A Statistical Analysis of IrisCode and its Security Implications

Adams Wai Kin Kong, *IEEE Member*
School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore, 639798
Email: adamskong@ntu.edu.sg

Abstract — IrisCode has been used to gather iris data for 430 million people. Because of the huge impact of IrisCode, it is vital that it is completely understood. This paper first studies the relationship between bit probabilities and a mean of iris images[1] and then uses the Chi-square statistic, the correlation coefficient and a resampling algorithm to detect statistical dependence between bits. The results show that the statistical dependence forms a graph with a sparse and structural adjacency matrix. A comparison of this graph with a graph whose edges are defined by the inner product of the Gabor filters that produce IrisCodes shows that partial statistical dependence is induced by the filters and propagates through the graph. Using this statistical information, the security risk associated with two patented template protection schemes that have been deployed in commercial systems for producing application-specific IrisCodes is analyzed. To retain high identification speed, they use the same key to lock all IrisCodes in a database. The belief has been that if the key is not compromised, the IrisCodes are secure. This study shows that even without the key, application-specific IrisCodes can be unlocked and that the key can be obtained through the statistical dependence detected.

Keywords: Biometrics, iris recognition, statistical dependence, Daugman algorithm, template protection

1. Introduction

IrisCode[2] is the most influential iris recognition algorithm in use [1]. As of Sept 2013, the total number of people in India alone who have had their iris patterns enrolled by IrisCode is approximately 430 million. The Unique Identification Authority of India (UIDAI) is enrolling about one million persons per day, at 36,000 active stations, and UIDAI plans to have the entire population of 1.2 billion people enrolled within 3 years. In addition to the unique identification project carried out by UIDAI, other flagship deployments include those at Amsterdam's Schiphol airport, ten UK airports, US/Canadian borders and all 32 air, land, and seaports in the United Arab Emirates [16]. The computational advantages of IrisCode, including its extremely high matching speed for large-scale identification and automatic threshold adjustment based on image quality (e.g., the number of effective bits) and database size [1-3], play an important role in its market success. In addition to its massive deployments, IrisCode influences numerous biometric recognition methods [4-15]. Because of its impact, IrisCode has to be understood clearly.

[1] The mean of iris images is defined as the average of independent iris images.

[2] In this paper, IrisCode is used interchangeably to refer to both the method and the features of the iris recognition algorithm developed by Daugman.

Many iris recognition papers have been published, but our understanding of this crucial algorithm, which will soon impact over 16% of the world's population, remains incomplete. In the original scientific paper describing IrisCode, Daugman noted that the bits "0" and "1" in an IrisCode are equiprobable and that its imposter distribution is binomial [1]. He also reported that the number of degrees of freedom of the imposter binomial distribution is much less than the number of bits because they are not independent [1, 24]. Hollingsworth et al.[3] analyzed bit stability in their iris codes to separate the best bits and fragile bits and thereby enhance recognition performance [17-18]. Santos and Proenca noted that the spatial distribution of concordant bits can be used as features for performance improvement [33]. Kong et al. mathematically derived the following points: IrisCode is a compression algorithm and also a clustering algorithm with four clustering centers; with respect to a phase parameter, the locus of a Gabor function is an ellipse in a two-dimensional space, which can often be approximated by a circle; the Gabor function can be considered a phase-steerable filter; the phase distance in a particular domain can be implemented through the bitwise hamming distance and a specially designed coding scheme; each IrisCode is a convex polyhedral cone in a hyperspace; the central ray of this cone is an optimal ray of an objective function on a set of distributions and also a very rough iris image; Gabor filters can be used as a Gabor atom detector; and the magnitude and phase of a target Gabor atom can be estimated from the magnitude and phase of the corresponding Gabor response [4, 19-21]. Nevertheless, a number of statistical properties of IrisCode have not been studied, e.g., the relationship between bit probabilities and a mean of iris images and the relationship between statistical dependence in IrisCodes and the Gabor filters. In addition to the properties of IrisCode, researchers have investigated security risks associated with iris templates. Kong used his theoretical results to reconstruct iris images from IrisCodes [20]. Venugopalan et al. and Galbally et al. used an information-embedding approach and a genetic algorithm, respectively, to synthesize iris images from binary iris templates with sizes in the range of 9,600 to 21,600 bits [31-32]. These security studies were based on the assumption that iris templates are unprotected. Statistical information is always crucial in security analysis.

Though the IrisCode computational procedures are well known, a brief computational summary is presented here for notation consistency. Two-dimensional Gabor filters with zero direct current (DC) are applied to an iris image in a dimensionless polar coordinate system, $I_0(\rho,\phi)$, to extract phase information. The complex Gabor response is quantized into two bits by the following inequalities:

[3] Hollingsworth et al. mainly used 1D log-Gabor wavelets in their study [17-18].

$$b_r = 1 \quad if \quad \mathrm{Re}\left(\int_{\rho}\int_{\varphi} \boldsymbol{I}_0(\rho,\varphi) e^{-(r_o-\rho)^2/\alpha_o^2} e^{-(\theta_o-\varphi)^2/\beta_o^2} e^{-i\omega_o(\theta_o-\varphi)} \rho d\rho d\varphi \right) \geq 0 , \tag{1}$$

$$b_r = 0 \quad if \quad \mathrm{Re}\left(\int_{\rho}\int_{\varphi} \boldsymbol{I}_0(\rho,\varphi) e^{-(r_o-\rho)^2/\alpha_o^2} e^{-(\theta_o-\varphi)^2/\beta_o^2} e^{-i\omega_o(\theta_o-\varphi)} \rho d\rho d\varphi \right) < 0 , \tag{2}$$

$$b_i = 1 \quad if \quad \mathrm{Im}\left(\int_{\rho}\int_{\varphi} \boldsymbol{I}_0(\rho,\varphi) e^{-(r_o-\rho)^2/\alpha_o^2} e^{-(\theta_o-\varphi)^2/\beta_o^2} e^{-i\omega_o(\theta_o-\varphi)} \rho d\rho d\varphi \right) \geq 0 , \tag{3}$$

$$b_i = 0 \quad if \quad \mathrm{Im}\left(\int_{\rho}\int_{\varphi} \boldsymbol{I}_0(\rho,\varphi) e^{-(r_o-\rho)^2/\alpha_o^2} e^{-(\theta_o-\varphi)^2/\beta_o^2} e^{-i\omega_o(\theta_o-\varphi)} \rho d\rho d\varphi \right) < 0 , \tag{4}$$

where ω_o is the spatial frequency, α_o and β_o control the shape of the Gaussian function and (r_o, θ_o) is the center/location of the filter in the spatial domain [2]. One thousand and twenty-four Gabor filters with different parameters $(r_o, \theta_o, \omega_o, \alpha_o, \beta_o)$ generate 2,048 bits to form an IrisCode, and a mask excludes the corrupted bits, e.g., from the eyelashes and eyelids [2]. The bitwise hamming distance is employed for high-speed matching. Although Eqs. 1–4 are presented as integrals, in the rest of this paper, the two-dimensional functions are expressed in discrete form as matrices, and these matrices are expressed as vectors after lexicographic ordering. Therefore, all filtering operations are expressed as inner products.

To prevent cross-matching of biometric templates stored in different application systems and to issue new templates to replace compromised templates, Braithwaite and his coworkers, including Daugman, patented a framework, owned by Iridian Technologies Inc., to produce application-specific templates [22-23]. This framework is also expected to preclude obtaining original templates from application-specific templates. This framework has been applied to IrisCode and deployed in commercial systems [25]. Two realizations of this framework for IrisCode were proposed. The first realization computes an application-specific IrisCode $B_{\otimes j\bullet}$ through

$$B_{\otimes j\bullet} = S_u \otimes B_{j\bullet} , \tag{5}$$

where $B_{j\bullet}$ is an original IrisCode computed from Eqs. 1-4, S_u is a random bit vector generated by a key u, and $\otimes$ is a bitwise XOR operator. $B_{j\bullet}$, $B_{\otimes j\bullet}$ and S_u can be regarded as binary row vectors with a length of 2,048. The subscript $\otimes$ in $B_{\otimes j\bullet}$ is only a symbol, not an operator. When a bit in S_u is one, the corresponding bit in $B_{j\bullet}$ will be changed; otherwise, it will remain the same. In this realization, the masks of $B_{j\bullet}$ and $B_{\otimes j\bullet}$ are the same. Using this scheme, $2^{2048} \approx 3.2 \times 10^{616}$ application-specific IrisCodes can be generated from one original IrisCode. To guarantee that $B_{\otimes j\bullet}$ and

$B_{j\bullet}$ are sufficiently different, Braithwaite et al. suggested retaining at least half of the bits in S_u as one. Thus, the total number of effective application-specific IrisCodes is 1.6×10^{616} [22]. The second realization computes an application-specific IrisCode $B_{\times j\bullet}$ and its mask $M_{\times j\bullet}$ through

$$B_{\times j\bullet} = B_{j\bullet} \times P_u \text{ and } M_{\times j\bullet} = M_{j\bullet} \times P_u, \tag{6}$$

where P_u is a 2,048-by-2,048 random permutation matrix generated by a key u and $M_{j\bullet}$ is the mask of $B_{j\bullet}$. The total number of different permutation matrixes is $2048! \approx 10^{5894}$. In this study, Eqs. 5 and 6 are called the XOR protection scheme and the permutation protection scheme, respectively.

These two schemes have many advantages. Application-specific IrisCodes and original IrisCodes have the same format, and therefore, the hamming distance can still be used as a dissimilarity measure. Another advantage is that these schemes do not cause any performance degradation. The speed of issuing new application-specific IrisCodes is very high. System administrators can regularly replace old templates with new templates in very short periods of time. Doing so significantly reduces the risk of attacks based on compromised templates. If the same key is used to lock all the IrisCodes in a database, high identification speed can be retained. Eqs. 5 and 6 are bijective functions. Once P_u and S_u are known, the original IrisCodes can be revealed. Braithwaite et al. utilize network and software approaches to secure P_u and S_u. It is assumed that without P_u and S_u, application-specific IrisCodes cannot be unlocked, and application-specific IrisCodes in different application systems cannot be matched. These two protection schemes are much more important than other iris template protection schemes because they have been deployed in commercial systems [25]. IrisCode has enrolled over 400 million users in approximately 170 countries. Any security risk in these protection schemes may endanger numerous people and organizations. This study first describes an examination of the relationship between bit probabilities and a mean of iris images and the statistical dependence between bits in IrisCodes. The statistical information is used to analyze security risks in the two protection schemes in the case that all IrisCodes in a database are locked by the same key for real-time large-scale identification but the key is not compromised.

The rest of this paper is organized as follows. Section 2 presents the testing databases, the relationship between bit probabilities and a mean of iris images and the statistical dependence between bits detected using the Chi-square statistic, the correlation coefficient and a resampling algorithm. Section 3 presents two graph-based algorithms to unlock application-specific IrisCodes without keys. Section 4 reports the experimental results. Section 5 discusses the implications of the statistical and experimental findings presented in this study.

2. Statistics and Graphs Derived from IrisCodes

2.1. Notations and Graph Representations

For purposes of presentation clarity, a set of notations is presented. Let an IrisCode database be a matrix B, with a size of n by 2048, of n IrisCodes from s different irises. When $n = s$, meaning that each iris has only one IrisCode in the database, the database is denoted by B_s. Each row of B containing one IrisCode is denoted by $B_{j\bullet} = [b_{j1} \cdots b_{j2048}]$, where $b_{jk} \in \{0,1\}$, $k \in \{1,\cdots,2048\}$ and $j \in \{1,\cdots,n\}$, and each column of B containing bits at the same location from different IrisCodes is denoted by $B_{\bullet k}$. Let the corresponding mask database be M, whose j^{th} row, denoted by $M_{j\bullet} = [m_{j1} \cdots m_{j2048}]$, is the mask of $B_{j\bullet}$ and whose k^{th} column, denoted by $M_{\bullet k}$, is a column vector formed by the k^{th} bits in the masks. The masks are used to discriminate between uncorrupted bits from iris regions and corrupted bits from noise, including eyelids, eyelashes and reflections. As with B_s, when $n = s$, the mask database is denoted by M_s. If the IrisCodes in B are protected by Eqs. 5 and 6, the application-specific IrisCode databases are denoted by $B_{\otimes}$ and $B_{\times}$, respectively, and the corresponding mask databases are denoted by $M_{\otimes}$ and $M_{\times}$, respectively. Note that $M_{\otimes} = M$. To obtain B and M from $B_{\otimes}$, $B_{\times}$, $M_{\otimes}$ and $M_{\times}$, the algorithms presented in Section 3 require another IrisCode database and the corresponding mask database from unrelated iris images, e.g., images from public databases. These databases are denoted by B_U and M_U, respectively. The term b_{jk} in B is determined by the sign of the inner product, $< I_j, g_k >= I_j^T g_k$, where g_k is either a real part or an imaginary part of a zero-DC Gabor filter, I_j is a digital iris image in a dimensionless polar coordinate system, multiplied by ρ, and T represents a transpose. The inner product is in fact a discrete version of the integrals in Eqs. 1–4. Both I_j and g_k are column vectors. When I_j is considered to be a two-dimensional image, $I_j(x,y)$ is used to denote its pixel value at the location (x,y). The term 1_v represents a vector whose elements are all ones, i.e., $1_v = [1,\cdots,1]^T$.

