PRIVACY PRESERVING MULTI-FACTOR AUTHENTICATION WITH BIOMETRICS

bhilasha Bhargav-Spantzel and Anna C. Squicciarini and Elisa Bertino and Shimon Modi and Matthew Young and Stephan

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

Privacy Preserving Multi-Factor Authentication with Biometrics

Abhilasha Bhargav-Spantzel [a], Anna C. Squicciarini [a], Shimon Modi [b],
Matthew Young [b], Elisa Bertino [a], and Stephen J. Elliott [b]

[a] *Department of Computer Science, Purdue University*
[b] *Department of Industrial Technology, Purdue University*

Abstract. An emerging approach to the problem of identity theft is represented by the adoption of biometric authentication systems. Such systems however present several challenges, related to privacy, reliability and security of the biometric data. Inter-operability is also required among the devices used for authentication. Moreover, very often biometric authentication in itself is not sufficient as a conclusive proof of identity and has to be complemented with multiple other proofs of identity such as passwords, SSN, or other user identifiers. Multi-factor authentication mechanisms are thus required to enforce strong authentication based on the biometric and identifiers of other nature.

In this paper we propose a two-phase authentication mechanism for federated identity management systems. The first phase consists of a two-factor biometric authentication based on zero knowledge proofs. We employ techniques from the vector-space model to generate cryptographic biometric keys. These keys are kept secret, thus preserving the confidentiality of the biometric data, and at the same time exploit the advantages of biometric authentication. The second phase combines several authentication factors in conjunction with the biometric to provide a strong authentication. A key advantage of our approach is that any unanticipated combination of factors can be used. Such authentication system leverages the information of the user that are available from the federated identity management system.

Keywords. Identity Management, Biometric, Security, Privacy

Introduction

The problem of identity theft, that is, the act of impersonating others' identities by presenting stolen identifiers or proofs of identities, has been receiving increasing attention because of its high financial and social costs. Recent federated digital identity management systems (IdM) (or identity federation standards) do not aim at providing strong authentication but simply provide ways to convey authentication information, including that of strong authentication. One approach to such problem is the adoption of biometric *authentication* systems. These systems are automated methods for recognizing an individual based on some physiologically and behavioral characteristics, such as fingerprints, voice, or facial features.

Biometric authentication[1] provides some inherent advantages as compared to other non-biometric identifiers since biometric characteristics correspond to a direct evidence of the personal identity versus possession of secrets which can be potentially stolen. Moreover, most of the times biometric enrolment is executed in-person and in controlled environments making it very reliable for subsequent use.

Challenges in Biometric Authentication. Biometric authentication poses however several non-trivial security challenges because of the inherent features of the biometric data itself. Addressing these challenges is crucial for the large scale adoption of biometric authentication and its integration with other authentication techniques and with access control systems.

Biometric matching is probabilistic in nature, which means that two samples of the same individual are never exactly the same. If the two samples are encrypted for security reasons, they need to be decrypted before they can be matched. Unlike some password systems that perform a one-way hash function on the user input, biometric systems cannot rely on the same process. The reason is that the hash values will never be the same for the reference template value and current presented sample. This also raises the issue of key management to enable decryption, and represents a vulnerability point in the process. Moreover, it is very hard to revoke and change biometrics in case biometric data are compromised. At the time of enrolment the individual's biometric sample is processed into a template to be used for subsequent verification or identification attempts. This template is in the form of digital data and often stored in a database or on a token. These templates are often vendor-specific and therefore the interoperable use of such templates in a distributed system is very difficult.

Biometric authentication from an unsupervised location also presents the possibility for sensor spoofing attacks. The credibility of the output from a biometric matching process depends entirely on the integrity of the sample provided, and whether is a true sample provided by the owner of the biometric characteristic. Older generation biometric capture devices were vulnerable to spoofing attacks, and there is extensive work currently on-going to mitigate biometric sensor spoofing.

Biometric authentication can be implemented through systems performing the matching either on the *server* or on the *client side*. Depending on where the matching of the biometric template is executed - at the server or at the client - different security problems arise. In the former case the main issues are related with the large scale and distributed management of biometric templates. The creation of a database of a particular biometric at the server should itself be secure and possibly decentralized. Also, such database may depend on a particular template creation and matching algorithm as well as hardware and thus may not be interoperable. Such a system would also be CPU-intensive because of the matching operations.

Additionally, storing biometric information in repositories along with other personally identifiable information raises several security and privacy risks [21]. These databases are vulnerable to attacks by insiders or external adversaries and may be searched or used outside of their intended purposes. It is important to note that if the stored biometric

[1] Note that within the biometric community authentication is more specifically referred to as either identification or verification. In verification user makes a claim to identity and then matching of the user sample presented to the system is executed on a one-to-one basis. Identification, on the other hand, does not require a claim to identity; therefore the current sample is compared against a large number of templates in the database until a match is found. In this paper we mainly refer to authentication as verification.

identifiers of an individual are compromised, there will be severe consequences for the individual because of the lack of revocation mechanisms for biometrics.

Due to the security and privacy problems of server side matching, several efforts in biometric authentication technology have tried to develop techniques based on client side matching [34,33]. Such an approach is convenient as it is relatively simple and cheap to build biometric authentication systems supporting biometric storage at the client end able to support local matching. Nevertheless, systems of such type are not secure if the client device is not trusted; therefore additional security mechanisms are needed.

Several efforts have been undertaken to strengthen client side authentication. Previous approaches [9,27] have been developed based on Chaum and Pederson wallet-with-observer paradigm [15]. An interesting approach recently proposed focuses on key extraction from biometrics which entails the problem of "approximate equality" in biometric comparisons. Several approaches have been proposed for overcoming this difficulty, including the use of error-correcting codes [20], fuzzy commitments and fuzzy vaults [34,33] and fuzzy extractors [22]. However, several of these schemes may be vulnerable to replay attacks, non-repudiation and cryptanalysis.

Client side authentication systems also led to research on key generation mechanisms that use biometrics [48,22,35]. Key generation is executed by first extracting the biometric features from the biometric data based on the feature extraction module of the biometric authentication system. Then, the biometric features are sent to the system specific key-generation module to generate a key, that we refer to as*biometric-key*. The challenge in such an approach is to devise algorithms for reliable key generation. Such key generation algorithms must be able to generate the same key despite the noise in biometric readings. Moreover, the semantics of the usage of such a key should still retain the property of *"what you are"* versus *"what you have"*.

Desiderata. Based on the previous discussion we identify several crucial properties of a suitable biometric authentication system. The system must:

- preserve the privacy of biometric data;
- be convenient to use and interoperable with different authentication servers thus providing scalability;
- be able to perform client-side matching without requiring tamperproof or trusted hardware;
- support revocation of the biometric credentials;
- be resilient to the compromise of the biometric template itself;
- be resilient to replay attacks so that the replay of the biometric signal or the key generated based on the biometric cannot result in successful authentication;
- provide security for any cryptographic token associated with the biometric and efficiently manage keys;
- provide non-repudiation of biometric authentication credentials and accountability.

Our approach. In this paper we cast the problem of biometric authentication in the specific context of federated IdM systems [26,29,45], which typically rely on attributes and properties of the member users to enforce authentication. Our main objective is to achieve a privacy preserving methodology, in which use of credentials and biometric is achieved without loss or exposure of additional data. As such, assuring privacy in our context consists of providing anonymous and unlinkable authentications, even with the use of unique biometric credentials. The first problem we have to deal with in our effort toward a methodology for biometric authentication in federated systems is related with interoperability. If on one hand federated digital IdM systems need to support data heterogeneity, on the other hand, biometric vendors typically generate proprietary templates which may not be interoperable among each other. Obviously, this represents a major limitation to the large scale deployment of federated systems. In order to address this problem while at the same time assuring privacy, we develop a cryptographic key generation algorithm for use at the client side, which can thenceforth be used with other clients in the federation. Precisely, in this paper we use mechanisms from vector space modeling [46] to generate cryptographic biometric-keys.

To further preserve privacy we provide authentication protocols based on well known techniques called zero knowledge proof of knowledge (ZKPK's for brevity) [8,13]. ZKPK's allow a user to have a private secret, and prove its possession without releasing it. As such, biometric-keys are never released but are instead used to generate a *proof* of the ownership of the biometric. This proof is sufficient for the purposes of authentication as it would correspond to the biometric enrolled in the system. The use of ZKPK proof enables us to have a *two-factor* authentication by using information theoretically secure Pedersens commitments [37]. Using the Pedersens commitments, the biometric-key and an associated random secret do not have to be verified separately by the authenticating party, instead the respective values can be verified together in one ZKP. Such commitments also elegantly handle revocation of the generated biometric-keys.

We also show how the two-factors can be combined with other identity information available in the federation to provide *multi-factor authentication*. Providing proof of the biometric identifier itself is not sufficient as the proof of knowledge of other sensitive identifiers like social security number (SSN) or the credit card number (CCN) can be required to complete the authentication procedure. Note that our multi-factor authentication methodology can be performed very efficiently using aggregate zero-knowledge proofs introduced in [4]. Here, each of the factors does not have to be verified separately, and can be verified using a single ZKP. This results in significant improvement in terms of cost of deployment and efficiency as compared to a system which requires the verification of each of the factors separately. The following example introduces a scenario which we will use to illustrate the authentication phases.

Example 1 *Consider a federation including a Bank $CityBank$, and a Tax Authority $TaxAuthr$. $CityBank$ is the local bank for the user $Alice$ and contains all financial information concerning $Alice$. She also enrolls other information with the bank which can be potentially used at the time of authentication. $CityBank$ is essentially Alice's local identity provider.*

$Alice$ wants to fill her tax on line with $TaxAuthr$. However $TaxAuthr$ requires its on-line users to authenticate using two-factor biometric authentication to access such service. Further, if Alice wants to do money transactions, then $TaxAuthr$ requires her to perform multi-factor authentication by providing proof of ownership of a registered 1) biometric 2) CCN and 3) verified SSN. Thus depending on the service requested, Alice would either need to perform two-factor or multi-factor authentication.

