional embedding is $O(dn^2)$. Similar to ISOMAP, LLE is also unsupervised. The attacker only needs to set two parameters k and d.

4.3 Laplacian Eigenmaps

Laplacian Eigenmaps (LE), like LLE, is also a local feature preserving algorithm. It builds a graph from neighborhood information of the data set. If two nodes i and j are adjacent in the original data space X, they are also adjacent in the subspace Y. The procedures of LE are as follows:

Constructing Adjacency Graph. We put an edge between the node i and node j if two measurements x_i and x_j are "close". Two schemes, k-NN and $\epsilon - ball$ can be used to construct of the adjacency graph. For k-NN, nodes i and j are connected by an edge if i is among n nearest neighbors of j or j is among the n nearest neighbors of i. For $\epsilon - ball$, nodes i and j are connected by an edge if $\|x_i - x_j\| < \epsilon$ where $\|\cdot\|$ is the usual Euclidean norm in R^D. Finally, the adjacency graph G is constructed.

Computing Reconstruction Weight Matrix. For each point, the attacker computes the weight coefficient

$$w_{ij} = e^{-\frac{\|x_i - x_j\|^2}{t}} \tag{18}$$

if nodes i and j are connected. Otherwise, w_{ij} is set to 0. The parameter t is an adjustable parameter, which is always set to 2 in our experiments. The attacker can also simply set

w_{ij} to 1 if nodes i and j are connected. Otherwise, w_{ij} is set to 0, of which the case also corresponds to $t = \infty$.

Computing Low Dimensional Embedding. LE keeps the local property of manifold structure in the mean sense, which depends on the distances between adjacent points. The adjacent points in G are close after dimensionality reduction. Therefore, LE minimizes the objective function:

$$\varphi(y) = \sum_{ij} \|y_i - y_j\|^2 w_{ij}. \tag{19}$$

Let J denote a diagonal weight matrix, and its entries are column sums of w, $J = (J_{ii}) = \sum_j w_{ij}$ and $L = J - W$ is the Laplacian matrix. If a constraint $YJY^T = I$ is added, the objective function can be transformed into:

$$\varphi(y) = 2YLY^T. \tag{20}$$

This is equivalent to solving

$$LY = \lambda JY. \tag{21}$$

LE then sorts the eigenvalues of Laplacian matrix in ascending order and selects the d eigenvectors corresponding to the d minimum eigenvalues. The obtained low dimensional embedding coordinates are:

$$Y = \{y_1, y_2, \cdots, y_n\}, y_i \in R^d. \tag{22}$$

The complexity of constructing adjacency graph is $O(Dn^2)$, the complexity of computing low dimensional embedding is $O(dn^2)$. Similar to LLE, d here denotes the number of eigenvalues or eigenvectors. The complexity of computing reconstruction weight matrix is less than $O(kDn)$. So, compared to ISOMAP and LLE, LE only has a small amount of computation, of which the calculation speed is high.

Compared to ISOMAP and LLE, LE not only needs to set k and d, but also needs to set a parameter t for the weight of hot kernel function (see Equation 18). In our experiments, this parameter is set to 2. We have not done much research on the impact of this parameter.

5 Experimental Results

In order to facilitate the evaluation, our experiments are performed on simulated traces. Since the locations of shares and the noise are easy to control. Moreover, the lengths of non-leaky areas can also be set shortly, so as to shorten the time for evaluation. Since the complexity of ISOMAP, LLE and KDA are $O(n^3)$. When more than 8000 power traces (300 points for each) are used, several hours are hard to get the experimental results for a repetition. Let R denote the Gaussian distributed noise with mean 0.00 and variance σ^2, Z denote the intermediate value (e.g. the output of Sbox in the first round of AES), and HW denote the Hamming weight function. The leakage model of the S-box output implementation can be modeled as

$$\mathcal{L} = \mathsf{HW}(Z) + R. \tag{23}$$

Since we aim at higher-order attacks in this paper, we divide the intermediate values into several shares using Equation 2. The leakage of each share satisfies the leakage model described above. Here the variance of noise is set to 1.00. In this case, the SNR is about 2.00. Since the variance of Hamming weights is about 2.00. The leakage areas of shares in masking implementation are often not adjacent. So, we use noise to simulate the non-leaky

