

# Biometric Encryption Chapter from the Encyclopedia of Biometrics

*The following is a chapter on Biometric Encryption excerpted from the Springer Encyclopedia of Biometrics.*

**Ann Cavoukian and Alex Stoianov**

Office of the Information and Privacy Commissioner, Toronto, Ontario, Canada

## Synonyms

Biometric cryptosystem; Biometric key generation; Biometric locking; Fuzzy extractor; Secure sketch

## Definition

Biometric Encryption (BE) is a group of emerging technologies that securely bind a digital key to a biometric or generate a digital key from the biometric, so that no biometric image or template is stored. What is stored is the BE template otherwise known as a “bio-metrically encrypted key” or “helper data.” As a result, neither the digital key nor the biometric can be retrieved from the stored BE template. BE conceptually differs from other systems that encrypt biometric images or templates using conventional encryption, or store a cryptographic key and release it upon successful biometric authentication. With BE, the digital key is recreated only if the correct biometric sample is presented on verification. The output of BE verification is either a digital key or a failure message. This “encryption/decryption” process is fuzzy because of the natural variability of biometric samples. Currently, any viable BE system requires that biometric-dependent helper data be stored.

## Introduction

Biometric technologies may add a new level of authentication and identification to applications, but are not, however, without their risks and challenges. There are important technological challenges such as accuracy, reliability, data security, user acceptance, cost, and interoperability, as well as challenges associated with ensuring effective privacy protections. Some common security vulnerabilities of biometric systems include:

Spoofing; replay attacks; substitution attacks; tampering; masquerade attacks (creating a digital “artifact” image from a fingerprint template so that this artifact, if submitted to the system, will produce a match); Trojan horse attacks; and overriding Yes/No response (which is an inherent flaw of existing biometric systems).

In addition to the security threats that undermine the reliability of biometric systems, there are a number of specific privacy concerns with these technologies:

- function creep (i.e., unauthorized secondary uses of biometric data)
- expanded surveillance, tracking, profiling, and potential discrimination (biometric data can be matched against samples collected and stored elsewhere and used to make decisions about individuals)
- data misuse (data breach, identity theft, and fraud)
- negative personal impacts of false matches, non-matches, system errors, and failures (the consequences of system anomalies, especially in large-scale systems, often fall disproportionately on individuals, normally in the form of inconveniences, costs, and stigma)
- insufficient oversight, accountability, and openness in biometric data systems
- potential for collection and use of biometric data without knowledge, consent, or personal control

These types of risks threaten user confidence, which leads to a lack of acceptance and trust in biometric systems.

Biometric Encryption (BE) technologies can help to overcome the prevailing “zero-sum” mentality involved in traditional biometrics, namely, that adding privacy to authentication and information systems weakens security. With BE, it is possible to enhance both privacy and security in a positive-sum model.

## What is Biometric Encryption (BE)?

The concept of Biometric Encryption (BE) was first introduced in the mid-90s by G. Tomko et al. [1]. For more information on BE and related technologies, see the review papers in [2–4].

Biometric Encryption is a process that securely binds a digital key to a biometric or generates a key from the biometric. In essence, the key is “encrypted” with the biometric, and the resulting biometrically encrypted key, also called BE template or helper data, is stored. The digital key can be “decrypted” on verification if a correct biometric sample is presented. This “encryption/decryption” process is fuzzy by nature, because the biometric sample is different each time, unlike an encryption key in conventional cryptography. A major technological challenge is to have the same digital key recreated despite the natural variations in the input biometrics.

After the digital key is recreated on verification, it can be used as the basis for any physical or logical application. The most obvious use is in a conventional cryptosystem where the key serves as a password and may generate, for example, a pair of Public and Private keys. It should be noted that BE itself is not a cryptographic algorithm. The role of BE is to replace or augment vulnerable password-based schemes with more secure and more convenient biometrically managed keys.

BE should not be mistaken for other systems that encrypt biometric images or templates using conventional encryption, or store a cryptographic key in a trusted token/device and subsequently release it upon successful biometric verification (i.e., after receiving Yes response). However, BE is related to another family of privacy-enhancing technologies called ► Cancelable Biometrics (CB) (N. Ratha et al. in [3]; see also the Encyclopedia article on “Cancellable Biometrics”). CB applies a transform (preferably, noninvertible) to a biometric image or template and matches the CB templates in the transformed domain. This transform is usually kept secret. Unlike BE, the CB system does not bind or generate a key. CB remains inherently vulnerable to overriding Yes/No response and to a substitution attack.