The main contributions of our work can be summarized as follows. We propose a novel method for key-generation using vector-space modeling. This is a generic methodology to generate biometric-key where the vectors used could correspond to any defined combination of one or more biometrics. As compared to existing biometric-key generation work, our approach differs in how the biometric-keys are actually used. We describe the complete life-cycle of such biometric-keys and the protocols involved in each step of the life-cycle, namely biometric-key generation, usage and revocation. The actual key is never revealed as it could possibly leak further information about the individual. We therefore use it for ZKPK which directly assures unlinkability and replay avoidance. We provide a two-phase authentication mechanism for federated IdM systems. The first phase consists of a two-factor authentication with biometric data, and the second of a multi-factor authentication involving other user attributes. We show how we can use our protocols to secure biometric data itself, thus preventing its fraudulent use that would result in identity theft and other security and privacy breeches. In addition, we provide an entropy based argument showing the advantage of biometric-keys as compared to traditional passwords. We show how the strength of the secret can be significantly increased by the combination of biometric-keys with the traditional secrets. We also show how privacy preserving multi-factor authentication can be enforced in federated IdM systems with the ability to use biometric data just like the other identifying attributes of the user. Our approach is privacy preserving in that all the authentication steps are executed with limited disclosure of data that cannot be used for any other purpose other than the authentication decision itself.

The paper is organized as follows. In Section 1 we provide basic background information regarding biometric authentication followed by Section 2 where we provide a brief review of authentication in federated IdM systems. In Section 3 we describe our approach based on the biometric-key life-cycle. In Section 4 we present our key protocols for biometric authentication. In particular in Section 4.1 we elaborate on our approach towards biometric key generation, followed by Section 4.2 where we show how it can be used in ZKPK. In Section 4.3 we describe the revocation mechanisms of the enrolled biometric commitments. We provide an entropy based comparison of keys generated by biometrics versus passwords in Section 5. In Section 6 we show how we can support strong authentication using our multi-factor technique. Finally, we present a detailed analysis of our approach in Section 7 and then outline some conclusions.

1. Biometric Authentication Background

Biometric authentication adds a new paradigm in user authentication which, unlike conventional approaches, is not based on what an individual knows or possesses, but on some characteristics of the individual itself. We elaborate on the main concepts related to biometric in this section.

A detailed reference model for a biometric system has been developed by ISO/IEC JTC1 SC37 [42], which aides in describing the sub-processes, and the flow chart of a biometric system (See Figure 1). Typically there are four main subsystems in the biometric model, namely the *Data Capture, Signal Processing, Data Storage, Matching* and *Decision* subsystems.

- *Data capture subsystem:* It collects the subject's biometric data in the form of a sample that the subject has presented to the biometric sensor.
- *Signal processing subsystem:* It extracts the distinguishing features from a biometric sample to then either be stored as the reference template during registration

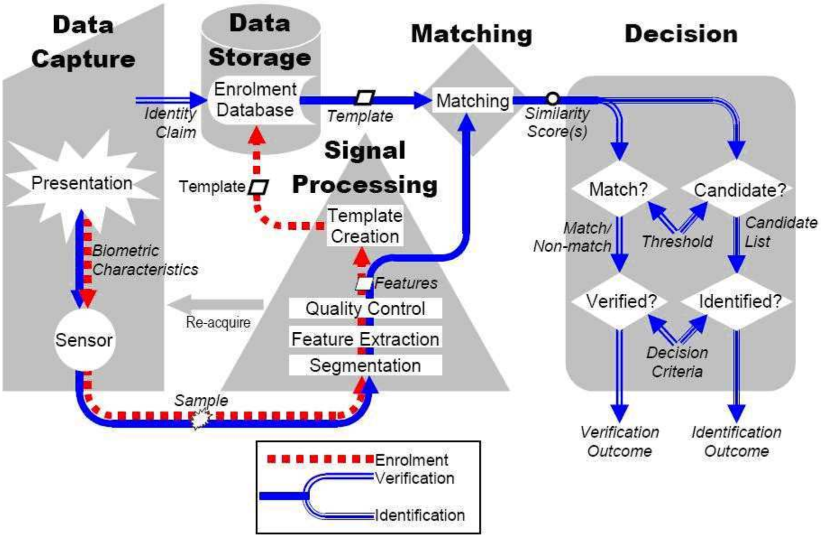
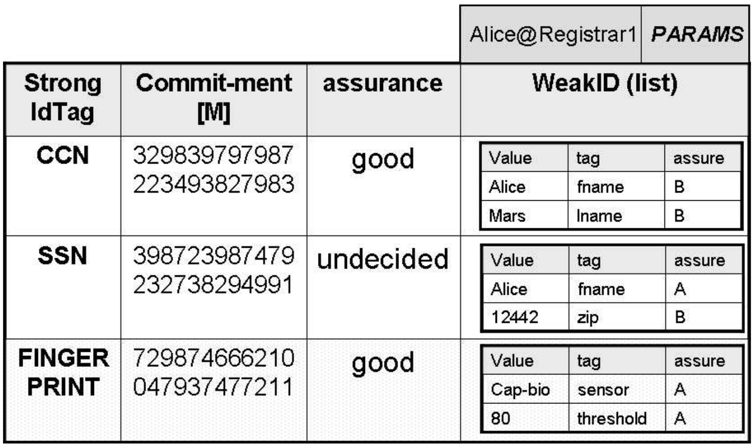


Figure 1. Reference Model for Generalized Biometric System developed by [42].

or be matched during verification and identification. A template is data, which represents the biometric measurement of an individual, used by a biometric system directly or indirectly for comparison against other biometric samples.

- *Data storage subsystem:* Reference templates are stored either at the server or at the client depending on the chosen architecture.
- *Matching subsystem:* It compares the features extracted from the captured biometric sample against one or more enrolment reference templates, the obtained similarity scores are then passed to the decision subsystem.
- *Decision subsystem:* It uses the similarity scores generated from one or more matching comparisons to make a decision about a verification or identification transaction. The features are considered to match a compared template when the similarity score exceeds a specified threshold.

A biometric system typically supports two sub-processes: registration (also called enrolment), and authentication (see Figure 1). *Enrolment* is the process of capturing the features from a biometric sample provided by an individual and converting it into a template. The effectiveness of enrolment strictly depends on the quality of the data submitted along with the biometric. Thus, the enrolment process has also to ensure that the verification documents (like passports and driver's licenses) are trustworthy so that a fake or false identity is not linked to a biometric. Additionally, no duplicate records have to be stored in the database for the same identity. Such enrolment mechanism is a key aspect of biometric authentication making it very reliable. Enrolment is the first interaction of the user with the biometric system, and misuses of such operation can affect the quality of sample being provided by the user, which in turn affects the overall performance of the system. Once the process of registration is successfully completed, the individual can use the biometric system for authentication. The *authentication* is performed when the individual presents his/her biometric sample along with some other identifier which



<table>
<tr><td></td><td></td><td></td><td>Alice@Registrar1 <i>PARAMS</i></td></tr>
<tr><th>Strong IdTag</th><th>Commit-ment [M]</th><th>assurance</th><th>WeakID (list)</th></tr>
<tr><td>CCN</td><td>329839797987
223493827983</td><td>good</td><td><table><tr><td>Value</td><td>tag</td><td>assure</td></tr><tr><td>Alice</td><td>fname</td><td>B</td></tr><tr><td>Mars</td><td>lname</td><td>B</td></tr></table></td></tr>
<tr><td>SSN</td><td>398723987479
232738294991</td><td>undecided</td><td><table><tr><td>Value</td><td>tag</td><td>assure</td></tr><tr><td>Alice</td><td>fname</td><td>A</td></tr><tr><td>12442</td><td>zip</td><td>B</td></tr></table></td></tr>
<tr><td>FINGER PRINT</td><td>729874666210
047937477211</td><td>good</td><td><table><tr><td>Value</td><td>tag</td><td>assure</td></tr><tr><td>Cap-bio</td><td>sensor</td><td>A</td></tr><tr><td>80</td><td>threshold</td><td>A</td></tr></table></td></tr>
</table>

Figure 2. Identity Record Example

uniquely ties a template with that individual. The matching process is performed against only that template.

2. Authentication in Federated Identity Management Systems

Digital identity can be defined as the digital representation of the information known about a specific individual or organization. As such it encompasses, not only login names, but many additional information, referred to as *identity attributes* or *identifiers*, about users. Managing identity attributes raises a number of challenges, due to conflicting requirements. On the one hand, identifiers need to be shared to speed up and facilitate authentication of users and access control. On the other hand, identity attributes need to be protected as they may convey sensitive information about an individual and can be targets of attacks like *identity theft*. In cyberspace preventing identity theft is especially hard because digital information can be copied, hence stolen unnoticed. Identity of an individual can be represented through different types and combinations of identifiers. Therefore when reasoning about authentication it is important to consider a combination of multiple identifiers. Such an authentication mechanism is called *multi-factor authentication* and is a prevalent mechanism to mitigate identity theft threats.

An emerging approach to address issues related to identity management is based on the notion of *federations* [26,29,45]. The goal of federations is to provide users with protected environments to federate identities by the proper management of identity attributes. Federations provide a controlled method by which federation members can provide more integrated and complete services to a qualified group of individuals within certain sets of business transactions.

