On Enabling Secure Applications Through Off-line Biometric Identification

George I. Davida
Univ. of Wisconsin-Milwaukee
Milwaukee, WI
davida@cs.uwm.edu

Yair Frankel
CertCo LLC
New York, NY
yfrankel@cs.columbia.edu

Brian J. Matt
Sandia National Laboratories*
Albuquerque, NM
bjmatt@cs.sandia.gov

Abstract

In developing secure applications and systems, the designers often must incorporate secure user identification in the design specification. In this paper, we study secure off-line authenticated user identification schemes based on a biometric system that can measure a user's biometric accurately (up to some Hamming distance). The schemes presented here enhance identification and authorization in secure applications by binding a biometric template with authorization information on a token such as a magnetic strip. Also developed here are schemes specifically designed to minimize the compromise of a user's private biometrics data, encapsulated in the authorization information, without requiring secure hardware tokens.

In this paper we furthermore study the feasibility of biometrics performing as an enabling technology for secure system and application design. We investigate a new technology which allows a user's biometrics to facilitate cryptographic mechanisms.

1 Introduction

Secure digital identification schemes are becoming increasingly important, as more security applications require identification based on physical characteristics rather than solely on a user's knowledge of a secret cryptographic key or password. The increased interest in such applications, ranging from door access to electronic commerce applications, has led to an increased interest in methods for secure and accurate identification [8, 5, 18, 17] of individuals as well as machines and objects. In this paper we are interested in systems of identification that use measurable biological features, biometrics, which can be readily measured at the point of application. It is desirable that such measurements be non-invasive and simple to perform. One biometric that has been suggested is the iris scan [3, 12, 6, 21].

On-line applications secured through the use of biometric authentication typically are based on a push or pull model. In both models, the first step is a user initialization, which occurs when the user's biometric template is registered with the on-line server. After initialization, when a user wants access that requires biometric identification, a *biometric authorization process* is performed. At this time the user's biometric is read by a reader. In the push model, the reader transmits (preferably via a private channel) the reading to the on-line server; the on-line server then verifies the validity of the reading based on the user's template in the server's directory; and finally the server sends an authenticated acceptance or rejection message back to the reader. In the pull model, the reader requests the template from the server, and the reader performs the verification steps after receiving the template over an authenticated and, preferably, private channel from the server. In both cases, an authenticated channel is necessary for some communications between the on-line database and the reader. The authentication can also provide for a binding of a user's biometric with some form of authorization, as established by trust relationships between the reader and the on-line database.

Here we are interested in developing biometric based identification systems which do not require the incorporation of an on-line database for the security infrastructure. Such databases are not always practical in mobile environments, such as military applications, and are often cost prohibitive since they require expensive wiring for connectivity or costly wireless devices. In order to remove the connectivity requirements, an *off-line* biometric system is achieved by incorporating a biometric template on a storage device / token (e.g., magnetic strip or smartcard) which provides for a reliable storage medium; however, there are no security requirements required of the token. We, therefore, will work in the pull model with the storage device containing sufficient information to validate the authenticity of the user's acquired biometric template to the biometric generated during user initialization. To provide for the user biometric/user authorization binding, a trusted authorization officer who authenticates (signs) the user's biometric template is incorporated into our infrastructure.

A biometric identification system which provides

*Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

DISCLAIMER

the user's biomeric template in the clear may not be acceptable to a user, because a user's biometric template could be used for unacceptable purposes if the template is obtained by an unauthorized individual. Biometric templates can provides information which a user may not want provided readily. For instance, a finger print reading can be used for law enforcement purposes and an eye scan (retinal or iris) may be able to detect medical conditions.

We study the feasibility of protecting a user's biometric on an insecure device. Such protection may be beneficial if the storage device holding the biometric template is lost or stolen. This added protection may provide for stronger user acceptance, since the user's template is not sent in the clear. In our study we propose a classification of secure off-line biometric systems according to who, if anyone, in the system has a private decryption key (when templates are encrypted).

An important model to consider is the case where *neither the user nor the reader* maintains private decryption keys, because it is a scalable solution when the user must have authorization amongst multiple readers and when password protection is inappropriate. Providing for authorization bound to a biometric template appears to be inherently difficult in this model, because the user's biometric template cannot exist in the clear on the storage device.

To achieve our result we had to overcome several hurdles. The first is to deal with errors which occur during the reading of biometrics. Variances from multiple readings of the same user often occur due to problems such a scratch on a finger, disease affecting blood vessels in the retina, variations in light causing changes in the pupil size during iris reading, and different positioning of the object being scanned (finger, head, etc.). In an off-line system if there are any discrepancies between the original template and later readings, the biometric template cannot be verified against the authentication officer's authentication information.