There are two BE approaches: key binding, when an arbitrary key (e.g., randomly generated) is securely bound to the biometric, and key generation, when a key is derived from the biometric. Both approaches usually store biometric dependent helper data. Some BE schemes (e.g., Fuzzy Commitment[5], Fuzzy Vault [6]) can equally work in both key generation and key binding mode; the key generation is also called “secure sketch” or “fuzzy extractor” as defined in [7]. Secure sketch implies that the enrolled biometric template will be recovered on verification when a fresh biometric sample is applied to the helper data (i.e., the enrolled template itself or a string derived from it, e.g., by hashing the template, serves as a digital key). Note, however, that this “key” is not something inherent or absolute for this particular biometric; it will change upon each re-enrolment. The size of the key space for the secure sketch is defined by the intraclass variations of the biometric as opposed to the key binding approach.

In the key binding mode, as illustrated in Fig.1, the digital key is randomly generated on enrollment so that neither the user nor anybody else knows it. The key itself is completely independent of biometrics, and therefore, can always be changed or updated. After a biometric sample is acquired, the BE algorithm securely and consistently binds the key to the biometric to create a biometrically encrypted key. The BE template provides privacy protection and can be stored either in a database or locally (smart card, token, laptop, cell phone, etc.). At the end of the enrollment, both the key and the biometric are discarded.

On verification, the user presents his or her fresh biometric sample, which, when applied to the legitimate BE template, will let the BE algorithm recreate the same key. At the end of verification, the biometric sample is discarded once again. The BE algorithm is designed to account for acceptable variations in the input biometric. On the other hand, an impostor whose biometric sample is different enough will not be able to recreate the key.

Many BE schemes also store a hashed value of the key (not shown in Fig. 1) so that a correct key is released from the BE system only if the hashed value obtained on verification is exactly the same. Also, good practice would be not to release the key, but rather, another hashed version of it for any application. This hashed version can in turn serve as a cryptographic key. With this architecture, an attacker would not be able to obtain the original key outside the BE system. Likewise, the biometric image/template should not be sent to a server; the BE verification should be done locally in most scenarios.