Federations are usually composed by two main entities: identity providers (IdP's), managing identities of individuals, and service providers (SP's), offering services to registered individuals. In a typical federated IdM system the individual registers with his/her local IdP and is assigned a username and password. Registration is usually based on an in-person verification at some registration office. Based on this information a registered individual can submit additional attributes and its corresponding attribute release policies, which are stored at the local IdP. The IdP is then contacted whenever the user interacts with any other SP in the federation when additional user information is needed. The IdP is mostly in charge of sending the SP the submitted user attributes in accordance to the attribute release policies. Note that IdP's and SP's are logically distinct, but they may correspond to the same organization. In fact, in some federation standards [26] any entity can play the role of an attribute provider at any given instance. In [8] a third type of entity was introduced, referred to as *Registrar*, which essentially captures the notion of proofs of identity for user private and sensitive attributes like SSN and Credit card numbers. These proofs are based on ZKPK protocols and Section 4 describes a modified version of the ZKPK which is specific for biometrics. Like the biometric system, the proposed IdM system also supports two main phases, namely enrolment and authentication. At the time of enrolment the zero knowledge commitments of the strong identifiers[2] are recorded with additional meta information about the state of the identifier. This record, referred to *identity record*, is stored under the single sign-on (SSO) identifier of the individual. The SSO ID is used by the user to authenticate itself to its Registrar, after which it can seamlessly access resources provided by the federation SP's if it satisfies the authorization requirements. Using the Registrar services the user can provide strong assurance of its identity as needed by the transactions. An example of an identity record (IdR) is shown in Figure 2. Typically the identifiers recorded in an identity record correspond to *what you know*, like CCN and SSN[3]. We show how we can use identifiers corresponding to *who you are* in such an identity record, as illustrated in the last row of Figure 2 with the tag "finger print".

Our goal is to use biometrics in combination with the other committed identifiers in the identity record at the time of authentication. As such, at the time of authentication, multiple commitments of strong identifiers and biometric commitments can be combined in an ad-hoc fashion, and verified by using exactly the same approach developed for the case on non-biometric data. This mechanism thus allows an individual to prove the knowledge of such proven identifiers. In the following sections we illustrate how biometric readings can be used so that the extracted information can be used across the federation, thus achieving interoperability.

3. Our Approach to Biometric-keys

In the following we outline our approach to the generation and usage of biometric-keys with the help of a biometric-key life-cycle (see Figure 3). There are four basic stages in such life-cycle: learning, generation, usage and revocation. Before a biometric-key can be generated, the biometric data needs to be analyzed to identify the various features to use at the time of key generation. This step, referred to as learning stage, is executed once by the providers of the biometric authentication system. The output of the learning stage is a list of features. These features are then tested with a database of actual biometric data to make sure that they satisfy the uniqueness and repeatability of the resultant biometric-key generated by those features. Uniqueness of biometric-keys is required to ensure that

[2] Strong identifiers are those that uniquely identify an individual. This is also known has personally identifiable information.

[3] If the enrolment is done in-person with the verification of the actual cards corresponding to the identifiers then those entries in the IdR would also correspond to *what you have.*

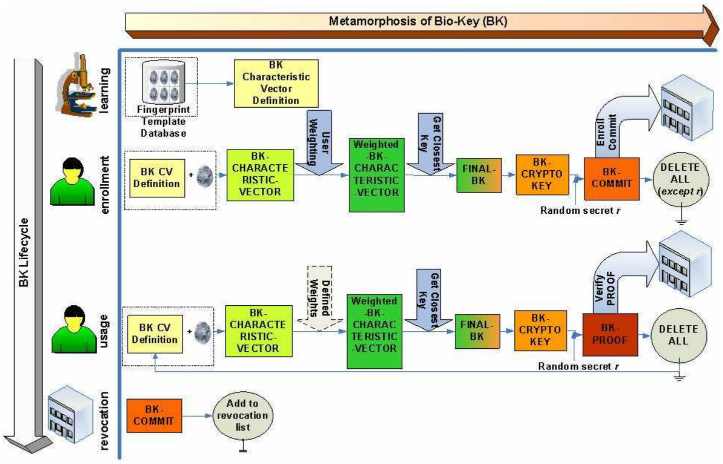


Figure 3. Biometric-key lifecycle stages and the metamorphosis of the biometric-key at each stage

two different individuals do not generate the same biometric-key. Repeatability, on the other hand, refers to the ability to re-generate the user's biometric-key by that user. The learning phase is executed only once or whenever there are significant changes in the user population, and the result provided to the users before they can enroll their biometric-key in the authentication system.

A particular user's biometric-key is generated at the time of user's enrolment. Depending on the policies of the authenticating party, the enrolment can be executed in-person at a (secure) physical location of the authenticating party or online. If the authenticating party wants to control which biometric characteristics are used, the enrolment needs to be executed in-person. However, it is possible that the authentication party just needs a key and does not prescribe a specific approach for generating the key. At the same time, the user wants to protect the key, by ensuring that it cannot be generated without the use of his/her choice of biometric characteristics. In this case, the biometric-key enrolment can be executed online. In both types of enrolment, the user uses his/her device to generate the biometric-key. As such, the user would record his/her biometric characteristic which is used by our algorithms to generate a unique biometric characteristic vector. The characteristic vector is then used by our cryptographic key generation algorithms which takes a unique vector as input and returns a cryptographic biometric-key satisfying the security parameters of the authentication system, for example the length of the key. After the enrolment, the characteristic vectors and the cryptographic biometric-key are permanently deleted from the user's machine. At the time of usage, the stored meta-data and the biometric data from the individual are used to re-generate the feature vector and hence the biometric-key. Such biometric-key, together with a random secret r, is then used for generating the commitments. After the enrolment, the random secret r is saved by the user along with the meta-data required for the re-generation. At the time of usage, the user needs to provide a proof of knowledge of the value committed at enrolment. Constructing such a proof requires that the user possesses both r and the cryptographic biometric-key. r could be stored in a secure location in the user machine. Alternatively r could also be stored in a portable user device, like a smartcard which and can be retrieved as needed. Moreover, it is possible that this random secret be (re) gener-

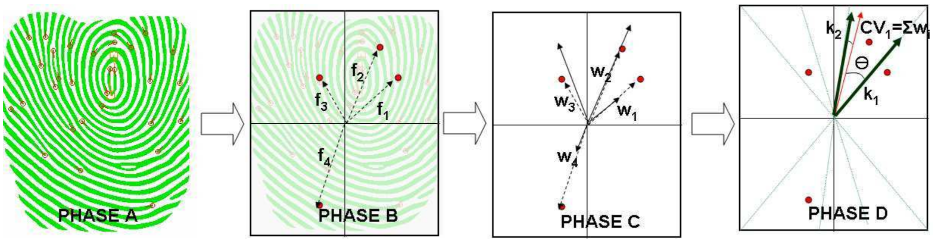


Figure 4. Phases of defining the key vector corresponding to a template

ated with a password or pass phrase that a user remembers and provides to his machine with a pseudo random generating algorithm, each time that r is needed. As the password or pass phrase provided to the random number generating algorithm is the same, the same random r is generated. In the case of the biometric-key, it is always regenerated, by using our algorithm, from a fresh reading of the biometrics through the biometric sensor. Once the re-generation is executed, the user would create a valid proof of knowledge to authenticate successfully.

Finally, to revoke the biometric-key, the commitment corresponding to the enrolled biometric data can simply be added to a revocation list, similar to the certificate revocation list (CRL) [25] in a public key infrastructure. CRL is typically a list of certificate serial numbers which have been revoked, hence are no longer valid, and should not be relied upon by any system user. In our approach, the revocation list just needs to record the commitments associated with the biometric-keys which have been revoked. After the publication of the list, even if the same biometric-key is generated, the user will not be able to do a proof of knowledge because the proof corresponds to a revoked commitment.

4. Biometric Authentication

There are two main parts in biometric authentication. The first one is the biometric-key generation and second is the generation of the proof based on recreation of the ZKP commitment. Biometric-key generation is difficult because every time particular biometric is read, the resulting template may be substantially different. Therefore reducing a biometric reading to a unique secret key poses several challenges [48,35]. The use of well known error correcting codes [20] on the fingerprint templates itself is not straightforward because the digital encoding of the two templates vary substantially if only the bit pattern is considered. Moreover, key generation differs from template matching because when re-generating the key the original fingerprint template is not available. We therefore investigate a technique based on fingerprint classification to define a characteristic vector for a fingerprint which is subsequently used by vector space modeling (VSM) [46] methodologies to retrieve a unique key.

4.1. Biometric Key Generation

Our goal is to define a key vector corresponding to the users fingerprint which can be repeatedly sampled and is reasonably unique with respect to other users. To achieve such goal we define a four-phased protocol as illustrated in Figure 4. Phase 'A' corresponds to the identification of the useful characteristics of the fingerprint template based on classification of the fingerprint itself. Fingerprint classification can be taken as a special case of pattern classification techniques which aim at reducing the computational overhead of pattern matching. If biometric patterns can be categorized, then given a pattern it may be possible to match the input pattern only against the stored templates in the same category of the input pattern. This is sometimes referred to as "binning". It has been shown [32] that fingerprints can be classified according to three different dimensions: by the shapes and contours of individual patterns, by the finger positions of the pattern types, and by relative size, determined by counting the ridges in loops and by tracing the ridges in whorls called minutiae points. The red circles corresponds to these minutiae points. At the end of Phase A, n such characteristics are identified.

In Phase B, the identified biometric characteristics are encoded according to a concise representation, referred to as *raw characteristic vector* represented as $\langle f_1, \ldots, f_n \rangle$ where n is the number of identified characteristics. Each characteristic itself is a vector f_i, $1 \leq i \leq n$, and has a distance from the core and an angle from the main axis. Note that such axis has to be aligned based on known heuristics [14].