In this section, the statistical relationships between bits in IrisCodes and between bits in their masks are examined, and graphs are used to represent this statistical information. The nodes in the graphs represent bit locations, and their edges represent the relationships between two bit locations. Fig. 1(a) illustrates a graph displaying the statistical dependence between bits in IrisCodes, where ψ is a function that uses two column vectors of B and the corresponding column vectors of M to compute the level of their statistical dependence. This function is in fact the proposed resampling algorithm based on the Chi-square statistic and the correlation coefficient. To understand the source of this dependence, another graph whose edges are the inner products of the filters, i.e., $< g_j, g_k >$, is constructed (Fig. 1(b)). One additional graph whose edges are defined by the correlation coefficients between mask bits is also discussed (Fig. 1(c)). These graphs are utilized to define their adjacency matrixes for the statistical analysis and the proposed graph-based algorithms. The first graph (Fig. 1(a)) is used to reveal the structure of the statistical dependence in IrisCodes; the second graph (Fig. 1(b)) is used to explain the source of the statistical dependence and the last graph (Fig. 1(c)) is used in Section 3 to analyze security risks in application-specific IrisCodes. These graphs are employed to study different relationships between bit locations. They are not simply for analyzing bit probabilities in individual bit locations. The adjacency matrix of the mask correlation graph depends highly on eyelashes, eyelids and segmentation processes. IrisCode, which only extracts information from a single channel, is the focus of this paper. Thus, dependence and correlation between different channels in color iris images are not considered. All these graphs are undirected because ψ, ϕ and the inner product are commutative. Note that their edge values can be negative. The precise mathematical definitions of ψ and ϕ are given in Fig. 7 and Eq. 15, respectively. Their adjacency matrixes are denoted by Ψ, Φ and G, respectively, and each has a size of 2,048 by 2,048. The element in the j^{th} row and the k^{th} column of Ψ is $\Psi(j,k) = \psi(B_{\bullet j}, B_{\bullet k}, M_{\bullet j}, M_{\bullet k})$. Similarly, $\Phi(j,k) = \phi(M_{\bullet j}, M_{\bullet k})$ and $G(j,k) =< g_j, g_k >$. To emphasize the adjacency matrixes computed from $(B_{\otimes}, M_{\otimes})$, $(B_{\times}, M_{\times})$ and (B_U, M_U), the subscripts $\otimes$, $\times$ and U are added. For example, $\Psi_{\otimes}$ is the adjacency matrix of the statistical dependence graph computed from $(B_{\otimes}, M_{\otimes})$. Section 2.2 presents the testing databases. Section 2.3 discusses the relationship between bit probabilities and a mean of iris images. Section 2.4 examines the statistical dependence between bits in IrisCodes and its relationship with $< g_j, g_k >$ and also discusses the correlation between bits in the masks.

2.2. Databases

Two public iris databases, the West Virginia University (WVU) iris database and the UBIRIS.v1 database [26-27], are employed to analyze the statistical properties of IrisCodes and evaluate the algorithms proposed in Section 3. The WVU iris database[4] contains 3,099 iris images from 472 irises, and the UBIRIS.v1 database contains 1,877 images from 241 irises. All the images in the WVU iris database are included in the analysis and experiments. However, 48 images from the UBIRIS.v1 database were removed because of their poor quality (some images do not even have irises). Fig. 2 gives

[4] Some mislabeled images were corrected.

examples of the removed iris images. The WVU iris images were captured in an infrared lighting environment, while the UBIRIS.v1 iris images were captured in a visible lighting environment. The original images in the UBIRIS.v1 database are color images. We only employ their red components for evaluation because the iris texture in this channel is the clearest (Fig. 3).

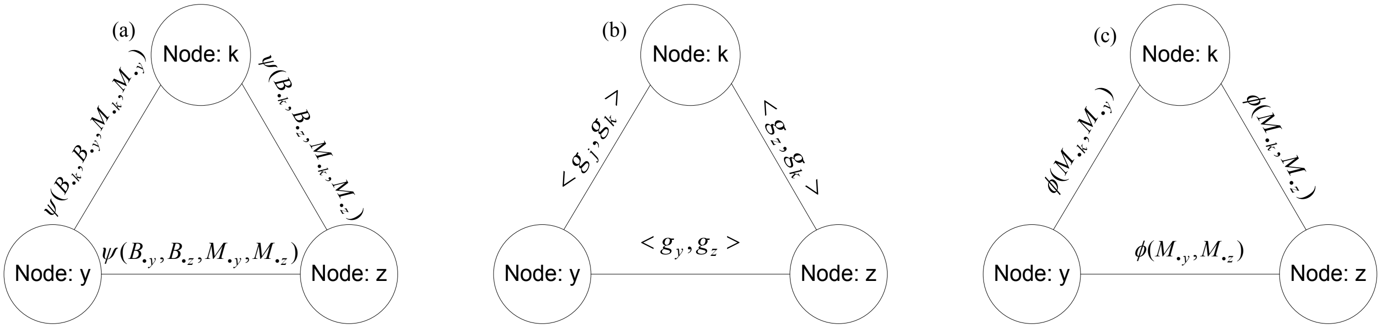

(a)
Node: k
Node: y
Node: z
$\psi(B_{\cdot k},B_{\cdot y},M_{\cdot k},M_{\cdot y})$
$\psi(B_{\cdot k},B_{\cdot z},M_{\cdot k},M_{\cdot z})$
$\psi(B_{\cdot y},B_{\cdot z},M_{\cdot y},M_{\cdot z})$
(b)
Node: k
Node: y
Node: z
$<g_j,g_k>$
$<g_z,g_k>$
$<g_y,g_z>$
(c)
Node: k
Node: y
Node: z
$\phi(M_{\cdot k},M_{\cdot y})$
$\phi(M_{\cdot k},M_{\cdot z})$
$\phi(M_{\cdot y},M_{\cdot z})$

Fig. 1 Three graphs that display information in IrisCodes, their masks and Gabor filters: (a) a statistical dependence graph, (b) a Gabor graph and (b) a mask correlation graph.

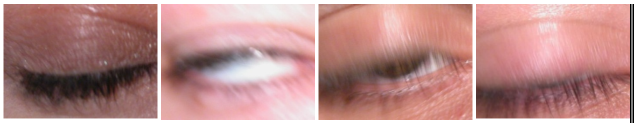


Fig. 2 Examples of images removed from the UBIRIS.v1 database. (Color figure)

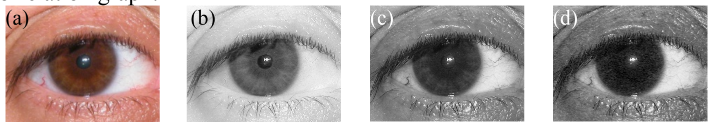


Fig. 3 Iris texture in different channels. (a) is a color image and (b), (c) and (d) are the R, G and B components of (a), respectively. (Color figure)

2.3. The Relationship between Bit Probabilities and a Mean of Iris Images

Bit probabilities are the most fundamental statistic in IrisCodes. Given an IrisCode database B_s containing n IrisCodes from n irises, the corresponding mask database M_s and the images $I_1,\cdots,I_n$ generating them, the relationship between their bit probabilities and the mean of the iris images are studied in this subsection. The sample mean of the iris images is $\bar{I}=\frac{1}{n}\sum_{j=1}^{n} I_j$, and the sample probability of the k^{th} bit being one is $\hat{p}_k=\sum_{j=1}^{n}(b_{jk}\cap m_{jk}) \Big/ \sum_{j=1}^{n} m_{jk}$, where $\cap$ is a bitwise AND operator. To reduce the influence of imperfect segmentation, 300 sample bit probabilities with the greatest $\sum_{j=1}^{n} m_{jk}$ values are selected for analysis. This number is based on some preliminary experiments on the databases. If more sample bit probabilities are selected, the results given below will have higher chances influenced by eyelashes and eyelids. Both the UBIRIS.v1 database and the WVU database contain a lot of very low quality iris images, and therefore the same number of sample bit probabilities is selected from them. This number is enough to eliminate most of the corrupted bits which are not detected by the segmentation algorithm and retain many uncorrupted bits for the statistical analysis. Their corresponding $<\bar{I},g_k>$ are also computed. The blue circles in Fig. 4 show the sample bit probabilities versus the corresponding $<\overline{I}, g_k>$. One image per iris in the testing databases is randomly selected in this analysis. This figure indicates that the bit probabilities are not always approximately 0.5 and that they are highly correlated with $<\overline{I}, g_k>$. The correlation coefficients for the UBIRIS.v1 database and the WVU database are 0.76 and 0.89, respectively. These large correlation coefficients can be explained by Eq. 9 and will be discussed. Some may expect that the blue circles in Fig. 4(a) should spread more than the blue circles in Fig. 4(b) because the images in the UBIRIS.v1 database are more heterogeneous. It should be noted that the blue circles in Fig. 4 are computed from very clean iris regions. Eyelashes and eyelids should not affect the result significantly. Another point should be emphasized that the red channels of the UBIRIS.v1 images are used in this study, while the WVU images are NIR images. Because NIR provides stronger iris signals than red light, $\left|<\overline{I}_{NIR}, g_k>\right|$ can be greater than $\left|<\overline{I}_{red}, g_k>\right|$, where $\overline{I}_{NIR}$ represents the mean of the NIR iris images and $\overline{I}_{red}$ represents the mean of the red channels of the color iris images. Furthermore, $\overline{I}_{NIR}$ and $\overline{I}_{red}$ are not normalized, meaning that their norms are not the same, i.e., $\left\|\overline{I}_{NIR}\right\| \neq \left\|\overline{I}_{red}\right\|$. These norms highly depend on imaging environments and have a great impact on $\left|<\overline{I}_{NIR}, g_k>\right|$ and $\left|<\overline{I}_{red}, g_k>\right|$. Thus, the blue circles in Fig. 4(b) spreading more than the blue circles in Fig. 4(a) are not abnormal.

To investigate the correlation between bit probabilities and the mean of the iris images, the intensities of the iris images are modified by the equation $\hat{I}(x,y) = (0.5 \times A(x,y) + 1)I(x,y)$, where $A(x,y) = \hat{A}(x,y) / \max_{(x,y)} \hat{A}(x,y)$. $\hat{A}(x,y) = \exp\left(-0.5\left[[x\ y] - \mu_1^T\right]\Sigma^{-1}\left[[x\ y]^T - \mu_1\right]\right) + \exp\left(-0.5\left[[x\ y] - \mu_2^T\right]\Sigma^{-1}\left[[x\ y]^T - \mu_2\right]\right)$, where $\mu_1 = [32\ 64]^T$, $\mu_2 = [32\ 488]^T$ and $\Sigma = \begin{bmatrix} 500 & 0 \\ 0 & 500 \end{bmatrix}$. A is a function composed of two Gaussian terms with a maximum value of one. Fig. 5 shows two modified images. The modification simulates extra light shining on the left and right parts of the irises. The same statistical analysis is repeated for the modified images, and the results are plotted as the blue circles in Fig. 6. The bit probabilities change significantly, although they are still positively correlated with $<\overline{I}, g_k>$. These changes indicate that bit probabilities can be influenced by lighting conditions and they can be far away from 0.5. The correlation coefficients computed from the blue circles in Figs. 6(a) and (b) are 0.93 and 0.94, respectively. The relationship between bit probabilities and $<\overline{I}, g_k>$ can be explained by the following equations. Using the mathematical expectation, the probability of the k^{th} bit in an IrisCode being one is defined as

$$E(b_k)=\int\frac{1}{2}(sign(g_k^T I)+1)\times f(I)dI\ , \tag{7}$$

where E represents an operator of the mathematical expectation, sign is the sign function and f is the probability density function of independent iris images. Eq. 7 assumes that $<I,g_k>=0$ is measure zero and can be ignored. Rewriting Eq. 7,

$$E(b_k)=\frac{1}{2}\int\frac{g_k^T I\times f(I)}{\left|g_k^T I\right|}dI+\frac{1}{2}, \tag{8}$$

is obtained. Assuming that $w_k\approx\left|g_k^T I\right|$ for all iris images, Eq. 8 can be further simplified as

$$E(b_k)\approx\frac{1}{2w_k}g_k^T\int I\times f(I)dI+\frac{1}{2}. \tag{9}$$

Note that $\int I\times f(I)dI$ is the mean of iris images and that w_k depends on g_k, meaning that w_k is not a constant for different bit locations. Eq. 9 clearly shows that $<\overline{I},g_k>$ influences $E(b_k)$, but their relationship is nonlinear because w_k depends on I. The red crosses in Figs. 4 and 6 show the approximate $E(b_k)$ values given by Eq. 9, plotted against $<\overline{I},g_k>$. These two figures validate Eq. 9 and demonstrate its predictive capability. The mean differences between the approximate $E(b_k)$ and $\hat{p}_k$ are 0.025, 0.025, 0.060 and 0.056 for the data in Figs. 4(a), 4(b), 6(a) and 6(b), respectively. If w_k in different bit locations are roughly the same, Eq. 9 becomes a linear equation and explains the large correlation coefficients of the blue circles in Figs. 4 and 6.

To demonstrate that the bit probabilities are in fact influenced by illumination, the following simple optical model is employed

$$I(x,y)=\int_0^\infty L(x,y,\lambda)R_{iris}(x,y,\lambda)S_c(\lambda)d\lambda\ , \tag{10}$$

where L is an illuminant, R_{iris} is the reflectance of an iris, S_c is the spectral response function of a camera and λ is a wavelength. Using Eq. 10 and $\hat{I}(x,y)=(0.5\times A(x,y)+1)I(x,y)$,

$$\hat{I}(x,y)=\int_0^\infty(0.5\times A(x,y)+1)L(x,y,\lambda)R_{iris}(x,y,\lambda)S_c(\lambda)d\lambda\ , \tag{11}$$

is obtained. Let $\hat{L}(x,y,\lambda)=(0.5\times A(x,y)+1)L(x,y,\lambda)$. $\hat{I}$ can be regarded as an image captured under the illuminant $\hat{L}$. Eqs. 9 and 11 indicate that illuminants can affect the bit probabilities. It agrees with the changes of the blue circles in Figs. 4 and 6. In addition, Eq. 9 shows that if $\int I\times f(I)dI$ and g_k are orthogonal, the expected probability $E(b_k)\approx 0.5$, which agrees with Daugman's result, published in 1993 [1]. If $\int I\times f(I)dI=e1_v$, where $e\in\Re$ and $1_v=[1,\cdots,1]^T$, the inner product of $\int I \times f(I)dI$ and g_k will be zero because g_k is a zero-DC filter. That is, when the intensity of the mean of the iris images is the same everywhere, the bit probabilities will be approximately 0.5.