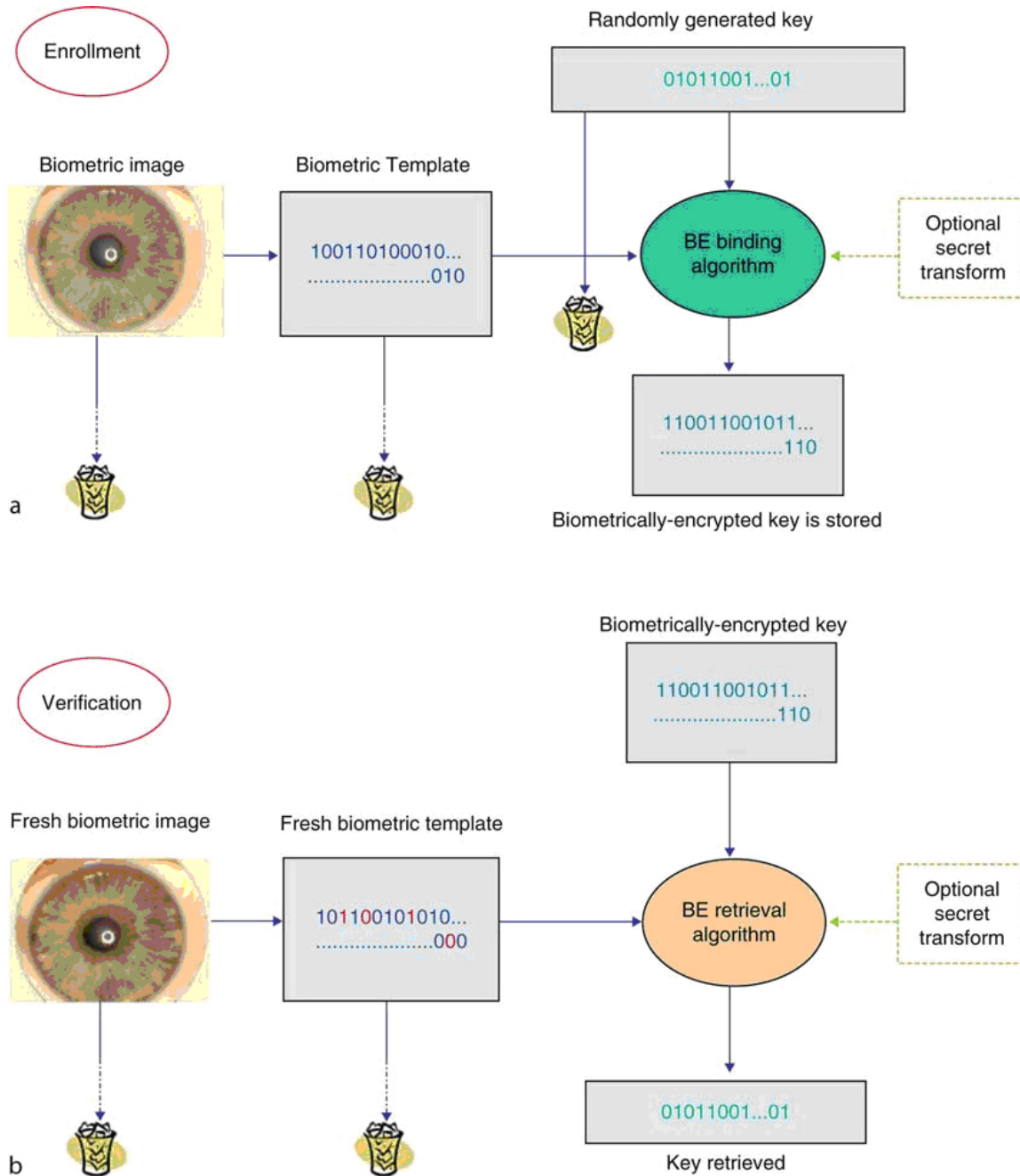


Figure 1. High level diagram of a Biometric Encryption process in a key binding mode. a) Enrollment; b) Verification

An important part of most BE algorithms is an Error Correcting Code (ECC). ECCs are used in communications, for data storage, and in other systems where errors can occur. Biometric Encryption is a new area for the application of ECC. For example, a binaryblock ECC, which is denoted  $(n, k, d)$ , encodes  $k$  bits with  $n > k$  bits by adding some redundancy. Those  $n$ -bit strings are called codewords; there are  $2^k$  of them in total, where  $k$  is the key length. The minimum distance (usually a Hamming distance is implied) between the codewords is  $d$ . If, at a later stage (in case of BE, on verification), the errors occur, the ECC is guaranteed to correct up to  $(d-1)/2$  bit errors among  $n$  bits. Ideally, the

legitimate users will have a number of errors within the ECC bound so that the ECC will decode the original codeword, and hence, the digital key. On the other hand, the impostors will produce an uncorrectable number of errors, in which case the ECC (and the BE algorithm as a whole) will declare a failure. In practice, BE, like any biometric system, has both false rejection and false acceptance rates (FRR and FAR). Note that BE does not use any matching score; instead, the FRR/FAR tradeoff may be achieved in some cases by varying the parameters of the BE scheme. Some ECCs may work in a soft decoding mode, that is, the decoder always outputs the nearest codeword, even if it is beyond the ECC bound. This allows achieving better error-correcting capabilities.

To improve the security of a BE system, an optional “transform-in-the-middle”(shown in the dashed square in Fig. 1) may be applied. Preferably, the transform should be non-invertible and kept secret. One of the ways would be employing a randomization technique, such as Biohashing [8] or “salting” in more general terms [2]. The transform can be controlled with the user’s password or can be separated from the rest of the helper data by storing it on a token or a server.

## Advantages and Possible Applications of BE

BE technologies can enhance both privacy and security in the following ways:

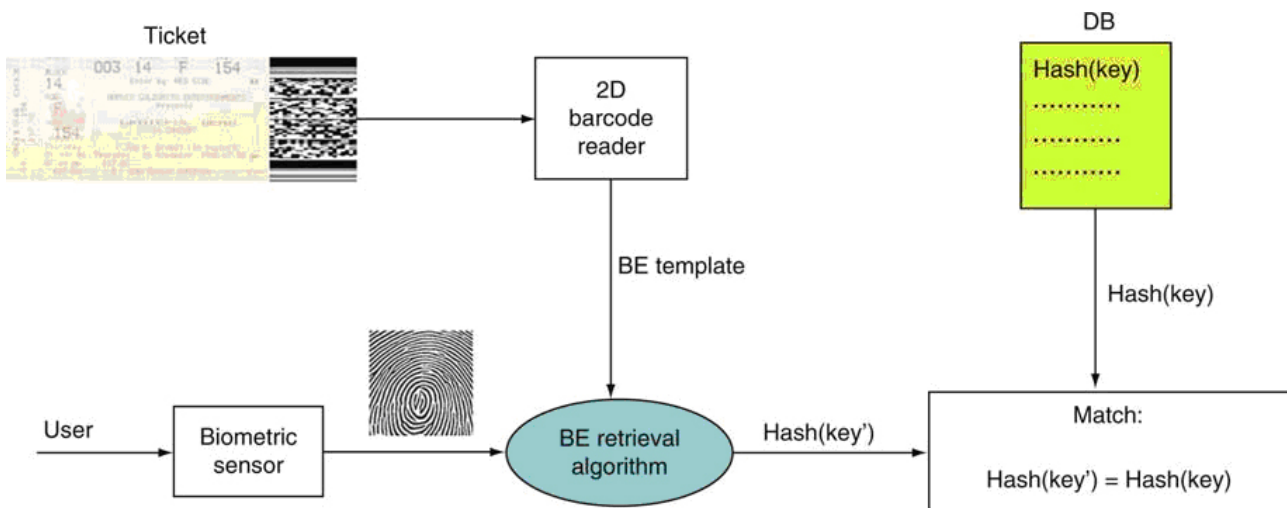
- There is no retention of biometric image or conventional biometric template, and they cannot be recreated from the stored helper data.
- They are capable of multiple identifiers: a large number of BE templates for the same biometric can be created for different applications.
- The BE templates from different applications cannot be linked.
- The BE template can be revoked or canceled.
- They can be easily integrated into conventional cryptosystems, as the passwords are replaced with longer digital keys, which do not have to be memorized.
- They provide improved authentication and personal data security through a stronger binding of user biometric and system identifier.
- The BE systems are inherently protected from substitution attack, tampering, Trojan horse attack, overriding Yes/No response, and less susceptible to masquerade attack.
- They are suitable for large-scale applications, as the databases will store only untraceable, yet sufficient, information to verify the individual’s claim.

