Biometric Cryptosystems: Issues and Challenges

UMUT ULUDAG, STUDENT MEMBER, IEEE, SHARATH PANKANTI, SENIOR MEMBER, IEEE, SALIL PRABHAKAR, MEMBER, IEEE, AND ANIL K. JAIN, FELLOW, IEEE

Contributed Paper

In traditional cryptosystems, user authentication is based on possession of secret keys, which falls apart if the keys are not kept secret (i.e., shared with nonlegitimate users). Further, keys can be forgotten, lost, or stolen and, thus, cannot provide nonrepudiation. Current authentication systems based on physiological and behavioral characteristics of persons (known as biometrics), such as fingerprints, inherently provide solutions to many of these problems and may replace the authentication component of the traditional cryptosystems. In this paper, we present various methods that monolithically bind a cryptographic key with the biometric template of a user stored in the database in such a way that the key cannot be revealed without a successful biometric authentication. We assess the performance of one of these biometric key binding/generation algorithms using the fingerprint biometric. We illustrate the challenges involved in biometric key generation primarily due to drastic acquisition variations in the representation of a biometric identifier and the imperfect nature of biometric feature extraction and matching algorithms. We elaborate on the suitability of these algorithms for the digital rights management systems.

Keywords*—*Authentication, biometrics, confidentiality, cryptography, entropy, fingerprints, invariance, key binding, key generation, key release, multibiometrics, privacy, secrecy, security.

I. INTRODUCTION

Content owners (such as authors and authorized distributors) are losing billions of dollars annually in revenues due to illegal copying and sharing of digital media [1], [2]. Digital rights management (DRM) systems are being deployed to address this problem. The user authentication, which is an essential part of a DRM system, determines whether a user is authorized to access the content. In a generic cryptographic system the user authentication is possession based. That is, possession of the decrypting key is a sufficient evidence to establish user authenticity. Because cryptographic keys are long and random, (e.g., 128 bits for the advanced encryption standard (AES) [3], [4]), they are difficult to memorize. As a result, the cryptographic keys are stored somewhere (for example, on a computer or a smart card) and released based on some alternative authentication (e.g., password) mechanism, that is, upon assuring that they are being released to the authorized users only. Most passwords are so simple that they can be easily guessed (especially based on social engineering methods) or broken by simple dictionary attacks [5]. It is not surprising that the most commonly used password is the word "password"! Thus, the multimedia protected by the cryptographic algorithm is only as secure as the passwords (weakest link) used for user authentication that release the correct decrypting key(s). Simple passwords are easy to crack and, thus, compromise security; complex passwords are difficult to remember and, thus, are expensive to maintain.[1] Users also have the tendency to write down complex passwords in easily accessible locations. Further, most people use the same password across different applications and, thus, if a single password is compromised, it may open many doors. Finally, passwords are unable to provide nonrepudiation; that is, when a password is shared with a friend, there is no way to know who the actual user is. This may eliminate the feasibility of countermeasures such as holding conniving legitimate users accountable in a court of law.

Many of these limitations of the traditional passwords can be ameliorated by incorporation of better methods of user authentication. Biometric authentication [7], [8] refers to verifying individuals based on their physiological and behavioral characteristics such as face, fingerprint, hand geometry, iris, keystroke, signature, voice, etc. It is inherently more reliable than password-based authentication, as biometric characteristics cannot be lost or forgotten (cf. passwords being lost or forgotten); they are extremely difficult to copy, share, and distribute (cf. passwords being announced in hacker websites) and require the person being authenticated to be present

Manuscript received September 12, 2003; revised February 21, 2004.

U. Uludag and A. K. Jain are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 (e-mail: uludagum@cse.msu.edu; jain@cse.msu.edu).

S. Pankanti is with the Exploratory Computer Vision Group, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 (e-mail: sharat@watson.ibm.com).

S. Prabhakar is with DigitalPersona Inc., Redwood City, CA 94062 (e-mail: salilp@digitalpersona.com).

Digital Object Identifier 10.1109/JPROC.2004.827372

[1]For example, anywhere between 25% and 50% of help desk calls relate to password resets; these calls cost as much as $30 per end user, with the help desk receiving at least five calls per end user every year [6].

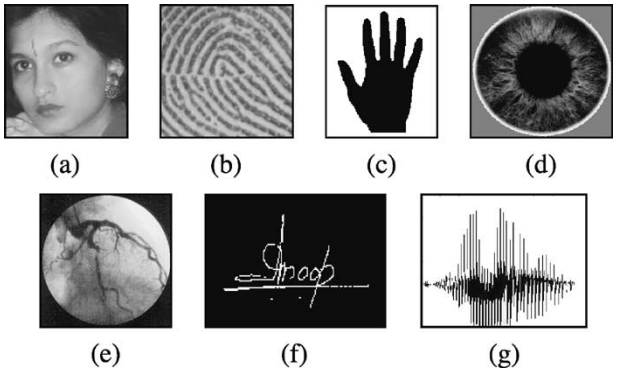

Fig. 1. Examples of biometric characteristics. (a) Face. (b) Fingerprint. (c) Hand geometry. (d) Iris. (e) Retina. (f) Signature. (g) Voice. From D. Maltoni, D. Maio, A. K. Jain, S. Prabhakar, *Handbook of Fingerprint Recognition* (New York: Springer-Verlag, 2003), Fig. 1.2, p. 8. Copyright by Springer-Verlag. Reprinted with permission.

at the time and point of authentication (cf. conniving users denying having shared the password). It is difficult to forge biometrics (it requires more time, money, experience, and access privileges) and it is unlikely for a user to repudiate having accessed the digital content using biometrics. Finally, one user's biometrics is no easier to break than another's; that is, all users have a relatively equal security level, hence, there are not many users who have "easy to guess" biometrics, that can be used to mount an attack against them. Thus, biometrics-based authentication is a potential candidate to replace password-based authentication, either by providing the complete authentication mechanism or by securing the traditional cryptographic keys that contain the multimedia file in a DRM system. In this paper, we attempt to present an analysis of implications of the existing biometric technologies to the containment process. We present a brief summary of biometric technology and dwell on the challenges involved in incorporating the biometric technologies to the cryptographic systems (Section II). We review the existing approaches for overcoming the challenges involved in designing biometrics-based cryptographic systems along with their strengths and limitations (Section III). Using fingerprint data, we present the limitations of the present approach to designing biometric cryptosystems (Section IV). Finally, in Section V, we summarize the advantages of biometric cryptosystems, challenges of designing such systems and stipulate on some of the promising directions for further research for a successful marriage of the biometric and cryptographic techniques.

II. Biometrics

A number of biometric characteristics have been in use in various applications (see Fig. 1). Each biometric has its strengths and weaknesses, and the choice depends on the application. No single biometric is expected to effectively meet all the requirements (e.g., accuracy, practicality, cost) of all the applications (e.g., DRM, access control, welfare distribution). In other words, no biometric is "optimal." The match between a specific biometric and an application is determined depending upon the requirements of the application and the properties of the biometric characteristic.