In Phase C, each characteristic of raw characteristic vector $\langle f_1, \ldots, f_n \rangle$ is weighted with respect to the importance of each characteristic. The weighting is done based on Salton's Vector Space Model (VSM) [41]. In particular to weigh the unique characteristics, the term weighting used in [47], called inverse document frequency (idf for brevity), can be used. Here the term rarity is a measure of its importance. It is calculated as $idf_i = log(B/bf_i)$ where $1 \leq i \leq n$. Here B is the number of biometric samples in a collection and bf_i is the frequency of the biometric characteristic f_i. Thus idf_i measures the importance of a characteristic f_i based on that feature rarity. Often this weighting is normalized to force values to fall in a particular range. Note that the higher the weight the greater is the impact on the cosine of any two vectors. The main idea is to give more weight to those characteristics of the biometric which are more constant and unique for an individual. Therefore this step requires a comparison of several samples of the several individual and can be pre-determined at the learning phase. Let $\langle w_1, \ldots, w_n \rangle$ the resultant weighted characteristic vector, where w_i, $1 \leq i \leq n$, is the weighted scalar multiple of the characteristic f_i.

In Phase D a vector sum of all the weighted characteristics is evaluated. Let $CV_j = \sum w_i$ represent the vector sum of the weighted characteristics recorded for the j^{th} sample, where $1 \leq j \leq q$ and q is the number of biometric samples provided by the user. Note that phases A to D are executed for each of the q samples. If q different samples of the same user are taken, then each calculated sum CV_j will differ slightly. If the variability of any two CV_j is less than some angle Δ, then the entire key space is divided such that any two vectors are separated by Δ angle from each other. For example, the bold arrows k_1 and k_2 of Figure 4 are two such key vectors. The number of valid key vectors would thus be $\frac{360}{\Delta}$. Finally the similarity of the summation vector CV_j is taken to any of the key space vectors. The similarity is essentially the *cosine measure* of the angle between two such vectors. For two vectors $\vec{k}$ and $\vec{b}$ the cosine similarity is given by:

$$\cos\theta = \frac{\vec{k} \times \vec{b}}{|\vec{k}||\vec{b}|} = \frac{\sum_j w_{i,j} \times w_{b,j}}{\sqrt{\sum_j w_{i,j}^2}\sqrt{\sum_j w_{b,j}^2}}$$

Here $\vec{k} \times \vec{b}$ is the vector product of $\vec{k}$ and $\vec{b}$, calculated by multiplying corresponding weights together. The cosine measure also calculates the angle between the vectors in a high-dimensional virtual space. The closest key vector to the summation characteristic vector corresponds to the resultant biometric-key (See Figure 4 Phase D).

Example 2 *Consider the fingerprint template shown in Figure 4. In Phase A the minutiae points are identified. In Phase B based on those minutiae points the raw characteristic vector is formed. For simplicity, only 4 such points are considered forming the characteristic vector $\langle f_1, f_2, f_3, f_4 \rangle$. In Phase C, for each of the characteristic f_i where $1 \leq i \leq 4$, the $idf_i = \log(B/bf_i)$ is calculated. This is used for weighting each of the f_i to result in the weighted vector $\langle w_1, w_2, w_3, w_4 \rangle$. In Phase D, a vector sum $CV_1 = w_1 + w_2 + w_3 + w_4$ is calculated. A cosine comparison of CV_1 is then executed with the pre-defined key space$\langle k_1, \ldots k_m \rangle$ with Δ angle apart, where m is the number of valid key vectors. In this case CV_1 is closest to k_2.*

If the user executes r rounds of all the phases then as a result $CV_1, \ldots, CV_r$ are computed. If the Δ angle used to generate the key space is less than the difference between any two $CV_1, \ldots, CV_r$, then each of them will be closest to k_2 and thus will generate the same biometric-key.

4.2. Using Biometric Keys for Zero Knowledge Proof

Once the biometric-key is generated it can then be used to perform a ZKPK to authenticate the individual to the authenticating server.

Preliminary ZKP Concepts. ZKP systems are interactive systems in which two parties, the prover and verifier, interact. The prover claims that a statement is true and the verifier wants to be convinced that this is true. At the end of the interaction, the verifier is either convinced that the statement is true, or alternatively, discovers that the statement is not true. ZKP's have extensively been used for identification purposes [8,13,16]. Using the ZKP for biometric data is not straightforward because the biometric template cannot be replicated exactly like cryptographic keys. We therefore present a semantically secure ZKP based on the final key generated according to the approach described in Section 4.1. ZKP's are based on secrets that are hidden in tokens provided to the authenticating party at the time of enrolment. Such tokens are called *commitments*. Commitments are cryptographic tokens that enables the user being bound to a secret the possession of which can be verified at a later stage without revealing the secret itself.

We now present the protocol for using the biometric-key at authentication. To incorporate the key generation phases described in the previous subsection, we provide in Table 1 the definitions of the main functions used in the protocol. We assume that there exists a valid key space definition$\vec{K}$ in a vector space model and there is a user U who provides his/her biometric x to biometric sensor S.

Protocol Description. The main steps of the key generation protocol can be summarized as follows. Note that the same key generation protocol is used both at the time of enrolment and authentication. Let U be the individual to be authenticated. First the biometric template x of U is read by sensor S. Then the client invokes function init_char_vector which extracts the desired features from the biometric template read. This is the initial characteristic vector which is weighed according to system defined heuristics in the function weigh_char_vector. At this point the input vector $\vec{b}$ is ready to be matched with the key space $\vec{K}$ to find the closest matching key vector$\vec{k_i}$. This key space is stored at

Phase	Function	Description
A	read_biometric $(x) \leftarrow t$	U sends its biometric data x to S and t is the result of the feature extraction.
B	init_char_vector $(t) \leftarrow \vec{F}$	This function outputs $\vec{F} = \langle f_1, \ldots, f_n \rangle$ where n is the number of characteristics.
C	weigh_char_vector $(\vec{F}) \leftarrow \vec{W}$	This function outputs $\vec{W} = \langle w_1..w_n \rangle$ with appropriate weights for each characteristic.
D	get_closest_key $(\vec{W}, \vec{K}) \leftarrow \vec{k_i}$	This function outputs $\vec{k_i}$ which is most similar to $\sum \vec{W}$.
	generate-crypto-key $(\vec{k_i}) \leftarrow m$	The key vector $\vec{k_i}$ is transformed to an integer m used as the cryptographic secret key.

Table 1. Ordered functions for biometric-key generation protocol: generate-biometric-key

the client device. This is executed by function get_closest_key which depends on the VSM technique. Once the key vector is identified, functiongenerate-crypto-key reads this key vector to obtain the final secret used as a cryptographic key in the ZKP. This function may use an expansion function to sample the biometric-key from a well spread or uniform distribution. The generated keys size is dictated by the group $Z_{2^{B+k}}$ where k and B are security parameters of the federation system. Additional secrets can also be incorporated in the last function if desired.

Protocol 1 ZKPK with Biometric Commitments

```
Require: $U$, Registrar $Reg$ and Federations Verifiers $V$ agree on a group $\mathcal{G}$, a large
    integer $Func(k)$ and $2^B \rangle ord(\mathcal{G})$, $T$ is a public constant chosen arbitrarily large, $k$
    and $B$ are security parameters.
Ensure: Private knowledge of $U$ is $m, r$.
    {Enrolment or Commitment Phase:}
 1: $Reg$ chooses $h \in \mathcal{G}$ which has a $Func(k)$ rough order, a random secret $s \in Z_{2^{B+k}}$.
    It sets $g \leftarrow h^s$ and sends the public key $K = (g; h)$ and proves that $g \in \langle h \rangle$.
 2: $U$ chooses random $r \leftarrow Z_{2^{B+k}}$.
 3: $U$ assigns $m :=$ generate-biometric-key .
 4: $U$ sends its public commitment $C_K(m, r) := g^m * h^r$ to $Reg$.
    {Authentication or Proving Phase:}
 5: $U$ picks random $y \in [0..T * 2^k], s \in [0..T * 2^{B+2k}]$ at random and sends $d = g^y * h^s$
    to $V$.
 6: $V$ sends random challenge $e \in [0..Func(k)]$ to $U$.
 7: $U$ assigns $m' :=$ generate-biometric-key $(x')$.
    {$m'$ should be equal to $m$}
 8: $U$ sends $u = y + em', v = s + er$ to $V$.
 9: $V$: accepts if $g^u * h^v = d * c^e$
10: return
```

Functions in Table 1, collectively referred to as generate-crypto-key are used by the Damgard Fujisaki Integer Commitment Scheme [19] as shown in Protocol 1. In the following we highlight the main steps which enable the use of biometric commitments. At the time of enrolment instead of sending the actual biometric x, to the authentication entity (Reg in this case), U at step 3 calculates the biometric commitment and generates the secret being committed as m :=generate-crypto-key(x). In a typical integer commitment scheme m would be the value of the sensitive data which is being committed.

Consequently, at the time of verification U will have to prove the knowledge of this m and the random number r which it generated at step 2 of Protocol 1. U has to store r as specified in a typical ZKP system, but m is generated at the time of authentication. More specifically, at step 7, U can generate m' using the same key generation function as enrolment. If the difference between x and x' is tolerable then the committed secret m will be equal to the retrieved secret m'. If U has m and r it follows that the proof at step 9 will succeed. The ZKP can be efficiently computed as shown in [13]. The challenge response can be made non-interactive using fiat shamir heuristics to enhance the efficiency. Other models in which the biometric-key is based on symmetric encryption would require the key to be known to the verifier, which is not necessary in our model.

4.3. Biometric Identifier Revocation

According to our schema, every biometric-key is uniquely generated and is associated with a commitment which is tightly coupled with the biometric feature. In case revocation becomes necessary, because of the individual leaving the federation or losing his/her privileges within the federation, the commitment needs to be publicly added to revocation lists.

The authentication system presented in this paper supports revocation of the enrolled biometric. Additionally, it also allows re-enrolment of the same biometric as a different biometric commitment. This implies that a user can enroll multiple times the same biometric identifier and the enrolments cannot be linked based on the registered values. This feature is desirable to guarantee freshness of the committed values, and ensure that stale or -possibly- forged commitments do not prevent a user to re enroll.

Revocation of user U's biometric commitment can be simply executed by including such commitment in a revocation list which is checked before authentication. Moreover, the IdR[4] associated with U can be updated to ensure that the status of this biometric commitment is stated as invalid. For further details corresponding to revocation of identifiers in IdR we refer the reader to [8], where a comprehensive description of the revocation mechanisms for the various types of attributes is provided.