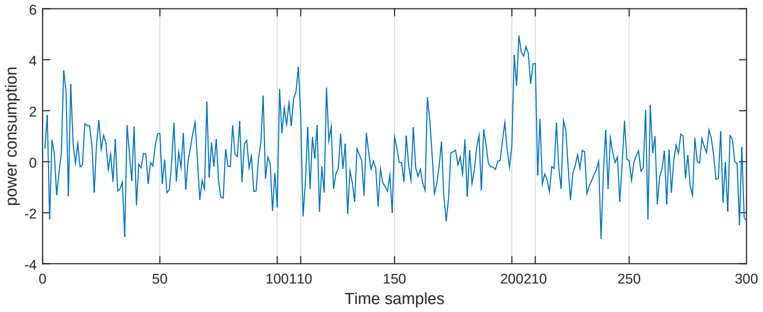

Figure 1: A simulated power traces with first-order mask.

areas that are not related to the operation of shares. The average power consumption is 0.00 and the noise standard deviation is 1.00 in these areas.

In our experiments, we insert a sample area of length 100 in front of and behind time samples of each share. Leakage area of each share includes 10 time samples. A simulated trace with first order mask is shown in Fig.1. The trace includes two leakage areas, of which the time samples range from 100 to 110 and from 200 to 210 respectively. Since Hamming weight model is used for the simulation, power traces are classified into 9 classes according to the Hamming weights of the intermediate values in KDA. Compared to KDA, our manifold learning solutions ISOMAP, LLE and LE are unsupervised, so dimensionality reduction is directly performed on the simulated power traces. Then, first order CPA is performed on the dimensionality reduced traces.

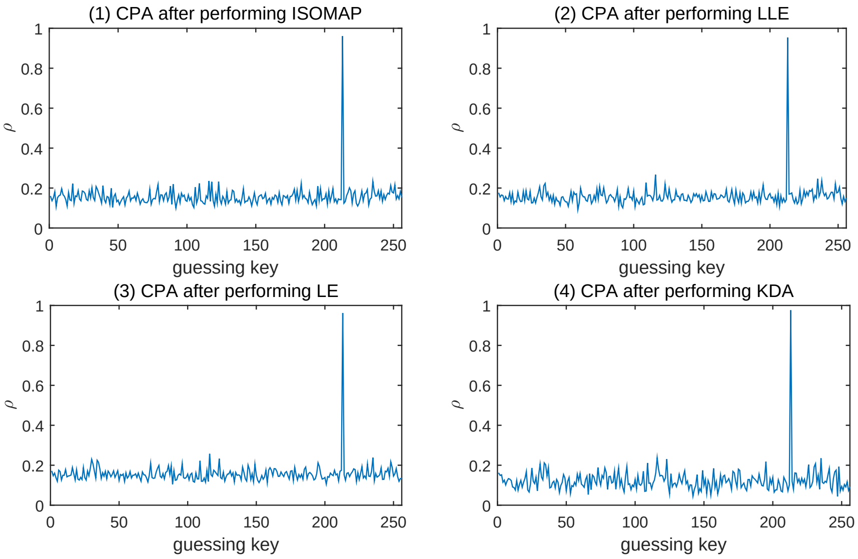

Figure 2: CPA after performing ISOMAP, LLE, LE and KDA on unprotected implementation respectively.

It is worth noting that nonlinear dimensionality reduction algorithms such as KDA and manifold learning schemes firstly introduced in this paper can also be used for linear dimensionality reduction. As shown in Fig.2, when 300 power traces are used, the correlation coefficients of the correct guessing key (213) are close to 1.00. This indicates that both KDA and manifold learning schemes (ISOMAP, LLE and LE) are very efficient when attacking unprotected implementations. However, this is not the focus of this article. The purpose of Fig.2 is to compare correlation coefficients under first-, second- and third-order attacks, and highlight the superiority of our manifold learning schemes ISOMAP, LLE and LE in comparison with KDA. We compare the experimental results of second-order attacks and third-order attacks on the simulated traces in Sections 5.1 and 5.2.

5.1 Second Order Attacks

μ is a very important parameter that needs to be optimized (see Equation 9). However, Eleonora et al. did not give the detailed method to optimize it in their paper [5]. They just tested μ for several times and chose the best one. In this paper, we also test this parameter several times and choose the best one, of which the results are shown in Table 1. We perform second-order attacks on 600 power traces and repeat this operation 200 times, the number of traces used to construct matrix M equals to 4000. The success rate (SR)[28] is the highest and equals to 0.155 when $\mu = 10$. So, in our experiments, the constant μ is always set to 10.