Table 1
Comparison of Various Biometric Technologies Based on the Perception of the Authors. High, Medium, and Low are Denoted by H, M, and L, Respectively

Biometric identifier	Universality	Distinctiveness	Permanence	Collectability	Performance	Acceptability	Circumvention
Face	H	L	M	H	L	H	H
Fingerprint	M	H	H	M	H	M	M
Hand geometry	M	M	M	H	M	M	M
Iris	H	H	H	M	H	L	L
Keystroke	L	L	L	M	L	M	M
Signature	L	L	L	H	L	H	H
Voice	M	L	L	M	L	H	H

A brief comparison of some of the biometric identifiers based on seven factors is provided in Table 1. Universality (do all people have it?), distinctiveness (can people be distinguished based on an identifier?), permanence (how permanent is the identifier?), and collectability (how well can the identifier be captured and quantified?) are properties of biometric identifiers. Performance (speed and accuracy), acceptability (willingness of people to use), and circumvention (foolproof) are attributes of biometric systems [9]. Use of many other biometric characteristics such as retina, infrared images of face and body parts, gait, odor, ear, and DNA in commercial authentication systems is also being investigated [7]. The following example illustrates how different biometric identifiers may be appropriate in different scenarios. If one would like to provide "just-in-time" secure access to the documents for "write/modify" operations to authorized users, e.g., brokers bidding on commodity items using a keyboard—both for repudiability as well as security—the most natural biometric for authenticating the bid document would be either keystroke dynamics or having fingerprint sensors on each key of the keyboard. If the brokers were bidding vocally, the bid voice segments could be authenticated using voice (speaker) recognition. If the application is intended for providing read-only access to a top secret "for your eyes only" document, ideal authentication would be iris or retina recognition of the authorized reader as she reads the document (contents can perhaps be projected directly onto their retina). Thus, depending upon the operational situation, different biometric characteristics are suitable for different DRM applications.

A. Biometric (In)Variance

Password-based authentication systems do not involve any complex pattern recognition and, hence, they almost always

Fig. 2. Top: variations associated with an individual's face image due to changes in pose. Bottom: variations in fingerprint images of the same finger over a period of six weeks due to wear and tear of ridges.

perform accurately as intended by their system designers. On the other hand, biometric signals and their representations (e.g., facial image and its computer representation) of a person vary dramatically depending on the acquisition method, acquisition environment, user's interaction with the acquisition device, and (in some cases) variation in the traits due to various pathophysiological phenomena. Below, we present some of the common reasons for biometric signal/representation variations.

Inconsistent Presentation: The signal captured by the sensor from a biometric identifier depends upon both the intrinsic identifier characteristic as well as the way the identifier was presented. Thus, an acquired biometric signal is a nondeterministic composition of physiological trait, the user characteristic behavior, and the user interaction facilitated by the acquisition interface. For example, determined by the pressure and contact of the finger on the image acquisition surface, the three-dimensional shape of the finger gets mapped onto the two-dimensional surface of the sensor surface. Since the finger is not a rigid object and since the process of projecting the finger surface onto the sensor surface is not precisely controlled, different impressions of a finger are related to each other by various transformations. Further, each impression of a finger may possibly depict a different portion of its surface. This may introduce additional spurious fingerprint features. In the case of a face, different acquisitions may represent different poses of a face (see Fig. 2). Hand geometry measurements may be based on different projections of hand on a planar surface. Different iris/retina acquisitions may correspond to different nonfrontal projections of iris/retina on to the image planes.

Irreproducible Presentation: Unlike the synthetic identifiers [e.g., radio frequency identification (RFID)], biometric identifiers represent measurements of biological trait or behavior. These identifiers are prone to wear and tear,

Fig. 3. Imperfect acquisition: three different impressions of a subject's finger exhibiting poor-quality ridges due to extreme finger dryness.

accidental injuries, malfunctions, and pathophysiological development. Manual work, accidents, etc., inflict injuries to the finger, thereby changing the ridge structure of the finger either permanently or semipermanently (see Fig. 2). Wearing of different kinds of jewelry (e.g., rings) may affect hand geometry measurements in an irreproducible way. Facial hair growth (e.g., sideburns, mustache), accidents (e.g., broken nose), attachments (e.g., eyeglasses, jewelry), makeup, swellings, cyst growth, and different hairstyles may all correspond to irreproducible face depictions. Retinal measurements can change in some pathological developments (e.g., diabetic retinopathy). The gait of a pregnant woman is significantly different from that of a woman who is not pregnant. Inebriation results in erratic signatures. The common cold changes a person's voice. All these phenomena contribute to dramatic variations in the biometric identifier signal captured at different acquisitions.

Imperfect Signal/Representational Acquisition: The signal acquisition conditions in practical situations are not perfect and cause extraneous variations in the acquired biometric signal. For example, nonuniform contact results in poor-quality fingerprint acquisitions. That is, the ridge structure of a finger would be completely captured only if ridges belonging to the part of the finger being imaged are in complete physical/optical contact with the image acquisition surface and the valleys do not make any contact with the image acquisition surface. However, the dryness of the skin, shallow/worn-out ridges (due to aging/genetics), skin disease, sweat, dirt, and humidity in the air all confound the situation, resulting in a nonideal contact situation (see Fig. 3). In the case of inked fingerprints, inappropriate inking of the finger often results in "noisy" low-contrast (poor-quality) images, which lead to either spurious or missing minutiae. Different illuminations cause conspicuous differences in the facial appearance. Backlit illumination may render image acquisition virtually useless in many applications. Depending upon ergonomic conditions, the signature may vary significantly. The channel bandwidth characteristics affect the voice signal.

The feature extraction algorithm is also imperfect and introduces measurement errors. Various image processing operations might introduce inconsistent biases to perturb feature localization. Two biometric identifiers extracted from two different people can be very similar because of the inherent lack of distinctive information in the biometric identifier or because the representation used for the biometric identifiers is too restrictive.

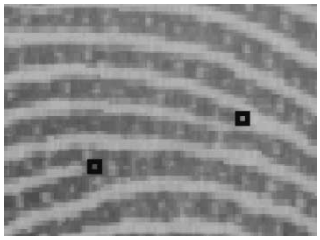

Fig. 4. Fingerprint minutiae. A ridge ending and a ridge bifurcation are shown.

As a result of these complex variations in the biometric signal/representations, determining whether two presentations of a biometric identifier are the same typically involves complex pattern recognition and decision making. Another ramification (compared to password-based authentication systems) is that the design of biometric cryptosystems must take into account the effects of these variations.

B. Biometric Matcher

For various reasons mentioned in the earlier section, unlike password or keys, the exact match of biometric identifiers is not very useful. Typically, a practical biometric matcher undoes some of the variations in the biometric measurements to be matched by *aligning* them with respect to each other. Once the two representations are aligned, an assessment of their similarity is measured based on acceptable variations within the aligned representations and is typically quantified in terms of a *matching score*; the higher the matching score, the more similar are the representations.

Let us consider a concrete example of fingerprint matching. The most widely used local features (ridge ending and ridge bifurcation) are based on minute details (*minutiae*) of the fingerprint ridges (see Fig. 4). The pattern of the minutiae of a fingerprint forms a valid, compact, and robust representation of the fingerprint and it captures a significant component of information in fingerprints. The simplest of the minutiae-based representations constitute a list of triplets (x, y, θ), where (x, y) represents the spatial coordinates in a fixed image-centric coordinate system and θ represents the orientation of the ridge at that minutia. Typically, a good-quality live-scan fingerprint image has 20–70 minutiae.

Only in the highly constrained fingerprint systems could one assume that the input and template fingerprints depict the same portion of the finger and both are aligned (in terms of displacement from the origin of the imaging coordinate system and of their orientations) with each other; given two (input and template) fingerprint representations, the matching module typically aligns the input and template minutiae and determines whether the prints are impressions of the same finger by identifying *corresponding* minutiae within an acceptable spatial neighborhood of the aligned minutiae. The number of corresponding minutiae is an effective measure of similarity between the matched prints. Fig. 5 illustrates a typical matching process. Even in the best of practical situations, *all* minutiae in input and template prints are rarely matched due to spurious minutiae introduced by dirt/leftover smudges, variations in the area of finger being imaged, and displacement of the minutia owing to distortion of the print from pressing the elastic finger against the flat surface of the acquisition device.