These features embody standard fair information principles, providing user control, data minimization, and data security.

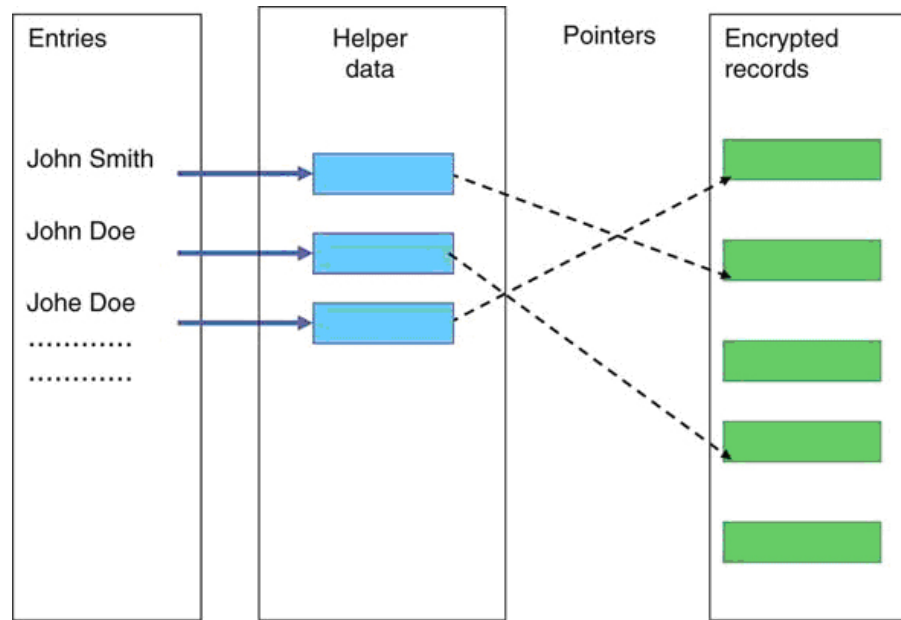
As such, BE technologies put biometric data firmly under the exclusive control of the individual, in a way that benefits the individual and minimizes the risk of function creep and identity theft. They provide a foundation for building greater public confidence, acceptance, and use, and enable greater compliance with privacy and data protection laws.

Possible applications and uses of Biometric Encryption include:

- Biometric ticketing (Fig.2) for events
- Biometric boarding cards for travel
- Drug prescriptions
- Three-way check of travel documents
- Identification, credit, and loyaltycard systems
- Anonymous databases (Fig.3), that is, anonymous (untraceable) labeling of sensitive records (medical, financial)
- Consumer biometric payment systems
- Remote authentication via challenge-response scheme
- Access control (physical and logical)
- Personal encryption products (i.e., encrypting files, BE Technologies drives, e-mails, etc.)
- Local or remote authentication of users to access files held by government and other various organizations



**Figure 2.** Biometric ticketing. A BE template is stored on a ticket as a 2D bar code, and a database stores the hashed value of a key, Hash (key), for each enrolled user. The key and the ticket are used only for this particular application. On a verification terminal: (i) The user presents his ticket to the system which reads in the BE template from the bar code; (ii) The live biometric sample is taken; (iii) The system applies the biometric to the BE template to retrieve the key; (iv) Hash (key') is sent to the database where it is compared to the stored version, Hash (key).



**Figure 3.** Anonymous database controlled by Biometric Encryption. The database contains anonymous encrypted records, e.g., medical files. The cryptographic keys and the links to the entries, which may be users’ names or pseudonyms, are controlled by BE. After the user enters his pseudonym, the associated BE template (helper data) is retrieved and applied to the user’s biometric. If BE successfully recovers the user’s digital key, it will recreate the pointer to the anonymous record and the encryption key to decrypt the record.