Another hurdle that had to be overcome is that cryptographic authentication mechanisms (e.g., a digital signature) that the trusted authorization officer invokes to bind authorization with a user's template do not necessarily hide all the information of the input (i.e., provide confidentiality of the message that is signed), thereby potentially leaking information about the user's biometrics. Let us give an example of a signature scheme SIG which leaks the acquired message completely. Let $sig(m)$ be the signature of a message m; observe as a simple example that one can generate a new secure (unforgeable) signature function $SIG(m) = (m, sig(m))$, (e.g. message/signature pair $(m', (m, sig(m))$ is valid if $m' = m$ and $Verify(m', sig(m)) = TRUE$). Hence, signature functions do not necessarily protect against information leakage of the input. A solution to this problem is simple, of course, if the trusted authorization officer and reader share a private key.

It should be noted that our system is also applicable to on-line systems where information is stored in an on-line database instead of on storage cards. By using our system in an on-line environment, one is able to reduce the security requirements imposed on the database. For example, our techniques prevent the database manager from reading biometric templates directly from the database or archives.

We also note that designers of secure systems are often hampered by the lack of mechanisms to satisfy the various requirements of a secure key management infrastructure. This infrastructure may have to deal with generation of both public and private keys, authenticated dissemination of keys, and the storage of keys, as well as other concerns such as maintaining privacy of users and trusted circulation of user authorizations. The security of this infrastructure is often hindered by insufficient mechanisms to secure private keys for users. We noticed that when one assumes that a user's biometric information has sufficient uncertainty, our technique also allows for the biometric template to be used as a private key. Since there may not be sufficient entropy (i.e., uncertainty) in a user's biometric, our system allows us to augment password encryption with the entropy provided in a biometric.

Our solutions are based on cryptography. We do not assume unproven, and usually expensive, physical protection mechanisms such as optical computers (see [20]).

The result we present here has many features:

- We present off-line identification systems based on any biometric technology that can be measured accurately (up to some Hamming distance).
- Enhancements also allow for incorporation of authorization information from a trusted authorization officer. In essence our system binds the user identity not only for simple access but for authorization.
- We classify off-line biometric systems according to which entity (e.g., reader, user, authorization officer), if any, must maintain a long term private decryption key for the purpose of hiding a user's biometric from compromise.
- Based on our classification of off-line biometrics, we discuss the feasibility of designing a system in which information stored in the the storage device does not compromise the biometric information of the individual involved when a card is lost or stolen.
- The techniques presented provide for on-line identification systems in which the privacy of a biometric template is protected on the database.
- We propose an infrastructure and mechanisms which allow biometrics to enable cryptographic applications when there is sufficient entropy in a user's biometric.
- In presenting our results, we shall relate them to the iris technology[3, 12, 6, 21].

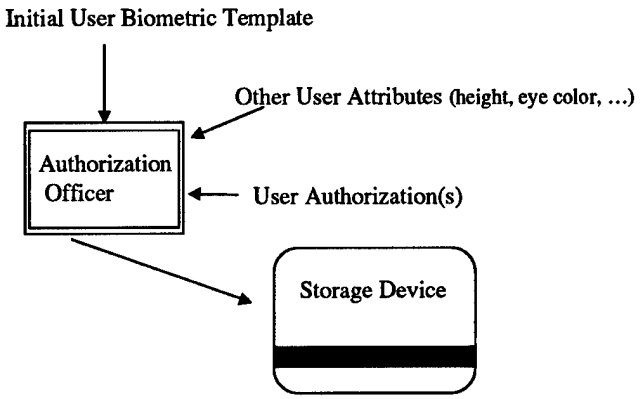

Figure 1: Storage device initialization

2 Model

We shall propose several models in which off-line biometrics can be incorporated into a security infrastructure. In order to motivate the design of our off-line system, we first analyze in Section 2.1 how an on-line system would work and the requirements which may be desired for such a system. We then investigate in Section 2.2 the off-line model for access control, authorization and private key storage.

In our models below we use an authorization officer entity in the architecture. The authorization officer's role is to certify (e.g., authenticate or sign) a binding between a user's biometric template and some other attributes of the user. The authorization officer is thereby the trusted third party attesting to authorization as well as to other user attributes. The authorization officer plays a role that is similar to the Certification Authority (CA) in a public key hierarchy (see [22]), except that the authorization officer binds biometrics to user attributes, while a CA binds a public key to user attributes.