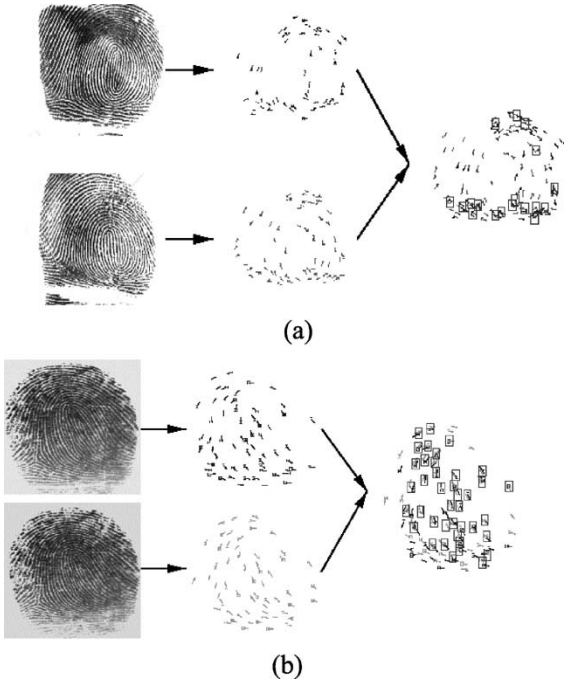

Fig. 5. Fingerprint matching. Here, matching consists of feature (minutiae) extraction followed by alignment and determination of *corresponding* minutiae (highlighted in boxes). (a) Matching two impressions of different fingers, matching score = 4. (b) Matching fingerprints from the same finger, matching score = 49. Maximum possible score is 100.

C. Performance Metrics

A biometric authentication system makes two types of errors: 1) mistaking biometric measurements from two different persons to be from the same person (called *false match*) and 2) mistaking two biometric measurements from the same person to be from two different persons (called *false nonmatch*). These two types of errors are often termed as *false accept* and *false reject*, respectively. There is a tradeoff between false match rate (FMR) and false nonmatch rate (FNMR) in every biometric system. In fact, both FMR and FNMR are functions of the system threshold t; if t is decreased to make the system more tolerant to input variations and noise, then FMR increases.[2]

The accuracy requirements of a biometric system are application dependent. Consider the following example: In a DRM application involving high-security top secret documents (e.g., in a nuclear reactor), the administration may want to ensure that all such documents are accessed only by authorized users. Further, unauthorized users should have a very little chance of accessing the documents. The requirement here translates to small FMR that may typically mean a large FNMR. In a less secure environment, the

[2]Besides the above two error rates, the failure to capture (FTC) rate and the failure to enroll (FTE) rate are also used to summarize the accuracy of a biometric system [8].

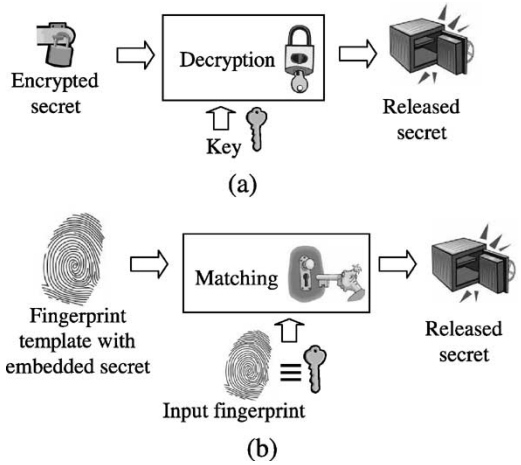

Fig. 6. A generic instantiation of simple conventional and biometric-based DRM systems. (a) In password-based authentication, a cryptographic key is the "secret" and the password is the "key." (b) In the fingerprint-based authentication, a cryptographic key is the "secret" and fingerprint is the "key." In both cases, the cryptographic key is released upon a successful authentication. From D. Maltoni, D. Maio, A. K. Jain, S. Prabhakar, *Handbook of Fingerprint Recognition* (New York: Springer-Verlag, 2003), Fig. 9.10, p. 306. Copyright by Springer-Verlag. Reprinted with permission.

primary objective of the DRM system design may be user convenience and user-friendly interface. That is, a user does not want to use engineered authentication systems (e.g., requiring badges or RFID tags) and would like to have reliable pervasive access to the documents. In this application, since user convenience is the primary criterion, the FNMR at the chosen operating point should be small, which may result in a large FMR.

III. Biometric Keys

The basic idea of biometric-based keys is that the biometric component performs user authentication (user authorization), while a generic cryptographic system can still handle the other components of containment (such as secure communication). For example, let us consider a straightforward implementation of a containment subsystem of a DRM system using biometric-based authentication. Alice, a legitimate user, wishes to access certain digital content; she offers her biometric sample to the system; if the biometric matcher successfully matches Alice's input biometric sample with her enrolled biometric template then a cryptographic key is released (see Fig. 6). The cryptographic key is used to decrypt the content and, thus, Alice is allowed access to the content. On the other hand, if Victor, an illegitimate user tries to access the same digital content posing as Alice, his biometric match with the biometric template of Alice will fail and Alice's cryptographic key would not be released by the system. We refer to this method of integrating biometrics into a cryptosystem as the method of *biometric-based key release*. Thus, in such systems, a cryptographic key is stored as part of a user's database record, together with the user name, biometric template, access privileges, and the like, that is only released upon a successful biometric authentication. Let us briefly outline the issues raised by the biometric-based key release system design.

The characteristics of the biometric key release system design are: 1) it requires access to biometric templates for biometric matching and 2) user authentication and key release are completely decoupled. Because the system stores biometric template locally, the design raises concerns about the theft of biometric data. That is, a stolen smart card gives access to the biometric template. In such systems, although biometrics eliminates the tedious task of maintaining different, complex, and changing passwords, this potential loss of biometric data is an important security issue. Further, once the biometric signals (measurements) are stolen from one DRM application, they may be used in other DRM applications (or other applications such as access control) using the same biometric identifier, thus making different applications vulnerable to the attack. Finally, since the biometric authentication is completely decoupled from the key release and outputs only an accept/reject answer, the system is vulnerable to Trojan horse attacks (e.g., a Trojan horse can replace the biometric authentication subsystem and simply inject a 1-bit accept/reject information to the key release subsystem).

In this context, solving the following problems becomes important.

1) Is it possible to design biometric systems such that if the biometric template in an application is compromised, the biometric signal itself is not lost forever and a new biometric template can be issued?
2) Is it possible to design biometric templates such that different applications are not able to use the same biometric template, thus securing the biometric signal as well as preserving privacy?
3) Is it possible to generate/release a cryptographic key using biometric information such that the cryptographic key management is secure and convenient?