## BE Technologies

The following are core BE schemes. The more detailed, files held by government and other various up-to-date overviews of BE technologies are presented organizations in [2,4].

### Mytec1

This is the first BE scheme [1]. It was developed using optical processing, but can also be implemented digitally. The key is linked to a predefined pattern,  $s(x)$ , which is a sum of several delta-functions. Using  $s(x)$  and a fingerprint,  $f(x)$ , one can create a filter,  $H(u) \propto S(u)/F(u)$ , in Fourier domain ( $S(u)$  and  $F(u)$  are the Fourier transforms of  $s(x)$  and  $f(x)$ ). It is difficult to obtain either  $S(u)$  or  $F(u)$  from the stored filter  $H(u)$ . On verification, if a correct fingerprint,  $F'(u) \approx F(u)$ , is applied to the filter, it will reconstruct a correct output pattern,  $s'(x) \approx s(x)$  so that the key will be regenerated from the locations of the output correlation peaks. Unfortunately, this scheme turned out to be impractical in terms of providing sufficient accuracy and security.

### Mytec2

This is the first practical BE scheme [9]. Unlike Mytec1, it retains phase-only parts of  $S(u)$  and  $F(u)$  in the filter,  $H(u)$ . The phase of  $S(u)$  is randomly generated, but not stored anywhere. As a result, the output pattern,  $c(x)$ , is also random. The key, normally 128 bit long, is linked to  $c(x)$  via a lookup table and ECC. The filter,  $H(u)$ , the lookup table, and the hashed key are stored in the helper data.

The system is error tolerant and translation invariant. The published version [9] used a simple repetition ECC, which makes the system vulnerable to several attacks, such as Hill Climbing [10].

However, a closer examination of the Mytec2 scheme shows that if the randomness of  $H(u)$  and  $c(x)$  is preserved on each step of the algorithm, the scheme is a variant of so-called “permutation-based fuzzy extractor” as defined in [7]. Therefore, if a proper ECC (preferably, single block) is used instead of the repetition ECC, the system will be as secure as those types of fuzzy extractors.

(Note that Mytec1 and Mytec2 schemes were originally called “Biometric Encryption”, which was a trademark of Toronto-based Mytec Technologies Inc., now Bioscrypt, a fully-owned subsidiary of L1 Identity Solutions Inc. The trademark was abandoned in 2005.)

## ECC Check Bits

This scheme, which was originally called “private template,” is a secure sketch (i.e., a key generation) [11].

A biometric template itself serves as a cryptographic key. To account for the template variations between different biometric samples, an  $(n, k, d)$  error correcting code is used. A number of  $(n-k)$  bits, called check bits, are appended to the template to map the  $k$ -bit template to an  $n$ -bit codeword. The check bits are stored into the helper data along with the hashed value of the template. The scheme is impractical, since it is required that  $n < 2k$  from the security perspective. Such ECC would not be powerful enough to correct a realistic number of errors for most biometrics, including iris scan.

## Biometrically Hardened Passwords

This technique was developed for keystroke dynamics or voice recognition [12]. A password that the user types or says is fused with a key (via a secret sharing scheme) extracted from a biometric component, thus hardening the password with the biometrics. The technique was made adaptive by updating a “history file” (which is, in fact, helper data) upon each successful authentication. However, the types of biometrics used did not allow for achieving good accuracy numbers.

## Fuzzy Commitment

This is conceptually the simplest, yet the most studied, BE scheme [5] A. Juels in [3]. A biometric template must be in the form of an ordered bit string of a fixed length. A key is mapped to an  $(n,k,d)$  ECC code word of the same length,  $n$ , as the biometric template. The codeword and the template are XOR-ed, and the resulting  $n$ -bit string is stored into helper data along with the hashed value of the key. On verification, a fresh biometric template is XOR-ed with the stored string, and the result is decoded by the ECC. If the codeword obtained coincides with the enrolled one (this is checked by comparing the hashed values), the  $k$ -bit key is released. If not, a failure is declared.

In a “securesketch” (i.e., key generation) mode [7], the enrolled template is recovered from the helper data on verification, if a correct (yet different) biometric sample is presented.

The scheme seems to be one of the best for the biometrics where the proper alignment of images is possible, such as iris scan [13,14] and facerecognition (T. Kevenaar in [3]). For iris, the reported results are  $FRR = 0.47\%$  at  $FAR < 10^{-5}$  for a 140-bit key mapped to 2048-bit code word [13], and  $FRR = 5.6\%$  at  $FAR < 10^{-5}$  (42-bit key) [14] for a poorer quality, yet more realistic, iris database.

## ECC Syndrome

In this spinoff of the Fuzzy Commitment scheme, a so-called ECC syndrome of  $(n-k)$  size is stored in the helper data [7, 2]. On verification, the enrolled template is recovered (i.e., the scheme works in the secure sketch mode).