In considering biometrics, we note that we need to make the following assumption:

Assumption 1 (Reproduction): *We assume that a biometric is not reproduceable. Hence it is unique to an individual, but even more importantly, one should not be able to artificially generate a "device" with sufficient characteristics to pass a biometric verification of a user.*

This assumption must be achieved in any high consequence application protected by a biometric system, in order to provide secure and unique identification. Otherwise, an adversary with sufficient probability will be able to impersonate a user by reproducing the authorized user's biometric. To provide for such protection, properties such as pupillary unrest of an iris and blood flow and heat from a finger scan have been used to support this assumption in some biometric systems. Throughout this paper we assume the biometric system we incorporate into our designs provides sufficient protection to provide the reproduction assumption.

2.1 On-line Model

Our architecture for an off-line system is motivated by the on-line system. We first briefly review the model for an on-line system.

The primary application of biometrics today involves the use of an on-line server. During *system setup* biometric readers are connected to a trusted on-line server through secure links which are either cryptographically secured channels or in which physical security is established. If cryptographic security is used, then a secure key distribution is required.

User initialization is performed by the user having his/her biometric template registered with the on-line server. Later, when a user wants access which requires the user to pass through a biometric identification, a *biometric authorization process* is performed. The user first has his/her biometric read by a reader; the reader transmits the reading to the on-line server; the on-line server then verifies the validity of the reading based on the user's template in the server's directory; and finally the server sends an authenticated acceptance or rejection message back to the reader. This is the push model for an off-line system. In the pull model, the reader requests the template from the server, and the reader perform the verification steps, after receiving the template over an authenticated and, preferably, private channel from the server.

Our off-line model below is inspired by the pull model. It simulates the on-line transmission of a user's template to the reader with storage device containing a user's biometric (or similar information) for verification authenticated by an authorization officer's signature.

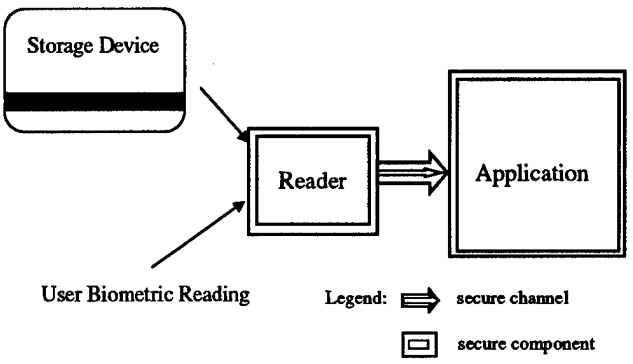

Figure 2: Secure application with biometric authorization

2.2 Off-line Model

In the off-line system, the biometric authorization process cannot have a direct (on-line) information retrieval mechanism. This requirement means that the push model cannot be used, because it requires a communication from the reader to the on-line database and back. The pull model, however, can be simulated by incorporating a storage token which replicates the information sent by the on-line reader. We should note, however, that as with any off-line identification system, immediate revocation of user privileges is not possible. This limitation must be taken into consideration by the system designer during the development of the security architecture.

We now discuss the workflow in the off-line model.

Initialization process:

The user initialization process for the off-line model is represented in Figure 1. The secure authorization officer takes as input an initial biometric reading, called the user biometric template, the authorization information defining the set of privileges granted the user by the authorization officer, and other user attributes. As output a storage device such as a magnetic strip card is encoded with information which establishes a binding between a user's biometrics (and, possibly, other user attributes) and the user's authorization granted by the authorization officer.

Application process:

During a secure application, as depicted in Figure 2, a reader takes as input the user's storage device (token) and reads the user's biometric. Given this information, which may also include other user attributes not represented in this figure, the user's authorization attributes can be obtained and linked to the authorization officer. This information may now be securely transmitted to the secure application. Note that the primary difference between an off-line and on-line system is that the storage device can be replaced by an authenticated transmission link to the authorization officer (or its database) in the on-line system.

Certain principles are incorporated in our model:

1. There must be a binding between a user's biometric and a trusted authorization officer. Hence, we require a storage device (e.g., magnetic strip or smartcard) to store the binding information.
2. There is a need for a scalable solution when privacy of a user's biometric must be protected in case a storage device is lost or stolen. The primary scalability issues are who must store private keys and how much storage must be provided on the cards.

Principle 2 suggests an interesting feasibility question. Is it possible to provide a scalable solution and protect a user's biometric, and if so, what requirement must be imposed on the security architecture? To answer the question, we now classify the off-line security architectures by who, if anyone, must hold a private key.

Private key in reader: If a reader has a private key to decrypt biometric information encrypted by the authorization officer, then there will be no leakage of biometric information when a card is lost or stolen. However, such a system is not scalable if the memory device has low storage capability and the application's architecture requires multiple readers (each with its own private key), because a separate encryption of the biometric template is required for each reader. This technique however, can be effective if there are few readers in the architecture.