Since in specific contexts many federations may require individuals to be permanently disallowed from enrolling or participating in activities within the federation, an alternative type of revocation can be supported.
Precisely, in our framework, we distinguish between two types of revocations namely *weak biometric revocation* and *strong biometric revocation*. In the former type, the revocation is of a particular fingerprint commitment. On the contrary strong biometric revocation revokes the actual user by revoking all the possible biometric commitments enrolled by the same user.

4.3.1. Weak Biometric Revocation

Once a commitment is revoked, the biometric-key m and the random secret r corresponding to this commitment cannot be used together. Therefore for a legitimate re-enrolment of a biometric B, a new pair m' and r' has to be used such that either $m' \neq m$ or $r' \neq r$, or both. Since the parameter r' will be chosen randomly, the re-enrolment of B with the same biometric-key is possible. However, if required a new m' can be generated

[4] Identity Record. For more details please see Section 2.

by executing the generate-crypto-key function in Protocol 1, if such a procedure uses different weighting at the time of generation of the biometric-key.

4.3.2. Strong Biometric Revocation

In case a specific identity management environment requires revocation to be tightly coupled with the actual individual, so that he/she cannot re-enroll a second time, a slight modification of the current schema should be done. The biometric-keys proposed in this work are used to create the Pedersen commitments, which by definition are unlinkable. To overcome unlinkability, every biometric should be tightly bound to a second form of unique identifier, which should be enrolled and stored at the time of user's registration. A user being revoked wishing to re-enroll could not succeed since the unique identifiers required would be associated to a revoked biometric. An example of such identifiers having the desired uniqueness properties are SSN or Passport Number. This information should be stored in the IdR illustrated in Figure 2.

An alternative solution would be adopting a deterministic commitment [6], which would be uniquely linked to the biometric of the individual. To ensure uniqueness of the biometric, enrolment and verification should be conducted in a controlled environment or using a trusted computing framework. The weighting should be nullified to ensure the same biometric-key to be generated for a given biometric. The deterministic commitment would correspond to a unique signature which can be revoked throughout the system.

5. Entropy based Analysis of Biometrics

In this section we provide an entropy based analysis of biometrics. The aim is to provide a comparison of the strength of biometric tokens with respect to traditional secrets like passwords. Preliminary concepts to understand the methodology are introduced in Section 5.1. Section 5.2 then provides the related theoretical work on the comparison of biometrics and passwords and extends it to show some experimental results. We elaborate on the bit strength of the biometric versus different types of passwords. Finally Section 5.3 presents a discussion of the use of biometrics in combination with other alphanumeric secrets.

5.1. Preliminary Concepts

To calculate entropy H of message X Shannon's [43] equation $H(X) = \sum_x^n p(X) \log_2(\frac{1}{p(X)})$ is used. In the case of the fingerprint the*keyspace* (corresponding to the summation limits $x \rightarrow n$) is the number of sites in the fingerprint image where minutiae points could appear. This is derived by dividing the surface area of the image by the space consumed by a single minutiae point. As such, the keyspace is dependent on the dimensions of sample acquired by the system. Typically the 2-dimensional space created by the length and width of a fingerprint image are the bounds of the keyspace. While the dimensions of fingerprint images vary based on sensor technology, the dimensions of 300×300 pixels are assumed to be reasonably average image size for fingerprints [38]. In the case of typical alphanumeric passwords the keyspace is limited to the 94 characters available to the user on the keyboard as the input device.

The *probability of occurrence*, $p(X)$, of the minutiae points in the keyspace is not equally likely for all the minutiae points. While high-level analysis has shown that minutiae points are more likely to occur in the center of the fingerprint image as opposed to the outer regions [24], detailed investigations into exact probabilities of occurrence based on certain minutiae sites have not been conducted. Typical alphanumeric passwords also have non-uniform probabilities for the characters as they are heavily dependent on words or phrases which can be recalled over time by the users. Estimating these probabilities has led to dictionary attacks on password systems where the perpetrator uses a large number of known words in attempt to discover a password. Note that in the cases of several other strong identifiers like CCN and SSN, the string of numbers often is not totally random and may follow a particular pattern for several of the numbers.

Finally, entropy also varies based on the *thresholds* of the environment. These thresholds are policies and requirements on the level of assurance the authenticator enforces. These can either be imposed by the human or machine element. For instance, a password that is 5 characters long will have lower entropy than that of an password that is required to be 8 characters long. Combining these three factors (keyspace, probability of occurrence, and threshold), NIST created a table [11] comparing the entropy with the password length of various types of passwords. The table was generated using Shannon's entropy equation for the purposes of evaluating entropy in bits based on the length of both passwords (94 character alphabet) and PINs (10 character alphabet). Estimations of message entropy for both passwords and PINs are analyzed as either "user chosen" or "randomly chosen". Furthermore, user chosen passwords are broken into columns indicating the degree of policy requirements for password composition. The lowest level of policy being no checks at all, the medium level uses a dictionary rule to prevent users from including common words; high level uses a dictionary and also requires the password to include at least one upper case letter, lower case letter, number, and special character. Each one of the cells indicates the estimated entropy value in bits for a designated character length of a password.

5.2. Biometric/Password Entropy based Comparison

Theoretical work [38] has been carried out to determine the keyspace and bit strength of fingerprint biometrics. This approach focuses on a hypothetical brute force attack against minutiae based fingerprints. It takes into consideration the parameters provided in Table 2 and uses the following equation to derive the probability of verification for theoretical query sample matching the reference minutiae template.

$$\rho_{verification} = \frac{e^{-Np}}{\sqrt{(2\pi m)}}(\frac{(eNp)}{m})^m$$

The probability of verification is then converted to log base 2 which is the number of digital bits (either 0 or 1) required to represent the amount of informational content of the matching scenario. Note that while this example follows a similar methodology as the NIST analysis of passwords [11] in taking to account keyspace and threshold, probabilities of occurrence for all values in the keyspace are considered equal. In other words, all possible minutiae sites are considered to be equally probable to occur.

Using the above methodology a fingerprint system requiring 25 minutiae points to be matched would have 82 bits of information and equates to a 16-character nonsense pass-

Description	Symbol	Value
Image size (pixels)	S	$300 \times 300 = 90,000$
Ridge plus valley spread (pixels)	T	15
Total number of possible minutiae sites	$K = \frac{S}{T}^2$	$20 \times 20 = 400$
Total number of orientations allowed for the ridge angle at a minutiae point	d	4
Total number of minutiae present in the reference template	N	40
Minimum number of corresponding minutiae in a query and reference template	m	[10..35]
Probability of matching an individual minutiae in the reference template	$p = \frac{N}{(K-N+1)d}$	

Table 2. Theoretical parameters for the hypothetical brute force attacks.

word (such as "m4yus78xpmks3bc9"). Analogous to the NIST method for passwords [11], 82 bits of entropy for a randomly chosen password would be 12.5 characters long. The difference in character length of the hypothetical password is due to the fact that 16 character example only takes into account lowercase letters and numbers whereas the NIST method also uses uppercase letters and special characters.

In order to transform the theoretical approach into experimental data, we have collected fingerprint images from human subjects. Fifty study participants each submitted 3 consecutive samples of each finger (index, middle, ring, and little) on both hands. There was no user feedback provided on the quality of the fingerprint image; the only interaction users had with the sensor was a prompt to place their finger on the sensor. This intentional limited feedback eliminated possible extraneous variables caused by changing user behavior from entering into the later analysis. Providing feedback such as poor placement or quality would have been counter productive to the intent of the study. This process yielded a total of $1,200$ images (50 users $\times 8$ fingers$\times 3$ images per finger). The sensor used for image capture was the Identix, Inc. $DFR\circledR - 2080U2$ Single Finger Reader operating at 500 dpi. The 50-users test population included 33 (66%) males and 17 (34%) females. The age of the users ranged from 19 to 86 years old, with the average age being 32.87.

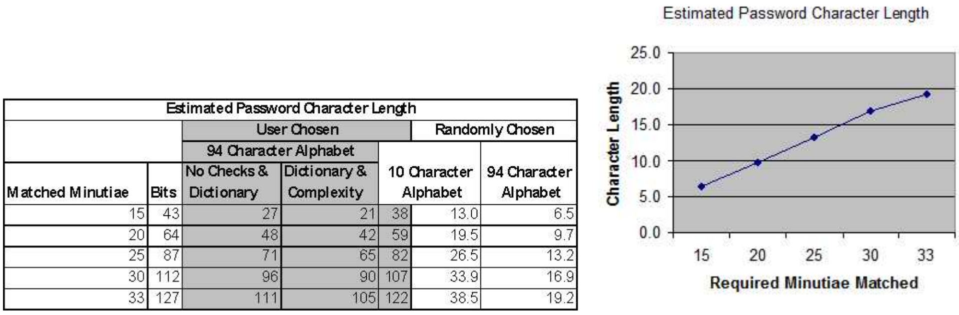
75 images were manually removed from the data set due to problems in image acquisition; these problems were not be detected when the data collection occurred because the user did not receive any feedback while placing their fingers. These images were either completely black or completely white, signifying a failure to acquire (FTA) and resulting in an FTA rate of 6.25%. After manual removal of the FTA images, the remaining images were processed using the WSQ by Aware, Inc. software package to analyze minutiae present in each image. Each of the images had the dimensions of 248×292 and the average number of minutiae for all images was 33. These values produced modifications to the evaluation parameters introduced in Table 2 to the values given in Table 3.