Table 1: Success rate under different μ-s.

μ	10^{-9}	10^{-7}	10^{-5}	10^{-3}	10^{-1}	10^{0}
SR	0.130	0.075	0.115	0.115	0.130	0.090
μ	10^{1}	10^{3}	10^{5}	10^{7}	10^{9}	10^{11}
SR	0.155	0.055	0.125	0.060	0.130	0.085

KDA is the first feasible supervised nonlinear dimensionality reduction scheme. It is not very efficient when attacking masking implementations. Take the simulated second-order masking implementation for example, to get a success rate close to 1.00, more than 3000 power traces are needed. If less than 1200 traces are used, the success rate is less than 0.50. Enlarging the noise or lengthening the number of time samples on the power traces reduces the success rate. Thus, the attacks require more traces.

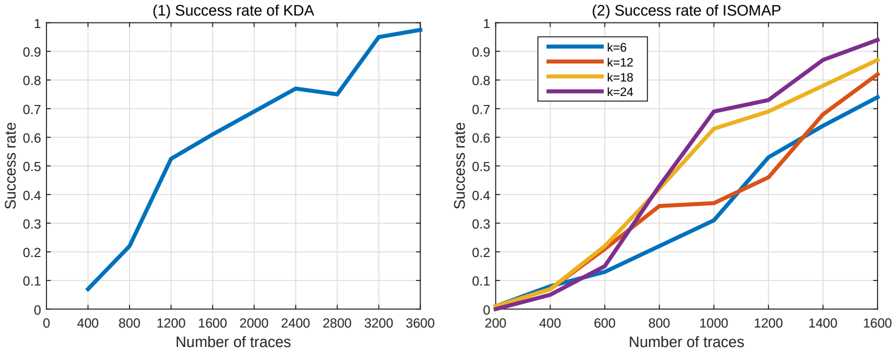

Figure 3: Success rates of KDA and ISOMAP when attacking second-order masking implementation.

Compared to KDA, LLE and LE, ISOMAP is more time-consuming when running the MATLAB code on our computer, the success rate of which is shown in Fig.3 (right). For ISOMAP, 4000 traces need more than 20 minutes on our ThinkPad X250 laptop computer with 2 Intel i5-5300U CPUs and 4GB RAM. This makes it very slow to use success rate to evaluate KDA, ISOMAP, LLE and LE. So, For ISOMAP, LLE and LE, we only repeat

200 times. This makes the success rate fluctuate, but this does not affect our analysis. As shown in Fig.3, the success rate of ISOMAP is much higher than that of KDA. The choice of constant k is crucial. When the number of traces used is small, a small k makes the attacker get a higher success rate. With increase of the number of power traces, k can be enlarged properly. When k is set to 6, 12, 18 and 24, and the number of traces is less than 400, the success rates are almost the sample. When $400 < n < 600$, the success rates corresponding to $k = 24$ is lower than these of $k = 12$ and $k = 18$. This indicates that if the threshold setting is too large, the attack efficiency will be affected. However, the threshold is relative and not fixed. It can be increased appropriately with n to achieve a higher success rate. When $n > 600$, the success rates corresponding to $k = 18$ and $k = 24$ are significantly higher than these corresponding to $k = 6$ and $k = 12$.

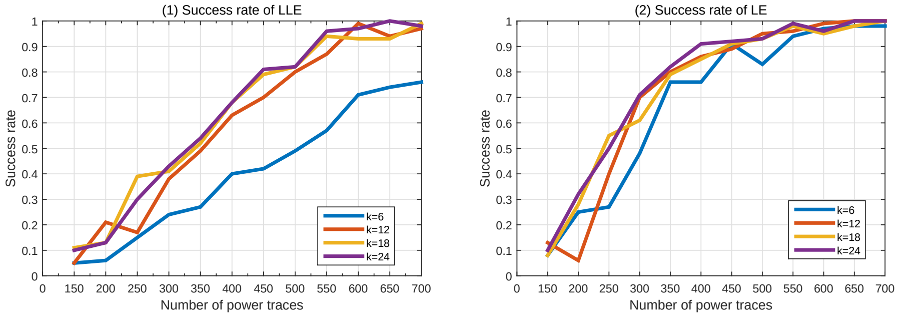

Figure 4: Success rates of LLE and LE when attacking first-order masking implementation.

