uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Brent Carrara
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.C.S.
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Achieving Non-Transferability in a Digital Identity Management System
TITRE DE LA THÈSE / TITLE OF THESIS

Carlisle Adams
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

Anil Somayaji

Guy-Vincent Jourdan

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Achieving Non-Transferability in a Digital Identity Management System

by

Brent Carrara

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.C.S. degree in Computer Science

School of Information Technology and Engineering
University of Ottawa

Your file Votre référence
ISBN: 978-0-494-65523-8
Our file Notre référence
ISBN: 978-0-494-65523-8

Abstract

In this work, we study the use of digital credentials in an online digital identity management setting. We focus on the problem of users sharing their digital credentials with one another, a problem called the "transferability" or "lending" problem. Further, we present the detailed mathematics showing how non-transferability can be achieved through biometrics. By tying the physical identity of a user to their digital identity, we are better able to prevent users from sharing their credentials. Additionally, we implement a state of the art biometric key generation algorithm and apply it to voice biometrics. In our results, we found that we are able to generate cryptographic keys significantly stronger than those of previous work.

Acknowledgements

First and foremost, I would like to thank Dr. Carlisle Adams for his guidance throughout my studies at the University of Ottawa. Professor Adams introduced me to digital credentials during his graduate level security course and I have been intrigued by the technology ever since. Without Professor Adams' help this work would not have been possible, thank you!

Secondly, I would like to thank both Dr. Fabian Monrose and Dr. Lucas Ballard. Both are world leaders in the area of biometric key generation and their guidance during my study of biometric key generators was invaluable. I would not have achieved the results that I did without their help.

Lastly, I would like to thank my support system very much. My wife, Julie, has supported me throughout my studies and for that I will be forever grateful. Also, I would like to thank my employer very much for supporting my initiative to go back to school. Last but not least, I would like to thank my parents. It goes without saying, but none of this would have been possible without them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Individuals have grown accustomed to disclosing personal information to organizations in order to receive a service in return. As a consequence, both fraud and identity theft have become major problems for organizations who offer these services and for the public who use them. A major source of these problems is organizations requiring their customers to reveal unnecessary, personal information about themselves. As an example, suppose Alice organizes a business meeting with Bob at her local pub, Trudy's Pub. In order for Alice to enter Trudy's Pub she is required to show a piece of identification, a driver's license or a Passport. In doing so Alice discloses a number of personally identifiable attributes about herself to Trudy's Pub, when the goal of Alice showing identification was simply to prove that she is over the age of majority. The problem of individuals unnecessarily disclosing personal information to organizations is exacerbated by the fact that people are no longer in control of what personal information about them is being held by organizations. If we go back to our example of Alice at Trudy's pub, it is clear that Alice has no knowledge of what personal information the doorman

at Trudy's Pub has either memorized, or electronically copied off of Alice's document, resulting in a violation of Alice's privacy.

Additionally, organizations are increasingly building data warehouses of their customers' personal information in an effort to provide targeted ads and promotions. This situation might not be of concern when organizations operate in silos where they do not communicate with one another; however, when organizations merge, form a partnership or collude with one another, privacy concerns must be raised. As an example, suppose Alice goes to Bob's Bank to receive a credit review because she is interested in obtaining a mortgage of $250,000 for a house. Alice is approved and takes Bob's positive credit review to Trudy's Mortgage Brokerage to obtain a mortgage. In doing so Alice reveals personal information about her relationship with Bob to Trudy, such as her bank account number at Bob's Bank, her credit score, historical financial information, etc., when all that was required by Trudy was confirmation from Bob's Bank that Alice is credit worthy. By revealing unnecessary information about Alice's relationship with Bob's Bank to Trudy's Mortgage Brokerage, both Trudy and Bob can now correlate the transactions of Alice to build a dossier of her actions. Correlation between organizations is a problem for clients of cooperating organizations, especially when medical and insurance information is at risk of being shared.

In recent years, two technologies have been proposed to address the privacy issues of modern identification methods: *Digital Credentials* [19] and *Anonymous Credentials* [22]. *Digital Credentials* combine public key encryption, blind signatures and zero-knowledge proofs to allow an individual to selectively disclose attributes about themselves in a privacy-preserving manner. Individuals generate a secret key, which is a binary encoded string of

their attributes, and a public key, which is a blinded version of their private key. Through the use of blind signatures an individual receives a signature on their public key from a Credential Authority (CA), and through the use of a zero-knowledge proof they are able to selectively disclose attributes about themselves to organizations. *Anonymous Credentials* on the other hand use pseudonymous transactions to preserve the anonymity of an individual. *Anonymous Credentials* allow an individual to posses a unique pseudonym with different organizations to prevent linkage between them. Credentials are obtained from organizations and are tied to their pseudonym with that organization. Through a proof of knowledge an individual is able to prove possession of a credential to verifiers as well as other organizations in a privacy-preserving and unlinkable manner.

While both *Digital Credentials* and *Anonymous Credentials* have received widespread research each system has their own advantages and disadvantages. Brands' *Digital Credential* scheme allows attributes within a credential to be selectively disclosed without unnecessarily revealing the other attributes in a credential; however, a credential that is shown multiple times can be linked to the same individual. *Anonymous Credentials*, on the other hand, does not have the selective disclosure property, as each attribute is itself a self-contained attribute; however, *Anonymous Credentials* can be shown multiple times in an unlinkable fashion. Two researchers, Persiano and Visconti [83, 84], have been able to build upon *Digital Credentials* and have proposed a scheme that can create credentials that have attributes that can be selectively disclosed **and** shown multiple times in an unlinkable fashion.

Credential schemes were invented to allow individuals to show their credentials to multiple organizations in an unlinkable way. Many different proposals

have been put forth which describe practical, efficient schemes to accomplish this task; however, in all of the proposed schemes a user's digital identity is simply a binary string of data, which can easily be transferred from one individual to another. Some solutions propose storing a user's digital credential on a tamper-proof device such as a smartcard or in a file on an individual's laptop; however, these media are trivial to share with other individuals. This problem has been deemed the "transferability" or "lending" problem, where a digital credential scheme that prevents sharing is referred to as having the property of "non-transferability". To date most schemes have proposed using discouragement to achieve non-transferability. By encoding some highly personal information in an individual's secret key, a credit card number for example, it is the hope that an individual would be reluctant to share their secret key. These solutions however, are insufficient for high security applications as two individuals who trust each other would have no problem sharing their credentials with each other.

Recently, researchers have proposed using biometrics to tie a digital credential to a specific individual in order to achieve non-transferability. Biometric authentication is an active area of research due to the permanence, non-repudiation, and portability of an individual's biometric reading. Biometric authentication has been studied as an alternative to password based authentication to compensate for an individual's tendency to choose an insecure password, share their password with someone, write their password down, or completely forget their password. Although biometric signals can provide an alternative to password based authentication there are a number of concerns about their use. A biometric signal is relatively invariant between readings of the same individual, which allows an individual to be authenticated with a high degree of confidence by comparing raw biometric readings against a

stored biometric template. However, the same invariant property of a biometric signal is also a deterrent to biometric systems being used; once an organization has an individual's biometric template there is no way for an individual to modify, reissue, or revoke their template. Privacy-preserving biometric authentication research aims to solve some of the problems that have been raised with traditional biometric authentication by combining their research with cryptography.

In this Thesis we show how the digital credential scheme of Persiano and Visconti [83,84] can be modified to allow for biometric-based non-transferability by combining digital credential research with biometric authentication research. In doing so we apply our new digital credential scheme to the problem of digital identity management in an online environment.

In Chapter 3, we lay out the model we will use to evaluate our solution as well as present the adversaries in the model. We also show how current online digital identity management systems fall short of preventing our adversaries from performing privacy-invasive attacks. Lastly, we introduce our proposed policy-based digital identity scheme which is designed to reduce the effectiveness of these adversaries. In Chapter 4, we present the detailed mathematics behind Persiano and Visconti's digital credential scheme, and present the extensions which allow us to extend their scheme to achieve our proposed policy-based digital identity scheme. In Chapter 5, we detail our proposed biometric key generation algorithm and present the empirical results of applying our proposed algorithm to a database of voice samples.

1.1 Contributions

The main objective of this Thesis is to show that biometrics can be used to achieve a non-transferable digital credential system. By applying the research in the areas of digital credentials and biometric key generation to the problem of online digital identity management we are able to show that non-transferable digital credentials can be used as a solution to a real-world problem. In doing so, we make the following contributions:

Chapter 3 : We present the problem that digital identity systems aim to solve and present the basic model used by many of today's digital identity management systems. We present the adversaries in the model and describe our new extended model as well as describe our contribution of a policy-based digital identity management system.

Chapter 4 : We present the detailed mathematics behind our policy-based digital identity management system. We describe, in detail, how we applied two existing proposals to the digital credential scheme of Persiano and Visconti. The extensions add biometric-based non-transferability as well as proxy support to their scheme.

Chapter 5 : We detail how we modified an existing biometric key generation algorithm to reliably create biometric keys from voice biometrics as well as present the experimental results of our study. Additionally, we detail our novel feature extraction algorithm and present the empirical results of our experiment to show that high-entropy keys can be created from voice-based biometrics.

We begin in the following chapter by presenting the background literature on digital credentials and biometrics necessary to understand the remainder of this Thesis.

Chapter 2

Background

Biometric authentication provides an efficient and practical solution to the transferability problem of digital credentials. This literature review will cover the area of digital credentials and privacy-preserving biometric authentication and conclude with how they can be brought together to achieve non-transferability.

2.1 Digital Credentials

In 1985, Chaum produced his seminal work in the area of pseudonymous based transactions by specifying a theoretical solution to the problem of linkable transactions [31]. In [31], Chaum was concerned with the simplicity with which organizations could link the electronic transactions of a particular individual, empowering organizations to easily build a dossier of a particular individual's actions. In his theoretical solution, Chaum introduced the idea of using pseudonym-based transactions. His scheme provided a means

for individuals to use unique pseudonyms when interacting with disparate organizations. As an example, an individual interacting with Organization A, Organization B, and Organization C would use three distinct pseudonyms (e.g. 762, 451, and 314), as opposed to a single pseudonym [31]. By using distinct pseudonyms, organizations are prevented from being able to work together to compile a dossier of an individual's information.