It is indeed possible to integrate biometric matching and cryptographic techniques to solve all of the above three problems. We illustrate this with the following simple example to address only problems 1) and 2) above. Consider that during enrollment in a biometric system, instead of storing the original biometric signal x in the system database, only its transformed version $H(x)$ is stored. Here, the transform is a change in the representation of an entity, where the new representation may comprise exactly the same information as in the previous one or may reflect a loss or augmentation of information contained in the original representation. During authentication, the biometric sensor would morph the signal using the same transform H and the biometric matching would be carried out in the transformed space. Different applications can use different transforms (or different parameters of the same transform) so that a template issued for a specific application can only be used by that application. If a biometric template is ever compromised, a new one can be issued by using a different transform. Since such a template does not reveal a user's biometric information, we call it a private template [10] (Ratha *et al.* [11] refer to this as *cancelable biometric*). If H

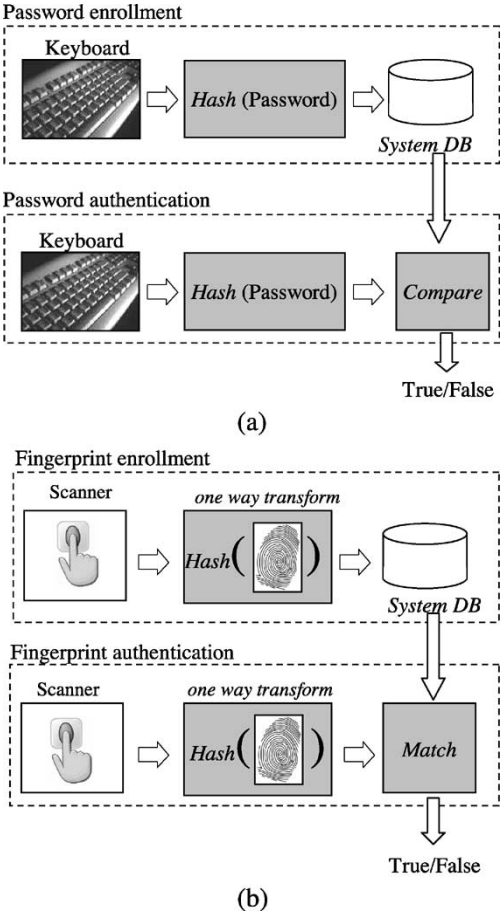

Fig. 7. Authentication based on "private templates" using hashing techniques. (a) Passwords are typically stored in the database after they are hashed; when a new password is received, it is hashed and compared with the password hashed at enrollment. If a person has access to the database of hashed passwords, a password is not compromised. In (b), a similar analogy is applied to fingerprints. Only one-way transformed representation is stored and thus, if an adversary has an access to the database, the biometric information is not compromised. From D. Maltoni, D. Maio, A. K. Jain, S. Prabhakar, Handbook of Fingerprint Recognition (New York: Springer-Verlag, 2003), Fig. 9.9, p. 303. Copyright by Springer-Verlag. Reprinted with permission.

is noninvertible (see Fig. 7), the security of the template can be assured, but the error rate of the authentication increases significantly as the matcher has difficulty in carrying out the matching in the transformed space (due to the dramatic variability in the biometric characteristic of a person). If H is invertible, then the biometric matcher can carry out the matching accurately, but the template is not secure.

Consider the following simple example that addresses problem 3) above. Instead of storing the cryptographic key in the user's record, we can hide a cryptographic key within the user's biometric template itself (e.g., via a trusted and secret bit-replacement algorithm that can replace, say, the least significant bits of the pixel values/features of the biometric template with the cryptographic key). Upon a successful biometric match, the correct cryptographic key is extracted from the biometric database template and released into the system. The security of this method is dependent on the secrecy of the key hiding and retrieval algorithms. If the key hiding and retrieval algorithms are deterministic (e.g., they always hide the key at the same locations), they can be easily compromised. For example, an attacker may enroll several people in the system using identical keys and locate the bits with common information across the enrolled biometric templates.

It is, therefore, important that the cryptographic key be monolithically bound with the biometric template in the stored database in such a way that it cannot be revealed without a successful biometric authentication. We refer to this method of integrating biometric into a cryptosystem as the method of *biometric key generation* or *binding*.[3] It is evident that such a solution would be secure inasmuch as it does not require access to the biometric features stored in the template. Further, the generation process seamlessly marries (*binds*) a private key into the user biometric information in such a way that both the cryptographic key and biometric information in the template are inaccessible to the attacker while the cryptographic key can be released to the appropriate application upon valid presentation of the user biometric template. Finally, the biometric matching does not have to be performed at all, thereby eliminating the need to access biometric information in the template.

Biometric key generation or binding still leaves several problems. As mentioned earlier in this paper, unlike a password, specific biometric signal/representations (e.g., fingerprint image and its minutiae representation) of a person vary dramatically. Consequently, it is not obvious how the inherently variant biometric signal can be used to generate cryptographic keys. The traditional cryptosystems (e.g., symmetric ciphers such as AES [3] and asymmetric ciphers such as RSA [4]) are designed to accept only identical keys used for encryption and decryption. Further, the accuracy performance of the existing biometric authentication technologies is not perfect (namely, nonzero FMR and FNMR) and there is a need to address the issues related to delivering perfect encryption/decryption performance (when the decrypted message is identical to the encrypted message), given the imperfect biometric authentication technology. Second, the "fuzzy" matching of biometrics cannot be performed in the encrypted domain because: 1) it is difficult (if not impossible) to engineer a meaningful similarity metric in the encrypted representation; 2) the biometric matchers need to align the representations before their similarity can be assessed—it is difficult to align the representations in the encrypted domain; and 3) typically biometric identifiers have variable and unordered representation; it is difficult to order the encrypted feature components so that there is a *correspondence* between the features of the two representations (i.e., order of the feature components should not matter). This implies that biometric key generation schemes have a challenging task of performing the biometric matching in an unencrypted domain without revealing significant information about the identifiers to the adversaries. Further, it is desirable that in biometric key generation schemes, there should not

[3]The biometric key generation does not necessarily imply that the same or similar key is generated by an identity all the time.

be a systematic correlation between the identity and the cryptographic key that could be exploited by the attacker (e.g., always generating the same cryptographic key for the same identity is undesirable). Finally, one needs to explore the implications of using biometric technology for key generation/release in the context of cryptographic issues such as nonrepudiation, vulnerability to cryptanalysis, etc.

A. Previous Work

Over the past several years, there have been a number of research efforts aimed at addressing the issues related to integration of biometrics into cryptosystems. Even though their number is limited to just a few, and the underlying biometric characteristics do not represent all the currently available pool of modalities, they convey the challenges related to this marriage.

Soutar *et al.* [12]–[14] proposed a key binding algorithm in an optical correlation-based fingerprint matching system. This algorithm binds a cryptographic key with the user's fingerprint images at the time of enrollment. The key is then retrieved only upon a successful authentication. By using several (training) fingerprint images of a finger (typically five), the algorithm first creates a correlation filter function $H(u)$ which has both the magnitude $|H(u)|$ and phase $e^{i\varphi(H(u))}$ components. The design criteria for this function include both distortion tolerance (in order to minimize FNMR) and discriminability (in order to minimize FMR). The algorithm also computes an output $c_0(x)$, which is obtained by convolution/correlation of the training fingerprint images with $H(u)$. Then, the complex conjugate of the phase component of $H(u)$, $e^{-i\varphi(H(u))}$ is multiplied with a randomly generated phase-only array of the same size, resulting in $H_{\text{stored}}(u)$ and the magnitude of $H(u)$, $|H(u)|$ is discarded. This process eliminates reverse engineering of the user's fingerprint image from $H(u)$. A given or randomly generated N-bit (typically 128-bit) cryptographic key k_0 is then linked with binarized correlation output c_0 by using an error-correcting code (in order to tolerate some expected variation in the biometric signal during authentication), resulting in a lookup table LT. The cryptographic key k_0 is also used as an encryption key to encrypt S bits of $H_{\text{stored}}(u)$ and the resultant encrypted message is hashed (using a standard hashing function such as SHA-1 or Triple-DES [15]) to form an identification code id_0. Finally, $H_{\text{stored}}(u)$, LT, and id_0 are stored in the database as the biometric template for the user (called *Bioscrypt* by the authors).

During authentication, the user inputs one or more (typically five) fingerprint images of her finger. The $H_{\text{stored}}(u)$ for this user is retrieved from her stored *Bioscrypt* and combined with the input fingerprint images to produce a correlation output $c_1(x)$. A cryptographic key retrieval algorithm then uses the LT for the user (stored in her *Bioscrypt*) to extract a key k_1 from the correlation output $c_1(x)$. The retrieved key k_1 is used to create a new identification code id_1 in exactly the same way as was done during enrollment. If $id_1 = id_0$, then k_1 is released into the system, else an "authentication failed" message is returned. Thus, the system never releases any (wrong) key into the system if the biometric authentication fails. The main criticism of Soutar *et al.*'s work in the literature [10], [16] is that the method does not carry rigorous security guarantees. The authors do not explain how much entropy is lost at each stage of their algorithm. Further, the resulting FMR and FNMR values are unknown. The authors also assume that the input and database templates fingerprint images are completely aligned. Even with a very constrained image acquisition system, it is unrealistic to acquire fingerprint images from a finger without any misalignment.