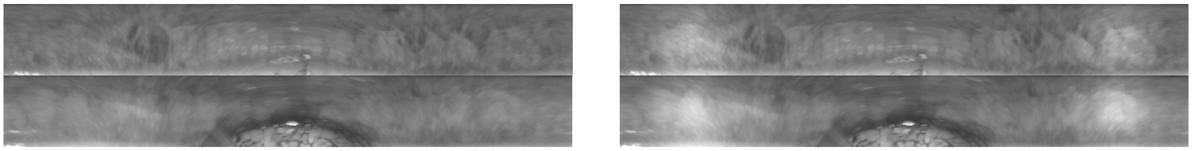


Fig. 5 The left column shows original images and the right column shows images modified by the equation $\hat{I}(x,y) = (0.5 \times A(x,y)+1)I(x,y)$.

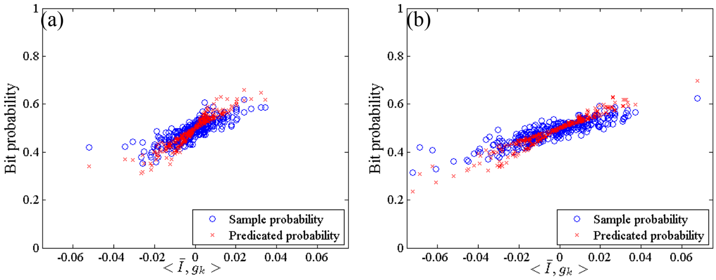


Fig. 4 The relationship between $< \bar{I}, g_k >$ and bit probabilities. (a) The results from the UBIRIS.v1 database and (b) results from the WVU database. One outline in (a) with $< \bar{I}, g_k > > 0.1$ is not shown. (Color figure)

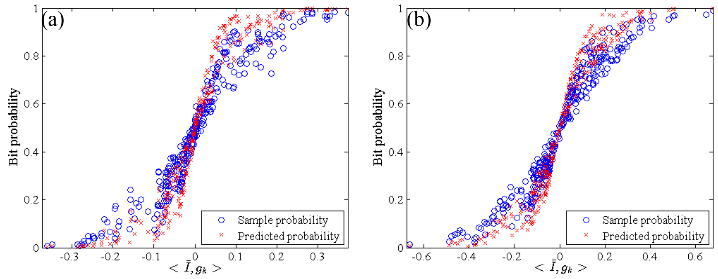


Fig. 6 The relationship between $< \bar{I}, g_k >$ and bit probabilities computed from the modified iris images. (a) The results from the UBIRIS.v1 database and (b) results from the WVU database. (Color figure)

2.3. Three Graphs Computed from IrisCodes, Masks and Gabor Filters

Section 2.1 briefly introduces three graphs that display different statistical and structural relationships between bits. This subsection presents the details of the computations for obtaining their adjacency matrixes, which are used in the next section to analyze application-specific IrisCodes. The statistical dependence graph is first discussed, and then, the Gabor graph is presented to explain the statistical dependence graph. Lastly, the mask correlation graph is discussed.

Given an IrisCode database B_s containing n independent IrisCodes from n irises and their mask database M_s, the Chi-square test is used to detect statistical dependence between bits. Because IrisCodes and their masks are binary, a bitwise implementation of the Chi-square statistic,

$$X^2(j,k) = \frac{\left(\sum_{i=1}^{n}(B_{sik} \cap B_{sij} \cap M_{sik} \cap M_{sij}) - n_{kj}\hat{p}_k\hat{p}_j\right)^2}{n_{kj}\hat{p}_k\hat{p}_j} + \frac{\left(\sum_{i=1}^{n}(\sim B_{sik} \cap B_{sij} \cap M_{sik} \cap M_{sij}) - n_{kj}\hat{q}_k\hat{p}_j\right)^2}{n_{kj}\hat{q}_k\hat{p}_j} + \frac{\left(\sum_{i=1}^{n}(B_{sik} \cap \sim B_{sij} \cap M_{sik} \cap M_{sij}) - n_{kj}\hat{p}_k\hat{q}_j\right)^2}{n_{kj}\hat{p}_k\hat{q}_j} + \frac{\left(\sum_{i=1}^{n}(\sim B_{sik} \cap \sim B_{sij} \cap M_{sik} \cap M_{sij}) - n_{kj}\hat{q}_k\hat{q}_j\right)^2}{n_{kj}\hat{q}_k\hat{q}_j}, \quad (12)$$

where B_{sij} and M_{sij} are the elements of B_s and M_s in the i^{th} row and the j^{th} column, respectively, $n_{kj} = \sum_{i=1}^{n} M_{sik} \cap M_{sij}$, $\hat{q}_k = (1-\hat{p}_k)$, $\hat{q}_j = (1-\hat{p}_j)$ and $\sim$ represents a bitwise NOT operator, can be derived. Note that $\cap$ is a bitwise AND operator. Note also that the analysis in this study does not depend on this bitwise implementation. A floating-point implementation can also be used. In Section 3, the notation $X^2(B_{s\bullet j}, B_{s\bullet k}) = X^2(j,k)$ is employed to emphasize the inputs. Although X_α^2 (defined as $\Pr(X^2(j,k) > X_\alpha^2 \mid B_{s\bullet j} \text{ and } B_{s\bullet k} \text{ are independent}) = \alpha$) is used as a decision threshold and α is set to a very small number, e.g., 0.01, the number of false detections is still very high because the number of pairwise dependencies to be tested is enormous. To be specific, 2,096,128 ((2048×2048-2048)/2) tests are required. In general, each iris has more than one IrisCode in a database, and therefore, a resampling algorithm is used to reduce the number of false detections. To extract more information from the dependent bits, their correlation coefficients, defined as

$$R(j,k) = \frac{\sum_{i=1}^{n} (B_{sij} - \hat{p}_j)(B_{sik} - \hat{p}_k) \times (M_{sij} \cap M_{sik})}{\sqrt{\sum_{i=1}^{n} (B_{sij} - \hat{p}_j)^2 \times (M_{sij} \cap M_{sik})} \sqrt{\sum_{i=1}^{n} (B_{sik} - \hat{p}_k)^2 \times (M_{sij} \cap M_{sik})}}, \tag{13}$$

are also computed. Only the sign of $R(j,k)$ is used to construct the statistical dependence graph, and therefore, the denominator is ignored. The numerator of $R(j,k)$, denoted by $R_u(j,k)$, can be simplified as

$$R_u(j,k) = \sum_{i=1}^{n} B_{sij} \cap B_{sik} \cap M_{sij} \cap M_{sik} - \hat{p}_j \sum_{i=1}^{n} B_{sik} \cap M_{sij} \cap M_{sik} - \hat{p}_k \sum_{i=1}^{n} B_{sij} \cap M_{sij} \cap M_{sik} + \hat{p}_j \hat{p}_k n_{jk}, \tag{14}$$

where $n_{kj} = \sum_{i=1}^{n} M_{sik} \cap M_{sij}$. In Section 3, $R_u(B_{s\bullet j}, B_{s\bullet k}) = R_u(j,k)$ is used to emphasize the inputs.

Now, let us present the proposed resampling algorithm for constructing the adjacency matrix Ψ of the statistical dependence graph. Given an IrisCode database B containing n IrisCodes from s irises, where $s \ll n$, and its mask database M , the number of iterations t, the minimum number of effective bits for the Chi-square statistic calculation m_e and the decision threshold for the Chi-square statistic X_α^2, the proposed resampling algorithm first randomly selects s independent IrisCodes and the corresponding masks from B and M to form B_s and M_s , respectively. Note that if s is very large, it is not necessary to select one IrisCode per iris. B_s and M_s are entered into Eq. 12 to compute all pairwise dependencies. If $X^2(j,k) > X_\alpha^2$, $R_u(j,k)$ is computed, and $\Psi(j,k)$ is updated through $\Psi(j,k) = \Psi(j,k) + sign(R_u(j,k))$. The process is repeated t times. The pseudo code of the resampling algorithm is given in Fig. 7. Fig. 8 shows the adjacency matrix of the statistical dependence graph of the UBIRIS.v1 database, where the gray pixels represent no statistical dependence; the white pixels represent statistical dependence with positive correlation and the black pixels represent statistical dependence with negative correlation. Fig. 9 shows the first 256 bits of three adjacency matrixes of the statistical dependence graphs. Figs. 9(a) and (b) are computed from the IrisCodes of the UBIRIS.v1 database. Fig. 9(a) is obtained from the standard Chi-square statistic and the correlation coefficient, while Fig. 9(b) is obtained from the proposed resampling algorithm with *t*=100. These figures show that the resampling algorithm effectively suppresses the noise from false detections. Figs. 8 and 9 illustrate that the statistical dependence in IrisCodes is very structural. If a suitable threshold is applied, they will become sparse matrixes. A large *t* produces more stable results if the number of images is enough for re-sampling. If *t* is too small, the adjacency matrixes will be very noisy such as Fig. 9(a), which is in fact computed from *t*=1. The preliminary experiments of this study indicate that when *t*=100, the adjacency matrixes are stable enough. The proposed resampling algorithm is also applied to the IrisCodes of the WVU database. Fig. 9(c) shows the first 256 bits of the corresponding adjacency matrix of the statistical dependence graph. Though the adjacency matrixes in Figs. 9(b) and (c) are computed from two different databases, they are very similar. Let Ψ_{UBIRIS} and Ψ_{WVU} be the two adjacency matrixes obtained respectively from the UBIRIS.v1 and WVU databases. 15,000 elements with the greatest $|\Psi_{\text{UBIRIS}}(j,k)|+|\Psi_{\text{WVU}}(j,k)|$ in Ψ_{UBIRIS} and Ψ_{WVU} are selected and each selected element pair is regarded as a point, i.e., $(\Psi_{\text{UBIRIS}}(j,k),\Psi_{\text{WVU}}(j,k))$ in a two dimensional space. Since Ψ_{UBIRIS} and Ψ_{WVU} are symmetric, the elements in the lower triangles of the matrixes are not selected. Fig. 10(a) is a plot of the points and Fig. 10(b) illustrates the locations of the selected elements. Fig. 10(a) shows clearly that even though the two adjacency matrixes are obtained from two different databases, their elements are highly correlated. The corresponding correlation coefficient is 0.99. These results imply that the dependence is generated from the same source.

To understand the source of this statistical dependence, the adjacency matrix G of the Gabor graph, whose edges are defined by $<g_j,g_k>$, is computed. The red pixels in Fig. 11(a) are the elements in $\Lambda_1=\{(j,k)\,|<g_j,g_k>\neq 0\}$, and their intensity represents $|<g_j,g_k>|$. The blue pixels in Fig. 11(a) are the elements in $\Lambda_2=\left\{(j,k)\mid \sum_{i=1,i\notin\{k,j\}}^{2048}|<g_j,g_i>||<g_i,g_k>|\neq 0 \wedge <g_j,g_k>=0\right\}$, where $\wedge$ is an AND operator, and their intensity represents $\sum_{i=1,i\notin\{k,j\}}^{2048}|<g_j,g_i>||<g_i,g_k>|$. Though Gabor filters have infinite support in the continuous domain, in discrete implementation, their supports are always finite. Thus, many $<g_j,g_k>$ are zero. The elements in Λ_1 indicate the nodes that are directly connected in the Gabor graph. The elements in Λ_2 are not connected in the Gabor graph, but they are connected through another node. For example, (j,k) are not connected, but both (j,i) and (i,k) are connected. Comparing the color pixels in Fig. 11(a) to the structural pattern in Fig. 8, two points can be concluded: 1) $< g_j, g_k >$ directly induces partial statistical dependence, and 2) this statistical dependence propagates through the Gabor graph and produces other statistical dependence, as shown in Fig. 8. The first point is much clearer when comparing Figs. 9(b)-(c) and Fig. 11(b). Fig. 11(b) shows the connections of the first 256 bit locations in the Gabor graph G. The grey pixels in Fig. 11(b) indicate $< g_j, g_k >= 0$; the white pixels indicate $< g_j, g_k >> 0$ and the black pixels indicate $< g_j, g_k >< 0$.

The last graph introduced in this section is the mask correlation graph, whose edges are defined by the correlation coefficient between mask bits in two locations. To be more precise, given a mask database M, the correlation coefficient between bit locations k and j is computed from

$$\Phi(j,k) = \frac{\sum_{i=1}^{n} (M_{ij} - \hat{p}_{mj})(M_{ik} - \hat{p}_{mk})}{\sqrt{\sum_{i=1}^{n} (M_{ij} - \hat{p}_{mj})^2} \sqrt{\sum_{i=1}^{n} (M_{ik} - \hat{p}_{mk})^2}}, \qquad (15)$$

where $\hat{p}_{mj} = \sum_{i=1}^{n} M_{ij} / n$ and $\hat{p}_{mk} = \sum_{i=1}^{n} M_{ik} / n$. The adjacency matrix of the mask correlation graph is denoted as Φ. Fig. 12 shows the adjacency matrixes of the mask correlation graphs computed from the UBIRIS.v1 database and the WVU database.

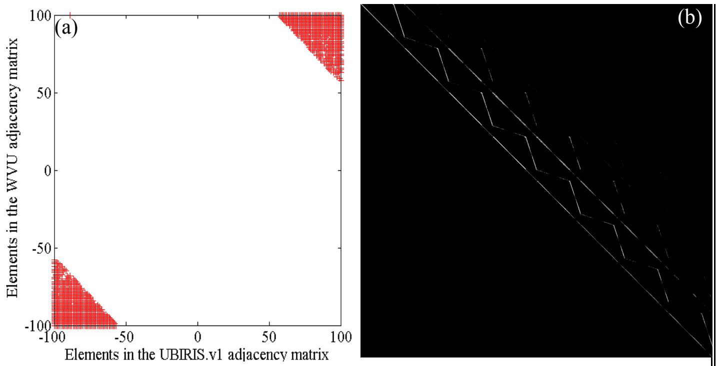


Fig. 10 The correlation between the elements of Ψ_{UBIRIS} and Ψ_{WVU}. (a) a plot of the selected elements of Ψ_{WVU} against the corresponding elements of Ψ_{UBIRIS} and (b) the locations of the selected elements to compute (a). The rows and columns of (b) represent bit locations and the size of (b) is 2048 by 2048. (Color figure. To see clearly the images, please read the electronic version.)

