A practical implementation of fuzzy fingerprint vault for smart cards

Daesung Moon · Yongwha Chung · Changho Seo · Sung-Young Kim · Jeong-Nyeo Kim

Abstract Biometric-based authentication can provide strong security guarantee about the identity of users. However, security of biometric data is particularly important as the compromise of the data will be permanent. To protect the biometric data, we need to store it in a non-invertible transformed version. Thus, even if the transformed version is compromised, the actual biometric data remain safe. Fuzzy vault is a cryptographic construct to secure critical data with the fingerprint data. In this paper, we implement the fuzzy fingerprint vault, combining fingerprint verification and fuzzy vault scheme to protect fingerprint templates, for the smart card environment. To implement the fuzzy fingerprint vault as a complete system, we have to consider several practical issues such as automatic fingerprint alignment, verification accuracy, template size for storing in the smart card, execution time, error correcting code, etc. Especially, we handled the fingerprints having a few minutiae by applying an adaptive degree of the polynomial, and thus our implementation result can be used for real, large-scale applications.

Keywords Cancelable biometrics · Fuzzy vault · Fingerprint template protection · Fingerprint recognition · Smart card

Introduction

The increasing demand for more reliable and convenient security systems generates a renewed interest in human identification based on biometric identifiers such as fingerprints, iris, voice, and gait (Maltoni et al. 2003; Ling and Masao 2011; Kryvinska et al. 2010). In this paper, the fingerprint has been chosen as the biometrics for user authentication because it is more mature in terms of the algorithm availability and feasibility (Maltoni et al. 2003).

In typical fingerprint verification systems, the fingerprint features are often stored in a central database. With the central storage of the fingerprint data, there are open issues of misuse of the fingerprint feature such as the “Big Brother” problem. To solve these open issues (Bolle et al. 2002; Uludag et al. 2004; Schneier 1999), the database can be decentralized into millions of security tokens such as smart cards, USB tokens (Janke 2001; Gil et al. 2001; Bamasak 2010). A smart card is a miniature computer that is composed of a microprocessor, RAM, ROM, EEPROM, etc, and we implement the e-ID card system using S3CC9GC (SAMSUNG) smart card chip. To identify the card holder, our e-ID card uses the facial and fingerprint data stored on the EEPROM. Because the size of the EEPROM for storing several data such as Java API Table, executable programs, and facial data is about 72 Kbytes, the fingerprint template size should be as small as possible.

In spite of the many advantages of using fingerprints, there are security and privacy problems (Maltoni et al. 2003; Bolle et al. 2002; Uludag et al. 2004; Schneier 1999). That is, once
the fingerprint data stored in the smart card is compromised, the data is compromised permanently and cannot be reissued whenever a user needs because the user has only ten fingers. This is an urgent problem in the biometrics community.

A fuzzy fingerprint vault (Clancy et al. 2003; Uludag et al. 2005; Nagar et al. 2008; Nandakumar et al. 2007; Yang and Verbauwhede 2005; Chung et al. 2005; Moon et al. 2007) is one of the most popular solutions for protecting the fingerprint data of a user stored in the database, by using the idea of the fuzzy vault (Juels and Sudan 2002). Some results using the fuzzy fingerprint vault have been reported (Clancy et al. 2003; Uludag et al. 2005; Nagar et al. 2008; Nandakumar et al. 2007; Yang and Verbauwhede 2005; Chung et al. 2005; Moon et al. 2007). However, these methods are not realistic for the smart card environment because they do not consider all possible issues such as automatic fingerprint alignment, verification accuracy, template size for smart card, execution time, etc. In this paper, we have implemented a fuzzy fingerprint vault for the smart card environment as a complete system.

**Background**

A fingerprint verification system has two phases: enrollment and verification (Maltoni et al. 2003; Chong and Tanaka 2010). In the off-line enrollment phase, a template fingerprint image is preprocessed, and features, called as minutiae, are extracted and stored. In the on-line verification phase, the similarity between the template minutiae and the input minutiae is examined.