Chaum's proposed pseudonym-based transaction system is based on his blind signature research [30]. In Chaum's blind signature scheme, an individual, the provider, possesses a value that requires signing by an organization, the signer, for verification by a third party, the checker. In the untraceable transactions analogy presented in [31], Alice, the individual (provider), possesses two pseudonyms, 523, and 965, for communication with organization X, the credential issuing organization (signer), and organization Y, the credential receiving organization (checker), respectively. In order to receive a credential from organization X, Alice produces a slip of paper with the pseudonym 523 written on it. The slip of paper is placed in a windowed envelope, which exposes the value 523. A piece of carbon paper is added to the envelope, and it is sealed. Alice then presents the envelope to the credential issuing organization, X. Organization X produces a stamp on the envelope to indicate that 523 has been granted the credential C. Since the envelope contained a piece of carbon paper the imprint of the stamp is made visible on the original slip of paper marked with the pseudonym 523. After applying the stamp, the envelope is returned to Alice, where she is able to verify the stamp on the envelope. At that point Alice opens the envelope, discards the carbon paper and writes down the pseudonym 965 on the slip of paper (in a different location on the paper, i.e. not where the pseudonym 523 is). The slip of paper is inserted into a second windowed envelope, which exposes the cre-

dential granted by X, and Alice's pseudonym with organization Y (hiding the pseudonym with organization X). The envelope is then sealed and delivered to Y. Upon receiving the envelope, Y is able to verify X's stamp and thus is assured that Alice does in fact possess a credential, C, from organization X. This example is analogous to the real blind signature scheme as proposed in [30, 31] and is shown pictorially in **Figure 2.1** [31]. The theoretical blind signature scheme relies on one-way functions, which are easy to compute, but whose inverse is believed to be infeasible to compute.

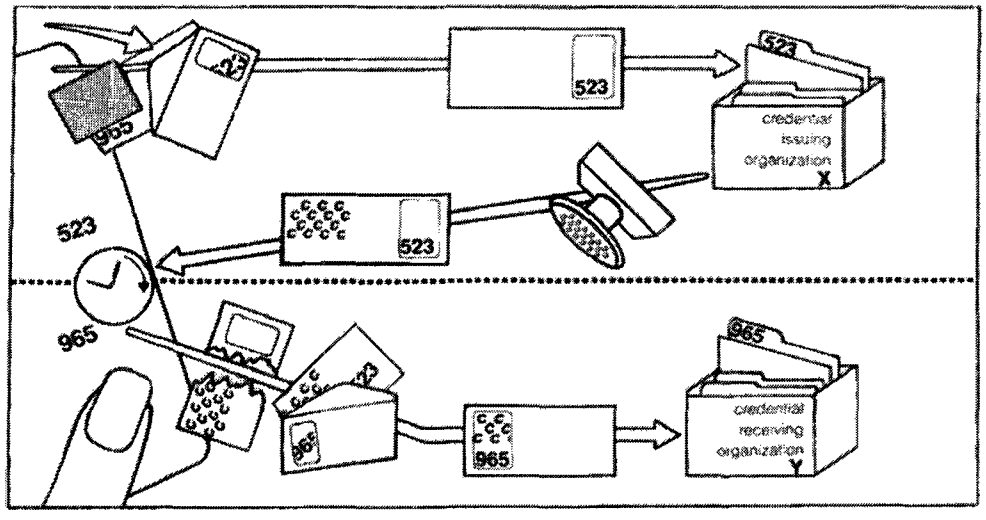

Figure 2.1: Untraceable Digital Credentials [31]

Chaum furthered his research into digital credentials in [33]. In [33], Chaum and Evertse looked at solving the problem of transferring an individual's personal information (credentials) from one organization to another in an unlinkable manner. Their scheme relies upon a trusted third party, the signing authority, Z, to produce pseudonyms and signatures. As a deterrent to forgeries, a trusted Z is responsible for creating pseudonyms to ensure

they are properly constructed (unforgeability property). An individual's pseudonym, as granted by Z, consists of a pseudonym and validating material. The validating material is used by the organizations to ensure the pseudonyms have been properly constructed. Additionally, Z is responsible for issuing credentials to individuals so that they can be shown to multiple organizations in an unlinkable manner (unlinkability property). Although this scheme does theoretically solve the problem of unlinkable transactions, it requires a trusted party, Z, to be available at all times. Additionally, since Z is the creator of all the pseudonyms, it is possible for Z to reveal all the pseudonyms of a particular individual, thus de-anonymizing the individual. However, in [32], Chaum attempted to practically limit the power of Z by replacing the one-way function used in [33] with RSA signatures [92]. By including the individual's public key in his signature scheme, individuals are able to vindicate themselves if Z is believed to be acting maliciously. [32] is the first paper to introduce a scheme where the individuals are in complete control of their credentials. RSA signatures, as presented in [92], however, are not secure against adaptive chosen message attacks (CMA) [41].

After Chaum introduced the idea of digital credentials in the late 1980's a number of other researchers proposed theoretical solutions to the linkable transactions problem. In [41], Damgard proposed using claw-free, trapdoor functions that are secure against adaptive-CMA as part of his signature scheme. As defined in [72], a one-way function, f, is a trapdoor function if it has an associated secret, s_f, which makes f easy to invert. Additionally, two functions, f_0, f_1, are claw-free if there is no efficient algorithm to find, x, y, and z such that $f_0(x) = f_1(y) = z$ [72]. [41] relies on an entity Z, to create an individual's pseudonyms, however Z is not as trusted as it is in [33]. In [36], Chen proposed a new credential scheme based on the as-

sumed difficulty of the discrete logarithm problem. Chen enhanced the work done in [41], by requiring Z to only be available during the pseudonym construction phase. Chen's pseudonyms are based on the Fiat-Shamir signature scheme as described in [46]. In [41], credentials are granted by the organizations and managed by the individuals, a desirable property. Furthermore, the credentials issued by the organizations can be shown in an unlinkable manner to other organizations; however, a credential can only be used once. If a credential needs be shown multiple times, the user must obtain multiple versions of the same credential from the issuing organization. Additionally, in [36] a malicious Z has the ability to transfer credentials between users as pointed out by Lysyanskaya, et al., in [71].

By the late 1990's and early 2000's a number of researchers began proposing practical schemes to solve the linkable transactions problem, namely Lysyanskaya [71], Camenisch [22], and Brands [19]. In addition to solving the linkable transactions problem, the researchers were interested in deterring individuals from sharing their digital credentials; a problem termed the "transferability", or "lending" problem [22]. In [71], Lysyanskaya proposed the first digital credential scheme to achieve a level of "non-transferability". All previous constructions to [71] easily allowed individuals to transfer their digital credentials and pseudonyms from one individual to another without detection by organizations or detriment to their anonymity. In [71], users are motivated to not share their identity through the use of discouragement. Each user in [71] possesses a master public key and a corresponding master secret key, which the individual is highly motivated to keep secret. Sharing their master secret key would allow other individuals to act on their behalf in sensitive operations, such as signing legal documents or encrypting data. In the construction of [71], a user cannot share their credentials with-

out sharing their master secret key. The scheme as described in [71] relies on zero-knowledge proofs [44], bit commitment schemes [79] and signature schemes [50, 93]. Furthermore, in a similar fashion to [36], an individual in [71] must receive multiple credentials from its credential granting organization in order to show a credential multiple times. Although using multiple credentials does allow a credential to be used multiple times, it requires the individual to either manage a collection of credentials, or requires the credential granting authority to be available at all times.

In [22], Camenisch and Lysyanskaya introduced *Anonymous Credentials*, and proposed the first practical digital credential scheme that allowed credentials to be shown multiple times without being linked, a novel ideal at the time. Camenisch and Lysyanskaya built their scheme on the strong RSA assumption [38] and the decisional Diffie-Hellman (DDH) assumption [72]. By using the scheme proposed in [22], an individual is able to demonstrate to an organization their possession of a credential from a different organization as many times as required without being linked, through the use of zero-knowledge proofs. In [22], two separate solutions for achieving non-transferability were presented: *all-or-nothing sharing*, and *PKI-based non-transferability*. Both solutions however, continue to use the technique of discouragement. In *all-or-nothing sharing*, the act of an individual, A, sharing a single credential with a second individual, B, grants B access to all of individual A's credentials through the use of a novel technique entitled *circular encryptions* [22]. *PKI-based non-transferability* is similar to *all-or-nothing sharing* except it does not rely on circular encryptions and assumes a PKI infrastructure is in place, which in most cases is unreasonable. Both solutions are derived from the non-transferability solution of [71]. Unfortunately, the introduction of circular encryptions to achieve non-transferability adds significant computa-

tional overhead to the system, which negatively affects performance. As an additional property, the scheme in [22] boasts the ability to revoke credentials. The scheme in [22] offers two levels of revocation, global anonymity revocation, which allows an entity to globally reveal the identity of an individual whose transactions are illegal, and local anonymity revocation, which allows an entity to reveal a user's pseudonym with an issuing organization. Revocation is a problem of paramount importance in group signature schemes [6,35] and identity escrow schemes [64] which unfortunately adds additional entities, computational load, and thus overheard to *Anonymous Credentials*. A Java implementation of Camenisch and Lysyanskaya's protocol is given in [27].

Competing with the protocol of Camenisch and Lysyanskaya in [22] is Brands' protocol [19]. In [19], Brands introduced *Digital Credentials*, a scheme that allows individuals to selectively disclose specific attributes contained in their credential. The scheme in [19] relies on the discrete logarithm problem being hard and uses blind signatures to prevent the credential authority from being able to track an individual's credential use in the system. Brands' scheme relies on the presence of a credential authority (CA), which is responsible for validating an individual's attributes and producing a public, blinded certificate for the individual (the digital credential). In order for an individual to prove to an organization that they are in possession of a specific attribute, a user's blinded certificate, along with their associated secret key, is used in a proof of knowledge with the organization. Through a property termed *selective disclosure*, individuals in the *Digital Credential* scheme are able to prove linear Boolean relationships between their attributes without unnecessarily revealing any additional information about themselves. Furthermore, *Digital Credentials* address the transferability problem through the use of discouragement. In [19], non-transferability is achieved by encoding a highly secret

value into the individual's secret key. Sharing a *Digital Credential* requires sharing its associated secret key, and thus sharing the individual's highly secret value. Although this solution is used in [22, 44], it is undesirable as extremely close friends, family members, or members of a gang would have no problem sharing their secret information with one another. Although [19] provides for a practical digital credential scheme, it suffers from not being able to issue multi-show credentials. Additionally, in [19], extra care must be taken in the showing protocol as the credential authority can de-anonymize an individual if it sees the values exchanged.

