Information Fusion 32 (2016) 93–108

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Optimal sequential fusion for multibiometric cryptosystems

Takao Murakami [a,*], Tetsushi Ohki [a], Kenta Takahashi [b]

[a] National Institute of Advanced Industrial Science and Technology, Tokyo 135-0064, Japan
[b] Hitachi, Ltd., Yokohama 244-0817, Japan

ARTICLE INFO

Article history:
Available online 10 February 2016

Keywords:
Multimodal biometrics
Sequential fusion
Biometric cryptosystems
Sequential probability ratio test

ABSTRACT

Biometric cryptosystems have been widely studied in the literature to protect biometric templates. To ensure sufficient security of the biometric cryptosystem against the offline brute-force attack (also called the FAR attack), it is critical to reduce FAR of the system. One of the most effective approaches to improve the accuracy is multibiometric fusion, which can be divided into three categories: feature level fusion, score level fusion, and decision level fusion. Among them, only feature level fusion can be applied to the biometric cryptosystem for security and accuracy reasons. Conventional feature level fusion schemes, however, require a user to input all of the enrolled biometric samples at each time of authentication, and make the system inconvenient.

In this paper, we first propose a general framework for feature level sequential fusion, which combines biometric features and makes a decision each time a user inputs a biometric sample. We then propose a feature level sequential fusion algorithm that can minimize the average number of input, and prove its optimality theoretically. We apply the proposed scheme to the fuzzy commitment scheme, and demonstrate its effectiveness through experiments using the finger-vein dataset that contains six fingers from 505 subjects. We also analyze the security of the proposed scheme against various attacks: attacks that exploit the relationship between multiple protected templates, the soft-decoding attack, the statistical attack, and the decodability attack.

1. Introduction

Protecting biometric data is a critical issue in biometric authentication systems, since biometric information such as faces, fingerprints, irises and vein patterns are personal and privacy information. Although many conventional systems rely on standard encryption to protect biometric templates, the encrypted templates have to be decrypted at the time of verification to perform pattern matching, and thus a skilled attacker who aims at this timing can break them. Other systems use tamper-proof devices, such as hardware tokens, to protect biometric templates. However, these systems require a user to possess a token or to use limited devices in which a user's template is enrolled. These limitations can reduce the usability of the authentication systems.

To solve the problem fundamentally, various studies have been made on so-called *biometric template protection* techniques that keep biometric templates secret in the algorithm level even during verification. Among them, *biometric cryptosystems* have particularly attracted attention, in which a biometric feature is used as a source of a secret key of cryptosystems and verified without being revealed using cryptographic techniques [1–4]. Biometric cryptosystems typically take a strategy of embedding a secret key into a biometric feature yielding so-called *auxiliary data* (AD), and releasing the secret key from the AD using a genuine biometric feature. For authentication, the secret key is verified using a *pseudo identifier* (PI), which is a public key or a hash value of the secret key. A set (AD, PI) is referred to as a *protected template*, and is enrolled into a database or smart card, along with a user ID.

There are two possible models for storing a protected template (AD, PI) in the biometric cryptosystem. The first model is to store AD and PI separately. For example, AD is stored into a smart card of a user, while PI is stored into the database in the authentication server. Then, even if either of AD or PI is compromised, we can restore the security by updating both of the two. In this model, the security requirement for the biometric cryptosystem is that it is sufficiently hard to guess a biometric feature or impersonate a user by use of either of AD or PI.

The second model is to store both AD and PI into a single place (e.g. the authentication server). In this case, it can happen that both of AD and PI are compromised from the place at the same time. Thus, the security requirement for the biometric cryptosystem in this model is that it is sufficiently hard to guess

* Corresponding author. Tel.: +81 29 861 3744.
E-mail addresses: takao-murakami@aist.go.jp (T. Murakami), tetsushi.ohki@aist.go.jp (T. Ohki), kenta.takahashi.bw@hitachi.com (K. Takahashi).

http://dx.doi.org/10.1016/j.inffus.2016.02.002
1566-2535/

a biometric feature or to impersonate a user even if both of AD and PI are available to the adversary. If this security requirement is satisfied, we can realize an authentication system where a user is not required to have a smart card that stores AD secretly nor required to present the smart card at the authentication phase. We can even disclose the protected template (AD, PI) to the public, or share it across multiple organizations. Thus, we refer to this model as a *public template model*. The aim of this paper is to realize this model in a secure manner.

In the public template model, it is required that the biometric cryptosystem defends against the *offline brute-force attack (also called the FAR attack);* the attack where the adversary prepares a large number of biometric features, and matches each of them with a protected template (AD, PI) offline to find a biometric feature that successes in authentication (i.e. a correct secret key is reproduced). Let α be FAR (False Acceptance Rate) of the biometric cryptosystem. Then, this attack results in success after matching on average $1/(2\alpha)$ biometric features. Thus, it is required that FAR α is sufficiently small to realize the public template model.

It is difficult to achieve sufficient security (comparable to typical cryptographic keys) against the offline brute-force attack using only one source of biometric information (e.g. one finger-vein). For example, to achieve at least 64-bit security, FAR should be smaller than $2^{-64} \simeq 5.4 \times 10^{-20}$, whereas FAR of many commercial biometric authentication systems is 10^{-5}–10^{-7}.