In Figure 3 we show the information that must be stored on a storage device when multiple readers are used.

To be effective, this approach requires that the readers provide some form of protection for the reader's private key (e.g., FIPS PUB 140-1 standards [9]), because if the private key is stolen from

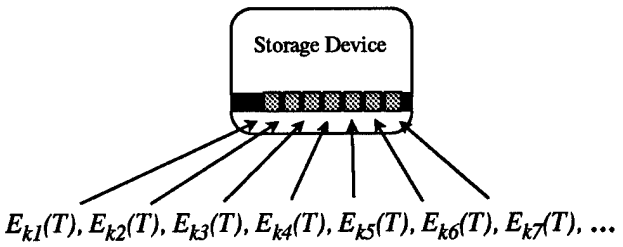

Figure 3: Scaling limitation of multiple (reader,key) architecture where each reader holds a different private key.

the device, the adversary is able to read the biometric from any user's storage device.

Password-protection: Password protection can hide information stored on a card if the password has sufficient entropy. This approach is a scalable solution (e.g., using password encryption [16] to encrypt the biometric template with a user memorized password), if revealing a password to a reader is considered safe and the readers have a user password input mechanism. Generally, password protection is considered insufficient, since it usually has low entropy and is therefore easily guessed. As a result FIP PUB 190 recommends the combination of PIN/password and a token for user authentication when feasible [10].

No keys or passwords: Potentially, this is the most scalable approach with minimal system component requirements for an off-line system. Such systems, as will be shown, are possible when the entropy in a biometric is large enough.

NOTE: It should be noted that the off-line systems we shall discuss are also applicable to on-line systems where information is stored in an on-line database instead of on storage cards. By using our system in an on-line environment, one is able to reduce the security requirements imposed on the database, where privacy restrictions on the information exit.

3 Background

We briefly present some background from Coding Theory and Cryptography that we will need in later sections.

3.1 Cryptography

In order to provide maximum protection of user biometric information, key material and other sensitive information on storage devices, we utilize mechanisms which prevent the storage device from leaking information (without the user's biometric) to an adversary of a specified strength. In order to do so we will use the tools which we informally discuss below.

Semantically Secure Encryption: In Shannon's theory [19] an encryption algorithm has *pefect secrecy* if a passive adversary, even with *un-bounded* computational power, cannot learn anything from any ciphertext about its corresponding plaintext, except possibly its length. An encryption algorithm is semantically secure [11] if a passive adversary cannot learn anything in *expected polynomial time* from any ciphertext about its corresponding plaintext, except possibly its length.

A **Random Oracle** is a publicly known function R with the property that when provided a value x the oracle produces a random number $R(x)$, that is totally independent of x (see, e.g., [1]).

A **Partial Information Hiding Function** (i.e. an oracle hashing function) [4] can be described informally as a hashing algorithm $H(x) = c$ and a verification function $V(x, c) \Rightarrow \{True, False\}$ with the following properties: 1) infeasible to find a collision, i.e. $V(x, c)$ and $V(y, c)$ cannot both be true if $x \neq y$ 2) information hiding, for a polynomial time adversary having $c = H(x)$, gives no further information on x beyond the ability to exhaustively search for x.

The final tool that we need is Universal One Way Hash Function families [14]. A **Universal One Way Hash Function** family is a family of hash functions $F_k(x) = c$ that utilizes a key k to select a member of the family. In addition a polynomial bounded adversary cannot choose an x, then upon learning k, find a collision, i.e. a pair x and $y, x \neq y$ such that $F_k(x) = F_k(y)$.

3.2 Coding Theory

Our interest in error correction codes stems from the fact that the biometrics acquired are not measured perfectly. Each measurement results in a vector that is at some Hamming distance (discussed below) from other measurements. Empirical work in measuring some biometrics, such as the iris, has shown that the expected hamming distance between any two biometric measurements is about 10 percent. These errors in the measured vectors appear to be independent. Hence error correction is critical to the computation of a biometric in this scheme.

We are interested in two types of error correction: Error correction at the point of acquiring the biomet-

ric, and error correction during the verification phase. Empirical measurements show that the errors in a biometric are independent, with a crossover probability of .016 [6]. This observation suggests that if several measurements of a biometric are subjected to majority decoding (discussed below) at the time of template creation, then that template can then be considered the "canonical" biometric template. Once this canonical biometric is obtained, error correction check digits are computed for this biometric, which will be used as will be shown below.

When a user presents for verification, the same procedure is used to arrive at a biometric that is then used in the rest of the process to verify identity. In this phase, error correction is used to remove residual errors, using the check digits computed above. This process will correct the measured biometric into the canonical biometric if the number of errors are within the tolerance.