## Quantization using Correction Vector

This method, which was also called “shielding functions”, is applied to continuously distributed and aligned biometric features (J.-P. Linnartz et al. in [3]). For each feature, a residual is calculated, which is the distance to the center of the nearest even-odd or odd-even interval, depending on the parity of the key bit. The correction vector comprising all the residuals is stored into the helper data. On verification, a noisy feature is added to the residual and is decoded as 1 or 0, if the resulting interval is odd-even or vice versa. The scheme can work with or without (if a noise level is low) a subsequent ECC. In general, storing a correction vector could make the scheme vulnerable to score-based attacks.

## Fuzzy Vault

This is, probably, the only BE scheme that is fully suitable for unordered data with arbitrary dimensionality, such as fingerprint minutiae [6, 15]. A secret message (i.e. a key) is represented as coefficients of a polynomial in a Galois field, for example,  $GF(2^{16})$ . In the most advanced version [15], the 16-bit  $x$ -coordinate value of the polynomial comprises the minutia locations and the angle, and the corresponding  $y$ -coordinates are computed as the values of the polynomial on each  $x$ . Both  $x$  and  $y$  numbers are stored alongside with chaff points that are added to hide real minutiae. On verification, a number of minutiae may coincide with some of the genuine stored points. If this number is sufficient, the full polynomial can be reconstructed

using an ECC (e.g., Reed-Solomon ECC) or Lagrange interpolation. The polynomial reconstruction means that the secret has been successfully decrypted. The scheme works both in the key binding and the key generation (secure sketch) mode. The version of [15] also stores fingerprint alignment information. The best results for fingerprints show  $FRR = 6\% - 17\%$  at  $FAR = 0.02\%$ .

The more secure version of Fuzzy Vault [7] stores high degree polynomial instead of real minutiae or chaff points. However, there are difficulties in the practical implementation of this version.

Unlike other BE schemes, the fuzzy vault actually stores real minutiae, even though they are buried inside the chaff points. This could become a source of potential vulnerabilities [2,4]. The system security can be improved by applying a secret minutiae permutation controlled by a user’s password [2]. This “transform-in-the-middle” approach is applicable to most BE schemes.

## Biohashing (with key binding)

An ordered biometric feature set is transformed into a new space of a lower dimension by generating a random set of orthogonal vectors and obtaining an inner product between each vector and the biometric feature set [8]. The result (called “Biohash”) is binarized to produce a bit string. The random feature vectors are generated from a random seed that is kept secret, for example, by storing it in a token. The key is bound to the Biohash via Shamir secret sharing with linear interpolation, or by using a standard Fuzzy Commitment scheme. Very good FRR/FAR numbers [8] were obtained, however, in an unrealistic “non-stolen token” scenario. Biohashing is referred more often as a CB scheme where Biohashes are matched directly, that is, without the key binding.

## Graph-based Coding

In this generalization of the ECC syndrome scheme, Low Density Parity Check (LDPC) ECCs are used in a graphical representation [16]. LDPC codes, which are the state-of-the-art channel ECCs  $(n, k, d)$ , can be designed with large numbers of  $n$  and  $k$ , and can handle high error rates. This makes them suitable for BE applications. The scheme can be applied to both ordered (e.g., iris) and unordered (e.g., fingerprint minutiae) feature sets. For the latter, a factor graph models the minutiae variability as a movement, an erasure, or an insertion (i.e., spurious generation) of minutiae. The scheme uses a Belief Propagation decoding algorithm and shows promising results.

## Attacks on BE

Despite the fact that many BE schemes have a formal proof of security, they may be vulnerable to low level attacks, such as when an attacker has access to helper data, is familiar with the BE algorithm, and can run the attack offline. By cracking a BE system, the attacker can pursue one or more of the following:

- Obtain the key bound to the biometrics
- Obtain the exact biometric template used on enrollment
- Obtain an approximate version of the template that, nonetheless, would defeat the system (masquerade template)
- Create a masquerade image of the biometrics
- Link BE templates generated from the same biometrics but stored in different databases

The known attacks on BE, as described in [4], are listed in the following paragraphs. Note that CB may also be vulnerable to most of the attacks.

*False Acceptance attack.* This is one of the “brute force” attacks. Offline, the attacker runs an impostor database of about  $FAR^{-1}$  biometric images or templates against the helper data to obtain a false acceptance. The database can be either real or computer-generated, such as *SFinGe*. The image that has generated the false acceptance will serve as a masquerade image.

*Reversing the hash.* This is another “brute force” attack. If a hashed key is stored into the helper data, the attacker may try to cryptographically reverse the hash. This attack should always be made more computationally expensive for an attacker than other attacks.