ISOMAP, LLE and LE is significantly more efficient than KDA. Compared to ISOMAP, LLE and LE is faster. The success rates of them is significantly higher than that of ISOMAP (as shown in Fig.4). When $n = 550$, the success rate corresponding to LE approaches 1.00, compared to nearly 700 power traces used in LLE. It is worth noting that the parameter k also needs to be set properly. If k is very small, the success rates of LLE and LE are low, especially for LLE, of which the success rate is significantly lower than these corresponding to $k = 12$, 18 and 24. However, if k is more than 12, the success rate no longer significantly improves.

The correlation between the intermediate values and the combined leakages drops very quickly. Similarly, correlation coefficient decrease rapidly in nonlinear dimensionality reduction. We use 4000 power traces to perform KDA, ISOMAP, LLE and LE, the success rates of which are about 1.00. However, the correlation coefficients of LLE and LE are about 0.30 (as shown in Fig.5). The ρ-s of ISOMAP and KDA are very small, the former is about 0.15, the latter is about 0.10. This is not a special case, the results are representative. We find that the ρ-s of LLE and LE in many experiments are about 0.30, and ρ-s of KDA and ISOMAP are close to 0.10. The small correlation coefficients can be easily drowned out by the ρ-s corresponding to wrong keys. This also means that it is difficult for the attacker to get success using the same number of traces. That's to say, in order to get the correct key, the attacker needs more power traces.

5.2 Third Order Attacks

The correlation between the intermediate values and the combined power consumption is very weak in higher-order attacks, which makes the attacker need a large number of traces to recover the key. Even if the attacker collects measurements with very small noise and uses KDA, ISOMAP, LLE and LE introduced in this paper to perform dimensionality reduction, this still happens. As shown in Fig.6, the correlation coefficients corresponding

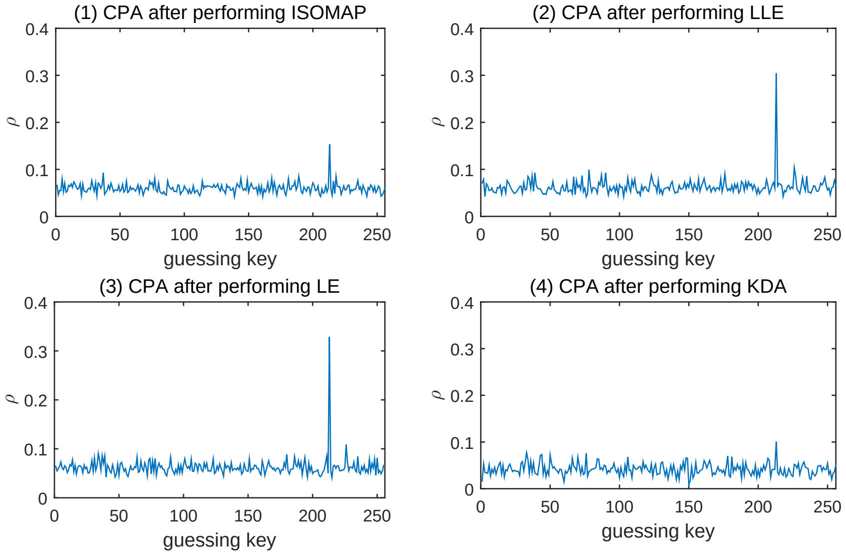

Figure 5: CPA after performing ISOMAP, LLE, LE and KDA on traces of first-order masking implementation respectively.

to KDA, ISOMAP, LLE and LE are all smaller than 0.10 when third-order attacks are performed on 4000 power traces.

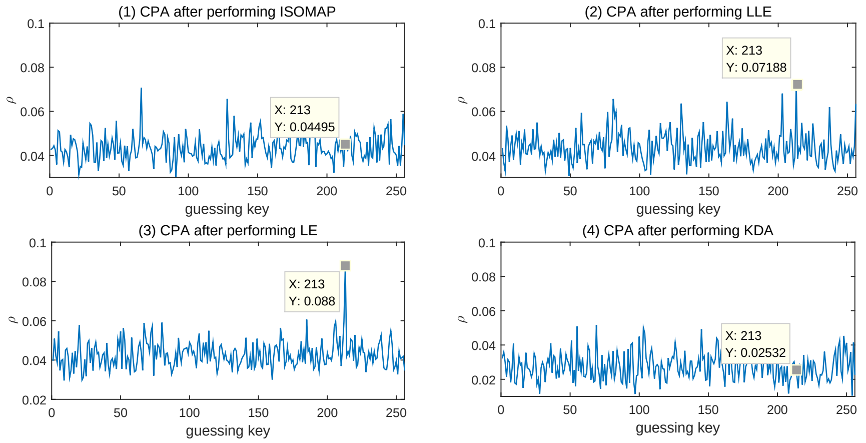

Figure 6: CPA after performing ISOMAP, LLE, LE and KDA on second-order masking implementation respectively.