Hamming Distance: For simplification, we shall restrict our discussion of error correcting codes to binary codes [2, 13, 15]. The (binary) Hamming weight of a codeword $\vec{c}$, denoted by $\mathrm{Hw}(\vec{c})$, is the number of one bits in the codeword. That is, for an n bit string [1] $\vec{c} = c_1||c_2|| \cdot ||c_n$ the Hamming Weight of $\vec{c} = \sum_{j=1}^{n} c_j$. The Hamming distance of two code words $\vec{c_1}$ and $\vec{c_2}$, denoted by $\mathrm{Hd}(\vec{c_1}, \vec{c_2})$, is the number of bits in which they differ. That is $\mathrm{Hd}(\vec{c_1}, \vec{c_2}) = \mathrm{Hw}(\vec{c_1} \oplus \vec{c_2})$. The minimal distance of a code C is the value $\mathrm{d}(C) = \min_{\vec{c_1}, \vec{c_2} \in C}(\mathrm{Hd}(\vec{c_1}, \vec{c_2}))$.

Majority decoding: Let $\vec{c_i} = c_{i,1}||c_{i,2}|| \ldots ||c_{i,n}$ be n bit code vectors. Given odd m vectors $\vec{c_i}$, a majority decoder computes vector $\vec{C} = C_1||C_2 \ldots ||C_n$, where $C_j = \mathrm{majority}(c_{1,j}, \ldots, c_{m,j})$, i.e., C_j is the majority of 0's or 1's of bit j from each of the vectors. We shall use majority decoding primarily to get the best biometric reading possible, thus reducing the Hamming distance between successive *final* readings $\vec{C}$.

Algebraic decoding: An (N, K, D) code is a code of N bit codewords (vectors) where K is the number of information digits and D is the minimum distance of code. It should be noted that an error correcting code ECC with rate K/N can correct $T = (D-1)/2$ errors.

An (N, K, D) code can be represented by a $K \times N$ generator matrix G of dimension K. G is said to be in canonical form if G has the form

$$G = [I_{K \times K} \ : \ P]$$

where I is a $K \times K$ identity matrix and P is a $K \times (N-K)$ sub-matrix. An information vector $\vec{U}$ of K bits is encoded into a code vector $\vec{V} = \vec{U} \cdot G$. $\vec{V}$ has the form $\left[\vec{U} \ : \ \vec{C}\right]$, where $\vec{C}$ is a vector of check digits of size $N-K$. Alternately, given a *generator* (binary) polynomial $G(X)$ over $GF(2)$ for a cyclic (binary) code, one can encode $U(X)$ into a codeword $V(X) = X^{N-K}U(X) + (X^{N-K}U(X)) \bmod G(X)$.

[1] Let || denote string concatenation.

3.2.1 Bounded Distance Decoding

To allow for error correction of a biometric, we encode a K bit biometric into an N bit code vector, with $N-K$ redundant (or check) digits.

The description of an (N, K, D) error correcting code with rate $K/N > \frac{1}{2}$, (*using bounded distance decoding of up to* $\frac{D-1}{2}$ *errors*), is provided to the authorization officer and biometric readers. To ensure that an impostor is not accepted, it is important to set the error correction capability of the error correcting code to a level that prevents an impostor's biometric from being "corrected" into a valid biometric (i.e., that no more than the allowed number of errors will be corrected).

4 Identification Scheme Assuming Public Biometrics

We now discuss an off-line identification protocol in which we assume that there is no requirement to hide one's biometric. Based on the reproduction assumption, the protocol below only protects against an adversary trying to prove that its potentially falsified biometric is the same as one signed by the authorization officer. Hence, we assume that the biometric can be read with sufficient accuracy in the amount of time available for the scan (possible with majority decoding, as discussed in Section 3.2) such that an (N, K, D) algebraic code will suffice to remove the remaining errors from the biometric[2].

The protocol below provides a framework for the rest of our discussion:

System Setup: The authorization officer generates its public and private keys and disseminates its public key to the biometric readers. The system also sets up an algebraic (N, K, D) code.

User Initialization: To register, M biometric templates of length K are independently generated for the user. These M vectors are put through a majority decoder to obtain the user's K bit template $\vec{T}$. Given the K information bits $\vec{T}$, an N bit codeword $\vec{T}||\vec{C}$ is constructed, where $\vec{C}$ are the check bits in the (N, K, D) code defined in system setup. The following four items go on the card:

1. Name of the individual, NAME
2. Other public attributes ATTR, such as the issuing center and a user's access control list
3. Check digits $\vec{C}$
4. Sig(NAME, ATTR, $\vec{T}$), where Sig(x) denotes the authorization officer's signature of x.

Biometric Authorization Process (verification) When a user presents a card, M biometric templates are independently generated for the user.