*Hill Climbing attack* [10]. Based on the knowledge of the algorithm, the attacker derives an intermediate matching score during the verification process, even though the BE algorithm does not use any score. By making small changes in the input impostor’s image or template, the attacker retains the change, if the score increases, or rejects it, if not. After a number of iterations, the attacker may be able to retrieve a key and create a masquerade image/template.

The BE schemes that divide helper data into short chunks of ECC (e.g., a repetition ECC), and the schemes with a correction vector may be especially vulnerable to this and to the Nearest Impostors attack.

*Nearest Impostors attack* [4]. This is another score-based attack. The attacker derives a partial matching score for each ECC chunk (if any) of the helper data and a global intermediate score (like in the Hill Climbing attack). By running a small impostor database against the helper data, the attacker identifies several “nearest impostors”, that is, the attempts with the highest global score, or alternatively, with the highest partial score for a given chunk. By applying a voting technique to the nearest impostors, the attacker retrieves the key bits associated with the chunk. If successful, the attack yields the entire key or at least reduces the search space for the key.

*Using statistics of ECC output* [4]. A small impostor database (with various distortions, rotations, and shifts applied) is run against the ECC chunks of the helper data. The number of appearances of each possible output codeword for all impostor attempts is counted to create a histogram. The codeword corresponding to the histogram maximum is declared a winner.

Using an information leak from helper data. This group of attacks may directly exploit

- Nonrandomness of the helper data [4] (e.g., if clusters in the helper data are identified, the attacker may interconnect the same parity bits)
- Alignment information and minutiae angles in the Fuzzy Vault
- A method for generating the chaff points[17]
- Nonuniformity of the output bits distribution in quantization schemes, etc.

*Re-usability attack* (X. Boyen in [3]). If the same biometric is re-used for different applications and/or keys, the attacker may combine several versions of the helper data to retrieve both the biometric and all the keys. Fuzzy Vault is especially vulnerable to this attack.

Among all BE schemes, it seems that one of the most secure would be a Fuzzy Commitment (or other related fuzzy extractors, such as ECC syndrome) scheme with a single block  $(n, k, d)$  ECC, where  $n$  and  $k$  are large (e.g.,  $n > \sim 1000$ ,  $k > \sim 100$ ). From the security perspective, the amount of any additional side information that is stored (e.g., alignment data) should be kept to a minimum.

The resilience to some of the attacks may be improved by employing the “transform-in-the-middle” approach, especially if the transform is controlled by a password/token.

## Current State of BE

Many different approaches have been developed for BE, but currently few systems have been deployed or implemented into products. Until now, little work has been done to analyze the security of BE systems.

The authors’ consider the following technologies as the state of the art of BE:

- Philips (the Netherlands) priv-ID™ for the face recognition (2D and 3D) and fingerprints (T.Kevenaarin[3])
- Hao et al for iris [13]
- Nandakumar et al (fuzzy vault for fingerprints) [15]
- Draper et al. of Mitsubishi Electric Research Laboratories (U.S.) for iris and fingerprints [16]
- Bringer et al. of Sagem Sécurité (France) for iris [14]
- Genkey (Norway) BioCryptic® for fingerprints (unfortunately, not much information about the technology is available)

The Philips priv-ID™ technology is ready for deployment. It is part of the EU 3D Face project and of the 3-year EUTURBINE project [18]. The latter has been given significant funding and aims at piloting a finger-print-based BE technology at an airport in Greece.

The Genkey BioCryptic1 technology has been deployed for a Rickshaw project in New Delhi (India). Both Philips and Genkey systems can fit the helper data into a 2D bar code.

## BE Challenges

Technologically, BE is much more challenging than conventional biometrics, since most BE schemes work in a “blind” mode (the enrolled image or template are not seen on verification). As BE advances to the next phase of creating and testing a prototype, the following issues need to be addressed:

- Biometric modalities that satisfy the requirements of high entropy, low variability, possibility of alignment, and public acceptance should be chosen. At present, the most promising biometric for BE is iris followed by fingerprints and face.
- The image acquisition process (the requirements are tougher for BE than for conventional biometrics) must be improved.
- BE must be made resilient against attacks.

- The overall accuracy and security of BE algorithms must be improved. Advances in the algorithm development in conventional biometrics and in ECCs should be applied to BE.
- Multimodal approaches should be exploited. BE applications should be developed.

## Summary

Biometric Encryption is a fruitful area for research and is becoming sufficiently mature for prototype development and the consideration of applications.

BE technologies exemplify the fundamental privacy and data protection principles that are endorsed around the world, such as data minimization, user empowerment, and security.

Although introducing biometrics into information systems may result in considerable benefits, it can also introduce many new security and privacy vulnerabilities, risks, and concerns. Novel Biometric Encryption techniques can overcome many of those risks and vulnerabilities, resulting in a win-win, positive-sum model that presents distinct advantages to both security and privacy.