Based on the experimental values, for each calculated entropy per given number of matched minutiae points, the corresponding length of the alphanumeric password was calculated. This provided a comparison of the required length of a password versus the number of matched minutiae points.The results are provided in Figure 5. For further analysis, randomly chosen passwords were used. This is because, while users can chose the nature and order of the characters in their password, users have no discretionary ability to chose the nature and order of the minutiae on their fingerprint. Based on the

Description	Symbol	Value
Image size (pixels)	S	$248 \times 292 = 72,416$
Ridge plus valley spread (pixels)	T	15
Total number of possible minutiae sites	$K = \frac{S}{T}^2$	321
Total number of orientations allowed for the ridge angle at a minutiae point	d	4
Total number of minutiae present in the reference template	N	33
Minimum number of corresponding minutiae in a query and reference template	m	$[15..33]$

Table 3. Experiment based parameters for comparison of biometrics and passwords.

experiments, we got a linear relationship of the character length to the required number of matched minutiae. In essence Figure 5 shows the conversion of the bit strength to an estimated password character length using the theoretical NIST methodology and our experimental data. In particular, 1 character was shown equivalent to 6.6 bits of entropy (for 94 character alphabet randomly chosen password).



Estimated Password Character Length						
		User Chosen			Randomly Chosen	
		94 Character Alphabet				
Matched Minutiae	Bits	No Checks & Dictionary	Dictionary & Complexity		10 Character Alphabet	94 Character Alphabet
15	43	27	21	38	13.0	6.5
20	64	48	42	59	19.5	9.7
25	87	71	65	82	26.5	13.2
30	112	96	90	107	33.9	16.9
33	127	111	105	122	38.5	19.2

Figure 5. Entropy Based Comparison of Matched Minutiae Points and Password Lengths

5.3. *Biometric/Password Combined Entropy*

Biometrics for the purpose of authentication is desired because it cannot be easily forged, shared or stolen providing repudiation and reliable linkability with the users unique identity. However, it is preferred that they are used in combination with a traditional password for several reasons. First, stable biometric signals can be forged if the system is not implemented correctly. Second, attacks on authentication systems are growing, hence requiring increasing entropy of the challenged secret. Passwords, if kept secret, provide an additive entropy on top of the biometrics. Increasing the keyspace and hashing protect these passwords against host attacks; the use of challenge-response protocols protects against replay, eavesdropping, and other attacks in transmission of the resultant secrets. Third, passwords provide a convenient and inexpensive methodology. However, remembering multiple passwords and changing them frequently lead to the usage of lower-entropy passwords which are susceptible to search attacks. Moreover, passwords can be compromised without this being detected and therefore do not defend well against repudiation. As illustrated in Figure 5, a typical biometric sample adds significant entropy as compared to the passwords. For example, most sensors match approximately 25 minutiae points, which corresponds to the resultant entropy of 87 bits. To achieve 87 bits of entropy, a user chosen password has to be 65 characters long, and a randomly generated password has to be 13.2 characters long. The increased length requirements for the passwords make the use of biometric more desirable.

In a two-factor authentication scenario where secrets and biometrics are used, it follows that the entropy of the resultant authentication transaction is greater than if only one authentication factor was used. In the case of biometric and passwords, the combined entropy of the two-factor authentication is simply the sum of the individual entropies (i.e. $H(Biometric)$ + $H(Secret)$) for each of the factors. This is because there is no inter-dependence between the biometric and secret presented by the user. For instance, if a user chooses a password that contains below average entropy, this does not mean the fingerprint of the same user will also have below average entropy. This rationale is focused on the example of fingerprints and doesn't take into accounts the fact that some biometrics such as signature/sign, voice and keystroke dynamics include a textual component in the biometric sample.

6. Multi-factor Authentication

Several biometric mechanisms have been recently considered and proposed, such as facial recognition, iris recognition and fingerprints. However, most commercially viable physical biometric set in the foreseeable future are fingerprints [44]. A multi-factor authentication may become essential with the growing security needs. In Section 4 we showed how we can preserve the privacy of a biometric used for authentication. In this section we show how we can achieve privacy preserving multi-factor authentication providing relatively stronger authentication as compared to systems that are solely dependent on either *what you know* identifiers or*who you are* identifiers. This is because all identifiers have the threat of being stolen or shared. However, if a collection of these different types of identifiers are challenged in an ad-hoc, unpredictable manner, such threats are sufficiently mitigated.

6.1. Two-Factor Authentication using ZKP

In Protocol 1 there are two specific secrets which have to be known to the user, namely m and r^5. m corresponds to the number generated by generate-crypto-key function in Protocol 1, from the fingerprint of the prover. Thus,m covers the *who you are* criterion of the authentication. The second secret r is the random number generated by the prover during the initial enrolment. The random number generation is depicted in step 2 of Protocol 1. Without the knowledge of r the proofs u and v cannot be generated at step 8. r is chosen such that its bit length is at least twice that of the order of h and the final commitment is statistically close to the uniform distribution in $\langle h \rangle$, for any value of m. This is essential for the hiding property of the ZKP. Hence, we conclude that the random r serves as a second factor essential for a successful ZKPK used for authentication. r thus corresponds to the *what you have* aspect of authentication. Note that even if one

[5] All symbols and variables used in this section correspond to the ones presented in Protocol 1

of the secrets, either m or r but not both, is compromised, the adversary cannot generate the correct proofs. We therefore assert that Protocol 1 provides a secure two-factor authentication.

6.2. Multi-Factor Authentication using Federation Registrar

Additional privacy preserving multi-factor authentication is possible by leveraging the registrar of the IdM system mentioned in Section 2. Typical enrolment of attributes omits step 3 of Protocol 1. An *identity record* corresponding to the proofs of identity is thus created. An example of such identity record is shown in Figure 2. Here, additional information regarding each committed identifier is also recorded which provides additional security properties [8]. Depending on the policy of the authenticator, multiple commitments of the identity record can be used for authentication. We omit the details of such proofs as they have been provided in our previous work [8].

Referring to Example 1, Alice can enroll with her identity provider *CityBank*. Now when using non-financial services from$TaxAuthr$, with the help of Protocol 1, proof of the fingerprint itself provides the required two-factor authentication. To satisfy the authentication requirements at the time of financial transactions with$TaxAuthr$, Alice refers to the identity record stored at her local registrar (see Figure 2). She combines the proof of knowledge of the identifiers with tags CCN, SSN and Fingerprint and uses the same proof of knowledge as in Protocol 1 for each of the identifiers. Efficient combinations of such proofs have been explored in [4,13] where knowledge of multiple factors can be proven to the authenticating party in one ZKP round. Note that our way of providing multi-factor authentication differs from the case where the biometric data is used to unlock another security token, for example a smart card containing a private key. In this case, the biometric data is verified by the smart card itself. The actual authentication to an authenticating party is based on one factor only (the private key). In this case the authentication may be considered consisting of two-factors. In our case however, the biometric-key is an integral part of the proof and only the authenticating party is responsible for the verification of the various factors.

7. Analysis

We now analyze the security, privacy and architectural aspects of the proposed authentication protocols. In particular we assess the level of protection provided by our mechanism against identity theft in the presence of malicious parties.

7.1. Security Analysis

Before presenting the security properties of our system, we illustrate the key assumptions which our solution builds on.

Assumption 1 *The raw characteristic vector $\langle f_1, \ldots, f_n \rangle$, where n is the number of characteristics, is correct and sufficient.*

The VSM technique adopted for the key generation in the 4 phases of key generation requires that correct and sufficient features are recorded by the function init_char_vector. The output vector is then weighed by the function weigh_char_vector resulting in the final characteristic vector $\vec{W}$.

Assumption 2 *The characteristic vector used in* init_char_vector *is sufficiently expressive to uniquely identify an individual with a high probability.*

This assumption relates to the expressiveness of the characteristic vector so that it can capture the uniqueness of a given biometric. We require that this is done with a sufficiently high probability, although we show in the security analysis that in the presence of collision generate-biometric-key is still resistant to collision.

Assumption 3 *The initial enrolment of the biometric is secure.*

This assumption is especially true considering the current day biometric enrolment where the individual is required to come in person and the enrolment is performed in a controlled environment by the designated authorities. If the enrolment is executed in an insecure fashion, it would lead to serious repercussions, especially in systems where the enrolled biometric is the only factor checked during authentication.

Based on the above assumption the following security properties hold.

Theorem 1 **Soundness***: Let U be an individual, and B be the biometric associated with it. If U has enrolled using Protocol 1, then it can execute the authentication phase successfully.*

PROOF. At the time of enrolment U generates a biometric-key m and also chooses a random r. r is the only value stored with U. Then at authentication time, because of Assumption 1, the biometric-key $m' = m$ can be regenerated. As evident from step 8 of Protocol 1 the final proof can be constructed correctly based on the retrieved values m and r. Therefore Protocol 1 is sound. □

We now show an interesting result on the correctness of our protocols. Correctness refers to the incapacity of any attacker to execute the protocol successfully, to be authenticated as another user, even with that users biometric sample.

Theorem 2 **Correctness** *(Two-factor): Let U be an individual with biometric B and associated random secret r, and A be an adversary with biometric B'. If r is not known to A, then Protocol 1 is correct.*

PROOF. Let the keys generated at step 3 of Protocol 1 be m_U for U and m_A for A. Under Assumption 2, with a high probability m_U is not equal to m_A. If the enrolment has been executed correctly, then m_U is the biometric enrolled. Therefore, the proof generated would be incorrect and step 9 of Protocol 1 will return false.