[2] Recall that the code is set up such that it can remove enough errors to allow the system to recognize the legitimate user of the card but not someone else, i.e. bounded distance decoding.

These M vectors are put through majority decoding and bounded distance decoding using the check digits $\vec{C}$ to obtain the user's K bit current reading $\vec{T}'$. Then Sig(NAME, ATTR, $\vec{T}$) is verified with the authorization officer's public key and message NAME, ATTR, $\vec{T}'$. Successful signature verification implies successful user identification.

5 Identification Schemes with Private Templates

We now discuss several off-line identification protocols. We remind the reader that in the model discussed in Section 2.2, the user obtains a storage device containing information on the user's template and a secure authenticated binding with an authorization officer. The two trivial cases are when there exists a private key in the reader and when password protection is used (See Section 2.1).

For the rest of this section we make the following additional assumption:

Assumption 2 Privacy: *It is assumed that a digital representation of the biometric cannot be produced with sufficient accuracy to pass a biometric authorization process (with respect to a user's template only and not to other biological tests such as pupilary unrest) without the cooperation of the subject involved. Hence, we assume that the biometric being measured can only come from an individual submitting to the measurement.*

We therefore now assume that there is a strong physical binding of a biometric to an individual, and that the biometric template cannot be "taken" (copied, stored, etc.) readily. Observe that information held by only one person and not obtainable by others is a property of a private key. This assumption inspired us to investigate how biometrics can enable cryptographic mechanisms.

One may argue that this assumption 2 is not acceptable, especially against a strong adversary. But in practice, much as passwords protect computer systems, this assumption can be beneficial for systems whose adversaries do not have such strengths. Moreover, if one does not accept this assumption, then one should also not believe that biometric information should be kept confidential, since it is readily available anyway.

5.1 Private Biometric

We now discuss an off-line biometric system which provides for privacy of a user's biometric, assuming the privacy assumption holds and sufficient entropy in biometric templates.

System Setup: The authorization officer generates its public and private keys and disseminates its public key to the biometric readers. The system also sets up an algebraic (N, K, D) code. We remind the reader that we use bounded distance decoding (See Section 3.2.1).

User Initialization: To register, M biometric templates of length K are independently generated for the legitimate user. These M vectors are put through a majority decoder to obtain the user's K bit template $\vec{T}$. Given the K information digits $\vec{T}$, an N digit codeword $\vec{E} = \vec{T}||\vec{C}$ is constructed, where $\vec{C}$ are the check digits, in the (N, K, D) code defined in system setup. A storage device is constructed with the following information:

1. Name of the individual, NAME
2. Other public attributes ATTR, such as the issuing center and a user's access control list.
3. The check digits $\vec{C}$, of the biometric
4. Sig(Hash(NAME, ATTR, $\vec{T}||\vec{C}$)) where Sig(x) denotes the authorization officer's signature of x, and Hash($\cdot$) is a partial information hiding hash function [4] (e.g., Sig(Hash($\cdot$)) is a content-hiding signature) or a random oracle (See [1]).

Biometric authorization process (verification): When a user presents a card, M biometric templates are independently generated for the user. These M vectors are put through majority decoding to obtain the user's K bit template $\vec{T}'$. Error correction is performed on codeword $\vec{E}' = \vec{T}'||\vec{C}$ to obtain the corrected biometric $\vec{T}''$. The signature Sig(Hash(NAME, ATTR, $\vec{T}''||\vec{C}$)) is then verified. Successful signature verification implies the user passed the identification step.

We next prove the correctness and security of the protocols above.

Theorem 1 *The biometric identification system above correctly accepts a valid subject whose $\vec{T}'$ has less than $\frac{D}{2}$ errors.*

Proof. Let $\vec{T}'$ be a scanned biometric, using majority decoding of M readings. Applying the (N, K, D) algebraic decoding to $E' = \vec{T}'||\vec{C}$ we obtain the corrected biometric $\vec{T}''$. If $\vec{E}'$ has less than $\frac{D}{2}$ errors, then E' is correctly decoded, resulting in a corrected biometric $\vec{T}''$ that matches the original biometric $\vec{T}$. The signature is then verified using the public key of the authorization officer. □

Theorem 2 *If an imposter is accepted, the reproduction assumption is violated or the signature scheme forgeable.*

Proof. **(Sketch)** This proof reduces to two cases. First, if the information on the memory device was at some time signed by the authorization officer (whether on this card or another one), then being accepted implies that either:

- Displaying a biometric which is close enough (within bound defined for the biometric) to the one that was signed by the authorization officer invalidates the reproduction assumption.

- The signature scheme accepts two different messages, implying the signature scheme is forgeable.