Davida *et al.* [10], [17] propose an algorithm based on the iris biometric. They consider binary representation of iris texture, called IrisCode [18], which is 2048 bits in length. The biometric matcher computes the Hamming distance between the input and database template representations and compares it with a threshold to determine whether the two biometric samples are from the same person or not. The authors assume that the IrisCodes from different sampling of the same iris can have up to 10% of the 2048 bits (204 bits) different from the same iris's template IrisCode. The authors also assume that the IrisCodes of different irises differ in as many as 45% of the 2048 bits (922 bits).

During enrollment, multiple scans of the iris of a person are collected and K-bit IrisCodes are generated for each scan. The multiple IrisCodes are then combined (through a majority decoder) to arrive at a canonical IrisCode T of the same length. An [N, K, D] bounded distance decoding error-correcting code [19] is then constructed by adding C check bits to the K-bit IrisCode (C is determined such that 10% of K bits can be corrected) resulting in an N-bit codeword, denoted by $T||C$. The codeword $T||C$ is hashed and digitally signed, denoted by $\text{Sig}(\text{Hash}(T||C))$, and together with the check bits C, stored as the database template for the user. At the time of authentication, again, multiple samples of iris of a person are collected and T' is estimated. The check bits C from the database template are used to perform error correction on the codeword $T'||C$ and the corrected IrisCode T'' is produced. Then $T''||C$ is hashed and signed (just like during enrollment), resulting in $\text{Sig}(\text{Hash}(T''||C))$. If $\text{Sig}(\text{Hash}(T''||C))$ is exactly the same as the $\text{Sig}(\text{Hash}(T||C))$ stored in the database template, authentication succeeds. Davida *et al.* [10], [17] argue that the database template of a user itself can be used as a cryptographic key (note that this key would always be the same for the same biometric identifier in contrast to cryptographic key binding algorithms such as Soutar *et al.*'s algorithm [12]–[14] that can bind any random/given key with the biometric data). If a chosen biometric does not provide the desired entropy (cryptographic strength), the authors propose that a password, a personal identification number, or multiple biometrics be added to the system to increase the entropy. Davida *et al.*'s algorithm [10], [17] is very fast and has provable security. However, they propose to store error-correcting bits C in the database, and this leads to some leakage of information about the user's biometric data. Further, the error tolerance of their scheme is rather small. The authors' assumption that only 10% bits of IrisCode change among different presentation of the iris of a person is too restrictive. In fact up to 30% bits of IrisCode could

be different between different presentations of the same iris [18]. The authors assume that by acquiring a large number of samples of the iris, the errors in the IrisCode would be significantly minimized. Finally, the authors assumed that the input and database template IrisCodes are completely aligned. Although constrained iris image acquisition systems can limit the misalignment among different acquisitions of the same iris, some degree of misalignment is natural. The authors have ignored this fact in their algorithm.

Monrose *et al.* [20] proposed a method to make passwords more secure by combining keystroke biometrics with passwords. Their technique was inspired by password "salting," where a user's password (*pwd*) is salted by prepending it with an s-bit random number (the "salt"), resulting in a hardened password (*hpwd*). During enrollment, the following information is stored in the user's database template: 1) a randomly chosen k (typically 160)-bit number r; 2) an "instruction table" encrypted with *pwd*—the instruction table is created as follows: first, the user's keystroke features (typically 15 in number) are thresholded to generate a binary feature descriptor, then the binary feature descriptor and r are used to create the instruction table using Shamir's secret sharing scheme [15] (the instruction table essentially contains instructions on how to generate *hpwd* from the feature descriptor r and *pwd*); and 3) a "history file" encrypted with *hpwd*. At the time of authentication, the algorithm uses the r and the instruction table from the user's template and the authentication password *pwd*′ and keystroke features acquired during the authentication to compute *hpwd*′. The *hpwd*′ is then used to decrypt the encrypted history file. If the decryption is successful, the authentication is successful, and the r and history file of the user are modified in the template; if the authentication is unsuccessful, another instance of *hpwd*′ is generated from the instruction table in a similar way but with some error correction, and the authentication is tried again. If the authentication does not succeed within a fixed number of error-correction iterations, the authentication finally fails. The authors claim that the hardened password itself can be used as an encryption key. A weakness of this work is that it only adds about 15 bits of entropy to the passwords, thus making them only marginally more secure. However, in their subsequent publications [21]–[23], Monrose *et al.* made some minor modifications to their original scheme, applied it to voice biometrics (which is more distinctive than keystroke biometrics), and were eventually able to generate cryptographic keys of up to 60 bits, which although much higher than the 15 bits achieved in their earlier work, is still quite low for most security applications. One of the strengths of Monrose *et al.*'s work is that they have demonstrated the practicality of their algorithm through experiments. The authors even implemented their scheme on a resource-constrained device (Compaq's off-the-shelf IPAQ Personal Digital Assistant) [23]. Different applications can use different cryptographic keys for a person (or the same application can change the cryptographic key upon reenrollment) by using different content (a different typed or uttered password).

Tuyls *et al.* [24], [25] assume that a noise-free template X of a biometric identifier is available at the enrollment time and use this to enroll a secret S to generate a helper data W. Assume that each dimension (of a multidimensional template) is quantized at q resolution levels. In each dimension, the process of obtaining W is akin to finding residuals that must be added to X to fit to odd or even grid quantum depending upon whether the corresponding S bit is zero or one. At decryption time, the (noise-prone) biometric template Y is used to decrypt W to obtain a decrypted message S', which is approximately the same as S. In each dimension, the process of decryption guesses whether a particular bit of secret S is zero or one, depending upon whether the sum of Y and W resides in an even or odd quantum of the corresponding dimension. It is hoped that the relatively few errors in S' can be corrected using error-correction techniques. The proposed technique assumes that the biometric representations are completely aligned and that noise in each dimension is relatively small compared to the quantization Q. Due to variability in the biometric identifier, different W's may be generated for the same message S. The authors prove that very little information is revealed from W by appropriately tuning the quantization scheme with respect to the measurement noise.

In their "fuzzy commitment" scheme [26], Juels and Wattenberg generalized and significantly improved Davida *et al.*'s methods [10], [17] to tolerate more variation in the biometric characteristics and to provide stronger security. In the fuzzy commitment scheme, the user at the enrollment time selects a secret message C. Let d denote the difference vector between the user biometric key X and C. The encrypted message (which is considered as a *fuzzy commitment*) then consists of d and $y = \text{hash}(C)$, where hash is a one-way hash function such as SHA-1 [15]. At the decrypting end, with biometric representation Y, $Y + d$ is used to decode the nearest codeword C'. Again, with the help of error-correcting techniques, it is hoped that the error in C' can be corrected to obtain the original message C. The authors acknowledge that one of the major shortcomings of the fuzzy commitment scheme is that it requires the biometric representations X and Y to be ordered so that their correspondence is obvious. In order to overcome this difficulty, Juels and Sudan [16] proposed a *fuzzy vault* scheme. Let us consider what happens at enrollment time in the fuzzy vault scheme. In this scheme, the secret message C to be transmitted is embedded in a (say, single variable x) polynomial $P(x)$ as its coefficients. The polynomial value $y = P(x)$ can be computed for different values of x. Each value of x can be chosen such that it represents a component of biometric representation, A. Let us call the set of pairs of values (x, y) lying on $P(x)$ as the genuine set G. Let us introduce a set of extraneous points (x', y') called N. The union of G and N constitutes the encrypted message. At the decrypting end, it is hoped that a biometric representation B which substantially matches A can help determine most of the genuine points G with little contamination from N. The recovered set of points R is then used for a polynomial fitting exercise to recover C'. With the help of error-correction schemes, it is hoped that the errors in C' can be corrected to obtain the transmitted message C. Juels and Sudan [16] prove the security of the fuzzy vault scheme in an infor-