## Related Entries

- ▶ Biometric Security, Overview
- ▶ Biometric Vulnerabilities
- ▶ Cancelable Biometrics
- ▶ SFinGe

## References

- 1 Tomko, G.J., Soutar, C., Schmidt, G.J.: Fingerprint controlled public key cryptographic system. U.S. Patent 5541994, July 30, 1996 (Filing date: Sept. 7, 1994)
- 2 Jain, A.K., Nandakumar, K., Nagar, A.: Biometric Template Security. EURASIP J. Adv. Signal Process. v. 2008, Article ID 579416, pp. 1–17 (2008)
- 3 Tuyls, P., Škorić, B., Kevenaar, T. (eds.): Security with Noisy Data: Private Biometrics, Secure Key Storage and Anti-Counterfeiting. Springer, London (2007)
- 4 Cavoukian, A., Stoianov, A.: Biometric Encryption: The New Breed of Untraceable Biometrics. In: Boulgouris, N.V., Plataniotis, K.N., Micheli-Tzanakou, E. (eds.): Biometrics: fundamentals, theory, and systems. Wiley, London (2009)
- 5 Juels, A., Wattenberg, M.: A fuzzy commitment scheme. In: Tsudik G. (ed.) Sixth ACM Conference on Computer and Communications Security, pp. 28–36. ACM Press, New York (1999)

6. Juels, A., Sudan, M.: A fuzzy vault scheme. In: Lapidoth, A., Teletar, E. (eds.) Proceedings of IEEE International Symposium on Information Theory, p. 408. IEEE, Lausanne (2002)
7. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy Extractors: How to Generate Strong Keys from Biometrics and other Noisy Data. In Cachin, C., Camenish, J., Proc. Eurocrypt 2004, pp. 523–540 Springer-Verlag, NY (2004)
8. Teoh, A.B.J., Ngo, D.C.L., Goh, A.: Personalised cryptographic key generation based on FaceHashing. *Comput. Secur.* 23, 606–614 (2004)
9. Soutar, C., Roberge, D., Stoianov, A., Gilroy, R., Vijaya Kumar B.V.K.: Biometric Encryption (Chapter 22). In: Nichols, R.K. (ed.): *ICSA Guide to Cryptography*, McGraw-Hill New York, (1999)
10. Adler, A.: Vulnerabilities in Biometric Encryption Systems. In: *Audio-and video-based Biometric Person Authentication (AVBPA2005)*. Lecture Notes in Computer Science, vol. 3546, pp. 1100–1109. Springer, New York (2005)
11. Davida, G.I., Frankel, Y., Matt, B.J.: On enabling secure applications through off-line biometric identification. In: *Proceedings of the IEEE 1998 Symposium on Security and Privacy*, pp. 148–157, Oakland, CA (1998)
12. Monroe, F., Reiter, M.K., Wetzel, S.: Password hardening based on keystroke dynamics. *Int. J. Inform. Secur.* 1(2), 69–83 (2002)
13. Hao, F., Anderson, R., Daugman, J.: Combining Crypto with Biometrics Effectively. *IEEE Trans. Comput.* 55(9), 1081–1088 (2006)
14. Bringer, J., Chabanne, H., Cohen, G., Kindarji, Z'emor, G.: Optimal iris fuzzy sketches. In: *IEEE First International Conference on Biometrics: Theory, Applications, and Systems, BTAS'07*, Washington, DC, 27–29 Sept, (2007)
15. Nandakumar, K., Jain, A.K., Pankanti, S.C.: Fingerprint-based Fuzzy Vault: Implementation and Performance. *IEEE Trans. Inform. Forensics Secur.* 2(4), 744–757 (2007)
16. Draper, S.C., Khisti, A., Martinian, E., Vetro, A., Yedidia, J.S.: Using Distributed Source Coding to Secure Fingerprint Biometrics. In: *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, pp. 129–132 (2007)
17. Chang, E.-C., Shen, R., Teo, F.W.: Finding the Original Point Set Hidden among Chaff. In: *Proceedings of the 2006 ACM Symposium on Information, computer and communications security. ASIACCS'06*, Taipei, Taiwan, pp. 182–188 Sept, (2006)
18. Delvaux, N., Bringer, J., Grave, J., Kratsev, K., Lindeberg, P., Midgren, J., Breebaart, J., Akkermans, T., van der Veen, M., Veldhuis, R., Kindt, E., Simoens, K., Busch, C., Bours, P., Gafurov, D., Yang, B., Stern, J., Rust, C., Cucinelli, B., Skepastianos, D.: Pseudo identities based on fingerprint characteristics. In: *IEEE fourth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2008)*, August 15–17, Harbin, China (2008)