The other case is that the information on the memory card was not at some time signed by the authorization officer, but this case also reduces to the signature scheme being forgeable. □

We now argue the privacy of our system. First, the hash is necessary when one does not know if the signature system leaks information about its input. Therefore, in order not to have an information hiding requirement of the signature function, we incorporate a random oracle or a partial information hiding hash function.

We cannot make the standard cryptographic reduction proof showing a polynomial time adversary is unable to attack the system. A reduction proof could be achieved if we assume that one can develop a biometric system in which the entropy in templates grows as the security parameter of the system grows. (That is, the reader can make finer and finer readings with the growth of a security parameter.) Without such an assumption, there is a "constant" size of uncertainty (remember we do not assume the reader has a private key or other private information) on the storage device representing the biometric information. As the security parameter grows, the adversary is able to eventually try all possibilities and check for correctness using the authorization officer's authentication information.

We can argue that since the hash function is a random oracle or partial information hiding hash function, the signature leaks no information. The check bits leak, as a conservative estimate, $N - K$ bits of information, which is small. As will be shown for iris scans (See Section 6), the entropy of the biometric template is around 173 bits. By applying majority decoding in the biometric reading process, one can use an algebraic code with $N = 2074$ and $K = 2048$, leaving 147 bits of entropy.

5.2 Biometrics as an Enabler

If the biometric has sufficient entropy, than the biometric itself can be used as a key. In fact, the template becomes a key for encrypting other private keys and private information. Thus, biometrics can be an enabler of cryptographic functions, if there exists sufficient entropy in the biometrics.

We are able to enable cryptographic applications through biometrics, since biometrics can hide private information such as keys. It may be worthwhile to encrypt other valuable information, such as cryptographic keys (Keys), private attributes (Private) including private access control lists, and other biometric information (Bio) including physical descriptions (e.g., Brown hair, Hazel eyes, 5' 11", 200 lbs.).

There, of course, is concern that a biometric is a lifetime key that cannot be revoked easily. Therefore, we suggest augumenting passwords, PINs, etc., with biometric entropy, in essence taking multiple sources with weak entropy to produce a key with a larger entropy. We included a PIN in this protocol to allow the user to add entropy into the final key. This addition is especially important when one does not believe in the privacy assumption.

Let UOWHF denote a universal one way hash function[14] and K_A be a key for application A known by the reader and the authorization officer for application A. Instead of a signature as in item 4 in the protocol from Section 5.1, the following encryption is encoded for each application A (where K_A is application A's private key and PIN_A is the user's PIN for application A).

New item 4. $\text{enc}_A =$
$\text{ENC}_{\text{UOWHF}_{K_A}(\text{PIN}_A,\vec{T})}(\text{Keys}, \text{PrivateBio}, \text{Sig}(msg))$
where
$\text{Sig}(msg) =$
$\text{Sig}(\text{NAME}, \text{ATTR}, \text{Keys}, \text{PrivateBio}, \text{Hash}(\vec{T}||\vec{C}))$.

Correctness of the above is trivial to prove. Informally we can prove security in a manner similar to that used in the last section. Moreover, privacy of the private attributes is due to the large entropy of either PIN_A and/or $\vec{T}$ and the security of the encryption scheme. The UOWHF maximizes the amount of entropy obtained from combining the PIN and template as a key.

Incorporating Multiple Biometrics When faced with adversaries with sufficient motivation and resources, Assumption 2 and even Assumption 1 may be called into question on a given biometric. To address such situations one can extend the previous work to provide support for two or more biometrics.

6 Iris Scan Biometric

As discussed above, our scheme depends on the existence of biometric systems that reduce a stable characteristic of individuals to a binary encoding with high entropy and significant Hamming distance between individuals. One such system that has received extensive study is iris scans [3, 12, 6, 7, 21].

The human iris is the colorful doughnut-shaped organ surrounding the pupil, as distinguished from the retina which is the hemispherical organ behind the cornea, lens, iris and pupil. The iris has highly detailed texture and is unique for each individual, differing between identical twins and between left and right eyes of the same individual. It has been determined that the iris imparts the same singularity to individuals as does the fingerprint [6].

A biometric system developed by IriScan Inc. performs the following functions to acquire an iris scan. When a user presents himself/herself, the system performs image analysis to determine if an iris is visible, the degree of occlusion of the iris by the eyelid, and the degree of spectral reflection; it also assesses the quality of the focus and locates the iris. The system adjusts for pupillary constriction, overall image size, head tilt and cyclovergence of the eye.

The system then proceeds to compute the encoding (scan) for the iris.