In [19], Brands' presented a novel technique that allows individuals to selectively disclose specific attributes in their digital credential. Although [19] boasts this desirable property, it suffers from not being able to show a credential multiple times in the same way that [22] does. In [83], Persiano and Visconti proposed a solution to the multi-show problem within *Digital Credentials* [19], by introducing the notion of master and slave chameleon certificates. Chameleon certificates allow for individuals to selectively disclose their attributes during the showing protocol while allowing credentials to be shown multiple times. The scheme in [83] is based on group signatures [26] and relies on the discrete logarithm problem being hard. The scheme in [83] consists of organizations, which issue master chameleon certificates; users, who receive the master chameleon certificates based on their attributes and are able to derive slave chameleon certificates from them; and service providers, which require users to prove they can satisfy some linear Boolean function with their slave chameleon certificate in order to access a protected resource. Users, once granted a master chameleon certificate, are able to perform a refresh operation on their certificate to derive a slave chameleon certificate. Slave chameleon certificates allow credentials to be

multiple-use by empowering users to derive them on-demand, without any intervention by a third-party. By using the derived certificate only once, users are able to satisfy a service provider's authentication policy in an unlinkable way. The authors of [83] again address non-transferability through discouragement and integrate revocation into their scheme by supporting short lived certificates, which can be checked to ensure the credential is still valid. Although [83] enhances [19], it suffers from a major vulnerability as pointed out and patched by Yang, et al., in [107]. In [107], Yang, et al., revealed that credential authorities could easily link the credentials of their users by cleverly constructing their public parameters. In [84], Persiano and Visconti presented a more efficient credential scheme than [83] that allows individuals to create digital credentials which can be shown multiple times. Although their scheme does require a number of PoKs it does allow credentials to be selectively disclosed and shown multiple times.

As another solution to the multiple-use credential problem, Verheul introduced the idea of using elliptic curve cryptography and Weil pairings for certificate construction in [104]. Verheul extended the use of traditional certificates to be self-blinding certificates, which allow a user to translate a single certificate into a new certificate during the credential issuing protocol. [104] uses the Chaum and Pedersen [34] digital signature scheme along with elliptic curve cryptography and Weil pairings to construct the self-blinding certificates. In [104], users acquire a first pseudonymous certificate (FPC), and a self-blinding certificate from a pseudonym provider during initial registration. During the credential issuing protocol, users transform their FPC into a random pseudonymous certificate (RPC) and obtain a credential in the form of a credential pseudonymous certificate (CPC). During the showing protocol, users are able to demonstrate to service providers possession of a

credential by proving they possess the secret key associated with their CPC. Verheul discusses a basic revocation policy, which enforces that the signing keys used for the certificates be short-lived; however, non-transferability is not discussed.

Peripherally relevant to the discussion of digital credential schemes is the research on group signatures [35] and identity escrow [64]. Group signature schemes as introduced in [6], allow members of a group to sign messages on behalf of the group. Members of the group are essentially anonymized with respect to an outside observer validating the signature; however, a group manager is able to determine which member of the group actually signed the message. Related to group signatures is identity escrow. Identity escrow schemes allow a party, A, to supply a party, B, with information that would allow a third party, E, to identify A. The information supplied to party B by A, however, does not allow party B to solely identity party A without the help of E. Both group signatures and identity escrow use cryptographic certificates to achieve their individual goals. Many privacy-preserving, certificate based credential systems have been built on these technologies [6,23–25,83]. Digital credential schemes are immediately available from both group signatures and identity escrow; however, a solution to the transferability problem is not. While providing efficient and cryptographically secure credential schemes, identity escrow and group signatures do not solve the transferability problem.

Directly related to achieving non-transferability is research in the area of k-times authentication schemes. k-times authentication schemes look to ensure that a specific individual cannot access a resource more than k times. Schemes, such as [13], which apply a k-times authentication solution to online subscriptions are directly relevant to the transferability of credentials. In [13],

Blanton discussed a scheme in which an online service can provide anonymous access to its resources, while preserving unlinkability through the use of blind signatures and zero-knowledge proofs. In Blanton's scheme, users are provided with an authentication token, which is composed of a commitment on a random value, m, a subscription type, and an expiration time. When accessing a resource, a user structurally randomizes their authentication token to provide unlinkability, and supplies it to the service provider. Upon receiving the authentication token the service provider is able to check that the randomized token has not been used before and that it is not expired. Upon validating the token, the service provider computes a new randomized value, m', and provides the user with a new authentication token for later use. [13] shows the results of an implementation of their system and discusses how it can theoretically be extended to many different subscription types.

In 2004, Teranishi, et al., introduced a k-times anonymous authentication (k-TAA) scheme in [101]. The k-TAA scheme is built upon group signatures and identity escrow. In [101], a group manager is responsible for granting a user access to a protected resource, managed by a service provider. The scheme allows for a user to be able to anonymously, and unlinkably, access the service provider's resource a fixed number of times, k. There is no mechanism for either the group manager or the service provider to de-anonymize a user if they only access the resource a maximum of k times. Malicious users, however, who attempt to access the service provider more than k times can be de-anonymized. Although not built upon digital credentials, this scheme does present a solution to the transferability problem by limiting the number of times a user can access a resource; however, for close-knit groups of malicious users this merely forces them to keep track of their usage and does not prevent them from sharing access to the service provider's resources. [81] extended the

research of Teranishi, et al., to allow service providers to revoke individual users. The research of Nguyen and Safavi-Naini, however, is not privacy preserving, as it allows service providers to keep track of a list of users in order to provide revocation. Furthermore, in [37], Chen, et al., extended the research of [81, 101] and introduced the concept of a multi-coupon (MC). In the MC scheme proposed in [37], a user is issued a collection of k coupons from a vendor. The coupons allow a user to authenticate at most k times, using each coupon at most once. In Chen, et al.'s, scheme the coupons must be used sequentially to receive access to a protected resource. The scheme has the property that each of the user's coupons must be shown sequentially, and is built upon the *all-or-nothing* sharing scheme. In [37], sharing a single coupon implies sharing all the coupons. Although the authors argue their solution is practical it is still subject to the transferability problem, as groups willing to share their MC can do so easily.

In [21], Camenisch, et al., combined their research in *Anonymous Credentials* with the research in k-times authentication schemes. [21] presents a practical credential scheme that allows a user to anonymously authenticate at most n times during a predefined time period. A dispenser is used to make n e-tokens available to a user each time period. Malicious users who attempt to reuse their e-tokens or use $n+1$ e-tokens during a time period can have their transactions linked. Although relevant to the discussion of non-transferability, the research in [21] does not adequately solve the problem. Under this scheme it is still possible for a malicious group of users to share the n e-tokens they receive each time period. Furthermore, this scheme does not encode any attribute values into the e-tokens; simply presenting an e-token allows a user to be authenticated. Since attribute values are not encoded in the tokens this system cannot be used as a practical credential scheme.

Although the schemes in [13, 21, 37, 81, 101] prevent users from using their credentials, tokens, and coupons an unlimited number of times, they do not adequately prevent users from sharing their credentials. In order to prevent users from sharing their credentials, the schemes in [13,21,37,81,101] merely make it slightly more cumbersome for a group of malicious users to do so. Using these methods, two highly motivated users could still easily share their credentials by transferring, perhaps via E-Mail, their refreshed, or remaining authentication tokens back and forth between each of their uses. In order to make transferring credentials more difficult, a number of researchers have proposed using biometrics as a technique to ensure non-transferability.

In 1998, Bleumer introduced the idea of using biometric data to enforce non-transferability in [14], by building on the wallet-with-observer research of [34, 39]. To our knowledge, [14] marked the first paper to present a scheme other than discouragement or k-times authentication to enforce non-transferability. In his paper, Bleumer assumed each user is equipped with a personal communication device, a wallet, which runs a trusted process, the observer. In his scheme the issuer, the wallet, and the observer participate in a binding protocol that presents the issuer with a pseudonym for the user, while the wallet and observer both obtain secret information that is required in later protocols. In Bleumer's scheme it is the responsibility of the observer to verify the authenticity of the user's biometric identity. During the credential issuing protocol the user is granted a credential (or credentials) on an interim pseudonym where again the wallet and observer are provided with secret information. During the showing protocol, a user is able to demonstrate their possession of a credential once they have provided a valid biometric sample to the observer. In Bleumer's theoretical scheme, both the user and any service provider must trust the observer to perform legitimate opera-

tions. Although Bleumer's scheme addresses the transferability problem, it does not discuss any practical biometric algorithms that can be used to implement his solution. Additionally, since Bleumer's solution is built upon the theoretical wallet-with-observer architecture, no practical biometric scheme is immediately available for the implementation of his solution.

In [57], Impagliazzo and More expanded on the research of Bleumer by combining anonymous credentials and biometric authentication. In [57], Impagliazzo and More introduced the concept of *strongly subliminal-free zero-knowledge proofs*. According to Impagliazzo and More, a *strongly subliminal-free zero-knowledge proof* leaks a single bit of information, which is intended to notify the user that one of the parties involved in the protocol is cheating. Through *strongly subliminal-free* protocols, a misbehaving party can be detected while preventing information from being leaked. In the construction of [57], there are credential-issuing authorities and users. The authority's goal is to grant authorization in the form of credentials to the users, while the users are interested in protecting their privacy. The biometric data used to authenticate the user is never shown to the authority and is stored on a tamper-resistant smartcard, which is issued by a proxy agent for the authority. The wallet-with-observer architecture is used in [57], where the wallet is referred to as the warden, and the observer is implemented as a tamper-resistant smartcard, which is issued by the trusted authority. During distribution, the tamper-resistant smartcard is loaded with the user's biometric data. During authentication the card is responsible for checking a fresh biometric sample against its stored template. The card together with the user form what Impagliazzo and More refer to as the prover. The role of the warden is to ensure that the prover abides by the protocols of the scheme, by using *strongly subliminal-free* protocols, while the card ensures that the

warden cannot tamper with the cryptographic material and biometric data stored on the tamper-resistant smartcard. Although [57] is a more secure and advanced solution to the problem of transferability it still does not address the practical issue of what biometric should be used, nor is it based on the pseudonymous based protocols of [19,22], which allow for credentials issued by one organization to be shown to another. Additionally, [57] suffers from the problem that the wallet-with-observer architecture is built upon the unrealistic assumption that the warden and prover are unable to communicate their secret information to the outside world. Lastly, by the admission of the author, the scheme in [57] is only as secure as the tamper-resistant smartcard.

Most recently, in [2], Adams proposed a generic biometric solution to the transferability problem of digital credentials. Adams' scheme is based on bit-commitments and can be applied to both *Anonymous Credentials* [22] and *Digital Credentials* [19], although his paper focuses on an implementation using Brands' *Digital Credentials*. Adams' scheme relies on the presence of a tamper-resistant, trusted device that is used to generate commitments on a user's biometric readings. During the issuing protocol, the device takes the user's initial biometric reading, b_I, and outputs a commitment, c_I. The produced commitment is then encoded in the user's digital credential. In Adams' example, the commitment, c_I, is encoded as one of the user's attributes in their blinded certificate (using Brand's scheme [19]). During the showing protocol, the user supplies a fresh biometric sample, b_F, to the trusted device and receives a commitment, c_S, while computing a second commitment, c_F, on the user's initial reading, b_I. Through the use of single-bit commitments the user is able to demonstrate to the verifier that both b_I, and b_F are biometric readings from the same individual. A verifier is able to determine

if b_I, and b_F are from the same individual if $c_I and c_F$ agree in t or more bits, where t is a system parameter. Adams' scheme is not reliant on any single biometric and can be used with any biometric algorithm that produces binary strings, as long as they can be compared for "closeness" by using a simple Hamming distance calculation. Although [2] is a novel solution to the transferability problem, Adams' protocol is computationally intensive and relies on a tamper-resistant device. Despite these drawbacks, Adams' work presents the most practical solution to date for addressing non-transferability.