If however, m_A happens to be equal to m_U to complete the proof in step 8 A would have to guess the random value r. Since this random value is chosen from $Z_{2^{B+k}}$ it is infeasible for an adversary to guess it. This condition holds true provided the secrecy condition on r (as given in the theorem). Thus the two-factor authentication is correct with respect to the biometric commitment and the thesis holds. □

Theorem 3 Correctness *(Multi-factor): Let U be an individual with biometric B included with other identifiers in an Identity Record IdR, and A be an adversary. Let I be a subset of identifier commitments in IdR including the commitment of B be used for authentication. If at least one of the secrets associated with I is not known by A, then A cannot execute the authentication phase of Protocol 1 successfully.*

PROOF. Let I be a set defined as $\{i_1, .., i_n\}$ where $i_k, 1 \leq k \leq n$, is a biometric commitment. Note each identifier is also associated with random secrets $\{r_1, .., r_n\}$ generated at the time of the enrolment. Multi-factor authentication in essence executes Protocol 1 separately. If anyone of the $2n$ identifiers in I and random secrets together is not known to the adversary, then the proof of at least one of the ZKPK will fail. This would result in failing the authentication process. Note that the length and the content of I is chosen in an ad-hoc fashion at the time of authentication, thus making the challenge fresh and not pre-determined. This further makes the probability of forging minimal. □

Identity Theft Protection. Biometric identifiers correspond to the physical characteristics of a person and as such are harder to steal as compared to other identifiers which are normally stored in external devices. Furthermore, the enrolment procedure for biometrics is typically very strong thus leading to higher assurance on the enrolled biometric commitment. In our system, biometric keys are generated on the fly and are not stored either at the client or the server. The biometric templates in fact represent intrinsic information about the user, therefore theft of the template leads to identity theft.

It would be important that the freshness and liveness of a biometric be ensured by the biometric scanner (Figure 1). If, however, an adversary manages by using a duplicate latex fingerprint to generate the biometric-key, then the identity theft attempt would be prevented as the attacker would still need to have the random secret to generate the proofs as required by the authentication Protocol 1. Even if the client device and ZKPK modules are compromised so as to maliciously store previous legitimate proofs, still the adversary cannot execute Protocol 1 successfully. This is because of the nature of the ZKPK itself, that requires a fresh random challenge and to reconstruct the proof each time.[6] Furthermore, from Theorem 3 we get strong authentication, which is in fact the predominant solution to mitigate the threat of identity theft.

7.2. *Privacy*

Our approach of key generation based on the biometric maintains the advantages of the biometric authentication and at the same time prevents any leakage of additional personal information. Thus the data collected at the time of authentication cannot be used for any other purpose other than the authentication decision itself.

The biometric-key generated is also secured based on the ZKPK proofs. Thus the biometric-key cannot be reverse engineered to guess the characteristic vector used to generate that key. This further preserves privacy of the biometric. Moreover, as illustrated in the previous section, the correctness of the protocols also help in preserving the privacy and misuse.

[6] This is true under the assumption that the biometric-key is not stored in the device.

7.3. Architectural Issues

In our scheme, the actual biometrics template is never stored anywhere, thus providing storage efficiency and preventing the need of databases storing biometric templates. Accordingly, the database threats by external and internal attackers, and tampering stored templates are prevented.

A federated identity system has to handle heterogeneity of the various clients and the servers. Using the methodology presented, the clients can generate the keys using any proprietary biometric scanner or software, and the server can still authenticate based on the same ZKPK. In fact, the level of interoperability is even higher, in that adding biometric authentication does not require additional verifications at the server end. As such, the server can verify the proofs of a biometric just like other identifiers stored in the users IdR. This also helps addressing deployment and scalability concerns.

The computational overhead at the client is also minimal since the vector space methods are efficient. The ZKPK proofs can be efficiently implemented and aggregation techniques have been proposed to further compute the multi-factor proofs in a concise manner.

8. Related Work

Client side authentication have been extensively investigated in the past years. We provide an overview of the work closely related to our approach. We first present the cryptographic mechanisms proposed in the context of biometrics, followed by descriptions of specific fingerprint recognition methods and technologies.

Several previous crypto-based approaches like [9,27] build on Chaum and Pederson wallet-with-observer paradigm. Here a tamper-proof device available at the client's end is required where the comparison is made. In this model a smart card provided to the user acts as the wallet, and the wallet runs a local process which is called the observer. The wallet is assumed to be tamper-resistant and powerful enough to carry out relatively expensive cryptographic computations. Security of the wallet in possession of the user is the key. Indeed, if the client side device is compromised, replay attacks or fraudulent authentication results may be sent to the server. Furthermore this model is not scalable nor interoperable. It can potentially be used by a party other than the owner which is an undesired property for biometric authentication.

Other efforts in cryptography based on biometrics are fuzzy commitments and fuzzy vaults [34,33] and fuzzy extractors [22] have been proposed. However, several of these schemes may be vulnerable to replay attacks, non-repudiation and the vulnerability of the resultant keys generated to cryptanalysis. Many of the schemes mentioned depend on the fuzzy commitment scheme which was developed by Juels and Wattenberg [34] who have proposed an improvement and generalization of the approach by Davida et. al. [20] where a synthesis of error correcting codes with cryptographic techniques was proposed.

Unlike traditional commitment schemes, the fuzzy commitment scheme was designed mainly to be resilient to small corruptions of the committed value, which is also called the *witness*. This means that if prover P originally committed value x, then P can potentially open the original commitment or decommit successfully with a value x' very close to x. One of the distance functions to measure closeness of two values is the Hamming distance. Such cryptographic primitive of fuzzy commitments was proposed to be achieved with the help of error correcting codes (ECC). ECC enable transmission of a message m intact over a noisy communication channel. An approach based on ECC is not trivial to implement because of the complexity. Therefore, in our work we investigate an approach based on recording specific characteristics of a biometric, instead of dealing with the biometric template as a simple bit pattern. We believe such a mechanism is highly practical. We not only use the biometric-key as it is, but we also combine it with ZKPK to assure the privacy of the biometric-key itself.

Key Generation based on biometric aggregation has been investigated in [18]. Several invariant features of different types of biometric are used to derive a biometric-key that is used to encrypt a plain text message with a header information. The decryption is based on a new generated biometric-key which may not be exactly the same as the initial key. Different permutations of the newly computed biometric-key are used to decrypt the header of the encrypted message after which the rest of the message is inferred. This approach addressed the non-repudiation problems and was proven to be efficient. However, to be robust this scheme needs several biometrics per user. We provide a more fine granular approach which can generate different keys for closely looking biometrics. Our approach greatly depends on the characteristic vector as highlighted in Section 4.1. We further use information theoretic ZKPK that provide additional privacy and security properties as shown in the analysis of Section 7.

Finally, as compared to traditional cryptographic secret key based systems like public key system RSA [5] and private key system [40], the biometric-key based authentication proposed in this paper was used to construct commitments to perform zero-knowledge proofs [13,23]. The use of ZKP provides additional security and privacy properties as highlighted in [7]. Some example properties are unlinkability, anonymity, data minimization and efficient revocation of biometric tokens without the loss of privacy. Due to the use of ZKPs with the biometric-key, these properties are achieved in our system in an elegant manner. Traditional signature schemes requires only one signature to be generated and validated thus proving to be very efficient. However, current anonymous credential system based on ZKPs [10,13] have been developed to be highly efficient. For example in the idemix anonymous credential system [12], only three multi-base exponentiations for the user and one for the authenticating party are needed [7]. Moreover, using aggregate ZKPs multi-factor authentication [4], it has shown that efficient ZKPs can be performed with multiple commitments. Here if a proof for t commitments is needed to be provided then not more than $5t$ exponentiations are needed. The available arithmetic techniques for optimization, and several possible pre-computations in such schemes make this approach practical.

In traditional fingerprint based biometric authentication systems [31,39], authentication based on matching of fingerprints. One way to do such matching is to extract the minutiae points of the fingerprint and compare it against the second fingerprint template minutiae's. The effectiveness of such systems are based on evaluating error rates such as False Accept Rate (FAR), False Reject Rate (FRR), and Equal Error Rate (EER). The processing time for such matching has shown to be efficient for practical purposes. For example for Neurotechnologija Verifinger system [36] the fingerprint enrolment time is 0.2-0.4 seconds, and VeriFinger can match 40,000 fingerprints per second in one to many identification mode. In our proposed system, there is no matching of the actual fingerprint template, therefore the efficiency of the biometric-key system are reliant primarily on the time needed to generate the biometric-key. This has been shown to be efficient using the protocols described in Section 4. Note that we use existing equipments and software like the Neurotechnologija Ltd.'s VeriFinger 4.2 SDK to evaluate the characteristic vector to calculate the distances of minutiae points from the fingerprint core. This SDK allows for the necessary information regarding the core and closest minutiae distances to be extracted and calculated. Most computation is performed in the learning phase (i.e. before enrolment), and the weights particular to the user are evaluated at the time of enrolment. Henceforth, at verification only the weighting has to be applied and the cosine measure be calculated and compared against the stored key space.

In the past years alternative pattern based fingerprint matching systems [28] have been proposed. These systems do not function on minutiae points, but rather segments of the entire image which are examined individually. More specifically, this global approach to matching fingerprints uses characteristics such as space between ridges and power spectrum analysis. This approach can use a smaller area of the fingerprint to perform the matching, but does not provide the type of informational content that our approach requires. Unlike pattern based fingerprint matching systems, our system only utilizes minutiae based information from fingerprints.

Concerning standards related to biometric technologies, the Biometric Application Programming Interface (BioAPI) is both an American National Standard as ANSI-INCITS 358-2002 [2] and International Standard as ISO/IEC 19784-1 [30] which defines an API framework through common function calls. The BioAPI Consortium is a non for profit organization consisting of over 100 members that promotes the adoption of the BioAPI framework and brings platform and device independence to application programmers and biometric service providers [17]. While BioAPI is an application layer standard there also exists minutiae data interchange format standards. In 2004 ANSI-INCITS 378-2004 was approved as an American National Standard [3]. The ISO International version was approved the following year in 2005 as ISO/IEC 19794-2 [1]. The purpose of these standards is to facilitate interoperability of minutiae data between different software applications utilizing the BioAPI framework. A method for documenting the information of x and y coordinates as well as direction angle and type are for each minutiae in a given sample are defined by these standards. Research has been conducted to test interoperability of various fingerprint template and matching algorithms according to ANSI-INCITS 378-2004 [24]. For the purposes of this research, only one template generation algorithm was used, and as mentioned before, no matching is performed. The only constraint on the system would be to use the same template generation algorithm for each instance of the bio-key generation. Although not proven in this research, using multiple template generation algorithms could result in variations in the minutiae details significant enough to effect performance of the system. This system complies with the minutiae data interchange format standards mentioned and future work includes integration into the entire BioAPI framework.