6.1 Remarks on Scan Sizes and Iris Scan Time

In [6, 21] it has been found that reliable iris scans can be computed from an individual in about 100 milliseconds. The scans that are computed are 256-byte vectors. These vectors have an error rate of 10 percent; that is, for a given user, repeated sampling results in biometric vectors that have a Hamming distance of 10 percent on the average. Thus one can say a vector has an "error" of about 10%. In the discussion above, we considered multiple scans and majority decoding to reduce the "errors" in the scan. If the time needed for multiple scans is prohibitive for an application, then one can reduce the need for costly error correction by reducing the size of the scanned vector. The 256-byte vectors have a high degree of redundancy. It has been determined empirically that $H(IRSSCANS) \approx 173$ bits. This entropy guarantees that iris scans have a probability of duplicates of about 1 in 10^{52}. Given that the entropy is large, it is possible to reduce the size of the scanned vector T without reducing the selectivity of the scans among the world population.

Consider the final scanned vector T. We then compute a reduced vector T' as follows:

1. Apply a permutation to the vector T
2. Let $T' =$ least L bits of T

Reducing the size of the scanned vector has the advantage of reducing the cost and time of the identification systems. If the time to perform a scan (about 100 milliseconds) is not an undue burden for the application, then multiple scans of the iris result in the following error rates, using majority decoding.

No. of scans	Per bit prob. of error	Expected no. of errors in a 2Kb scan
1	0.1	205
3	0.028	58
11	0.000306	1
21	0.00000135	.002

Acknowledgements

The authors wish to acknowledge Dale McDermott for improving the readability of this paper, Moti Yung for several helpful discussions, and the anonymous referees for their comments.

References

[1] M. Bellare and R. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computers and Communications Security*, 1993.

[2] E. R. Berlekamp. *Algebraic Coding Theory.* McGraw-Hill, 1968.

[3] F. Bouchier, J. S. Ahrens, and G. Wells. Laboratory evaluation of the iriscan prototype biometric identifier. Technical Report SAND96-1033, Sandia National Laboratories USA, April 1996.

[4] R. Canetti. Towards realizing random oracles: Hash functions which hide all partial information. In *"Advances in Cryptology. Proc. of Crypto'97*, pages 455–469, 1997.

[5] D. Chaum. Security without identification:transaction systems to make big brother obsolete. *Communication of the ACM*, 28(10):1030–1044, 1985.

[6] J. Daugman. High confidence personal identifications by rapid video analysis of iris texture. In *IEEE International Carnahan Conference on Security Technology*, pages 50–60, 1992.

[7] J. Daugman. High confidence personal identifications by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):648–656, November 1993.

[8] U. Feige, A. Fiat, and A. Shamir. Zero knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988.

[9] Security requirements for cryptographic modules(FIPS PUB 140-1). Technical Report FIPS 140-1, National Institute of Standards and Technology, Gaithersburg, MD, 1994.

[10] Guideline for the use of advanced authentication technology(FIPS PUB 190). Technical Report FIPS 190, National Institute of Standards and Technology, Gaithersburg, MD, 1994.

[11] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.

[12] J. P. Holmes, R. L. Maxell, and L. J. Wright. A performance evaluation of biometric identification devices. Technical report, Sandia National Laboratories, July 1990.

[13] F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes.* North – Holland Publishing Company, 1978.

[14] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 33–43, 1989.

[15] W. W. Peterson and E. J. Weldon. *Error Correcting Codes.* The MIT Press, 1988.

[16] Password-based encryption standard (PKCS5). Technical Report PKCS 5, RSA Laboratories, Redwood City, CA, 1993.

[17] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology. Proc. of Crypto'84 (Lecture Notes in Computer Science 196)*, pages 47–53. Springer–Verlag, 1985. Santa Barbara, California, U.S.A., August 19 – 22.

[18] A. Shamir. Interactive identification, March 23–29, 1986. Presented at the Workshop on Algorithms, Randomness and Complexity, Centre International de Rencontres Mathématiques (CIRM), Luminy (Marseille), France.

[19] C. E. Shannon. A mathematical theory of secrect systems. *Bell System Technical Journal*, 28:656–715, 1949.

[20] C. Soutar and G. J. Tomko. Secure private key generation using a fingerprint. In *CardTech/SecurTech Conference Proceedings Vol. 1*, pages 245–252, May 1996.

[21] G. O. Williams. Iris recognition technology. In *IEEE International Carnahan Conference on Security Technology*, pages 46–59, 1996.

[22] The directory - authentication framework. - X.509, International Telecommunications Union, Geneva, Switzerland, 1993.

Report Number (14) SAND--98-0989C
CONF-980534--

Publ. Date (11) 199804
Sponsor Code (18) DOE/DP, XF
UC Category (19) UC-706, DOE/ER

DOE