A minutia can be specified by its $x$, $y$-coordinate, angle, and its type (i.e., ending/bifurcation) and represented as (1).

$$A = \{(x_i, y_i, \theta_i, t_i) | i = 1, \ldots, n\}$$

(1)

As explained in Sect. “Introduction”, some problems with fingerprint verification systems can arise when the template minutiae have been compromised. For verification systems based on physical tokens such as keys and badges, a compromised token can be easily canceled and the user can be assigned a new token. Similarly, user IDs and passwords can be changed as often as required. However, once minutiae stored in the smart card are compromised, the minutiae are compromised permanently and cannot be reissued whenever a user needs because the user has only ten fingers.

Juels and Sudan (Juels and Sudan 2002) proposed the fuzzy vault, a new architecture with applications similar to Juels and Wattenberg’s fuzzy commitment scheme (Juels and Wattenberg 1999), but is more compatible with partial and reordered data.

The procedure for constructing the fuzzy vault is as follows: Secret value $S$ is first encoded as the coefficients of some degree $k$ polynomial in $x$ over a finite field $GF(p)$. This polynomial $p(x)$ is now the secret to protect. The locking set $L$ is a set of $n$ values $l_i \in GF(q)$ making up the fuzzy encryption key, where $n > k$. The locked vault contains all the pairs $[l_i, p(l_i)]$ and some large number of chaff points $(\alpha_j, \beta_j)$, where $f(\alpha_j) \neq \beta_j$. After adding the chaff points, the total number of items in the vault is $r$.

In order to crack this system, an attacker must be able to separate the chaff points from the legitimate points in the vault. The difficulty of this operation is a function of the number of chaff points, among other things. A legitimate user should be able to unlock the vault if they can narrow the search space. In general, to successfully interpolate the polynomial, they have an unlocking set $U$ of $m$ elements such that $L \cap U$ contains at least $k + 1$ elements. The details of the fuzzy vault can be found in Juels and Sudan (2002).

Using the error-correction coding, it is assumed that the legitimate user can reconstruct $p$ (and hence $k$). The security of the scheme is based on the infeasibility of the polynomial reconstruction problem (i.e., if Bob does not know many points that lie on $p$, he cannot feasibly find the parameters of $p$, hence he cannot access $k$). Since this fuzzy vault can work with unordered sets (common in biometric templates, including fingerprint minutiae data), it is a promising candidate for crypto-biometric systems.

Since the introduction of the fuzzy vault scheme, some implementations results for fingerprint have been reported (Clancy et al. 2003; Uludag et al. 2005; Nagar et al. 2008; Nandakumar et al. 2007; Yang and Verbauwhede 2005; Chung et al. 2005; Moon et al. 2007). Clancy et al. (2003) proposed a fuzzy fingerprint vault scheme based on the location of minutiae in a fingerprint. However, the False Reject Rate (FRR) of their system was $20 - 30\%$. Uludag et al. (2005) introduced a modification to the fuzzy vault scheme, which uses a simple Cyclic Redundancy Check (CRC) for error-correction, instead of the Reed-Solomon polynomial decoding. Also, the FRR of their system was $21\%$, and execution time was $52s$ with a $3.4GHz$ processor. An additional problem with these implementations (Clancy et al. 2003; Uludag et al. 2005) is that they assumed fingerprints were pre-aligned. This is not a realistic assumption for fingerprint-based authentication systems, and limits the applicability of their implementations.

Nagar et al. (2008) and Nandakumar et al. (2007) also proposed the fuzzy fingerprint vault based on the helper data for the auto-alignment. Nagar et al. (2008) reported a FRR of $15.5\%$ at $0\%$ false accept rate (FAR), and Nandakumar et al. (2007) got a significant improvement in the genuine accept rate (GAR). However, they have some limitations such as a slow decoding time (e.g., the median and mean decoding time was $3$ and $8s$, respectively) or a possibility of failure to find the core point in some images. Furthermore, the helper data may leak some information of the fingerprint data.