Table 2
Comparison of Various Biometrics-Based Key Generation and Key Release Algorithms Based on the Perception of the Authors

Algorithm	Biometric (representation)	Classification	Privacy protection	Practicality	Sensitivity to invariance	Security
Soutar et al.	Fingerprint (image)	R	H	M	H	U
Davida et al.	Iris (IrisCode)	G	H	H	L	U
Monrose et al.	Keystroke, Voice	G	H	H	H	M
Linnartz and Tuyls	No evaluation	G	H	L	L	H
Juels and Sudan	No evaluation	G	H	H	L	H
Clancy et al.	Fingerprint (minutiae)	G	H	H	M	H

mation-theoretic sense. Although the authors specifically mention application of their scheme to biometric keys, it is not clear how robust the algorithm is to typical variations in the biometric signal. Further, although this scheme takes into account unordered feature representations, it is not obvious how this algorithm will handle the alignment requirements of the feature representations.

Clancy *et al.* [27] propose a "fingerprint vault" based on the scheme of Juels and Sudan [16]. At the enrollment time, multiple (typically five) fingerprints of users are acquired. The fingerprint representation (minutiae position) is extracted from each fingerprint. Correspondence between the feature points (minutiae) extracted from the multiple prints is established based on a bounded nearest-neighbor algorithm. That is, when different prints of a finger are overlaid on top of each other, the minutiae in one print which are within a close spatial proximity of minutiae in other print are considered as the same ("corresponding"). Thus, corresponding minutia form clusters and these clusters are used to estimate the variance of minutia location (d). The minutia for which the correspondence is found in at least a predetermined (typically two) set of prints constitutes the effective feature representation (locking set G). Given the fingerprint impression size and d, they add the maximal number of random (chaff) points (N) to the feature representation that are at least d distance away from all the other feature points. As in Juels and Sudan's work [16], the union of G and N constitutes the abscissa of the encoded message; the ordinates are determined by the polynomial embedding of the secret to be shared. Unlike Juels and Sudan [16], Clancy *et al.* [27] propose a concrete prescription for the degree of polynomial for fingerprint domain. At the decrypting end, given a user fingerprint, the features are extracted. The features are used to find the corresponding points within the encoded message using the bounded nearest-neighbor algorithm based on abscissa alone. The corresponding ordinates with encoded message are fed into Reed Solomon error-correcting codes to figure out the encoded polynomial correctly. The strength of this work is that it concretely describes the fuzzy vault implementation in the fingerprint domain and concludes that it is possible to achieve a security (FMR) of 69 bits at a false negative rate (FNMR) of about 20%–30%. One of the major shortcomings of the work is that, like [16], it assumes the fingerprints are prealigned; it is not clear how many fingers were involved in their experiment or whether they captured the fingerprint impressions under realistic conditions.

In summary, the published literature primarily attempts to address the issue of how biometric-based key schemes should handle the variability in the biometric representation. The most promising approaches in the literature tolerate only a limited amount of variation in the biometric data and offer relatively few insights into practical feasibility of the proposed solution. In the next section, we attempt to provide a feel for the biometric variability in realistic data by objectively quantifying the variability parameters in the domain of fingerprints-based user authentication and discussing its practical implications. In Table 2, we provide a comparison of various algorithms: Soutar *et al.* [12]–[14], Davida *et al.* [10], [17], Monrose *et al.* [20]–[23], Linnartz and Tuyls [24], [25], Juels and Sudan [16], and Clancy *et al.* [27]. The third column in Table 2 indicates the key release (R) or key generation (G) classification. Practicality deals with the complexity of the algorithm. The sensitivity of a scheme was assessed based on our perception of whether the algorithm can tolerate realistic variations in the biometric signal such as noise, variable-length representation, unordered representation, unaligned representation, etc. In the last four columns, high, medium, and low are denoted by H, M, and L, respectively. Rigorous security analysis for the first two algorithms has not been provided (U).

Table 3
Statistics for Minutiae Distributions

Set	Mean	Standard Deviation
Original minutiae	41.7	11.8
Matching minutiae	37.9	11.9
Added/missed minutiae	4.1	5.0

IV. Experiments

Using fingerprint domain as a representative biometric, we first quantify the variations in the fingerprint identifiers. We used a database (denoted as GT henceforth) consisting of 450 mated fingerprint pairs collected in a realistic setting and acquired in multiple sessions with a DBI optical sensor with 500 dots per inch (dpi) resolution. A human expert identified the minutiae position/orientation in all of the images. Further, the expert visually found the minutiae correspondence information between minutiae of every mated pair. Table 3 shows the statistics of minutiae distributions for three sets: total number of minutiae in the images, number of matching minutiae in the images, and the number of minutiae added to/missed from the originals. Note that the missing and added minutiae may eliminate some possibilities for using minutiae representation as keys, since even if all of the translational, rotational, and nonlinear distortions in the prints are eliminated, the representations for template and query will not be the same. Using the features identified by the expert, we computed the optimal (in the least-squares sense) rigid transformation (e.g., rotation and translation) between the mated pairs of prints and estimated the variation in the minutiae positions among the *registered* prints using optimal transformation. Table 4 summarizes the (in)variance information in real fingerprints using a perfect feature extractor (e.g., human expert). It is easy to note that to accommodate *almost all* prints, a significant variation needs to be tolerated even within the registered prints. Since the biometric features (minutiae) are identified by a human expert (and not an automatic program that can introduce measurement errors), the statistics given here form a lower bound for the biometric variance.

Let us now use the ground truth marked minutiae in the fingerprints to estimate the performance of a state of the art biometric cryptosystem, e.g., Clancy *et al.*'s [27] implementation of Juels and Sudan's fuzzy vault algorithm [16] for the fingerprint key generation based on minutiae representations. Let us note up front that this key generation scheme assumes prealigned representation and the algorithm is almost of no use if a significant transformation (e.g., elastic distortion) exists between the query and template prints. Let us restrict ourselves to a 512×512 fingerprint image at 512 dpi (i.e., $p = 509$, where p is a sufficiently large prime number such that field F_{p^2} can encode the message as coefficients of some degree k polynomial).

The pixel locations can be encoded in 18-bit numbers. Suitable Reed Solomon error-correction codes can be generated from $509^2 - 1 = 259\,080$ coefficients, where each coefficient holds 17.98 bits. For encoding a 128-bit key, one would require at least eight coefficients and, thus, the degree of polynomial should be $k = 8$. Let us assume the size of minutiae dispersion to be conservatively $d = 15$ pixels [28] for *prealigned* mated fingerprints. Let the number of minutiae required to be submitted by the user be $t = 40$ (e.g., average number of minutiae per print). Following the prescription in [27], at a packing density of $\rho = 0.45$, the vault size will be

$$r = \frac{4\rho p^2}{\pi d^2} \approx 660 \tag{1}$$

randomly packed points. The complexity of the brute force attack is given by

$$C_{\mathrm{bf}}(r,t,k) = \binom{r}{\delta}\binom{t}{\delta}^{-1} \tag{2}$$

where δ refers to the minimum number of correct points that need to be decoded for successfully retrieving the secret key, where

$$\delta \geq \left\lceil \frac{\log \frac{1}{3}p^{2k}}{\log \frac{kp^2}{r}} \right\rceil. \tag{3}$$

The probability of FNMR due to the vault is given by

$$P_e \geq \sum_{i=\delta}^{t} \binom{t}{i} \exp\left(\frac{-\rho p^2}{2\pi r\sigma^2}\right)^i \left(1 - \exp\left(\frac{-\rho p^2}{2\pi r\sigma^2}\right)\right)^{(t-i)}. \tag{4}$$

FMR(δ) for the fingerprint matcher [29] is computed as the probability of δ or more minutiae to match in a nonmatching pair of fingerprints by the matcher. FNMR(δ) is similarly computed by estimating the probability of the matcher to match fewer than δ minutiae in a mating fingerprint pair. Note that the performance of the matcher used in our experiments [29] is not as good as those reported in the literature more recently [30].