We get a conclusion similar to the one of second order attack in Fig.5: the attacker will get higher correlation coefficients if he uses LLE or LE to perform dimensionality reduction rather than KDA and ISOMAP. The attack efficiency of LLE and LE is still much higher than that of KDA and ISOMAP in our third-order attack experiments. When the number of nearest neighbors k is set to 24, the success rate of LE can reach 0.90 under 5000 power traces (as shown in Fig.7). Compared to LE, the success rate of LLE is less than 0.40. k has great impact on the attack efficiency. For example, the success rate of LLE is almost 0 under 5000 power traces if k is set to 6. When k is set to 18 or 24, the success rate is higher than 0.30. The similar impact of threshold k on attack efficiency also happens in LE.

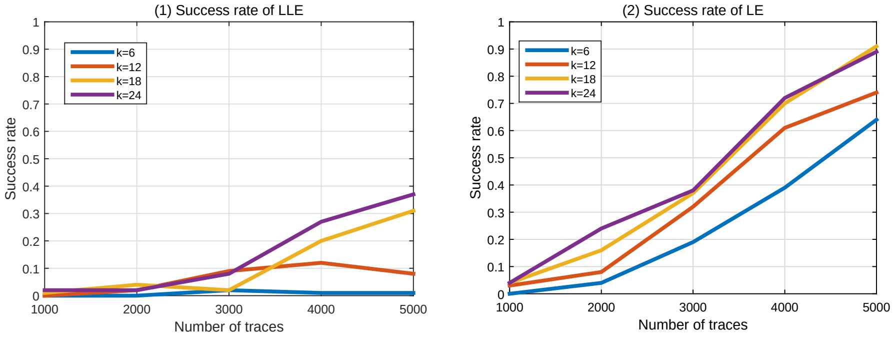

Figure 7: Success rates of LLE and LE when attacking second-order masking implementation.

We perform our experiments on 6000 power traces for each repetition to test the success rate of KDA. However, the success rate is still very close to 0. So as to ISOMAP. ISOMAP is very time-consuming in our experiments since it uses Dijkstra algorithm to find the shortest path between any two power traces. So, we do not give the corresponding graphs for both of them. This paper only considers using the most basic methods. A lot of improvements have been proposed to enhance KDA, ISOMAP, LLE and LE, other manifold learning schemes can also be used to reduce dimension. These work may also be introduced into attacking masking implementations. There also exists other side channel attacks more efficient than CPA. These improvements can be left as open problems here.

6 Conclusions

Informative extraction (e.g. POI detection scheme PP [10] and KDA) for masking implementation is one of the most challenging problems of side channel attacks. It is also crucial for higher-order attacks and leakage assessment. However, it is rarely studied. In this case, we think KDA in [5] is a milestone for informative extraction towards masking implementation, since it makes the dimensionality reduction of high-dimensional nonlinear leakage samples practical for the first time. However, this scheme is supervised, noise-sensitive and inefficient, requires a large number of power traces to perform attacks.

In this paper, we introduce manifold learning into nonlinear dimensionality reduction for masking implementations. We use un-supervised schemes ISOMAP, LLE and LE to realize this target. Compared to KDA, these three methods are much more efficient at second-order attacks. The correlation between intermediate values and the components after dimensionality reduction in these three schemes are higher than that of KDA, too. We also compare ISOMAP, LLE and LE with KDA on third-order attacks. ISOMAP is very time-consuming and the success rate of KDA is very low. In this case, LLE and LE still work very well. Manifold learning schemes introduced in this paper are unsupervised, which makes the attacks easy to perform. The attacker needs to set fewer parameters. These above advantages make manifold learning meaningful when attacking masking implementations.

In this paper, we directly use ISOMAP, LLE and LE to perform dimensionality reduction, any improvement is made. Moreover, original CPA is directly performed here. This makes the correlation between intermediate values and dimensionality reduced samples drop quickly. Manifold learning towards masking implementations is still worthy of further study. For example, better kernel functions, more accurate intrinsic dimension estimation and more efficient dimensionality reduction algorithm implementations. It may be the most efficient way to use non-linear dimensionality reduction methods to attack the high-dimensional non-linear time samples of masked implementations.