2.2 Biometrics

Research in the area of biometric authentication has been active for a number of years. Of interest is research in the area of privacy-preserving biometric authentication. Over the last decade quite a bit of research has been done to answer the question: how can an individual be biometrically authenticated, using a stored template that does not leak any information about them? The research in privacy-preserving biometric authentication is split up into four categories: biometric salting, biometric key generation, fuzzy schemes, and non-invertible transforms [87].

In the category of fuzzy schemes is the novel work done by Davida, et al., [42]. In [42], Davida, et al., proposed the first privacy-preserving biometric authentication scheme to our knowledge. The authors proposed using majority decoding and error correcting codes (ECC) to allow distinct biometric readings to be compared using a Hamming distance calculation. Their scheme was applied to iris scans and the resultant biometric templates were stored on a portable card with a magnetic strip. Although the work in [42] was ground

breaking, it suffers from a number of problems: the majority-decoding algorithm proposed is not practical in reality, and the ECC information produced by the algorithm leaks information about the individual. Furthermore, since no randomness is introduced during the execution of their algorithm, their scheme is susceptible to a simple replay attack.

In [61], Juels and Wattenberg further built on the idea of using ECC by proposing a novel fuzzy commitment scheme [61]. The proposed scheme uses bit commitments as a mechanism to preserve the privacy of an individual's biometric reading. The construction of fuzzy commitments forgives small perturbations in the individual's biometric reading during validation. Furthermore, ECC is used to ensure that multiple readings are corrected to the same value. Although novel, the scheme in [61] does not allow extracted features to be reordered, a common problem in fingerprint biometrics. Additionally, multiple commitments have been demonstrated to be linkable [51]. As a follow-up, Juels proposed a new scheme, fuzzy vaults, to address the problem of feature reordering [60]. In [60], Juels outlined a scheme in which Alice places a secret value, k, in a fuzzy vault and locks it using a set of elements, A. Alice's fuzzy vault is then unlocked by a set, B, if the set B is sufficiently close to the set A. Although Juels' scheme addressed the problem of reordering through the use of Reed-Solomon codes, an implementation showed the algorithm to be memory intensive and to leak a significant amount of information about the individual. An implementation of a fingerprint based fuzzy vault scheme can be found in [78], in which helper-data is used to align the fingerprint images. An analysis of the fuzzy commitment schemes of Juels can be found in [56], where Ignatenko and Willems conclude that any commitment scheme which relies on ECC will always leak information about an individual's biometric. In [56], Ignatenko and Willems

proposed using a masking layer to protect the individual's privacy.

In [43], Dodis, et al., proposed a new construction, the secure sketch. Their construction relies on a fuzzy extractor to extract nearly uniform randomness from an individual's biometric. The extracted randomness is then used to create a secure sketch for the individual. Secure sketches are intended to allow information about a biometric to be communicated publicly, while protecting it at the same time. Additionally, secure sketches allow the exact reconstruction of an individual's biometric during subsequent readings when the readings are "close" to the user's enrollment data. Reconstruction in [43] is facilitated using ECC and a hash function. Most notably, Dodis, et al., discussed how the extracted randomness could be used for cryptographic purposes. However, in [17], Boyen was able to demonstrate that fuzzy extractors were not practical for multiple uses. To solve the problem of multiple-use, Boyen proposed introducing random perturbations into the fuzzy extraction algorithm. Information about the random perturbations could then be stored as public information and used to perturb further readings in the same way. A security analysis of fuzzy extractors is given in [68] by Li, et al., in which they evaluate the entropy loss of [43]. Entropy loss is a measure of the advantage a secure sketch gives to an adversary trying to reconstruct an individual's biometric. Furthermore, Li, et al., proposed a scheme whose entropy loss is nlog3, where n is the number of points of randomness extracted from an individual's biometric. An implementation of a secure sketch algorithm can be found in [20], where Bringer, et al., uses fingerprint biometrics as their case study. Bringer, et al., were able to achieve a FRR of 3% and a FAR between five and six percent.

In 2001, Ratha, et al., evaluated the state of biometric systems and com-

mented on the ways a biometric system could be attacked [88]. In their seminal work, Ratha, et al., proposed a new biometric scheme entitled cancelable biometrics, a novel scheme in the realm of non-invertible transforms. In their cancelable biometric scheme, biometric readings are distorted in a repeatable way based on a chosen non-invertible transform so that they can be compared in the transformed space. The construction of cancelable biometrics prevents an adversary from synthesizing an individual's biometric from their distorted template. The algorithm also allows for the stored, transformed templates to be revoked. By modifying the way an individual's biometric is transformed, a new template can be created by simply adapting the transformation. Ratha, et al., later realized their idea of cancelable biometrics by applying three distinct transformations (Cartesian, polar, and functional) to fingerprints [87]. Whereas both Ratha, et al.'s Cartesian and polar transformations were built on random permutations, their functional transformations were built on surface folding which allowed for a random key to be used as a source of randomness. Ratha, et al.'s empirical results showed that functional transformations were the most effective empirically in terms of FAR and FRR and summarized that functional transformations were the most promising transformation they evaluated due to their increased randomness.

Ratha, et al.'s idea of using non-invertible transformations has been applied to other biometrics as well. Savvides, et al., were one of the first groups of researchers to apply Ratha, et al's non-invertible transformation idea to facial biometrics [95]. In [95], the authors proposed distorting facial images with a random convolution kernel. During enrollment, a number of training images are convolved with a chosen random kernel and then combined into a minimum average correlation energy (MACE) filter. During authentication,

a fresh biometric reading is correlated with the individual's stored MACE filter. The authors of [95] claimed 100% acceptance, however their scheme is susceptible to the hill-climbing attack of Adler [3]. Further to the work done by Savvides et al. is the work done by Boult [15,16]. In [15,16], Boult introduced the idea of Biotopes. Biotopes are applied to facial images and function by breaking up feature vectors into an integral and fractional part. The integral part, which is believed to be fairly static between readings, is encrypted, or distorted in a non-invertible way, whereas the fractional part, which is believed to vary between readings, is left untouched. An individual's stored template is composed of the virgin fractional portion, as well as the distorted integral portion of the individual's features. Boult's algorithm relies on the effectiveness of his robust distance measure for comparison during authentication.

A number of other non-invertible schemes have also proposed. In Teoh and Yuang [100], a novel technique, multi-space random projections (MRP) keyed on a pseudorandom number, was introduced. Unfortunately the authors of [100] were unable to demonstrate practical results, achieving an equal error rate (EER) of 16%. Lee, et al., proposed distorting facial biometrics by using principal component analysis (PCA) and independent component analysis (ICA) and were able achieve extremely good results, with an EER of 0.02% [67]. The distorted images of Lee, et al., however, do not distort the face enough and allow discernible facial characters to be easily seen by an observer. In [63], Kang, et al., proposed transforming an individual's facial image by permuting it. The random permutation applied to the individual's image is randomized by a password, which facilitates revocation by simply choosing a new password. [63] however, simply outlines a theoretical biometric authentication scheme and provides no empirical results.

Although the research into non-invertible transformations has produced a number of practical schemes, their method of creating a comparable biometric template in the transformed space is incompatible with the requirements of Adams' scheme [2]. Related to the work done in secure sketches is the work done in biometric key generation. Early work done by Soutar, et al. [97], considered protecting a pre-generated cryptographic key by linking it to an individual's biometric. During the enrollment phase, a cryptographic key is generated and fundamentally linked to the individual's biometric through a process called *Biometric Encryption*. During authentication, the key is extracted if the individual is properly verified. Individuals in [97] are verified using correlation, which makes [97] susceptible to a hill-climbing attack as described in [3]. Uludag, et al., continued with the proposed idea of embedding an individual's cryptographic key into biometric templates in [103] and provided a survey of some of the work done in the area.

A much more promising avenue of research, however, is the research into generating cryptographic keys from a biometric. In [49], Goh and Ngo proposed using eigenanalysis, discretization and Shamir secret-shares to generate cryptographic keys from facial images using a technique they called *bio-hashing*. Goh and Ngo were able to achieve extremely low FRR, and a FAR of 0%. In [51], Golic and Baltatu demonstrated that fuzzy schemes are insecure with respect to template protection. Further, the authors proposed a novel scheme built on two levels of ECC to achieve better EER by combing code-offset construction and code-redundancy. Although they claimed their scheme is optimal in the Shannon-entropy sense, their scheme suffers from high FAR and FRR. In addition to generating biometric keys from facial images, a number of schemes for generating cryptographic keys from iris scans have also been proposed. In [110], Zheng, et al., proposed a scheme which

uses fuzzy commitments and lattice mapping, and were able to achieve an EER of 3%. Furthermore, in [53] Hao, et al., proposed a two-level technique that combines Hadamard and Reed-Solomon ECC techniques. Their scheme stores ECC information in their biometric templates and is able to achieve a FRR of 0.4% and a FAR of 0%.

In the same vein as generating cryptographic keys from biometrics is the research into biometric hashing. Sutcu, et al., first proposed a robust hash function for facial biometrics [99]. In their scheme a one-way transformation is generated by fitting a Gaussian function to extracted features, while hiding it with fake Gaussian functions. Their resultant one-way transformation is finally stored on a tamper resistant card. During authentication, fresh feature vectors are transformed using the stored one-way transformation and quantized. The quantized values are then concatenated together and hashed. Unfortunately, the algorithm of Sutcu, et al., suffers from extremely high FRR and FAR. In [102], Tulyakov, et al., proposed a proof of concept biometric hashing scheme for fingerprint biometrics. Although their scheme is based on solving a set of equations, the algorithm does not introduce any randomness and is therefore unsuitable for multiple showings. In [58], Jin, et al., proposed a practical scheme, termed *BioHashing*, based on introducing randomness and using the Fourier-Mellin transform. In their scheme two factors are required during authentication, the tokenized random number, and the individual's fingerprint. Jin, et al., were also able to demonstrated excellent EER results.