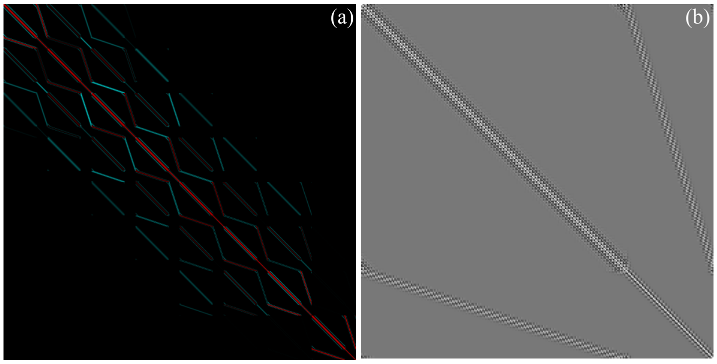


Fig. 11 (a) A matrix displaying the first-order connections and the second-order connections in the Gabor graph. The red pixels indicate the elements in Λ_1, and the blue pixels indicate the elements in Λ_2. (b) The first 256 bits of the adjacency matrix of the Gabor graph, whose elements are $< g_j, g_k >$. The rows and columns of these matrixes represent bit locations. The size of (a) is 2048 by 2048 and the size of (b) is 256 by 256. (Color figure. To see clearly the images, please read the electronic version.)

```
Input: An IrisCode database $B$ and its mask database $M$ from $n$ iris images and $s$ different irises, where $s \ll n$.
The minimum number of effective bits $m_e$ required to calculate the value of the Chi-square statistic.
The decision threshold of the Chi-square statistic $X_\alpha^2$ and the number of iterations $t$.
Output: The adjacency matrix $\Psi$ of the statistical dependence graph.
Algorithm:
    1. Set $\Psi = 0$, where $0$ is a zero matrix with a size of 2,048 by 2,048.
    2. Randomly select $s$ independent IrisCodes and their masks to form $B_s$ and $M_s$.
    3. Calculate the value of the Chi-square statistic, $\forall j \le k$ if $n_{kj} = \sum_{i=1}^{n} M_{sik} \cap M_{sij} > m_e$.
    4. If $n_{kj} > m_e$ and $X^2(j,k) > X_\alpha^2$, calculate $R_u(j,k)$ and update $\Psi$ through $\Psi(j,k) = \Psi(j,k) + sign(R_u(j,k))$.
    5. Repeat steps 2–4 $t$ times.
    6. Set $\Psi(k,j) = \Psi(j,k)$, $\forall j < k$.
```

Fig. 7 The pseudo code of the resampling algorithm.

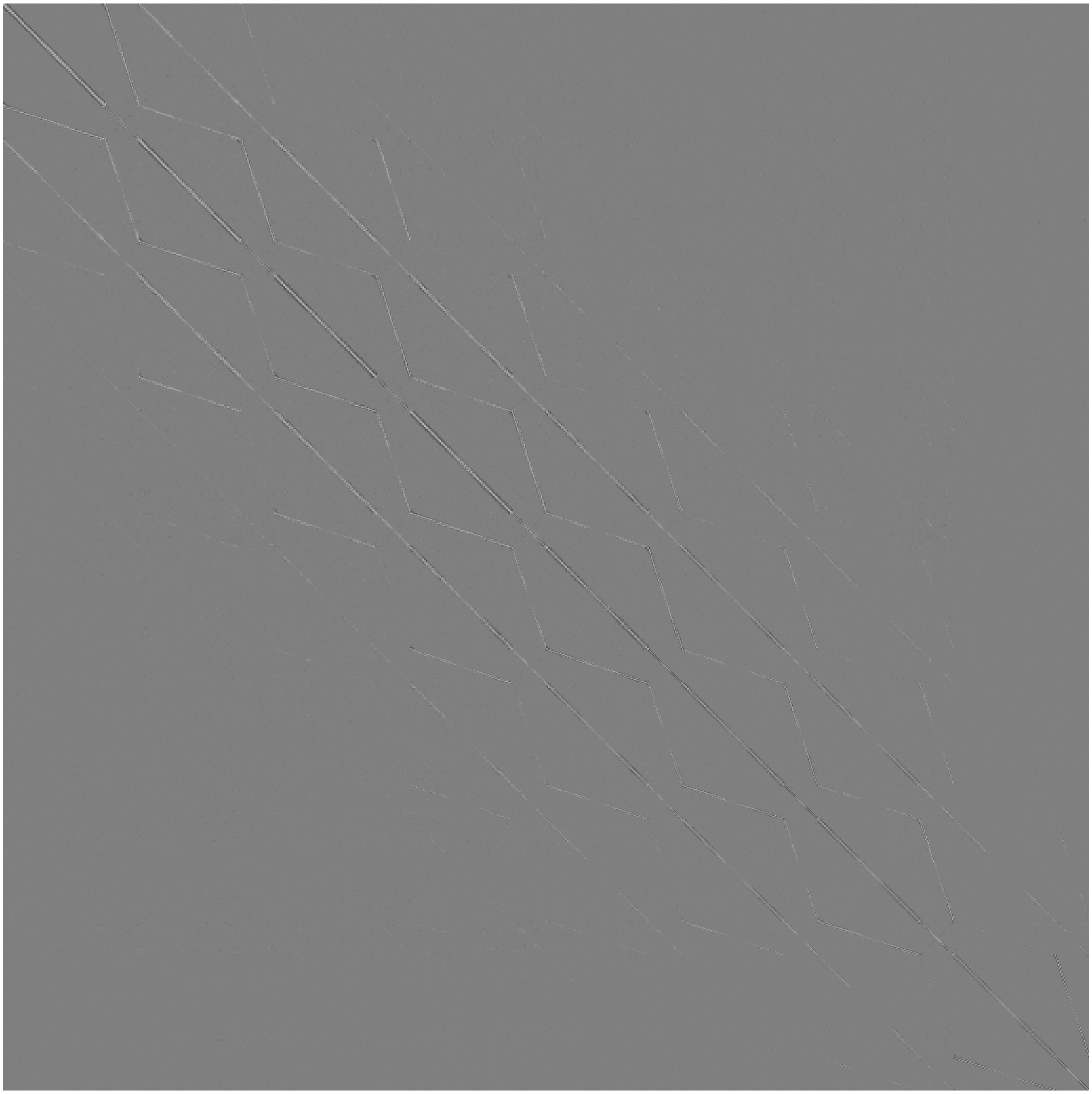


Fig. 8 The adjacency matrix of the statistical dependence graph of the UBIRIS.v1 database. The matrix is obtained from the proposed resampling algorithm (Fig. 7). Its rows and columns represent bit locations and its size is 2048 by 2048.

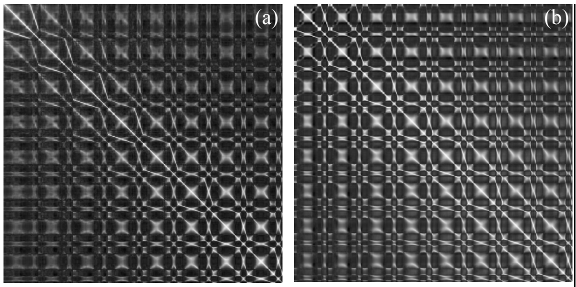


Fig. 12 The adjacency matrixes of the mask correlation graphs of (a) the UBIRIS.v1 database and (b) the WVU database. The elements of these matrixes are computed from Eq. 15. The rows and columns in these matrixes represent bit locations. Both matrix sizes are 2048 by 2048.

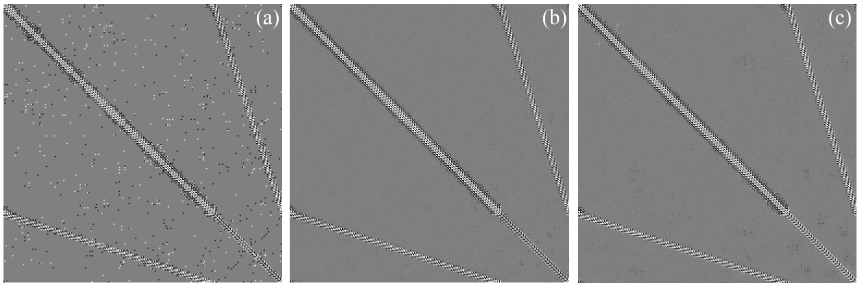


Fig. 9 A comparison of the first 256 bits of the adjacency matrixes of the statistical dependence graphs obtained from (a) the standard Chi-square statistic and correlation coefficient and (b)-(c) the proposed resampling algorithm (Fig. 7) on the UBIRIS.v1 and WVU databases, respectively. The rows and columns of these matrixes represent bit locations. All matrix sizes are 256 by 256. (To see clearly the images, please read the electronic version.)

3. Two Graph-based Algorithms

In this section, two graph-based algorithms called the X-algorithm and the P-algorithm are presented to analyze application-specific IrisCodes produced by the XOR and permutation protection schemes in Eqs. 5 and 6. The two algorithms are designed to study the case in which the same key is used to lock all IrisCodes in a database for large-scale real-time identification. Both algorithms assume that attackers have an application-specific IrisCode database, its mask database, an unrelated iris image database and all the filters, $\{g_1,\cdots,g_{2048}\}$, but do not have the key that locks the database, meaning that S_u in Eq. 5 and P_u in Eq. 6 are unknown to them. The first two assumptions are valid. If it is guaranteed that attackers obtain neither the application-specific IrisCodes nor their masks, there is no point in applying template protection schemes. If attackers are insiders, they may be able to obtain the application-specific IrisCodes and their masks directly. If iris recognition systems are connected to the Internet, attackers may be able to obtain the application-specific IrisCodes and their masks through networks. The unrelated iris images can be obtained from public iris image databases or collected from attackers' iris imaging systems. The two algorithms do not assume that the application-specific IrisCodes and the unrelated iris images are from the same eye. The last assumption concerning how attackers obtain $\{g_1,\cdots,g_{2048}\}$ is discussed in the last section. In fact, some attacks do not depend on $\{g_1,\cdots,g_{2048}\}$, which is also discussed in the last section. The two graph-based algorithms presented in this section depend heavily on the graphs and the statistical information given in the previous section. Section 3.1 presents the X-algorithm for analyzing the XOR protection scheme, and Section 3.2 presents the P-algorithm for analyzing the permutation protection scheme.

3.1. The X-Algorithm for Analyzing the XOR Protection Scheme

The X-algorithm for analyzing the XOR protection scheme requires the adjacency matrixes of two statistical dependence graphs as inputs. One is computed from unprotected IrisCodes and their masks of unrelated iris images, and the other is computed from application-specific IrisCodes and their masks, produced by the XOR protection scheme. The two adjacency matrixes are denoted by Ψ_U and $\Psi_\otimes$, respectively. Using the unrelated iris images, $\{g_1,\cdots,g_{2048}\}$ and the resampling algorithm given in Fig. 7, Ψ_U is obtained. Applying the resampling algorithm to the application-specific IrisCodes and their masks, $\Psi_\otimes$ is also obtained. The proposed X-algorithm uses Ψ_U and $\Psi_\otimes$ to estimate the random bit vector S_u that locks the application-specific IrisCodes.

Two mathematical properties of the Chi-square statistic and the correlation coefficient are derived to design the X-algorithm. Given two columns $B_{s\bullet j}$ and $B_{s\bullet k}$ in B_s and the corresponding columns $M_{s\bullet j}$ and $M_{s\bullet k}$ in M_s, it can be proven that the Chi-square statistic is invariant to the XOR operation, meaning that

$$X^2(B_{s\bullet j},B_{s\bullet k})=X^2(S_u(j)\otimes B_{s\bullet j},S_u(k)\otimes B_{s\bullet k}), \tag{16}$$

$\forall\, S_u(j)$ and $S_u(k)\in\{0,1\}$, where $S_u(j)$ and $S_u(k)$ are the j^{th} and k^{th} elements in S_u. $S_u(j)\otimes B_{s\bullet j}$ and $S_u(k)\otimes B_{s\bullet k}$ represent two column vectors whose elements are $S_u(j)\otimes B_{sij}$ and $S_u(k)\otimes B_{sik},\forall i$, respectively. When both $S_u(j)=0$ and $S_u(k)=0$, $S_u(j)\otimes B_{s\bullet j}=B_{s\bullet j}$ and $S_u(k)\otimes B_{s\bullet k}=B_{s\bullet k}$. Clearly, $X^2(S_u(j)\otimes B_{s\bullet j},S_u(k)\otimes B_{s\bullet k})=X^2(B_{s\bullet j},B_{s\bullet k})$. When both $S_u(j)=1$ and $S_u(k)=1$, $S_u(j)\otimes B_{s\bullet j}=\sim B_{s\bullet j}$ and $S_u(k)\otimes B_{s\bullet k}=\sim B_{s\bullet k}$. Substituting $\sim B_{s\bullet j}$ and $\sim B_{s\bullet k}$ into Eq. 12, $X^2(\sim B_{s\bullet j},\sim B_{s\bullet k})=X^2(B_{s\bullet j},B_{s\bullet k})$ is obtained. When $S_u(j)=1$ and $S_u(k)=0$, $S_u(j)\otimes B_{s\bullet j}=\sim B_{s\bullet j}$ and $S_u(k)\otimes B_{s\bullet k}=B_{s\bullet k}$. Substituting $\sim B_{s\bullet j}$ and $B_{s\bullet k}$ into Eq. 12, $X^2(\sim B_{s\bullet j},B_{s\bullet k})=X^2(B_{s\bullet j},B_{s\bullet k})$ is also obtained. Similarly, when $S_u(j)=0$ and $S_u(k)=1$, $X^2(B_{s\bullet j},\sim B_{s\bullet k})=X^2(B_{s\bullet j},B_{s\bullet k})$. Thus, $\forall\, S_u(j)$ and $S_u(k)\in\{0,1\}$, $X^2(S_u(j)\otimes B_{s\bullet j},S_u(k)\otimes B_{s\bullet k})=X^2(B_{s\bullet j},B_{s\bullet k})$.