References

[1] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The EM side-channel(s). In *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, pages 29–45, 2002.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[3] I. Borg and P. Groenen. Modern multidimensional scaling: theory and applications. *Journal of Educational Measurement*, 40(3):277–280, 2003.

[4] E. Brier, C. Clavier, and F. Olivier. Correlation power analysis with a leakage model. In *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, pages 16–29, 2004.

[5] E. Cagli, C. Dumas, and E. Prouff. Kernel discriminant analysis for information extraction in the presence of masking. In *Smart Card Research and Advanced Applications - 15th International Conference, CARDIS 2016, Cannes, France, November 7-9, 2016, Revised Selected Papers*, pages 1–22, 2016.

[6] S. Chari, J. R. Rao, and P. Rohatgi. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, pages 13–28, 2002.

[7] J. Coron, E. Prouff, and M. Rivain. Side channel cryptanalysis of a higher order masking scheme. In *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, pages 28–44, 2007.

[8] J. Coron, E. Prouff, M. Rivain, and T. Roche. Higher-order side channel security and mask refreshing. In *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*, pages 410–424, 2013.

[9] J. A. Costa and A. O. H. III. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Trans. Signal Processing*, 52(8):2210–2221, 2004.

[10] F. Durvaux, F. Standaert, N. Veyrat-Charvillon, J. Mairy, and Y. Deville. Efficient selection of time samples for higher-order DPA with projection pursuits. In *Constructive Side-Channel Analysis and Secure Design - 6th International Workshop, COSADE 2015, Berlin, Germany, April 13-14, 2015. Revised Selected Papers*, pages 34–50, 2015.

[11] T. Eisenbarth, C. Paar, and B. Weghenkel. Building a side channel based disassembler. *Trans. Computational Science*, 10:78–99, 2010.

[12] D. Genkin, A. Shamir, and E. Tromer. RSA key extraction via low-bandwidth acoustic cryptanalysis. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 444–461, 2014.

[13] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel. Mutual information analysis. In *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, pages 426–442, 2008.

[14] X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 153–160, 2003.

[15] P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 388–397, 1999.

[16] E. Levina and P. J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada*, pages 777–784, 2004.

[17] E. Oswald, S. Mangard, C. Herbst, and S. Tillich. Practical second-order DPA attacks for masked smart card implementations of block ciphers. In *Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2006, Proceedings*, pages 192–207, 2006.

[18] C. Rechberger and E. Oswald. Practical template attacks. In *Information Security Applications, 5th International Workshop, WISA 2004, Jeju Island, Korea, August 23-25, 2004, Revised Selected Papers*, pages 440–456, 2004.

[19] O. Reparaz. Detecting flawed masking schemes with leakage detection tests. In *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, pages 204–222, 2016.

[20] O. Reparaz, B. Gierlichs, and I. Verbauwhede. Selecting time samples for multivariate DPA attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, pages 155–174, 2012.

[21] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[22] B. Schölkopf, A. J. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[23] K. Schramm, G. Leander, P. Felke, and C. Paar. A collision-attack on AES: combining side channel- and differential-attack. In *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, pages 163–175, 2004.

[24] K. Schramm and C. Paar. Higher order masking of the AES. In *Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2006, Proceedings*, pages 208–225, 2006.

[25] K. Schramm, T. J. Wollinger, and C. Paar. A new class of collision attacks and its application to DES. In *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*, pages 206–222, 2003.

[26] K. Sricharan, R. Raich, and A. O. H. III. Optimized intrinsic dimension estimator using nearest neighbor graphs. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2010, 14-19 March 2010, Sheraton Dallas Hotel, Dallas, Texas, USA*, pages 5418–5421, 2010.

[27] F. Standaert and C. Archambeau. Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, pages 411–425, 2008.

[28] F. Standaert, T. Malkin, and M. Yung. A unified framework for the analysis of side-channel key recovery attacks. In *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, pages 443–461, 2009.

[29] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[30] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):40–51, 2007.

[31] J. Yang, X. Gao, D. Zhang, and J. Yang. Kernel ICA: an alternative formulation and its application to face recognition. *Pattern Recognition*, 38(10):1784–1787, 2005.

[32] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J. Scientific Computing*, 26(1):313–338, 2004.