Many practical algorithms to generate cryptographic keys and hashes from a variety of biometrics have been proposed, however, all of the schemes discussed so far have been based on physiological traits that cannot be adapted

by the individual supplying the biometric. A major problem with unsupervised biometric authentication is that a replay attack is trivial; an attacker simply observes an authentic biometric signal and replays it at a later time. Behavioural biometrics, such as handwriting, typing and voice, however, provide for adaptability and variability in the signals they produce. They allow an individual to be authenticated using a challenge-response protocol, where the individual is prompted to produce a different signal each time they are authenticated.

A number of biometric key generation algorithms have been proposed for different behavioural biometrics. In [76], Monrose, et al., proposed hardening an individual's typed password by adding the entropy from their typing dynamics to their password. By observing an individual's key-press duration and inter-key-press delays, the algorithm is able to select an individual's distinguishing, repeatable features. Once calculated, the distinguishing features are used to reconstruct Shamir secret-shares, which are ultimately combined with the individual's password to add entropy. Furthermore, Monrose, et al.'s scheme is able to adapt to an individual's changing typing patterns over time. The FAR, FRR, added entropy and the number of distinguishing features are determined by a sensitivity parameter, k. By varying the sensitivity parameter, varying levels of FAR, FRR, and entropy can be achieved.

Monrose, et al., further applied their technique for generating entropy to voice biometrics in [73–75]. In [74,75], Monrose, et al., described a novel technique to generate 46 bits of entropy from an individual's voice sample. The authors used traditional automatic speech recognition algorithms to extract cepstral feature vectors from audio samples. In their implementation, they used a text-independent corpus of speech samples to generate a vector codebook,

which was subsequently used to segment the user's speech samples through a technique termed *segmental vector quantization* (VQ). Once the speech samples were segmented, the VQ centroids were used to determine if the user's speech sample fell above or below a plane centered at the centroid's value. A value of one was assigned to features that fell below, and a value of zero was assigned to features that fell above. The bit-string generated was then used to look up the individual's secret shares in a lookup table. The shares output by the algorithm were used as the individual's cryptographic key, which was verified by decrypting a stored file. Similar to the result in [73], the statistics of Monrose, et al.'s voice algorithm are dictated by the selected sensitivity parameter, k. Increasing the value of k generally decreases the amount of entropy generated as well as the FRR, yet increases the FAR. Finding the optimal value of k is of paramount importance to their algorithm. In [73], Monrose, et al., furthered their research into generating cryptographic keys from voice by implementing their algorithm on hand-held PDAs. Their original scheme was modified to provide front-end speech processing and attempts to extract 60 bits of entropy. Although the authors discuss the added bits of entropy, they did not conclude that extracting 60 bits of entropy is possible for every individual. Monrose, et al.'s, attacker model is significant for the research presented in this work. In [73–75], Monrose, et al., assumed an attacker had full access to the stored biometric template of an individual; an assumption we also make in this research.

A number of researchers have proposed schemes to generate biometric keys from handwritten signatures as well. In [45], Feng and Wah introduced the idea of *BioPKI*, a novel algorithm for generating cryptographic keys from handwritten signatures. In their scheme 43 features are selected and coded. The concatenation of the coded features is then SHA-1 hashed and used as

the cryptographic key. Although the scheme boasts a respectable EER of 8%, it introduces no randomness and is only able to generate a single template per signature. In [11], Ballard, et al., looked at enumerating and evaluating the ways in which handwriting biometric templates could be attacked in an off-line model. Ballard, et al., were able to show that trained forgers who had access to many samples were able to achieve an EER of 20.6%. A major contribution of Ballard, et al., in [11] is their work on evaluating the effectiveness of an adversary (*generative forger*), who is able to use a statistical attack against the stored templates of an individual. Ballard, et al., were able to demonstrate that a *generative forger* who is able to use statistical measures to concatenate m-grams from a corpus of samples is able to generate an individual's key with an EER of 27.6%. In [106], Vielhauer, et al., proposed a hashing algorithm for handwritten signatures. In their proposal, 24 features were identified and used in conjunction with user specific statistics to construct a hash. Although they demonstrated an excellent of FAR of 0% and FRR of 7.05%, they provided no analysis on the amount of entropy their algorithm extracted. In [105], however, Vielhauer, et al., analyzed their previous work and evaluated their intrapersonal feature deviation, interpersonal entropy, and the correlation between both.

In [9], Ballard, et al., proposed a novel handwriting biometric algorithm, entitled randomized biometric templates (RBTs), which produces templates in a non-verifiable manner. Ballard, et al.'s main goal in [9] was increasing the amount of entropy extracted from handwritten signatures, and argued that their RBTs could be applied to any biometric. The main adversary that Ballard, et al., were concerned with was an attacker generating an individual's cryptographic key from the individual's stored template. Through using system wide quantization values, calculated from the entire popula-

tion, and selecting the individual's features uniformly at random and independent of one another, Ballard, et al., are able to create stored templates that when decrypted are unverifiable by an attacker. In order to achieve an added level of secrecy, the templates are encrypted using an individual's low-entropy password. Through their novel algorithm Ballard, et al., were able to demonstrate that a large portion of the population studied achieved 44 bits of entropy while a portion of the population achieved 80 bits of entropy. In [9], Ballard, et al., compared the results of their new RBTs to the biometric hashing scheme of Vielhauer, et al., [105]. In their analysis, Ballard, et al., demonstrated that while Vielhauer, et al.'s algorithm has a maximum entropy of 40 to 45 bits, theirs has a maximum entropy of 80 bits.

Digital Credentials [19], as originally proposed by Brands, provides an elegant solution to the linkable transactions problem described by Chaum [31]. Unfortunately, *Digital Credentials* do not allow individuals to disclose their credentials more than once without being linked; however, modifications to Brands' original scheme allow credentials to be shown multiple times [83,84,107]. Furthermore, *Digital Credentials* use discouragement to prevent individuals from sharing their credentials. Unfortunately, in high security applications, or when individuals are highly motivated to share their credentials with one another, discouragement is insufficient. The research of Adams, however, has shown that biometrics can be used to make *Digital Credentials* non-transferable [2]. By encoding a commitment on a user's biometric into their digital credential, individuals can be biometrically authenticated while maintaining their privacy.

Compatible with Adams' solution are biometric authentication schemes which produce binary strings that can be compared for closeness by a Hamming

distance calculation. Both biometric hashes and biometric key generation algorithms have this property and are therefore viable alternatives for use in Adams' scheme. The work of Monrose, et al., on biometric key generation allows 46 bits of entropy to be extracted from a behavioural biometric, voice [73–75]. As a possible solution to increase the number of bits of entropy in Monrose, et al.'s algorithm, we will explore applying the work of Ballard, et al., [9] to voice biometrics. By combining the research of Monrose, et al., Ballard, et al., and Adams, we will show that a practical solution to the non-transferability problem can be achieved.

In this chapter we covered the background research in the area of digital credentials and biometric authentication. We have shown that research in these two disparate areas have the potential to be combined to produce a powerful solution to the non-transferability problem. In the next chapter we will outline the model in which we will apply our research and define the adversaries that are present. We will also look at what practical systems exist in this research space.

Chapter 3

Online Digital Identity Systems

Over the last nine years, the number of Internet users has grown 361%, from 360 million users to 1.73 billion users worldwide (as of September, 2009) [52]. Culturing this growth has been the improved quality and content of web applications. Web based applications have developed into dynamic, interactive, and feature-rich experiences that users now rely on to carry out their day-to-day tasks. Web based E-Mail, social networking applications, and online banking are just some of the examples of the web-based applications users have grown accustomed to using. It is estimated that there are 1.4 billion web-based E-Mail accounts worldwide [89], and over 40 million online banking users in the United States alone [70]. The number of registered users on Facebook has recently eclipsed the 250 million mark [111], while MySpace has 76 million registered users from the United States [82]. Further facilitating this growth is the insatiable appetite of users, who require access to their web-based applications on their PCs as well as on their mobile devices. It was estimated that at the end of 2008, there were 4 billion mobile phone subscribers [1]. The W3C estimated in 2004 that the Internet-enabled mobile

phone penetration rate was 49% worldwide at the time [80].

Typically, web applications protect their content by asking users to prove their identity through supplying a username and password in order to grant them access to their services. Users are using so many of these web-based applications that they are now expected to remember a growing number of username/password combinations. E-Mail accounts, online banking accounts, social networking applications, government websites, etc., all require separate authentication. In more complex schemes, a user is required to prove their identity by supplying more personal information during authentication, such as their date of birth, their country, etc. All of this information supplied by the user to the organization running the web-based service is referred to as the user's identity. This personal information, once supplied to the organization, is out of the user's control and can be used or disseminated by the organization in ways that the user did not anticipate or consent to. This web-based authentication paradigm has led to a number of very serious problems:

1. Users are required to manage a collection of online identities;

2. Organizations are in complete control of their users' data;

3. Users are easily able to transfer their identities to other individuals.

Digital identity management systems have been designed and implemented to address some of these problems. A number of online digital identity management systems are currently in use today, including OpenId [90], Shibboleth [77], and CardSpace [29].

In this chapter we describe our contribution of a policy-based digital identity management system, which is built upon the deployed systems that we will outline in this chapter. We start in the next section, with the basic model for web-based authentication. Following the basic model, three adversaries will be introduced, and each of OpenId, Shibboleth, and CardSpace will be evaluated. In section 3.4, our new digital identity system will be introduced and evaluated.

3.1 Basic Model

Web-based applications are built upon a client-server model. A webmail provider, such as Gmail, or Hotmail, acts as the server in the model, while a user accessing their E-Mail account acts as the client. In this model, users supply their credentials, in the form of a username and password, to the server and the server validates them. If the credentials are positively validated, the user is granted access to their E-Mail account. This same procedure is performed each time a user requires access to their web-based service. In this model, each account is referred to as an identity, whereas the problem of managing multiple identities is referred as the the "multiple identity problem".

To solve the "multiple identity problem", a single-sign-on (SSO) service has been proposed [77, 90]. SSO allows users to authenticate to one service, and access many applications. In addition to allowing users to authenticate to one service, specific implementations of SSO also allow users to store and manage their digital identities from a centralized location. SSO services that offer this service are referred to as identity providers (IdP). IdPs allow users

to store commonly accessed information about themselves such as their date of birth, their country, etc., online in a centralized location so they do not have to manage the information themselves. Identity creation at an IdP works as follows:

1. The user either chooses an identifier or is supplied one by the IdP.
2. The user chooses a password.
3. The IdP creates a new identity with the user's identifier and password.
4. The user adds attributes to their newly created identity (e.g., name, address, birth date, etc.) which are stored by the IdP.