The other mathematical property is that

$$R_u(S_u(j)\otimes B_{s\bullet j},S_u(k)\otimes B_{s\bullet k})=H(S_u(j)\otimes S_u(k))\times R_u(B_{s\bullet j},B_{s\bullet k}), \tag{17}$$

where $H(0)=1$ and $H(1)=-1$. Note that the subscript *u* in R_u and the subscript *u* in S_u have different meanings. Eq. 17 means that if $S_u(j)=S_u(k)$, $R_u(S_u(j)\otimes B_{s\bullet j},S_u(k)\otimes B_{s\bullet k})=R_u(B_{s\bullet j},B_{s\bullet k})$; otherwise, $R_u(S_u(j)\otimes B_{s\bullet j},S_u(k)\otimes B_{s\bullet k})=-R_u(B_{s\bullet j},B_{s\bullet k})$. Without loss of generality, $M_{s\bullet j}$ and $M_{s\bullet k}$ are regarded as vectors with all ones in the proof. Thus,

$$R_u(B_{s\bullet j},B_{s\bullet k})=\sum_{i=1}^{n}(B_{sij}-\frac{1}{n}\sum_{w=1}^{n}B_{swj})(B_{sik}-\frac{1}{n}\sum_{w=1}^{n}B_{swk}), \tag{18}$$

which is the numerator of Eq. 13. Eq. 17 is clearly valid when both $S_u(j)=0$ and $S_u(k)=0$ because $S_u(j)\otimes B_{s\bullet j}=B_{s\bullet j}$ and $S_u(k)\otimes B_{s\bullet k}=B_{s\bullet k}$. When $S_u(j)=1$ and $S_u(k)=1$, $S_u(j)\otimes B_{s\bullet j}=1_v-B_{s\bullet j}$ and $S_u(k)\otimes B_{s\bullet k}=1_v-B_{s\bullet k}$, where $1_v=[1,\cdots,1]^T$. Substituting these terms into Eq. 18 and simplifying, $R_u(1_v-B_{s\bullet j},1_v-B_{s\bullet k})=R_u(B_{s\bullet j},B_{s\bullet k})$. When $S_u(j)=1$ and $S_u(k)=0$, $S_u(j)\otimes B_{s\bullet j}=1_v-B_{s\bullet j}$ and $S_u(k)\otimes B_{s\bullet k}=B_{s\bullet k}$. $R_u(1_v-B_{s\bullet j},B_{s\bullet k})=-R_u(B_{s\bullet j},B_{s\bullet k})$ is obtained. Similarly, when $S_u(j)=0$ and $S_u(k)=1$, $R_u(B_{s\bullet j},1_v-B_{s\bullet k})=-R_u(B_{s\bullet j},B_{s\bullet k})$ is obtained. Combining these four cases, Eq. 17 is proven.

Let $\Psi_\otimes$ be the adjacency matrix of the statistical dependence graph computed from an application-specific IrisCode database $B_\otimes$ whose j^{th} row is $B_{\otimes j\bullet}=S_u\otimes B_{j\bullet}$ and the corresponding mask database $M_\otimes$. Note that $M_\otimes=M$. Using the mathematical properties given in Eqs. 16–17 and the resampling algorithm, the relationship between $\Psi_\otimes$ and Ψ is described by

$$\Psi(j,k)=H(S_u(j)\otimes S_u(k))\Psi_\otimes(j,k). \tag{19}$$

Note that Ψ is the adjacency matrix of the statistical dependence graph computed from the original IrisCode database B and the corresponding mask database M, which produce $B_\otimes$ and $M_\otimes$. Let the adjacency matrix computed from the IrisCodes of unrelated iris images be Ψ_U. Based on Eq. 19, a minimization defined as

$$\hat{S}_u=\arg\min_{V\in Z_2^{2048}}\sum_{j=1}^{2048}\sum_{k=1}^{2048}d(\Psi_U(j,k)-H(V(j)\otimes V(k))\Psi_\otimes(j,k)), \tag{20}$$

where d is a distance function and Z_2^{2048} is a set containing all binary vectors with a length of 2,048, can be used to estimate S_u.

The X-algorithm is a greedy algorithm used to perform this minimization. Because of the enormous search space, the global optimum is not expected to be found. The algorithm first assumes that one of the bits in S_u is known. Here, $S_u(1)$ is assumed to be zero, meaning that the first bits in the application-specific IrisCodes are not changed. Thus, $\hat{S}_u(1)$ is set to zero. The X-algorithm processes bit by bit. In each iteration, one bit location is selected based on the optimization

$$k_o=\arg\max_{k\in\backslash\Gamma}\sum_{j\in\Gamma}\left|\Psi_\otimes(j,k)\right|+\left|\Psi_U(j,k)\right|, \tag{21}$$

where Γ is the set of all bit locations that have been processed and \ is a complement operator. Eq. 21 selects the bit location whose statistical dependence on the processed bit locations is the highest. The minimization, defined as

$$\hat{S}_u(k_o) = \arg\min_{z\in\{0,1\}} \sum_{j\in\Gamma} \left(\left|\Psi_{\otimes}(j,k_o)\right| + \left|\Psi_U(j,k_o)\right|\right) \times \left|\Psi_U(j,k_o) - H(z\otimes \hat{S}_u(j))\Psi_{\otimes}(j,k_o)\right|, \tag{22}$$

is used to determine $\hat{S}_u(k_o)$. This minimization is based on using the processed bits in $\hat{S}_u(k)$ and their locations in Γ to determine $\hat{S}_u(k_o)$. $\left(\left|\Psi_{\otimes}(j,k_o)\right| + \left|\Psi_U(j,k_o)\right|\right) \times \left|\Psi_U(j,k_o) - H(z\otimes \hat{S}_u(j))\Psi_{\otimes}(j,k_o)\right|$ in Eq. 22 is a realization of the distance function *d* in Eq. 20. Direct optimization can be performed because the size of the search space in Eq. 22 is only two numbers. The proposed algorithm iteratively uses Eqs. 21 and 22 to estimate all bits in S_u. Initially, $S_u(1)=0$ is assumed. This assumption may be incorrect, meaning that the actual $S_u(1)$ may be one. According to Eqs. 19, 21 and 22, if this assumption is incorrect, the true estimate should be $\sim \hat{S}_u$. Using these two estimates to unlock the j^{th} application-specific IrisCode in the database, $\hat{S}_u \otimes B_{\otimes j\bullet}$ and $\sim \hat{S}_u \otimes B_{\otimes j\bullet}$ are obtained. Up to this point, neither estimate is known to be better than the other. To make the final decision, the sample bit probabilities $\hat{p}_k$ of B_U are calculated. A naïve Bayes classifier is constructed to calculate

$$\Pr(\hat{S}_u \otimes B_{\otimes j\bullet}) \approx \prod_{k\in\Theta}((\hat{S}_u(k)\otimes B_{\otimes jk})\hat{p}_k + (1-\hat{S}_u(k)\otimes B_{\otimes jk})\hat{q}_k), \tag{23}$$

where Θ is the set of *m* bit locations with the highest $\sum_i M_{Uik}$. That is, Θ is the set of the *m* clearest bit locations. Note that only the *m* clearest bit locations are used to estimate $\Pr(\hat{S}_u \otimes B_{\otimes j\bullet})$. Then,

$$\varsigma = \sum_{j=1}^{n_{\otimes}} \xi(\Pr(\hat{S}_u \otimes B_{\otimes j\bullet}) > \Pr(\sim \hat{S}_u \otimes B_{\otimes j\bullet})), \tag{24}$$

where $\xi = 1$ if the condition is true; otherwise, $\xi = 0$. If $\varsigma > n_{\otimes}/2$, where $n_{\otimes}$ is the number of application-specific IrisCodes in $B_{\otimes}$, $\hat{S}_u$ is selected to approximate S_u; otherwise, $\sim \hat{S}_u$ is selected. Fig. 13 gives the pseudo code of the X-algorithm.

3.2. The P-Algorithm for Analyzing the Permutation Protection Scheme

The P-algorithm for analyzing the permutation protection scheme requires the adjacency matrixes of the statistical dependence graph and the mask correlation graph, computed from unrelated iris images, and the adjacency matrixes of the statistical dependence graph and the mask correlation graph, computed from application-specific IrisCodes and the corresponding masks. These adjacency matrixes are denoted by Ψ_U, Φ_U, $\Psi_{\times}$ and $\Phi_{\times}$, respectively. Because $B_{\times j\bullet} = B_{j\bullet} \times P_u$ and $M_{\times j\bullet} = M_{j\bullet} \times P_u$, $\Psi_{\times} = P_u^T \Psi P_u$ and $\Phi_{\times} = P_u^T \Phi P_u$, where Ψ and Φ are computed from the unprotected IrisCode database *B* and the corresponding mask database *M*. Using these equations, an optimization

$$\hat{P}_u = \arg\min_{P\in\Xi} \left\| \Psi_\times - P^T \Psi_U P \right\|, \tag{25}$$

where Ξ is the set of all permutation matrixes of size 2,048 by 2,048 and $\|\bullet\|$ is a matrix norm, e.g., the Frobenius norm, is formulated to estimate P_u. If $\Psi = \Psi_U$, Eq. 25 is a graph isomorphism problem, whose complexity is NP. In general, $\Psi \neq \Psi_U$ because of statistical fluctuation and noise, and therefore, Eq. 25 becomes a graph matching problem. In preliminary attempts, a number of existing graph matching methods were tested [28-30], but no positive result was obtained. The huge adjacency matrixes, 2,048 by 2,048 in size, are the major challenge in estimating P_u. The differences between Ψ_U and Ψ make the problem more complicated.

The proposed P-algorithm has three components: the starting point generation, the core of the P-algorithm and the hierarchical search. The starting point generation and the hierarchical search are relatively simple, as discussed later. The core of the P-algorithm is iterative use of selection, optimization and update steps to estimate the permutation matrix P_u. Note that a permutation matrix can be regarded as a function $\wp(i) = j$ if $P(i,j) = 1$, and the function $\wp$ can be regarded as a vector $[\wp(1),\cdots,\wp(2048)]$. Once the function $\wp$ is known, the corresponding permutation matrix is obtained. The core of the P-algorithm takes two sets of bit location correspondences, i.e., $\hat{\wp}(i_1) = j_1$ and $\hat{\wp}(i_2) = j_2$, where the notation ^ signifies that $\hat{\wp}$ is an estimate of $\wp$ and $i_1, i_2, j_1, j_2 \in \{1,\cdots,2048\}$, as a starting point. Note that $\wp$ and $\hat{\wp}$ may not be the same. Here, it is assumed that these two sets of bit location correspondences are correct. The starting point generation component discussed later is designed to produce correct correspondences.

Let Γ and $\Gamma_\times$ be the sets of the processed bit locations in Ψ_U and $\Psi_\times$, respectively. Initially, $\Gamma = \{i_1, i_2\}$ and $\Gamma_\times = \{j_1, j_2\}$. The core of the P-algorithm first selects a bit location from $\backslash\Gamma$ based on Γ and Ψ_U. Note that \ is a complement operator, as mentioned previously. The core of the algorithm then uses an optimization with inputs Ψ_U, Φ_U, $\Psi_\times$ and $\Phi_\times$ to determine the corresponding bit location in $\Psi_\times$ of the selected bit location in Ψ_U. Lastly, $\hat{\wp}$, Γ and $\Gamma_\times$ are updated. For each given $j \in \backslash\Gamma$, the core of the P-algorithm constructs a set Υ_j of *c* bit locations in Γ that have the greatest statistical influence on the bit location *j*. Mathematically, $\Upsilon_j = \{R_\Gamma(j,1),\cdots,R_\Gamma(j,c)\}$, where R_Γ is an indexing function defined as $|\Psi_U(j,R_\Gamma(j,1))| \ge |\Psi_U(j,R_\Gamma(j,2))| \cdots \ge |\Psi_U(j,R_\Gamma(j,c))|$ and $R_\Gamma(j,i) \in \Gamma$, $\forall i \in \{1,\cdots,c\}$. The cardinality of Υ_j is controlled by the parameter $c = \min(t_1, |\Gamma|)$, where t_1 is a threshold and $|\Gamma|$ is the cardinality of Γ. The parameter t_1 is used to remove unreliable bit locations whose statistical dependence on the bit location *j* is small. In the experiments, t_1 is set to ten. In the first several iterations, $|\Gamma| < t_1$, and therefore, c is defined as $\min(t_1, |\Gamma|)$. Note that each $j \in \backslash\Gamma$ has different Υ_j, but their cardinalities are the same. To select a bit location $h \in \backslash\Gamma$ for processing, a maximin scheme,

$$h = \arg_j \max_{j\in\backslash\Gamma} \min_{\substack{k\in\backslash\Gamma \\ k\neq j}} \sum_{i\in\Upsilon_j} \left|\Psi_U(j,i)\right|\left|\Psi_U(j,i) - \Psi_U(k,i)\right|, \tag{26}$$

is used. Note that $k \notin \Upsilon_j$ because $k \notin \Gamma$. The term $\sum_{i\in\Upsilon_j} \left|\Psi_U(j,i)\right|\left|\Psi_U(j,i) - \Psi_U(k,i)\right|$ is the sum of the weighted absolute differences between $\Psi_U(j,i)$ and $\Psi_U(k,i)$ at the bit locations in Υ_j. This term is a measure of the dissimilarity between the two vectors $[\Psi_U(j, R_\Gamma(j,1)) \cdots \Psi_U(j, R_\Gamma(j,c))]$ and $[\Psi_U(k, R_\Gamma(j,1)) \cdots \Psi_U(k, R_\Gamma(j,c))]$. Given a fixed j, $\min_{\substack{k\in\backslash\Gamma \\ k\neq j}} \sum_{i\in\Upsilon_j} \left|\Psi_U(j,i)\right|\left|\Psi_U(j,i) - \Psi_U(k,i)\right|$ yields a vector $[\Psi_U(k, R_\Gamma(j,1)) \cdots \Psi_U(k, R_\Gamma(j,c))]$ whose difference from $[\Psi_U(j, R_\Gamma(j,1)) \cdots \Psi_U(j, R_\Gamma(j,c))]$ is a minimum. The core of the P-algorithm identifies a bit location $h \in \backslash\Gamma$ that yields the greatest minimum difference. This maximin scheme avoids selection of a bit location whose $[\Psi_U(j, R_\Gamma(j,1)) \cdots \Psi_U(j, R_\Gamma(j,c))]$ is close to another $[\Psi_U(k, R_\Gamma(j,1)) \cdots \Psi_U(k, R_\Gamma(j,c))]$, to prevent a decision error in the next step. All the operations in this maximin scheme are on the set $\Upsilon_j \subset \Gamma$, which guarantees that the selected bit location h and Γ have strong statistical dependence. Once the bit location h is chosen, the corresponding $\hat{\wp}(h)$ is determined by

$$\hat{\wp}(h) = \arg\min_{k\in/\Gamma_\times} \sum_{i\in\Upsilon_h} \left|\Psi_U(h,i) - \Psi_\times(k, \hat{P}(i))\right| + \iota\left|\Phi_U(h,i) - \Phi_\times(k, \hat{P}(i))\right|, \tag{27}$$

where ι is a parameter balancing the two terms. The upper limit of the elements in Ψ_U and $\Psi_\times$ is t, which is the number of iterations in the resampling algorithm given in Fig. 7, while the upper limit of the elements in Φ_U and $\Phi_\times$ is one. In the experiments, ι is set to t when Φ_U, and $\Phi_\times$ are obtained from the same database. Lastly, h is inserted into Γ, $\hat{\wp}(h)$ is inserted into $\Gamma_\times$, and $\hat{\wp}$ is updated. The core of the P-algorithm is summarized in Fig. 14.