If one considers vault and fingerprint matching as a conjunctive system, the FMR of the resultant system will be (simplistically) a product of the individual FMRs and the effective FNMR of the system will be sum of the individual FNMRs. Using these parameters, we have plotted the receiver operating characteristic (ROC) curves of the biometric-based cryptosystem of both, when implemented as a key release as well as a key generation method (see Fig. 8).

Note that the number of feasible operating points for the key generation system is limited (as noted qualitatively by Juels and Sudan [16]). Also note that the $\delta = 5$ line in Fig. 8 shows how the operating points of the two systems are related when the system requires that at least five minutiae correspondences are necessary. It appears that the key generation system is able to make the system more resistant to attacks with almost no deterioration in the FNMR performance. While this observation may be true for a naive brute force attack, it is easy to see that using fingerprint models of the representative dataset, a clever attacker should be able to bring the FMR of the attack very close to the corresponding

Table 4
A Summary of the Distributions

Quantity		Mean	Standard Deviation
Translation before alignment (pixels)	Minimum difference	30.7	33.3
	Mean difference	42.1	35.1
	Maximum difference	52.1	37.2
Rotation before alignment (degrees)	Minimum difference	0.1	1.1
	Mean difference	6.5	3.3
	Maximum difference	22.4	21.2
Best transform (translation) (pixels)		42.4	35.8
Best transform (rotation) (degrees)		3.3	4.1
Translation after alignment (pixels)	Minimum difference	0	0
	Mean difference	4.1	2.3
	Maximum difference	12.4	6.7
Rotation after alignment (degrees)	Minimum difference	0.4	0.5
	Mean difference	5.8	1.6
	Maximum difference	21.3	20.8

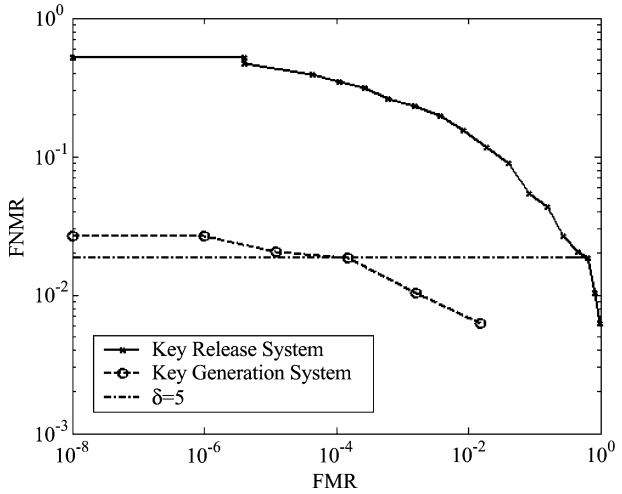

Fig. 8. ROC curves estimated using the minutiae matcher described in [29] in a key release and key generation [27] algorithm.

point on the key release system. In other words, both systems are equivalent when no restriction is placed on the types of attacks undertaken by the attacker. Further, the entire system is as robust as its weakest link (e.g., biometric matcher) and the overall system vulnerability can only be improved by making the biometric matcher as accurate as possible.

Operationally, however, the key generation technology suffers from many limitations in terms of: 1) requiring prealigned representations; 2) having a limited choice of flexible operating points; 3) resulting in a higher complexity of the overall system; and 4) requiring more intensive computation. The key release system, on the other hand, only requires a trusted host and a secure channel.

V. Conclusion

Biometrics are an *essential* component of any identity-based security system because no other technology can replace the requisite functionality of "identifying the authorized person based on their intrinsic distinctive traits." In this paper, we have presented the research issues related to incorporating biometrics into a cryptographic system in the context of DRM applications. While incorporation of biometrics for effective user authentication within a cryptographic system makes intuitive sense, there are a number of challenges involved in combining biometrics into a cryptographic system, primarily due to dramatic variations in the representations of a biometric identifier and due to imperfect nature of biometric feature extraction and matching algorithms. Existing research in biometric cryptosystems is focused on the brute force complexity of adversarial attacks. Within this limited context, simple methods based on biometric authentication to *release* a biometric key are not useful in many cryptographic applications because they involve sharing unencrypted biometric information over an insecure channel; it implies that such applications would require generation of biometric keys to release the transmitted secret encrypted message. While researchers have proposed many interesting and clever ideas for *generation* or *binding* of biometric keys, we believe that many critical problems peculiar to the biometric domain have not been satisfactorily solved. For example, although the complexity of successful intrusion can be made formidable, these systems can, in practice, be defeated using relatively simple strategies. A naive attack on a biometric system could be launched by successively presenting biometric samples from a representative population (either synthetically generated or actual samples) and the success of the attack is likely to be bounded by the weakest link in the security chain, i.e., operating point of the biometric matcher. In this regard, we believe it is more critical to focus on increasing the accuracy of the individual biometric matcher performance and on devising effective multibiometric strategies to deliver acceptable end-to-end system performance.

Biometrics are not secrets and are not revocable [31]; while revocability and secrecy have been critical requirements of conventional cryptosystem design, one then wonders whether it is possible to design a secure authentication system from the system components which in themselves are neither secrets nor revocable—for example, whether the methods of ensuring liveness of biometric identifiers [8] and challenge–response schemes [8] obviate fraudulent insertion of "stolen" biometric identifiers. Is it possible to nontrivially combine knowledge and biometric identifiers to arrive at key generation/release mechanisms where biometric identifiers are necessary but not sufficient for cryptographic key generation/release? Is it possible to require multiple biometrics to make it increasingly difficult for the attacker to fraudulently insert multiple biometrics into the system? Is it possible to make it unnecessary to revoke/update the cryptographic key in the event of a "stolen biometric"? Exploring challenges in designing such systems is a promising (yet neglected) avenue of research.

When cryptobiometric systems eventually come into practical existence, there is a danger that biometric components may be used as an irrefutable proof of existence of a particular subject at a particular time and place. Mere incorporation of biometrics into a system does not in itself constitute a proof of identity. We need to understand how these foolproof guarantees can be theoretically proved in a deployed cryptosystem and how to institute due processes that will provide both technological and sociological freedom to challenge the premises on which nonrepudiability is ascertained.

REFERENCES

[1] US global piracy losses estimated at $9.2B in 2002, IDG News Service. (2003, February 14). [Online]. Available: http://www.computerworld.com/securitytopics/security/cybercrime/story/0,10801,78545,00.html

[2] Four out of every ten software programs are pirated worldwide, Business Software Alliance. (2002, June 10). [Online]. Available: http://global.bsa.org/usa/press/newsreleases//2002-06-10.1129.phtml

[3] Advanced encryption standard (AES), Federal information processing standards publication 197, National Institute of Standards and Technology. (2001). [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[4] W. Stallings, *Cryptography and Network Security: Principles and Practices*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2003.