After an identity is created, the user can use it to sign in to their IdP and access any web-based application that supports SSO. In order to access a web-based service under this model, the user must first authenticate. A high level description of the basic authentication protocol is shown in **Figure 3.1**, (see [77, 90] for more details).

The steps of the authentication protocol are as follows:

1. The user visits the login screen of their web application just like they usually would (Step 1).
2. The user is redirected to their SSO provider either by choosing their SSO provider from a "where are you from" (WAYF) page, or by notifying the web application that they have an account at an SSO provider (Step 2).
3. The user is prompted to log into their SSO provider (Step 3).

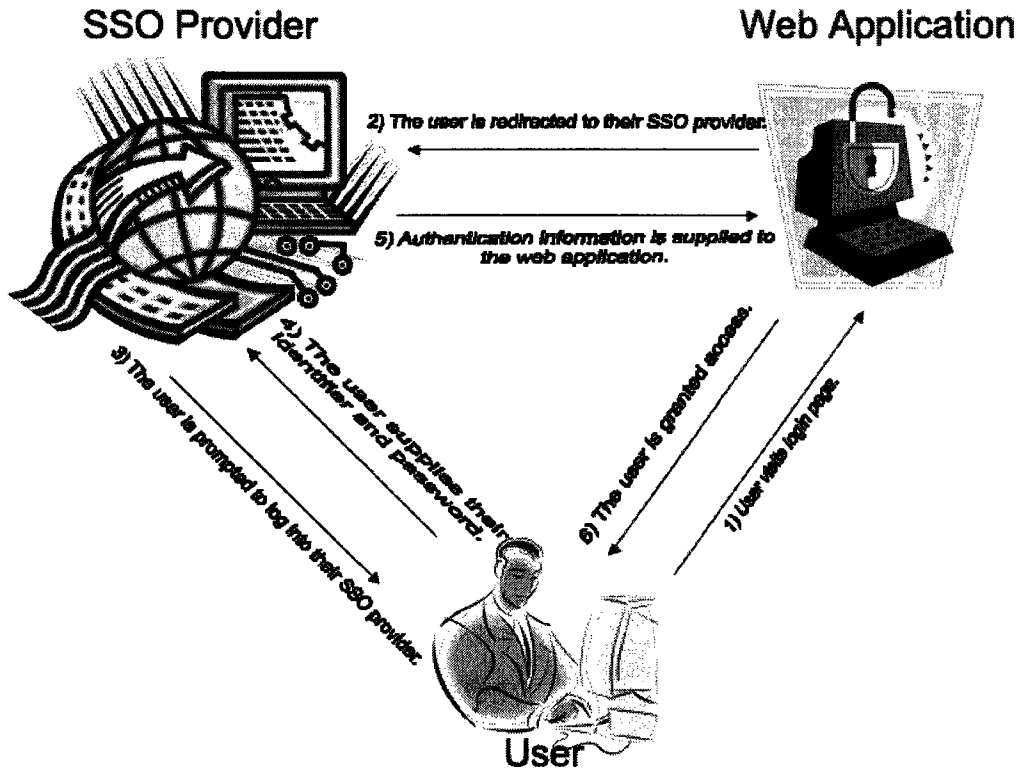

Figure 3.1: Basic SSO authentication protocol.

4. The user enters their unique identifier and password (Step 4).

5. Once the user has been authenticated by their SSO provider the SSO provider supplies the web application with a confirmation that the user has been properly authenticated (Step 5).

6. Once the web application receives the confirmation from the SSO provider the user is granted access to the web application's services (Step 6).

Using an SSO model obviously adds an additional party to the basic client-server model. However, the IdP in the SSO model alleviates the burden of

managing multiple identities from the user. By using an SSO service the user is only required to remember one identifier/password combination.

It should be noted that in most schemes, the web-based application is referred to as a relying party (RP), because it relies on the information supplied to it by an IdP. We will continue with this naming convention as it is consistent with the literature. We will also refer to the SSO model, and not the client-server model, as the basic model.

3.2 Adversaries

Using an IdP allows a user to access the services of many RPs from one centralized location. Implicit in this statement is the fact that a user must be registered with an RP in order to receive a service from it. Typically, during registration, users are required to supply personal information, which is stored at their IdP, to the RP. Depending on the type of service they are offering, RPs require users to submit varying levels of personal information about themselves. As an example, someone opening a Gmail account is required to tell Google their name, their country of residence, and their secondary E-Mail account, as well as answer a security question; as a second example, when registering for an ePass on the Canada Revenue Agency's website, users are required to supply their social insurance number, their date of birth, and their postal code, as well as information about their previous year's tax return. In this section we outline the capabilities of our three adversaries: a malicious RP, a malicious IdP and a group of malicious users.

3.2.1 Malicious RPs

Organizations typically require personal information to uniquely identify their users. In isolation, personally identifiable information revealed to a single organization may not be privacy invasive; however, when the same identifier is revealed to many different organizations a user's privacy loss can become an issue. By revealing the same identifier, or identifiers, to many different organizations, users allow organizations to efficiently track their online transactions. RPs who collude with one another, form partnerships, or merge can compromise a user's privacy under this model. This problem has been referred to as the linkable transactions problem by Chaum [31]. Once a user has divulged their information to multiple organizations, it is trivial for them to work together and build a dossier of the individual's transactions. In [31], Chaum offered a solution to the linkable transactions problem: instead of using a personal identifier when interacting with disparate organizations, a different pseudonym should be used with each organization.

In our basic model, we assume RPs are able to do the following:

1. Store and access all of their users' transaction history as well as any personal information that the user has supplied to the RP.
2. Collude with one another and correlate their users' data.

3.2.2 Malicious IdPs

As a motivating example for our next adversary, let us assume that a user, Alice, is registered with a malicious IdP, IdP_m, and two RPs, RP_1 and RP_2. Since IdP_m is Alice's IdP, it is involved in Alice's transactions with both RP_1

and RP_2 and is knowledgeable about Alice's relationship with both of them. This problem is unavoidable in our basic model, since SSO requires Alice to authenticate to a single location in order to gain access to many RPs. Malicious IdPs in our model, however, are able to perform a much more powerful attack against Alice's anonymity. The authentication mechanism employed at the IdP is extremely important. Let us assume that the authentication method used by IdP_m simply requires Alice to supply a username and password to her IdP. This authentication mechanism in no way requires the user to physically perform the login before being granted access to her web-based service. An IdP which implements this authentication scheme is able to easily access a user's RP by simply acting as the user in the transaction and supplying a fake confirmation that the "user" has been authenticated. With this authentication scheme, IdP_m is trusted by the RP and assumes the IdP is acting on the Alice's behalf. However, under this assumption it is trivial for IdP_m to masquerade as Alice and gain access to her web applications.

Additionally, in some more advanced authentication schemes the basic authentication steps discussed in section 3.1 are extended to allow RPs to grant access to users based on their attributes. After step six, the IdP is able to exchange a user's attributes with the user's RP (in conjunction with a security policy). Storing their attributes at their IdP alleviates users from storing and managing all of this information themselves. However, if stored insecurely, it not only allows a malicious IdP to masquerade as a user but it also allows a malicious IdP to correlate a user's personal information across their multiple identities.

In our model we expect an adversarial IdP to:

1. Attempt to masquerade as individual users and gain access to their

web-based services.

2. Correlate all the attributes for a given user from any of their identities.

3.2.3 Malicious Users

Continuing with our example, let us assume Alice's friend, Trudy, has been having some medical problems and requires some very expensive drugs. However, Trudy has fallen on hard times and is uninsured at the moment. Trudy explains her situation to Alice, and Alice offers to cover Trudy's medical expenses using her insurance policy. Alice describes to Trudy how to log into her insurer's website, and supplies Trudy with the username and password she requires in order to authenticate to Alice's IdP. Trudy gratefully accepts Alice's offer, authenticates to Alice's IdP to gain access to the insurance RP and files a claim for her expensive drugs. This problem has been discussed at length before and is referred to as the "transferability", or "lending", problem [19, 22]. This problem can be detrimental to any RP, especially to RPs providing a subscription based service. Subscriptions can be easily shared by a group of malicious users, by paying for one subscription and disseminating the associated login credentials to its group members, thus receiving the RP's service for a fraction of the cost. In our model we expect an adversarial user to share their login credentials with a group of other users.

3.3 Existing Digital Identity Systems

In this section we evaluate three well used digital identity management systems: OpenId, Shibboleth, and CardSpace. We analyze how effectively they

can prevent the basic model's adversaries from infringing on the privacy of their users.

3.3.1 OpenId

OpenId provides users with the ability to manage their identities from a centralized service, referred to as the OpenId Provider (OP), what has been referred to in this Thesis as an IdP. Users are able to create an account that can be used to authenticate to any RP that supports the OpenId protocol. RPs are able to rely on the OP to authenticate their users [98], as well as to provide an interface to fetch and store user attributes [54].

In the OpenId model, users identify themselves to an RP via their assigned identifier. A single identifier can be used when authenticating to multiple RPs, allowing users to operate under a single pseudonym. Since the pseudonym is potentially shown to many different RPs, a group of malicious RPs is able to correlate transactions from the same user. By allowing users to create multiple identities, OpenId does allow for a solution to this problem. However, the burden of managing these identifiers is left to the individual. Furthermore, user attributes, stored at their OP, are linked to their identifier. An individual showing an identifier to an RP allows the RP to access the attributes stored under that identifier. This allows a malicious RP to perform a number of attacks against the user's anonymity. Primarily, an RP can harvest any identity information stored by the user at their OP by requesting the user's attributes associated with their identifier. This allows attributes stored by a different RP, but under the same identifier, to be harvested by a malicious RP. Furthermore, colluding RPs can easily come together to link

the transactions and data of a user under the same identifier, thus completely de-anonymizing the user under our basic model. Access control on attributes is possible in OpenId; however, it is left to the OP to implement it.

OpenId offers many different mechanisms for authentication at the OP. Traditional username and password authentication, or biometric authentication can be used to authenticate users; the implementation details are left to the OP [98]. The authentication protocol used by the OPs, however, is in no way linked to the individual. Once a user has authenticated to their OP, the OP simply provides a signed assertion to the RP signifying the user has been authenticated. There is no mechanism to prevent an OP from fabricating the assertion and thus no mechanism for preventing an OP from masquerading as one of its users. Furthermore, since all identity attributes are stored at the OP, the OP is able to harvest and correlate all their users' data across multiple identities. An adversarial OP therefore, can perform both a masquerade attack against its users as well as correlate their identity information in the OpenId model.