Ψ_U and $\Psi_\times$ are very similar even when they are computed from databases with different characteristics (Fig. 9). However, Φ_U and $\Phi_\times$ can be different because they are affected mainly by the percentage of eyelids and eyelashes covering irises (Fig. 12). To reduce this difference, $\tilde{\Phi}_\times$ defined as $\tilde{\Phi}_\times(\Theta_{\times x}(j), \Theta_{\times y}(j)) = \Phi_U(\Theta_{Ux}(j), \Theta_{Uy}(j))$, where $\Theta_{\times x}$ and $\Theta_{\times y}$ are indexing functions such that $\Phi_\times(\Theta_{\times x}(1), \Theta_{\times y}(1)) \geq \cdots \geq \Phi_\times(\Theta_{\times x}(2048^2), \Theta_{\times y}(2048^2))$ and Θ_{Ux} and Θ_{Uy} are also indexing functions such that $\Phi_U(\Theta_{Ux}(1),\Theta_{Uy}(1)) \geq \cdots \geq \Phi_U(\Theta_{Ux}(2048^2),\Theta_{Uy}(2048^2))$, is used to replace $\Phi_\times$ in Eq. 27. In the experiments, $\tilde{\Phi}_\times$ is used when Ψ_U and $\Psi_\times$ are computed from different databases, meaning that one is UBIRIS.v1 database and the other is the WVU database. In addition, a small value of ι is employed.

The core of the P-algorithm relies on two sets of bit location correspondences, i.e., $\hat{\wp}(i_1) = j_1$ and $\hat{\wp}(i_2) = j_2$. If all possible correspondences are tested, the core of the algorithm must be run more than 8.78×10^{12} times (2,048!/(2!×(2,048-2)!)×2,048×2,047). To reduce the computation burden, the starting point generation component and the hierarchical search component are proposed. Instead of computing all possible cases, a pair of bit locations (i_{s1}, i_{s2}) with strong statistical dependence in Ψ_U is preselected. Thus, the number of bit location correspondences is reduced to 4,192,256 (2,048×2,047). To further reduce this number, features are extracted from all possible bit location pairs (k_1, k_2) in $\Psi_\times$ that potentially correspond to (i_{s1}, i_{s2}). The features include mask probabilities, bit probabilities, the rank of $\Psi_\times(k_1,k_2)$ in the list $\Psi_\times(k_1,1),\cdots,\Psi_\times(k_1,2048)$, $\sum_{i\in\mho_z}(|\Psi_\times(k_1,i)| + |\Psi_\times(i,k_2)|)/|\mho_z|$, where $z \in \{1,2,3\}$, $\mho_1 = \{i \mid i \neq k_1 \wedge i \neq k_2 \wedge \Psi_\times(k_1,i) > 0 \wedge \Psi_\times(i,k_2) > 0\}$, $\mho_2 = \{i \mid i \neq k_1 \wedge i \neq k_2 \wedge \Psi_\times(k_1,i) < 0 \wedge \Psi_\times(i,k_2) < 0\}$ and $\mho_3 = \{i \mid i \neq k_1 \wedge i \neq k_2 \wedge \Psi_\times(k_1,i) \times \Psi_\times(i,k_2) < 0\}$. If the features computed from (k_1, k_2) are within a predefined range, (k_1, k_2) will be considered a pair correspondence to (i_{s1}, i_{s2}). In the experiments, these features were found to reduce the average number of potential corresponding pairs to 176 for the UBIRIS.v1 database and 133 for the WVU database.

To increase the speed of the algorithm further, a hierarchical search is used. All selected corresponding pairs are entered into the core of the P-algorithm, but only l bit locations are processed. The best ϖ corresponding pairs, based on the criterion

$$s = \frac{\sum_{c_1\in\Gamma}\sum_{c_2\in\Gamma}\left|\Psi_U(c_1,c_2) - \Psi_\times(\hat{\wp}(c_1),\hat{\wp}(c_2))\right|}{\sum_{c_1\in\Gamma}\sum_{c_2\in\Gamma}\left|\Psi_U(c_1,c_2) + \Psi_\times(\hat{\wp}(c_1),\hat{\wp}(c_2))\right|}, \tag{28}$$

are chosen to process the rest of the bit locations. The core of the P-algorithm returns one permutation matrix for each corresponding pair. That is, ϖ permutation matrixes are produced. In the experiments, $l = 50$ and $\varpi = 15$. In general, attackers have more than one chance. Eq. 28 is used again to select three permutation matrixes from the ϖ permutation matrixes to finally unlock the application-specific IrisCodes in $B_\times$. The experimental results reported in Section 4 are calculated from the best of these three permutation matrixes, in terms of error bits and hamming distances.

```
Input: An IrisCode database $B_U$ and its mask database $M_U$ from unrelated iris images. An application-specific IrisCode database $B_\otimes$ protected by the XOR protection scheme and the corresponding mask database $M_\otimes$. The parameter $m$ that controls the number of effective bit locations used in the naïve Bayes classifier.
Output: The estimated bit vector $\hat{S}_u$.
Algorithm:
    1. Use the re-sampling algorithm in Fig. 7 to calculate $\Psi_U$ of $B_U$ and $\Psi_\otimes$ of $B_\otimes$.
    2. Calculate $\hat{p}_k$ and $\hat{q}_k$ from $B_U$ and select the $m$ clearest bit locations to form $\Theta$, according to $\sum_i M_{Uik}$.
    3. Set $\hat{S}_u(1) = 0$ and $\Gamma = \{1\}$.
    4. Use Eq. 21 to select a bit location $k_o$.
    5. Use Eq. 22 to obtain $\hat{S}_u(k_o)$ and insert $k_o$ into $\Gamma$.
    6. Repeat steps 5 and 6 until all bit locations are processed.
    7. Use $\hat{S}_u$ and $\sim \hat{S}_u$ to unlock $B_\otimes$.
    8. Use Eqs. 23–24 to compute $\varsigma$.
    9. Return $\hat{S}_u$ and $\hat{S}_u \otimes B_{\otimes j\bullet}, \forall j$ if $\varsigma > n_\otimes / 2$; otherwise, return $\sim \hat{S}_u$ and $\sim \hat{S}_u \otimes B_{\otimes j\bullet}, \forall j$.
```

Fig. 13 The pseudo code of the X-algorithm.

```
Input: The adjacency matrixes ($\Psi_U$ and $\Phi_U$) of the statistical dependence graph and the mask correlation graph from unrelated iris images.
The adjacency matrixes ($\Psi_\times$ and $\Phi_\times$) of the statistical dependence graph and the mask correlation graph from an application-specific IrisCode database $B_\times$ and $M_\times$.
A pair of bit location correspondences, i.e., $\hat{\wp}(i_1) = j_1$ and $\hat{\wp}(i_2) = j_2$.
Output: The estimated permutation matrix that locks $B_\times$ and $M_\times$ and the unlocked $B_\times$ and $M_\times$.
Algorithm:
    1. Set $\Gamma = \{i_1, i_2\}$ and $\Gamma_\times = \{j_1, j_2\}$.
    2. Compute $\Upsilon_j = \{R_\Gamma(j,1), \cdots R_\Gamma(j,c)\}$, $\forall j \in \backslash\Gamma$.
    3. Use Eq. 26 to select a bit location $h \in \backslash\Gamma$.
    4. Use Eq. 27 to compute correspondence bit location $\hat{\wp}(h)$.
    5. Update $\Gamma$, $\Gamma_\times$ and $\hat{\wp}$.
    6. Repeat steps 2–5 until $|\Gamma|$ is 2048.
    7. Use $\hat{\wp}$ to construct the corresponding permutation matrix and use it to unlock $B_\times$ and $M_\times$.
```

Fig. 14 The pseudo code of the core of the P-algorithm.

4. Experimental Results

The X-algorithm and the P-algorithm were applied to the two public iris image databases, the West Virginia University (WVU) iris database and the UBIRIS.v1 database, to evaluate the security risks in the XOR and permutation protection schemes. The UBIRIS.v1 database was collected in a visible light environment and the WVU database was collected in an NIR environment. For each algorithm and database, one hundred tests were performed, and in each test, the database was randomly divided into a training set and a testing set. The training set contained half of the irises, and the testing set contained the other half. These two datasets were disjoint, meaning that no iris had images in both datasets. To study the performance of the algorithms on training and testing sets with different characteristics, the UBIRIS.v1 database was used as a training set and the WVU database was used as a testing set, and vice versa. In each of these experiments, one hundred tests were also performed. The IrisCodes and the masks computed from the training sets are regarded as B_U and M_U, respectively, and the IrisCodes and the masks computed from the testing sets are regarded as B and M, respectively. The training sets were used to compute Ψ_U and Φ_U.

In each test of the XOR protection scheme, a random bit vector S_u with a length of 2,048 was generated to lock the IrisCodes and the masks in the testing set. Half of the random bits in S_u were one. The locked B and M, regarded as $B_\otimes$ and $M_\otimes$, respectively, were used to compute $\Psi_\otimes$. The X-algorithm was applied to $\Psi_\otimes$ to estimate the random bit vector S_u and to unlock $B_\otimes$ and $M_\otimes$. The histograms of the number of bit differences between S_u and the estimated S_u, denoted by $\hat{S}_u$, are shown in Figs. 15(a)-(d). Though S_u had 2,048 bits, the X-algorithm only made 62 and 11 bits of error, on average, for the UBIRIS.v1 and WVU databases, respectively, meaning that 97.0% and 99.5% of the bits, respectively, were correct (Figs. 15(a) and (b)). The X-algorithm even correctly obtained all bits in 18 tests on the WVU database. Figs. 15(c) and (d) show that in the experiments where training and testing sets were from different databases, the X-algorithm made respectively 5 and 11 bits of error, on average. Some may expect that when the training and testing sets had different characteristics, the performance of the X-algorithm should deteriorate. In these experiments (Figs. 15(c) and (d)), the entire UBIRIS.v1 and WVU databases were utilized to calculate Ψ_U and $\Psi_\otimes$, and therefore, they were more stable. Figs. 9-11 have shown that the adjacency matrixes of the statistical dependence graphs computed from the two databases are very similar because the dependence is from the same source. Though the adjacency matrixes of the mask correlation graphs computed from the two databases are different significantly (Fig. 12), they do not influence the X-algorithm (Eqs. 21 and 22). Figs. 15(e)-(h) show the distributions of the normalized hamming distance between two IrisCodes from the same image, one in B and the other unlocked from $B_\otimes$. The distribution in Fig. 15(e) was obtained from 91,521 matchings of the iris images in the UBIRIS.v1 database. The mean of these normalized hamming distances is 0.011, and 38% of them are zero. The distribution in Fig. 15(f) was obtained from 155,566 matchings of the iris images in the WVU database. The mean of these normalized hamming distances is 0.0066, and 85% of them are zero. The distribution in Fig. 15(g) was obtained from 309,900 matchings of the iris images in the WVU database and the mean of these normalized hamming distances is 0.0020. The distribution in Fig. 15(h) was obtained from 182,900 matchings of the iris images in the UBIRIS.v1 database and the mean of these normalized hamming distances is 0.0005. The genuine and imposter distributions of the databases that produce the corresponding testing sets are also given in Figs. 15(e)-(h). Note that the training set and the testing set of Figs. 15(g) and (h) were from different databases and iris orientation was taken into account in all these matchings [1-2]. The distributions of matching unlocked IrisCodes and the corresponding original IrisCodes are very sharp and their means are very close to zero. Figs. 15(i)-(l) show the corresponding receiver operating characteristic (ROC) curves, where the raw normalized hamming distances were scaled by the function given in [4]. The ROC curves obtained from the databases that produce the corresponding testing sets are also given. Fig. 15 indicates clearly that the XOR protection scheme is vulnerable and that, through the bit probabilities and the statistical dependence, the X-algorithm can effectively unlock the protected IrisCodes without the keys.