[5] D. V. Klein, "Foiling the cracker: a survey of, and improvements to, password security," in *Proc. 2nd USENIX Workshop Security*, 1990, pp. 5–14.

[6] I. Armstrong. (2003, June) Passwords exposed: Users are the weakest link. *SC Mag.* [Online]. Available: http://www.scmagazine.com/features/index.cfm?fuseaction=FeatureDetails&newsUID=43b97d8d-3605-4a18-b9bc-03c063008dc8&newsType=Features

[7] A. K. Jain, R. Bolle, and S. Pankanti, Eds., *Biometrics: Personal Identification in Networked Society*. Norwell, MA: Kluwer, 1999.

[8] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. New York: Springer-Verlag, 2003.

[9] J. L. Wayman, "Fundamentals of biometric authentication technologies," *Int. J. Image Graph.*, vol. 1, no. 1, pp. 93–113, 2001.

[10] G. I. Davida, Y. Frankel, and B. J. Matt, "On enabling secure applications through off-line biometric identification," in *Proc. 1998 IEEE Symp. Privacy and Security*, pp. 148–157.

[11] N. Ratha, J. Connell, and R. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Syst. J.*, vol. 40, no. 3, pp. 614–634, 2001.

[12] C. Soutar, D. Roberge, S. A. Stojanov, R. Gilroy, and B. V. K. Vijaya Kumar, "Biometric encryption using image processing," in *Proc. SPIE, Optical Security and Counterfeit Deterrence Techniques II*, vol. 3314, 1998, pp. 178–188.

[13] ——, "Biometric encryption—enrollment and verification procedures," in *Proc. SPIE, Optical Pattern Recognition IX*, vol. 3386, 1998, pp. 24–35.

[14] ——, "Biometric encryption," in *ICSA Guide to Cryptography*, R. K. Nichols, Ed. New York: McGraw-Hill, 1999.

[15] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed. New York: Wiley, 1995.

[16] A. Juels and M. Sudan, "A fuzzy vault scheme," in *Proc. IEEE Int. Symp. Information Theory*, A. Lapidoth and E. Teletar, Eds., 2002, p. 408.

[17] G. I. Davida, Y. Frankel, B. J. Matt, and R. Peralta, "On the relation of error correction and cryptography to an offline biometric based identification scheme," in *Proc. Workshop Coding and Cryptography (WCC'99)*, pp. 129–138.

[18] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1148–1161, Nov. 1993.

[19] W. W. Peterson and E. J. Weldon, *Error Correcting Codes*. Cambridge, MA: MIT Press, 1988.

[20] F. Monrose, M. K. Reiter, and S. Wetzel, "Password hardening based on keystroke dynamics," in *Proc. 6th ACM Conf. Computer and Communications Security*, 1999, pp. 73–82.

[21] F. Monrose, M. K. Reiter, Q. Li, and S. Wetzel, "Using voice to generate cryptographic keys," in *Proc. 2001: A Speaker Odyssey, Speaker Recognition Workshop*, 2001, pp. 237–242.

[22] ——, "Cryptographic key generation from voice," in *Proc. 2001 IEEE Symp. Security and Privacy*, pp. 202–213.

[23] F. Monrose, M. K. Reiter, Q. Li, D. P. Lopresti, and C. Shih, "Toward speech-generated cryptographic keys on resource constrained devices," in *Proc. 11th USENIX Security Symp.*, 2002, pp. 283–296.

[24] J.-P. Linnartz and P. Tuyls, "New shielding functions to enhance privacy and prevent misuse of biometric templates," in *Proc. 4th Int. Conf. Audio- And Video-Based Biometric Person Authentication*, 2003, pp. 393–402.

[25] E. Verbitskiy, P. Tuyls, D. Denteneer, and J. P. Linnartz, "Reliable biometric authentication with privacy protection," presented at the SPIE Biometric Technology for Human Identification Conf., Orlando, FL, 2004.

[26] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proc. 6th ACM Conf. Computer and Communications Security*, G. Tsudik, Ed., 1999, pp. 28–36.

[27] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcard-based fingerprint authentication," in *Proc. ACM SIGMM 2003 Multimedia, Biometrics Methods and Applications Workshop*, pp. 45–52.

[28] S. Pankanti, S. Prabhakar, and A. K. Jain, "On the individuality of fingerprints," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 1010–1025, Aug. 2002.

[29] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An identity authentication system using fingerprints," *Proc. IEEE*, vol. 85, pp. 1365–1388, Sept. 1997.

[30] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2002: Fingerprint Verification Competition," in *Proc. Int. Conf. Pattern Recognition (ICPR)*, 2002, pp. 744–747.

[31] B. Schneier, "Biometrics: uses and abuses," *Commun. ACM*, vol. 42, no. 8, p. 136, Aug. 1999.

Umut Uludag (Student Member, IEEE) received the B.S. and M.S. degrees in electrical and electronics engineering from Bogazici University, Istanbul, Turkey, in 1999 and 2001, respectively. He is currently working toward the Ph.D. degree in the Department of Computer Science and Engineering, Michigan State University, East Lansing.

He was a researcher at the Marmara Research Center from 1999 to 2001. His research interests include biometrics, pattern recognition, digital security, watermarking, multimedia, image processing and computer vision.

Mr. Uludag is a Member of the IEEE Computer Society.

Sharath Pankanti (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from Michigan State University, East Lansing, in 1995.

From 1995 to 1999, he worked on the Advanced Identification Solutions Project dealing with reliable and scalable fingerprint recognition systems. He is currently with the Exploratory Computer Vision and Intelligent Robotics Group, IBM T. J. Watson Research Center, Yorktown Heights, NY. For the past few years he has been working on analysis and interpretation of video depicting human activities. His research interests include biometrics, pattern recognition, computer vision, and human perception.

Salil Prabhakar (Member, IEEE) received the B.Tech. degree in computer science and engineering from the Institute of Technology, Banaras Hindu University, Varanasi, India, in 1996 and the Ph.D. degree in computer science and engineering from Michigan State University, East Lansing, in 2001.

From 1996 to 1997, he worked with IBM India as a software engineer. He currently leads the Algorithms Research Group at DigitalPersona Inc., Redwood City, CA, where he works on fingerprint-based biometric solutions. He is coauthor of more than 25 technical publications and coauthor of *Handbook of Fingerprint Recognition* (New York: Springer-Verlag, 2003), and has two patents pending. His research interests include pattern recognition, image processing, computer vision, machine learning, biometrics, data mining, and multimedia applications.

Anil K. Jain (Fellow, IEEE) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1969 and the M.S. and Ph.D. degrees in electrical engineering from Ohio State University, Columbus, in 1970 and 1973, respectively.

He is currently a University Distinguished Professor in the Departments of Computer Science and Engineering and Electrical and Computer Engineering at Michigan State University, East Lansing. He was the Department Chair of the Computer Science and Engineering Department between 1995 and 1999. Several of his papers have been reprinted in edited volumes on image processing and pattern recognition. He is coauthor of *Handbook of Fingerprint Recognition* (New York: Springer-Verlag, 2003). He holds six patents in the area of fingerprint matching. His research interests include statistical pattern recognition, exploratory pattern analysis, Markov random fields, texture analysis, three-dimensional object recognition, medical image analysis, document image analysis, and biometric authentication.

Dr. Jain is a Fellow of ACM and the International Association of Pattern Recognition (IAPR). He has received a Fulbright Research Award, a Guggenheim Fellowship and the Alexander von Humboldt Research Award. He delivered the 2002 Pierre Devijver Lecture sponsored by the IAPR. He received the best paper awards in 1987 and 1991, and received certificates for outstanding contributions in 1976, 1979, 1992, 1997, and 1998 from the Pattern Recognition Society. He also received the 1996 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award.