It is easy to see that an OP who performs authentication strictly based on username and password allows a malicious user to share their OpenId credential. Biometric authentication, however, can help alleviate the problem of transferability; instead of authenticating a user based on their username and password, they can be authenticated via one of their biometric identifiers, such as their face or voice. Biometrics can help solve the problem of transferability; however, the chosen authentication algorithm must be properly evaluated to ensure non-transferability, as a naïve biometric authentication protocol is susceptible to a replay attack. Although it is not clear if the OpenId protocol itself is susceptible to a malicious user attempting to trans-

fer their identity, it is clear that no OpenId specification has been proposed to adequately protect against this type of adversary.

3.3.2 Shibboleth

Shibboleth provides much of the same functionality that OpenId does, but with more focus on security. Shibboleth also supports SSO, but as an extension to their protocol. The protocol is designed to be used in a federated identity setting where multiple IdPs are used to manage multiple identities (it should be noted that OpenId can also be used in this setting). Federated identity, however, requires a user to not only manage multiple identities, but also manage where those identities are located. Only the SSO, or basic model, is considered in this chapter.

In comparison to OpenId, Shibboleth takes a different approach to identifying which IdP a user belongs to. Instead of a user supplying their identifier to an RP, the user is presented with a *where are you from* (WAYF) page when visiting their RP (it should be noted that RPs are referred to as *service providers* and users are referred to as *principals* in Shibboleth). The user is given the opportunity to choose their IdP from a list, at which point they are redirected to their IdP's authentication portal. Once the user has authenticated, their IdP redirects them back to the RP with a Security Assertion Markup Language (SAML) assertion(s) [86], signifying the user has been properly authenticated. Optionally, the IdP and RP can further exchange attributes, in the form of SAML tokens. Details of the protocol can be found in [28].

Shibboleth does an excellent job of protecting the user's anonymity with re-

spect to a malicious RP by allowing users to be identified by a pseudonym. Instead of identifying a user by a single pseudonym, Shibboleth allows different pseudonyms to be used with different RPs, thus protecting the user against colluding RPs. Shibboleth also enforces access control on their users' attributes. By enforcing access control, users are able to decide which RPs can access their attributes. A malicious RP is handcuffed by the access control mechanisms of Shibboleth, and users are protected from RPs harvesting their attributes without their consent. By providing access control and pseudonymous based transactions, Shibboleth adequately protects against an adversarial RP.

Unfortunately, Shibboleth stores all user attributes under the control of the IdP, allowing a malicious IdP to harvest a user's attributes. Furthermore, Shibboleth's authentication mechanism is in no way tied to the actual identity of the user and thus does not prevent a malicious IdP from masquerading as one of its users. Shibboleth IdPs, much like OpenId OPs, are able to harvest their users' attributes as well as masquerade as their users and thus must be completely trusted by both their users and by the RPs in the system. Additionally, in the same way that OpenId is susceptible to users sharing their identity, Shibboleth is as well.

3.3.3 CardSpace

The digital identity system proposed in [29], Cardspace, takes a different approach to managing multiple identities. As opposed to storing and managing identities from a central location, Cardspace distributes the storage of its identities, while centralizing the management of them. In CardSpace,

user identities are granted and stored at each of the separate IdPs but are managed at a high level by the individual users. During authentication, a policy is downloaded from an RP. Based on the policy, the user locally selects which of their identities they wish to use during authentication. The associated IdP of the selected identity is contacted and the IdP, based on the policy of the RP, creates a fresh security token. The security token is then passed through the user to the RP in order to authenticate the user.

CardSpace centralizes the management of multiple identities, which prevents users from having to sign in to an IdP in order to manage their identities. Centralizing the management of identities also allows users to manage which of their identities they have used with each RP. Furthermore, separating each of the users' identities across multiple IdPs prevents a malicious IdP from harvesting their data across identities. Additionally, CardSpace allows users to create their own self-issuing IdP locally, in order to create security tokens for commonly shown attributes such as their name, address, phone number, etc. Individual public/private keys are generated for each RP as a means to prevent linkage.

Although CardSpace does boast a number of attractive properties, it suffers from a significant drawback. Since users manage their digital identities centrally, meta-data about their identities is tied to a specific machine or device. Migrating a user's identities from one device to another requires exporting the meta-data from one device and importing it on another. This procedure can become cumbersome, and could potentially make identities difficult to keep up to date. Importing identities might even be impossible if a user is using a machine that they cannot install the required CardSpace software on.

CardSpace does protect against a malicious RP by using pseudonymous transactions. Additionally, CardSpace protects against malicious IdPs by physically separating them and removing their SSO service. However, CardSpace provides no mechanism to tie the digital identities of a user to the actual user. Users can easily transfer their identities to their friends for their own personal use, by performing the same import/export operation required when migrating their identities.

While OpenId provides for a centrally managed, easy to use, digital identity system, it fails to provide any protection against malicious RPs, IdPs or users. Shibboleth on the other hand takes greater care with respect to their users' anonymity; however, it provides no protection against a malicious IdP, nor does it prevent co-operating users from sharing their identity. Cardspace does provide for protection against malicious IdPs; however, it provides no protection against users sharing their identity. A more advanced and privacy-preserving digital identity system is required to address these adversaries. In the next section we propose a high level architecture aimed at minimizing the effectiveness of all three adversaries.

3.4 Privacy-Preserving Digital Identity System

In this section, a privacy-preserving digital identity system, which builds on some of the ideas implemented by OpenId, Shibboleth, and CardSpace, is introduced. The system allows for users to sign in to one location and select their identity, while reducing the effectiveness of our three adversaries. Since

this scheme builds on top of the basic model it will be referred to as the extended model.

3.4.1 Key Players

The key players in the scheme are essentially the same, however some of their roles have slightly changed. Additionally, one party has been added in the extended model, the identity manager (IdM). The players in the extended model are as follows:

Identity Provider (IdP) : Identity providers are solely responsible for granting digital identities to users in the form of *Digital Credentials* (see [83,84]) and are no longer responsible for managing multiple identities as they are in the OpenID and Shibboleth schemes. Each digital identity consists of a public and private portion, where the private portion contains the user's attributes and the public portion contains a blinded and signed version of the attributes.

Users : Users are granted digital identities from identity providers. Users are able to store their digital identities at their identity manager according to their selected trust policy. Three policies exist:

Complete Trust : Both the public and private portion of the user's identity is stored at their identity manager.

Partial Trust : The public portion of the user's identity is stored at their identity manager. From the private portion a blinded portion is created (see [96]); the blinded version is stored at their identity manager, while the private portion is stored by the user.

No Trust : The public portion of the user's identity is stored at the identity manager, while the private portion is stored and managed by the user.

When authenticating to a relying party, the user is empowered to select their identity manager via a WAYF page, as well as which identity and attributes they wish to disclose.

Relying Party (RP) : In the extended model, the relying party changes slightly. Instead of redirecting their users to a specific identity provider during authentication, the relying party now redirects their users to their identity manager (after presenting them with a WAYF page). Relying parties grant access to users by evaluating the user's ability to prove a linear Boolean formula defined over their attributes; users are positively authenticated if they are able to satisfy the formula.

Identity Manager (IdM) : The identity manager is responsible for managing the identities of its users and providing a proxy service as well as an SSO service to its users through biometric authentication. Since users are able to select the complete trust model, the identity manager is a trusted party who has a vested interested in preserving the privacy of its users. There are many identity managers in the model; however, each user only has one identity manager. During authentication, the IdM is responsible for creating a fresh commitment to a user's biometric signal as a means to prevent transferability [2]. The details of how this is done is the subject of Chapter 5.

In the extended model, the IdM allows users to manage multiple identities from one location. Additionally, the model allows relying parties to authen-

ticate users based on the attributes encoded in the user's digital credentials by formulating a linear Boolean formula over their attributes and biometric identifier. Furthermore, IdMs authenticate their users through biometric authentication. By using Adams' non-transferability scheme [2], IdMs and IdPs are able to link the physical identity of a user to their digital identity through biometrics.

3.4.2 Trust Model

The extended model is designed for there to be a small number of IdMs, n_{IdM}, many more IdPs, n_{IdP}, and a large number of users, n_U, thus $n_{IdM} \ll n_{IdP} \ll n_U$. It is also assumed that IdMs are run by large, reputable companies and are trusted by both their users and the relying parties. A more thorough discussion of all the trust relationships is explained in the following chapter. In our new model, we introduce the IdM to physically separate the creation of identities from the management of them. Some metrics which support these assumptions can be extracted from the OpenID project:

> "OpenID is growing quickly and becoming more popular as large organizations like AOL, Facebook, France Telecom, Google, LiveDoor, Microsoft, Mixi, MySpace, Novell, Sun, Telecom Italia, Yahoo!, etc., begin to accept and/or provide OpenIDs. Today, it is estimated that there are over **one billion OpenID enabled user accounts** with **over 40,000 websites supporting OpenID** for sign in [48]"

It is important to note that it is not a requirement for the IdPs to necessarily trust the IdMs. *Digital Credentials* rely on public key encryption and use dig-

ital signatures to sign all credentials, therefore any digital credential issued by an IdP is independently verifiable by any RP. The RPs, however, must trust the IdMs to properly create their users' biometric commitments during authentication. Additionally, by design, a user in the system has many identities, on average μ_{ID} identities, where $\mu_{ID} \geq 1$. For convenience, it is also assumed that users only have one IdM. By design then $n_{IdM} \ll (n_U * \mu_{ID})$, and thus a profiling attack against an IdM that has a trivially low number of users is not expected to be viable. It is conceivable that a user could be de-anonymized by a group of malicious RPs who know that a user's IdM only has one user; however, this event is expected to be mitigated by only allowing an IdM to operate with a sufficiently high number of users. Furthermore, all communication between entities in this scheme is expected to occur over encrypted channels (using SSL or TLS). The use of these channels ensures that the user is communicating with the correct entities (e.g. the RPs and IdM) over secure channels.

All identities issued to a user are stored at their IdM with their own trust policy. The system supports three types of policies, each with varying levels of trust. The proposed policies are:

Complete Trust : Using this policy, an identity, I_P, supplied to user, U, by IdP, P, would have both its public and private portions, PK_P and SK_P respectively, stored at the user's IdM, M. This policy allows for ubiquitous use of the user's identities, however, it gives the IdM complete control.

Advantages : The major advantage of this policy is that U is not required to store any of its identity material locally (except maybe a list of its identities). This allows U to easily access its identities

from any Internet enabled device. Another advantage of this policy is that all the computations required during authentication as well as the storage of the digital identity material are offloaded to M.

Disadvantages : This policy allows M to masquerade as U. Since M has both PK_P and SK_P, M could conceivably masquerade as U in the authentication protocol. This would require M to act in a malicious fashion and store U's biometric reading; however, this action is neither difficult nor expensive for M to perform. If an identity is stored using this policy, M also learns all the attributes of the identity I_P. This setting allows an IdM to behave exactly like a malicious IdP in the OpenId or Shibboleth systems. A malicious IdM is able to masquerade as U, as well as correlate all of their attributes.