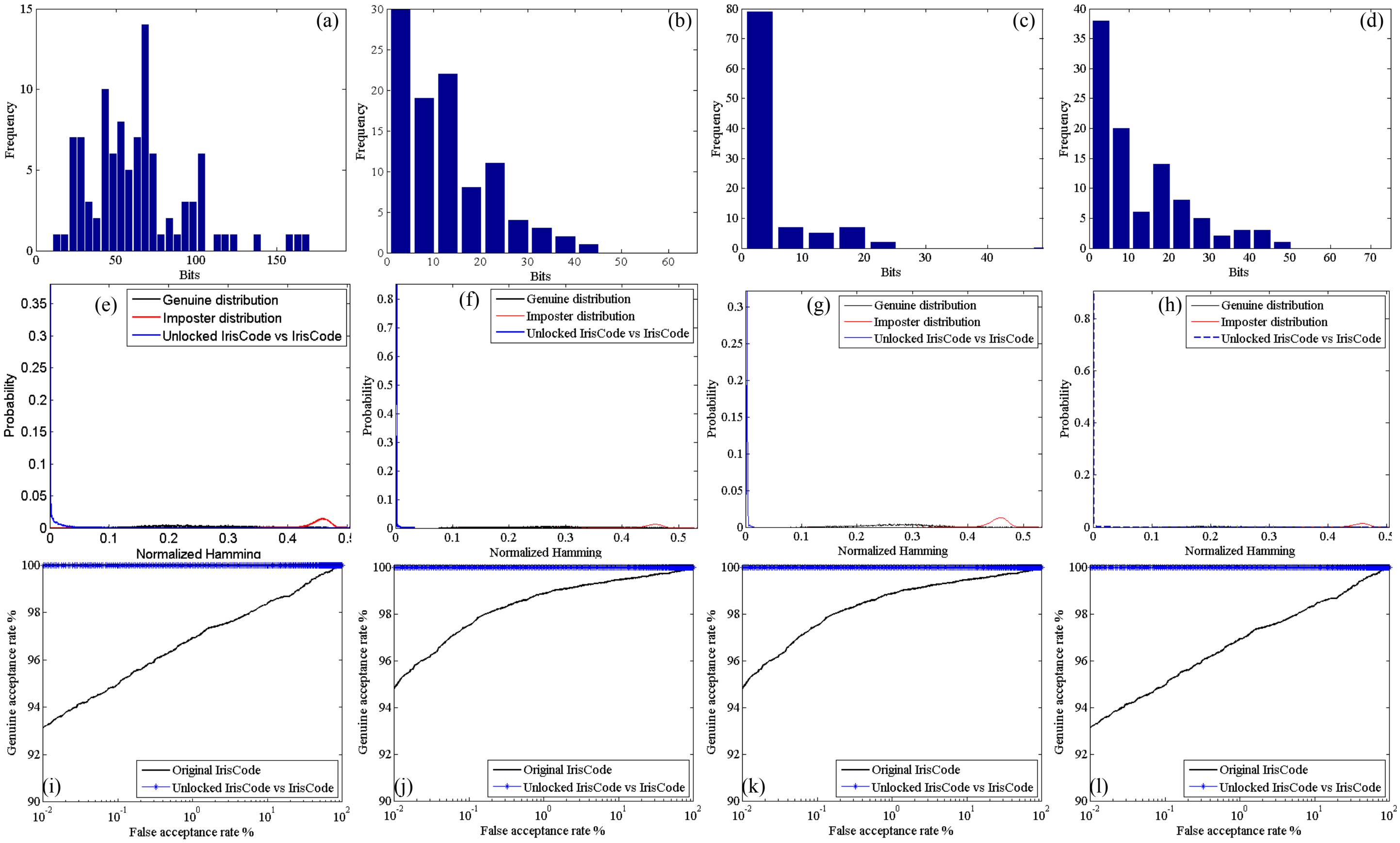


Fig. 15 Results given by the X-algorithm. The first row (a-d) shows the histograms of the number of bit differences between S_u and $\hat{S}_u$ estimated by the X-algorithm. The second row (e-h) shows the distributions of the normalized hamming distance between two IrisCodes from the same image, one in B and the other unlocked from $B_\otimes$ by the X-algorithm. The third row (i-l) shows the ROC curves of the X-algorithm. The first column is results from the UBIRIS.v1 database. The second column is results from the WVU database. The third column is results from using the UBIRIS.v1 and WVU databases as the training and testing sets, respectively and the fourth column is results from using the WVU and UBIRIS.v1 databases as the training and testing sets, respectively. (Color figure. To see clearly the distributions, please read the electronic version.)

In each test of the permutation protection scheme, a random vector $\wp_u$ with a length of 2,048, containing integers from 1 to 2,048, was generated to construct a permutation matrix P_u, where $P_u(j,i)=1$ when $\wp_u(j)=i$; otherwise, $P_u(j,i)=0$. P_u was used to lock the IrisCodes and the masks in the testing set to produce $B_\times$ and $M_\times$, and $\Psi_\times$ and $\Phi_\times$ were obtained by applying the resampling algorithm and the correlations coefficients to them. The P-algorithm was applied to $\Psi_\times$ and $\Phi_\times$ to estimate P_u, B and M. From the estimated P_u, the corresponding estimated key, denoted by $\hat{\wp}_u$, was obtained. Figs. 16(a)-(d) show the histograms of the number of different elements between $\wp_u$ and $\hat{\wp}_u$, i.e., $2048-\sum_{j=1}^{2048}\delta[\wp_u(j)-\hat{\wp}_u(j)]$, where δ is the Kronecker delta function. Fig. 16(a) shows that in 5% of the tests on the UBIRIS.v1 database, the P-algorithm correctly obtained more than 70% elements in $\wp_u$. Fig. 16(b) shows that in 9%, 16% and 26% of the tests on the WVU database, the P-algorithm correctly obtained more than 90%, 80%, and 70% elements in $\wp_u$, respectively. The function $\tilde{\Phi}_\times(\Theta_{\times x}(j),\Theta_{\times y}(j))=\Phi_U(\Theta_{Ux}(j),\Theta_{Uy}(j))$ was not used in these two experiments. Figs. 16(c) and (d) show that when the training set and the testing set used to compute $\{\Psi_U,\Phi_U\}$ and $\{\Psi_\times,\Phi_\times\}$ had different characteristics, the differences between $\wp_u$ and $\hat{\wp}_u$ increased. $\Psi_\times$ and Ψ_U were stable for the two databases, but $\Phi_\times$ and Φ_U were not. In these experiments, the parameter ι in Eq. 27 was lowered down and the function $\tilde{\Phi}_\times(\Theta_{\times x}(j),\Theta_{\times y}(j))=\Phi_U(\Theta_{Ux}(j),\Theta_{Uy}(j))$ was also used to reduce the impact from the differences between $\Phi_\times$ and Φ_U. Figs. 16(e)-(h) show the distributions of the normalized hamming distance between two IrisCodes from the same image, one in B and the other unlocked from $B_\times$. The genuine and imposter distributions of the databases that were used to compute the corresponding $\Psi_\times$ and $\Phi_\times$ are also given. Figs. 16(e) and (f) show that a large amount of normalized hamming distances are shorter 0.33 and some of them are even shorter than 0.1. However, when $\{\Psi_U,\Phi_U\}$ and $\{\Psi_\times,\Phi_\times\}$ were computed from different databases, the normalized hamming distances shorter than 0.33 reduced (Figs. 16(g) and (h)). Figs. 16(i)-(l) show the corresponding ROC curves. The ROC curves of the databases that were used to compute $\{\Psi_\times,\Phi_\times\}$ are also given. The ROC curves in Figs. 16(i) and (j) were obtained from the UBIRIS.v1 and WVU databases, respectively. They indicate that when the threshold is set at the false acceptance rate of 10^{-2}%, more than 40% of the unlocked IrisCodes can match their corresponding unprotected IrisCodes. When $\{\Psi_U,\Phi_U\}$ and $\{\Psi_\times,\Phi_\times\}$ were computed from different databases, the ROC curves dropped (Figs. 16(k) and (l)). However, for the same threshold, approximate 20% of the unlocked IrisCodes can still match their corresponding unprotected IrisCodes. Fig. 16 demonstrates that the permutation scheme is also vulnerable.

The experimental results show that the permutation protection scheme is more secure than the XOR protection scheme. Comparing the security keys S_u and $\wp_u$ in the two schemes and analyzing the two algorithms, this result is not surprising, because the key space of $\wp_u$ (2048!) is much larger than that of S_u (2^{2048}), and the selection space $(/\Gamma_\times)$ in the optimization step (Eq. 27) of the P-algorithm is also much larger than the selection space $(\{0,1\})$ in the optimization step (Eq. 22) of the X-algorithm. The experimental results fit the a-priori expectation. A comparison of the results shown in Figs. 16(a)-(d) and 16(e)-(h), obtained from the P-algorithm, indicates that though the errors between the keys $\wp_u$ and $\hat{\wp}_u$ are large, the normalized hamming distance between the IrisCodes in B and the IrisCodes unlocked from $B_\times$ is still short. This phenomenon can be attributed to the robustness of IrisCode. Roughly speaking, even for two unrelated IrisCodes, half of their bits are the same. Thus, only a portion, e.g., 50% of the elements in $\hat{\wp}_u$ being correct is sufficient to produce a normalized hamming distance shorter than a decision threshold, e.g., 0.33.

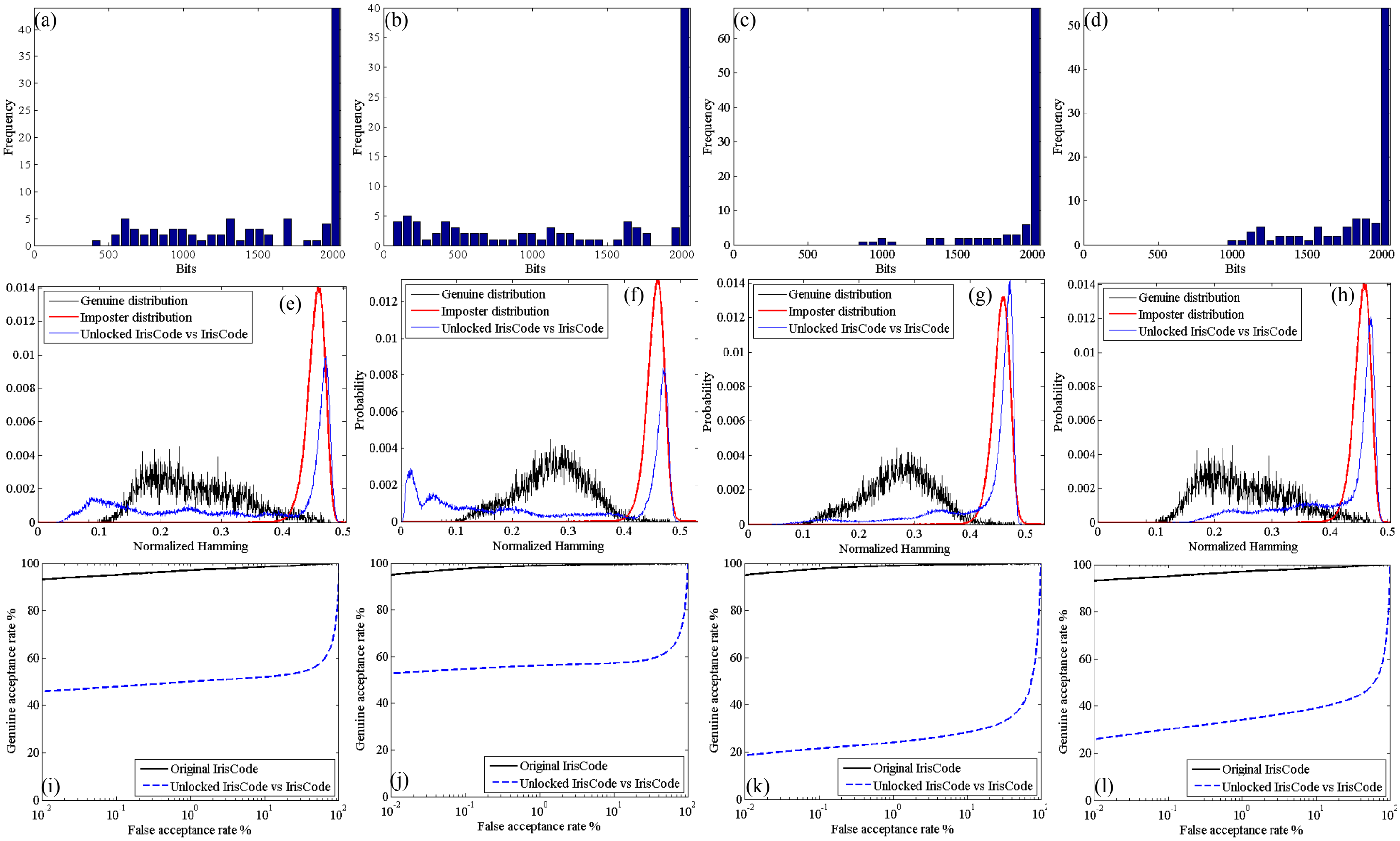


Fig. 16 Results given by the P-algorithm. The first row (a-d) shows the histograms of the number of different elements between $\wp_u$ and $\hat{\wp}_u$ obtained by the P-algorithm. The second row (e-h) shows the distributions of the normalized hamming distance between two IrisCodes from the same image, one in B and the other unlocked from $B_\times$ by the P-algorithm. The third row (i-l) shows the ROC curves of the P-algorithm. The first column is results from the UBIRIS.v1 database. The second column is results from the WVU database. The third column is results from using the UBIRIS.v1 and WVU databases as the training and testing sets, respectively and the fourth column is results from using the WVU and UBIRIS.v1 databases as the training and testing sets, respectively. (Color figure)

5. Summary, Discussion and Future Work

Over 400 million users have been mathematically enrolled by IrisCode, and this number is increasing at a dramatic rate. Detailed analyses of IrisCode, especially aspects related to security and privacy, are greatly needed. Although a number of papers have been published on the theoretical and geometric properties of IrisCode, its statistical characteristics have not been fully revealed. This paper first studies the relationship between bit probabilities and a mean of iris images, as described by Eq. 9, and shows that bit probabilities are not always approximately 0.5 and that they are highly dependent on the mean of iris images. That is, zero-DC filters do not guarantee that bit probabilities are approximately 0.5. However, when the mean of iris images is constant everywhere, they will be approximately 0.5, which matches the well-known results reported in Daugman's paper, published in 1993 [1]. To study the statistical dependence between bits, the Chi-square statistic, the correlation coefficient and a resampling algorithm are utilized. These statistical tools successfully suppress detection noise and reveal the statistical dependence in IrisCodes. The adjacency matrix of the graph formed by the statistical dependence is very structural. A comparison of this adjacency matrix and that of the Gabor graph shows that partial statistical dependence between bits is induced by the Gabor filters and propagates through the graph to produce other statistical dependence. Using this newly derived statistical information, two graph-based algorithms were developed to examine the security risk in the patented XOR and permutation protection schemes proposed by Braithwaite et al. [22-23] and deployed in commercial applications [25]. The experimental results demonstrate that both schemes are vulnerable. Even without the keys, these algorithms can unlock the protected IrisCodes and estimate the keys through the bit probabilities and the statistical dependence.