Multibiometric fusion [5], which combines multiple sources of biometric information (e.g. fingerprint, face, and iris; multiple finger-veins) for authentication, is expected to fill this gap. Some studies also applied multibiometric fusion to the biometric cryptosystem [6–9]. However, most of the conventional fusion schemes (including [6–9]) require users to input *all* the enrolled biometric information (e.g., 10 finger-veins) at the authentication phase. Such schemes are referred to as *parallel fusion schemes* [10], and cause inconvenience for users.

To make the authentication system convenient for users, some studies proposed a *sequential (or serial) fusion scheme* [10–13], which makes a decision each time a user inputs his/her biometric sample. Some sequential fusion schemes [12,13] can optimize the trade-off between the accuracy and the number of biometric inputs required before acceptance, and thus realize a secure and convenient biometric system. These schemes combine multiple sources of biometric information at the matching score level (i.e. score level fusion) to make such an optimal decision.

However, it must be noted that biometric cryptosystems must not output a matching score for a security reason; an attacker can reconstruct the original biometric feature by using the score as a clue (i.e. *hill-climbing attack* [14]). Thus, the conventional sequential fusion schemes using scores cannot be applied to biometric cryptosystems. To reduce FAR as much as possible without disclosing scores during verification, we should construct a multibiometric cryptosystem based on *feature level fusion*, which combines biometric features into a single (but large-sized) feature. To the best of our knowledge, no studies have attempted to develop a sequential fusion scheme at the feature level.

In this paper, we propose an optimal sequential fusion scheme at the feature level to realize secure and convenient biometric cryptosystems in the public template model. The main contributions are as follows:

- We first propose a general framework for feature level sequential fusion. To the best of our knowledge, this is the first framework that enables secure and convenient biometric cryptosystems in the public template model (Section 3.1).
- Based on this framework, we second propose an optimal algorithm for feature level sequential fusion. This algorithm is based on the SPRT (sequential probability ratio test) [15,16], and minimizes the average number of input while keeping FAR less than or equal to the required value. We also provide a formal proof of this optimality (Sections 3.2–3.4).
- We apply the proposed scheme to the fuzzy commitment scheme, and demonstrate its effectiveness through experiments using the finger-vein dataset in [17], which contains six fingers from 505 subjects (33,298 finger-vein images in total) (Section 4).
- We finally analyze the security of the proposed scheme against various attacks: attacks that exploit the relationship between multiple protected templates, the soft-decoding attack [18,19], the statistical attack [18,20], and the decodability attack [21,22] (Sections 5 and 6).

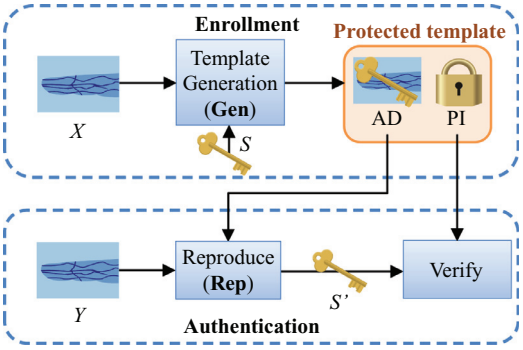

Fig. 1. Architecture of a biometric cryptosystem (X, Y: biometric feature, S, S': secret key, AD: auxiliary data, PI: pseudo identifier). If $\mathrm{dis}(X, Y) \leq \delta$, then $S = S'$ (δ: distance threshold).

2. Related work

2.1. Biometric cryptosystems

Biometric cryptosystems have been widely studied in the literature [23]. Examples include fuzzy commitment [3], fuzzy vault [2], and fuzzy extractor [1].

Fig. 1 shows an architecture of a typical biometric cryptosystem (we also describe the fuzzy commitment scheme [3] as an example of the biometric cryptosystem in Section 2.2). At the enrollment phase, a template generation algorithm **Gen** receives a biometric feature X from a user. It then encodes a secret key S (typically a random string) using an ECC (error-correcting code), and embeds it into X to make AD (auxiliary data; it is also called helper data). AD is designed so that if the user inputs a biometric feature Y that is sufficiently close to X according to some distance metric dis (i.e. $\mathrm{dis}(X, Y) \leq \delta$ for a predetermined distance threshold δ), a secret key S is reproduced from AD. **Gen** also makes PI (pseudo identifier), which is used to verify the reproduced secret key. PI is a public key, or a hash value of S. A protected template (AD, PI) is enrolled into a database (or smart card), along with a user ID.

At the authentication phase, a reproduce algorithm **Rep** receives a new biometric feature Y and AD, and reproduces a secret key S' from AD using Y. As described above, if X and Y are sufficiently close (i.e. $\mathrm{dis}(X, Y) \leq \delta$), a correct secret key is reproduced (i.e. $S = S'$). The system authenticates a user using S' and PI.

As described in Section 1, we do not have to store the protected template (AD, PI) secretly in the public template model. We can even disclose it to the public, or share it across multiple organizations. To achieve this ultimate goal, FAR needs to be sufficiently small.