9. Conclusion and Future Work

In this paper we have developed a privacy preserving biometric authentication methodology. Our approach has several security and privacy advantages for the authentication mechanism and the biometric itself. We provide a new application of vector-space model to efficiently generate cryptographic biometric keys. We preserve privacy and unconditional security of the biometric key by employing information theoretically secure ZKPK. Our notion of privacy relates to the amount of data disclosed and the controlled usage of it, for the specific authentication purposes it was designed for. Moreover, we show how the biometric authentication can be combined with other identifiers used in a federated IdM system for authentication, thus resulting in a multi-factor authentication.

We plan to extend this work in several directions especially with respect to experimentation. First we plan to work on simulations and evaluations of fingerprints, and voice prints biometrics to develop examples of the characteristic vectors. We will investigate how this characteristic vector differs in the various vendor specific biometric scanners. The second direction is to use this characteristic vector to generate a key space in VSM which is sufficiently expressive to generate different keys from closely related biometrics. The third direction is to explore the feasibility of the given approach without any user-specific state stored at the client machine. In this way the client would not be able to determine if the correct biometric-key was generated, and would have to rely on a successful completion of the ZKPK to determine this. The resulting scheme would be able to support portable authentication by users at trusted client terminals which they have not previously interacted with. We need further investigation and experimental results to ensure the feasibility of such extension. The fourth direction is to extend the concept of strong revocation discussed in Section 4.3 using deterministic commitments of the biometric-keys. Finally, we would like to investigate combining various biometrics in the VSM model as investigated in [18].

10. Acknowledgement

The work reported in this paper has been sponsored by NSF under the ITR Project 0428554 'The Design and Use of Digital Identities". We thank Prof. Samuel Wagstaff for his useful suggestions.

References

[1] ISO/IEC JTC 1/SC 37 Working Group 3. Information technology - biometric data interchange formats - part 2: Finger minutiae data. `http://isotc.iso.org/livelink/livelink/3917020/JTC001-N-7225.pdf?func=doc.Fetch&nodeid=3917020`, 2005.

[2] ANSI-INCITS 358-2002. Biometric Application Programming Interface (BioAPI), 2002.

[3] ANSI-INCITS 378-2004. Finger Minutiae Format for Data Interchange, 2004.

[4] Abhilasha Bhargav-Spantzel, Anna C. Squicciarini, Rui Xue, Elisa Bertino. Practical identity theft prevention using aggregated proof of knowledge. Technical report, CS Department. CERIAS TR 2006-26.

[5] Accredited Standards Committee X9. *Working Draft: American National Standard X9.31-1992: Public Key Cryptography Using Reversible Algorithms for the Financial Services Industry: Part 1: The RSA Signature Algorithm*, March 7, 1993.

[6] D. Beaver, J. Feigenbaum, and V. Shoup. Hiding instances in zero-knowledge proof systems. In A. J. Menezes and S. A. Vanstone, editors, *Proc. CRYPTO 90*, pages 326–338. Springer-Verlag, 1991. Lecture Notes in Computer Science No. 537.

[7] Abhilasha Bhargav-Spantzel, Jan Camenisch, Thomas Gross, and Dieter Sommer. User centricity: a taxonomy and open issues. In *DIM '06: Proceedings of the second ACM workshop on Digital identity management*, pages 1–10, New York, NY, USA, 2006. ACM Press.

[8] Abhilasha Bhargav-Spantzel, Anna Squicciarini, and Elisa Bertino. Establishing and protecting digital identity in federation systems. *Journal of Computer Security*, 13(3):269–300, 2006.

[9] Gerrit Bleumer. Biometric yet privacy protecting person authentication. *Lecture Notes in Computer Science*, 1525:99–110, 1998.

[10] Stefan Brands. Electronic cash systems based on the representation problem in groups of prime order. In *Preproceedings of Advances in Cryptology — CRYPTO '93*, pages 26.1–26.15, 1993.

[11] William E. Burr, Donna F. Dodson, and W. Timothy Polk. Electronic authentication guideline: Recommendations of the national institute of standards and technology. In *NIST SP800-63*, page 64. NIST, 2006.

[12] Jan Camenisch and Els Van Herreweghen. Design and implementation of the *idemix* anonymous credential system. acm press, 2002.

[13] Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor,*Advances in Cryptology — EUROCRYPT 2001*, volume 2045, pages 93–118. Springer Verlag, 2001.

[14] Raffaele Cappelli, Alessandra Lumini, Dario Maio, and Davide Maltoni. Fingerprint classification by directional image partitioning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):402–421, 1999.

[15] David Chaum and Torben P. Pedersen. Wallet databases with observers. In *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, pages 89–105, London, UK, 1993. Springer-Verlag.

[16] David Chaum and Hans van Antwerpen. Undeniable signatures. In *Advances in Cryptology — CRYPTO '89*, volume 435, pages 212–216. Springer-Verlag, 1990.

[17] BioAPI Consortium. http://www.bioapi.org/.

[18] Christopher Ralph Costanzo. Biometric cryptography: Key generation using feature and parametric aggregation. Online Technical Report, 2004.

[19] Ivan Damgård and Eiichiro Fujisaki. An integer commitment scheme based on groups with hidden order. In *Advances in Cryptology — ASIACRYPT 2002*, volume 2501. Springer, 2002.

[20] G. Davida, Y. Frankel, and B. Matt. The relation of error correction and cryptography to an offline biometric based identication scheme, 1999.

[21] Rachna Dhamija and J. D. Tygar. The battle against phishing: Dynamic security skins. In *SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security*, pages 77–88, New York, NY, USA, 2005. ACM Press.

[22] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Eurocrypt 2004*, 2006.

[23] Uriel Feige, Amos Fiat, and Adi Shamir. Zero knowledge proofs of identity. In *Proc. 19th ACM Symp. on Theory of Computing*, pages 210–217, May 1987.

[24] P. Grother. Minex - performance and interoperability of the incits 378 fingerprint template. In*National Institute of Standards and Technology*, page 49. NIST, 2006.

[25] R. Housley, W. Ford, W. Polk, and D. Solo. Internet x.509 public key infrastructure certificate and crl profile, 1999.

[26] Identity-Management. Liberty alliance project. http://www.projectliberty.org.

[27] Russell Impagliazzo and Sara Miner More. Anonymous credentials with biometrically-enforced non-transferability. In *WPES '03: Proceedings of the 2003 ACM workshop on Privacy in the electronic society*, pages 60–71, New York, NY, USA, 2003. ACM Press.

[28] Identix Incorporated. Minutia vs. pattern based fingerprint templates. www.ibia.org/membersadmin/whitepapers/pdf/9/M_vs_P_White%20Paper_v2.pdf.

[29] Internet2. Shibboleth. http://shibboleth.internet2.edu.

[30] ISO/IEC 19784-1 ISO/IEC JTC 1/SC 37 Working Group 2. Information Technology - Biometric Application Programming Interface (BioAPI). 2005.

[31] A. Jain and L. Hong. On-line fingerprint verification.*icpr*, 03:596, 1996.

[32] A. Jain, S. Prabhakar, L. Hong, and S. Pankanti. Filterbank-based fingerprint matching, 2000.

[33] A. Juels and M. Wattenberg. A fuzzy vault scheme. In *Proceedings of IEEE International Symposium on Information Theory, 2002.*, 2002.

[34] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *ACM Conference on Computer and Communications Security*, pages 28–36, 1999.

[35] F. Monrose, M. Reiter, Q. Li, and S. Wetzel. Using voice to generate cryptographic keys. 2001.

[36] Neurotechnologija. http://www.neurotechnologija.com.

[37] Torben Pryds Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO '91*, volume 576, pages 129–140. Springer Verlag, 1992.

[38] N. K. Ratha, J. H. Connell, and R. M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. In *IBM Systems Journal*, volume 3, page 40. IBM, 2001.

[39] Nalini K. Ratha, Kalle Karu, Shaoyun Chen, and Anil K. Jain. A real-time matching system for large fingerprint databases.*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):799–813, 1996.

[40] M. Roe. Performance of symmetric ciphers and one-way hash functions. In Ross Anderson, editor, *Fast Software Encryption*, volume 809 of *Lecture Notes in Computer Science*, pages 83–89, Berlin, 1994. Springer-Verlag.

[41] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[42] SC 37 Secretariat. Text of fcd 19795-2, biometric performance testing and reporting Ű part 2: Testing methodologies for technology and scenario evaluation. In *ISO/IEC JTC 1/SC 37*. ANSI, 2006.

[43] C. E. Shannon. Communication theory of secrecy systems. *Bell Sys. Tech. J.*, 28:657–715, 1949.

[44] U. Uludag, S. Pankanti, S. Prabhakar, and A. Jain. Biometric cryptosystems: Issues and challenges, 2004.

[45] IBM Microsoft RSA VeriSign. Web Services Federation Language (WS-Federation). version 1.0. July 8 2003. `http://www-128.ibm.com/developerworks/library/specification/ws-fed/`.

[46] Z. W. Wang, S. K. M. Wong, and Y. Y. Yao. An analysis of vector space models based on computational geometry. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 152–160, New York, NY, USA, 1992. ACM Press.

[47] Harry Wu and Gerard Salton. A comparison of search term weighting: term relevance vs. inverse document frequency. In *SIGIR '81: Proceedings of the 4th annual international ACM SIGIR conference on Information storage and retrieval*, pages 30–39, New York, NY, USA, 1981. ACM Press.

[48] Wende Zhang, Yao-Jen Chang, and Tsuhan Chen. Optimal thresholding for key generation based on biometrics. In *ICIP*, pages 3451–3454, 2004.