The experiments which the color images (the UBIRIS.v1 database) were used as a training set and the NIR images (the WVU database) were used as a testing set, and vice versa are extreme cases of evaluating the X-algorithm and the P-algorithm because the training and testing images have totally different characteristics. It should be highlighted that commercial iris recognition systems have quality checkers to control image quality and ISO/IEC JTC 1/SC 37 working group has set up iris image standards. If attackers, e.g., insiders know the model of the iris recognition system that produces the protected IrisCodes, they can purchase the same system to collect iris images as a training set. It will increase similarity between the training set and the testing set.

The Gabor filters used in the X-algorithm and the P-algorithm should not be considered secrets because attackers can estimate them from large iris image databases. This estimation may not require many unprotected IrisCodes and the corresponding images. To obtain excellent performance, the parameters of the Gabor filters in commercial iris recognition systems are optimized. Attackers can use a large iris image database to optimize their parameters and validate them on publicly available IrisCodes and the corresponding images, which can be found in Daugman's scientific papers and homepage. For the XOR protection scheme, attackers can even use the statistical dependence in the protected IrisCodes to estimate the Gabor filters. One simple method is $\arg\min_{G_s} f_d(|\Psi_\otimes| - |\Psi_{U_{Gs}}|)$, where f_d is a matrix norm, e.g., the Frobenius norm, $\Psi_{U_{Gs}}$ is the adjacency matrix of the statistical dependence graph obtained by applying the Gabor filters G_s to unrelated iris images and $|\Psi_\otimes|$ and $|\Psi_{U_{Gs}}|$ are two matrixes whose elements are $|\Psi_\otimes(j,k)|$ and $|\Psi_{U_{Gs}}(j,k)|$, respectively. In this minimization, G_s is a set of zero-DC Gabor filters controlled by free parameters.

To link application-specific IrisCodes in different databases locked by different keys, attackers may not even need the Gabor filters. Let $B_{\otimes j\bullet} = S_u \otimes B_{j\bullet}$ and $B'_{\otimes j\bullet} = S_{u'} \otimes B_{j\bullet}$ be two application-specific IrisCodes generated by the XOR protection scheme with two different keys S_u and $S_{u'}$. Using the XOR operator, their relationship is described by $B_{\otimes j\bullet} = S_u \otimes S_{u'} \otimes B'_{\otimes j\bullet}$, where $S_u \otimes S_{u'}$ is regarded as a single key and $B'_{\otimes j\bullet}$ is regarded as the original IrisCode in Eq. 5. Thus, the X-algorithm can be used to estimate the key $S_u \otimes S_{u'}$, and lastly, the relationship between $B_{\otimes j\bullet}$ and $B'_{\otimes j\bullet}$ can be revealed. To be more precise, attackers only need to replace Ψ_U in the X-algorithm with the adjacency matrix of $B'_\otimes$ for this attack. A similar equation, $B_{\times j\bullet} = B'_{\times j\bullet} \times P_{u'}^{-1} \times P_u$, where $B_{\times j\bullet} = B_{j\bullet} \times P_u$, $B'_{\times j\bullet} = B_{j\bullet} \times P_{u'}$ and $P_{u'}$ and P_u are two random permutation matrixes, is derived from the permutation protection scheme. $P_{u'}^{-1} \times P_u$ is regarded as a single permutation matrix, but estimating this matrix demands more computational power. The core of the P-algorithm can still be used. However, the number of corresponding pairs, i.e., $\hat{\wp}(i_1) = j_1$ and $\hat{\wp}(i_2) = j_2$, needed to run the core is greater because the statistical properties of the starting point (i_1, i_2) cannot be estimated from unrelated iris images to set the thresholds in the starting point generation component. Assuming that (i_1, i_2) is a pair of bit locations with the strongest statistical dependence in $B'_\times$, for each bit location $v_o \in \{1, \cdots, 2048\}$ in $B_\times$, $(v_o, v_1), \cdots, (v_o, v_{\hat{n}})$, where $v_1, \cdots, v_{\hat{n}}$ are the $\hat{n}$ bit locations with the strongest statistical dependence on v_o, are considered to be corresponding pairs, i.e., $\hat{\wp}(i_1) = v_0$ and $\hat{\wp}(i_2) = v_k$, where $k \in \{1, \cdots, \hat{n}\}$. Because most of the bit pairs are independent, $\hat{n} \ll 2048$. The sign of their correlation coefficients can be used to further reduce this number. To be more precise, if (i_1, i_2) is positively (negatively) correlated, the corresponding pair (v_o, v_k) must also be positively (negatively) correlated. Let the total number of (v_o, v_k) pairs be 20 after the reduction based on the sign of the correlation coefficients. Thus, the total number of corresponding pairs needed to start the core is 61,440 (2,048×20×2). Although this number is much greater than the original number, attackers with rich computational resources can still accomplish the attack. Cloud computing is one inexpensive way to access huge computational power in a short period of time.

Braithwaite et al. mentioned that using the XOR and permutation protection schemes simultaneously increases the security level. This scheme is described by

$$B_{\otimes\times j\bullet} = B_{j\bullet} \otimes S_u \times P_u \, . \tag{29}$$

Undoubtedly, this scheme is more difficult to break because P_u randomly changes the dependent bit locations in the adjacency matrix of the statistical dependence graph and S_u randomly changes the sign of its elements. However, it would be a mistake to think that simply combining the two vulnerable schemes would achieve perfect security. One possible attack can be based on a modification of the P-algorithm and the X-algorithm. To break the scheme described by Eq. 29, the P-algorithm can be used to recover the positional information, and then the X-algorithm can be used to recover the sign information. However, the original $\Psi_{\times}$ in the P-algorithm must be replaced with $\left|\Psi_{\otimes\times}\right|$, where $\Psi_{\otimes\times}$ is the adjacency matrix of the statistical dependence graph computed from $B_{\otimes\times}$, because S_u randomly changes the sign of its elements. Due to the loss of the sign information, some features employed in the starting point generation component cannot be used. Consequently, more computational resources are required.

There are a number of approaches available to eliminate the security risk reported in this study. The easiest approach is to use different keys to lock different IrisCodes and store them in a distributed database (e.g., smartcards). The statistical dependence could not then be detected. Braithwaite et al. also mentioned this approach [22-23], but they did not notice the security risk reported in this study. The price of this approach is that it only supports verification. Another approach is to store IrisCodes in a central database but use different keys to lock different IrisCodes. For identification, all of the keys would be applied to input IrisCodes to match IrisCodes locked by different keys. Although this approach can support identification, the identification is much slower than in the original approach. How to protect the keys is another problem because the number of keys, which are stored in the central database, is equal to the number of IrisCodes. Using the XOR protection scheme in these scenarios is in fact an implementation of one-time pad, which was proven to offer perfect security [34], if the mask information cannot be utilized to infer any bit in an IrisCode. To encrypt the mask information, the permutation protection scheme should also be applied. Note that the permutation protection scheme alone does not offer perfect security even in these scenarios. Assuming that an IrisCode whose all bits are one is processed by the permutation protection scheme, the protected IrisCode is still same as the original one. This simple example demonstrates that the permutation protection scheme does not offer perfect security because it retains the bit frequency in the protected IrisCode. To maintain the identification speed of the original approach, stronger template protection methods are greatly needed. When researchers design these methods, the statistical results presented in this study must be taken into account. In addition to template protection methods, other security measures, e.g., database security and hardware measures, should be used simultaneously. Although this study concentrates on the use of statistical information in security analysis, this information should be further exploited to enhance the performance of IrisCode.

Acknowledgements

The author would like to thank West Virginia University and the University of Beira Interior for sharing their databases. This work is partially supported by the Academic Research Fund Tier 1 (RG6/10) of the Ministry of Education of Singapore.

References

[1] J.G. Daugman, "High confidence visual recognition of persons by a test of statistical independence", *IEEE TPAMI*, vol. 15, no. 11, pp. 1148-1161, 1993.

[2] J. Daugman, "How iris recognition works", *IEEE TCSVT*, vol. 14, no. 1, pp. 21-30, 2004.

[3] J. Daugman, "New methods in iris recognition", *IEEE TSMC, B*, vol. 37, no. 5, pp. 1167-1175, 2007.

[4] A.W.K Kong, D. Zhang and M. Kamel, "An analysis of IrisCode", *IEEE TIP*, vol. 19, no. 2, pp. 522-532, 2010.

[5] A.W.K. Kong and D. Zhang, "Competitive coding scheme for palmprint verification", *ICPR*, vol. 1, pp. 520-523, 2004.

[6] A.K. Jain, S. Prabhakar, L. Hong and S. Pankanti, "Filterbank-based fingerprint matching", *IEEE TIP*, vol. 9, no. 5, pp. 846-859, 2000.

[7] Z. Sun, T. Tan, Y. Wang and S.Z. Li, "Ordinal palmprint representation for personal identification", *CVPR*, vol. 1, pp. 279-284, 2005.

[8] L. Ma, T. Tan, Y. Wang and D. Zhang, "Efficient iris recognition by characterizing key local variations", *IEEE TIP*, vol. 13, no. 6, pp. 739-750, 2004.

[9] Z. Sun and T. Tan, "Ordinal measures for iris recognition", *IEEE TPAMI*, vol. 31, no. 12, pp. 2211-2226, 2009.

[10] L. Zhang, L. Zhang, D. Zhang and H. Zhu, "Online finger-knuckle-print verification for personal authentication", *Pattern Recognition*, vol. 43, pp. 2560-2571, 2010.

[11] H.A. Park and K.R. Park, "Iris recognition based on score level fusion by using SVM", *Pattern Recognition Letters*, vol. 28, pp. 2019-2028, 2007.

[12] H. Proença and L.A. Alexandre, "Toward noncooperative iris recognition: a classification approach using multiple signatures", *IEEE TPAMI*, vol. 29, no. 4, pp. 607- 612, 2007.

[13] E. Krichen, M.A. Mellakh, S. Garcia-Salicetti and B. Dorizzi, "Iris identification using wavelet packets", *ICPR*, vol. 4, pp. 226-338, 2004.

[14] S.I. Noh, K. Bae, Y. Park and J. Kim, "A novel method to extract features for iris recognition system", *LNCS*, Springer*,* vol. 2688, pp. 861-868, 2003.

[15] W.K. Kong and D. Zhang, "Palmprint texture analysis based on low-resolution images for personal authentication", *ICPR*, pp. 807-810, 2002.

[16] J. Daugman, History of iris recognition, http://www.cl.cam.ac.uk/~jgd1000/history.html.

[17] K.P. Hollingsworth, K.W. Bowyer and P.J. Flynn, "Improved iris recognition through fusion of hamming distance and fragile bit distance", *TPAMI*, vol. 33, no. 12, 2465-2475, 2011.

[18] K.P. Hollingsworth, K.W. Bowyer and P.J. Flynn, "The best bits in an iris code", *IEEE TPAMI*, vol. 31, no. 6, pp. 964-973, 2009.

[19] A. Kong, "An analysis of Gabor detection", *International Conference on Image Analysis and Recognition*, pp 64-72, 2009.

[20] A.W.K. Kong, "IrisCode decompression based on the dependence between its bit pairs", *IEEE TPAMI*, vol. 34, no. 3, pp. 506-520, 2012.

[21] A.W.K. Kong, "Modeling IrisCode and its variants as convex polyhedral cones and its security implications", *IEEE TIP*, vol. 22, no. 3, pp. 1148-1160, 2013.

[22] M. Braithwaite, U.C. von Seelen, J. Cambier, J. Daugman, R. Class, R. Moore and I. Scott, "Applications-specific biometric template", *IEEE Workshop on Automatic Identification Advanced Technologies*, pp. 167-171, 2002.

[23] M. Braithwaite, U.C. von Seelen, J. Cambier, J. Daugman, R. Class, R. Moore and I. Scott, "Authentication using application-specific biometric templates", WO2002/095657, 2002.

[24] J. Daugman, "The importance of being random: statistical principles of iris recognition", *Pattern Recognition*, vol. 36, no. 2, pp. 279-291, 2003.

[25] J.R. Matey, D. Ackerman, J. Dergen and M. Tinker, "Iris recognition in less constrained environments", pp. 114, in Advances in Biometrics: Sensors, Algorithms and Systems edited by N.K. Ratha, V. Govindaraju, Springer, 2007.

[26] H. Proenca and L.A. Alexandre, "UBIRIS: a noisy iris image database", *13th ICIAP*, vol. 1, pp. 790-977, 2005.

[27] A. Ross and S. Shah, "Segmenting non-ideal irises using geodesic active contours", *Biometrics Symposium*, pp. 1-6, 2006.

[28] S. Umeyama, "An eigendecomposition approach to weighted graph matching problems", *IEEE TPAMI*, vol. 10, no. 5, pp. 695-703, 1998.

[29] H.A. Almohamad and S.O. Duffuaa, "A linear programming approach for the weighted graph matching problem", *TPAMI*, vol. 15, no. 5, pp. 522-525, 1993.

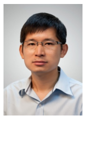
[30] M. Zaslavskiy, F. Bach and J.P. Vert, "A path following algorithm for the graph matching problem", *IEEE TPAMI*, vol. 31, no. 2, pp. 2227-2242, 2009.

[31] S. Venugopalan and M. Savvides, "How to generate spoofed irises from and iris code template", *IEEE TIFS*, vol. 6, no. 2, pp. 385-395, 2011.

[32] J. Galbally, A. Ross, M. Gomez-Barrero, J. Fierrez and J. Ortega-Garcia, "From the Iriscode to the iris: a new vulnerability of iris recognition systems", *Black Hat USA*, 2012.

[33] G. Santos and H. Proenca, "Iris recognition: analyzing the distribution of the Iriscodes concordant bits", *3rd International Congress on Image and Signal Processing*, vol. 4, pp. 1873-1877, 2010.

[34] A. Sinkov, Elementary Cryptanalysis A Mathematical Approach, The Mathematical Association of America, Second Edition (Revised and Updated by T. Feil), 2009



Adams Wai-Kin Kong received his PhD from the University of Waterloo, Canada. Currently, he is an associate professor at the Nanyang Technological University, Singapore. His research interests include biometrics, forensics, image processing, and pattern recognition.