Partial Trust : Using this policy, the public portion, PK_P, along with a blinded version of the private portion, SK'_P are stored at M. U is responsible for storing the private information, SK_P. This policy allows users to offload the intense computations required during authentication and removes control from the IdM to both masquerade as the user as well as see the user's attributes.

Advantages : This policy does not allow M to masquerade as U in the authentication protocol, nor does it allow M to learn the attribute values in U's digital identity. The other advantage is that quite a few of the intense computations associated with digital credentials are offloaded to M; this is especially important when U is using a low-power mobile device to authenticate.

Disadvantages : This policy requires U to store SK_P locally. This makes migrating U's identity more cumbersome, as identity material must be transferred to each device U wishes to use during the authentication protocol.

No Trust : Using this policy, no private identity information is stored at M. U stores the entire private portion, SK_P, of its identity locally; M only stores the public portion, PK_P.

Advantages : This policy does not allow M to masquerade as U in the authentication protocol. In this setting M is only involved in generating a fresh biometric commitment during the authentication protocol.

Disadvantages : This policy requires U to shoulder all the intensive computations required during the authentication protocol. It also requires U to manage and store SK_P, prohibiting U from easily using its digital identities ubiquitously.

The policy selected by the user will highly depend on the type of digital identity being issued. The protocols should be able to handle identities that are created using any of the policies. The Liberty Alliance Foundation (LAF) has a specification, the Liberty Identity Assurance Framework [40], which specifies the trust level associated with the credentials granted by IdPs. The LAF outlines a list of four different levels of assurance associated with varying levels of requirements the IdPs must meet. Identities with a higher level of assurance are offered by IdPs that meet a higher level of trust. This is the motivation for the proposed trust model in this chapter. As an example, a birth certificate offered by a government IdP is presumed to have a high level of assurance as it obviously contains extremely sensitive information about

the individual. Since a birth certificate has a high level of assurance, and contains sensitive data, it is most likely going to be managed by a *No Trust* policy. On the other hand, an identity offered by an IdP, such as Google, that simply contains a user's E-Mail address, is most likely going to have a mid-range assurance level and be managed with a *Complete* or *Partial Trust* policy.

Continuing with this argument, users are able to manage many identities each with their own level of trust. A hybrid approach to managing a user's identities is extremely beneficial to a user. Imagine that this scheme was used to authenticate to a social networking site, an action that is presumed to occur on a fairly regular basis. The attributes the social networking site requires for authentication are assumed to be of low assurance and thus are managed using a *Complete Trust* policy, perhaps only requiring the user to show a credential containing their primary E-Mail address. In this scenario a user is easily able to log into their social networking site from any Internet enabled device (provided a biometric reading can be taken) as all the identity information required is stored at their IdM. Now, imagine a user is renewing their driver's license online, an operation that is presumed to occur infrequently. The attributes required by the Ministry of Transportation (MTO) are contained within the user's birth certificate identity and car insurance identity. These attributes obviously contain very sensitive information about the user and are stored using a *No Trust* policy. In this scenario the user would be required to authenticate to the MTO using their private identity information stored on a secure device, perhaps their home PC, thus protecting the user's sensitive information from their IdM.

By adopting the hybrid approach to managing identities, users can enjoy the advantages of the *Complete Trust*, *Partial Trust* and *No Trust* policies while not being burdened with memorizing any login information such as a username or password. Additionally, when either the *Partial* or *No Trust* policy is used, a malicious IdM is no longer able to masquerade as one of its users, nor is it able to correlate any of its users' sensitive attributes, thus rendering a malicious IdM ineffective.

In this section we introduced the basic and extended model and discussed how it is used by various digital identity management systems on the Internet today. We introduced three different adversaries and showed how all three studied digital identity systems, OpenId, Shibboleth, and CardSpace, fail to prevent all the adversaries from infringing on the privacy of the users in each of the systems. We also introduced a policy based system that allows users to mitigate how much of their privacy is lost in using our new system. In the next chapter we go over the detailed mathematics of how our privacy-preserving digital identity management system implements the three different policies. We also analyze our theoretical system and show how it addresses the privacy concerns presented by the three adversarial groups presented in this section.

Chapter 4

Detailed Digital Identity Management System

In order to realize the privacy-preserving digital identity system in section 3.4, a digital credential scheme must be used. A number of digital credential systems have been proposed over the last decade; however, two have primarily been chosen by the research community as the most practical schemes. Both the scheme of Camenisch [22] and Brands [19] have enjoyed acceptance and focused research. Although both systems achieve a digital credential system, they do so in very different ways.

Camenisch's scheme is built upon the RSA assumption and the DDH assumption which allows organizations to produce digital multi-show credentials for their users. Brands' system also relies on the RSA and DDH assumptions, however, his system is built upon a public key infrastructure that allows users to prove, in zero-knowledge, that the attributes encoded in their certificate satisfy some linear Boolean formula. Where Brands' system excels in allow-

ing users to prove properties about their attributes it fails to provide the multi-show capability that Camenisch's scheme does. However, where Camenisch's scheme allows for credentials to be multi-show, it does not allow properties about individual attributes to be shown. In Camenisch's scheme, each attribute requires a different credential that must be explicitly disclosed during the showing protocol. Although Brands and Camenisch have outlined novel schemes, they both have significant drawbacks. A scheme that allows for properties of attributes to be shown in zero-knowledge, while having the credentials themselves be multi-show, is required.

In [84], Persiano and Visconti proposed a novel scheme based on Brands' *Digital Credentials*. In their proposal, Persiano and Visconti outlined a digital credential scheme that allows credentials to be multi-show, while allowing their attributes to be shown in zero-knowledge. Persiano and Visconti's scheme boasts a number of desirable properties [84]:

Security : It is hard for a coalition of users to mount an attack against the protocol and get access to a service (provided by an RP) without having the requested credentials.

Multi-show privacy : A user, during a transaction, can prove possession of credentials, and at the same time, the RP does not obtain any private user information. This holds even if the user interacts using the same credential certificate several times with the same (or other) RPs.

Usability : A user that possesses a credential certificate should be able to prove general statements (in our case the satisfaction of linear Boolean formulae) over the credentials while preserving multi-show privacy.

Efficiency : The overhead in terms of communication and computation

imposed by the credential system to users and RPs must not heavily affect their performance.

Persiano and Visconti's scheme also boasts another property, non-transferability, a property achieved by encoding highly personal information in the private key of the user's digital credential. By sharing a credential, a user must then share some highly personal piece of information. The mechanism employed by Persiano and Visconti to achieve non-transferability is discouragement, a mechanism which does nothing to prevent users from sharing their credentials with other members of a trusted group. A more rigorous mechanism is required to ensure non-transferability.

In this chapter we apply two existing proposals to our policy-based digital identity management system. In doing so, we extend Persiano and Visconti's scheme with the biometric non-transferability work of Adams [2] and the proxy support work of Shapiro, et al. [96].

4.1 Background

The scheme put forth by Persiano and Visconti includes a number of protocols: *SetUp, Enroll, IssueCred, ProveCred, VerifyCred.* Each protocol requires a subset of the parties in their model to interact in order to achieve a specific goal. Persiano and Visconti's model consists of three players: users, organizations, and service providers. The players are described as follows [84]:

Organizations : Release credential certificates to users.

Users : Receive, from an organization, credential certificates encoding their

credentials that will be used to access services.

Service Providers : Offer services and have access control policies for their services. The access control policies for each resource of each service is represented by a linear Boolean formula, Φ, over the required credentials.

The players in Persiano and Visconti's scheme have analogous players in the privacy-preserving digital identity system in section 3.4. Organizations are actually identity providers, while service providers are the relying parties. Although Persiano and Visconti's model closely resembles the basic model, their protocols must be extended to achieve the privacy-preserving digital identity system described in section 3.4. Before describing the extensions in detail, a basic math background must be first introduced. The definitions presented in section 4.2 are all from [84] unless otherwise indicated.

4.2 Math Background

Definition 4.2.1 *The* integers modulo n, *denoted by* $\mathbb{Z}_n$, *is the set of (equivalence classes of) integers* $\{0, 1, 2, \ldots, n-1\}$. *The* multiplicative group *of* $\mathbb{Z}_n$ *is* $\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n | gcd(a, n) = 1\}$ *[72].*

Definition 4.2.2 *Given a group G of order n (formed by* $\mathbb{Z}_n$*), and two elements g and y of G, the discrete logarithm of y to the base g, if it exists, is the integer* $0 \leq x \leq n-2$ *such that* $y \equiv g^x \pmod{n}$. *The discrete logarithm of y to the base g exists if and only if y belongs to the subgroup generated by g.*

It should be noted that if n is a composite integer of unknown factorization, then calculating the discrete logarithm of y to the base g is considered hard in $\mathbb{Z}_n^*$ [84].

Definition 4.2.3 *Let G be a group of order n and let $y, g_1, \ldots, g_m \neq 1$ be elements of G. A $(G, g_1, \ldots, g_m)$-DL representation (DLREP) of y is a tuple $(x_1, \ldots, x_m)$ such that $0 \leq x_i \leq n-1$ for $i = 1, \ldots, m$ and $y = g^{x_1} \ldots, g^{x_m}$. Moreover, for $i = 1, \ldots, m$, we call x_i the g_i-part of the $(G, g_1, \ldots, g_m)$-DL representation $(x_1, \ldots, x_m)$ of y.*

Definition 4.2.4 *Let e be an element of $\mathbb{Z}_n^*$ co-prime with $\phi(n)$ (Euler's totient function). A $(\mathbb{Z}_n^*, e)$-root of $y \in \mathbb{Z}_n^*$ is an element $x \in \mathbb{Z}_n^*$ such that $x^e \equiv y (mod\ n)$. The RSA assumption states that if the factorization of n is unknown then computing the $e-th$ roots in $\mathbb{Z}_n^*$ is assumed to be infeasible.*

Definition 4.2.5 *Let $e \in \mathbb{Z}_n^*$ be co-prime with $\phi(n)$ and let $y, g_1, \ldots, g_m \neq 1$ be elements of $\mathbb{Z}_n^*$. A $(\mathbb{Z}_n^*, e, g_1, \ldots, g_m)$-RSA representation (RSAREP) of y is a tuple $(x_1, \ldots, x_m, x)$ such that $y \equiv g_1^{x_1} \ldots g_m^{x_m} x^e (mod\ n)$, $0 \leq x_i \leq e$ for $i = 1, \ldots, m$ and $x \in \mathbb{Z}_n^*$.*

Since Persiano and Visctonti's scheme is built upon Brands' scheme it relies heavily on proofs of knowledge (PoKs). Three primary PoKs are used throughout this discussion: *PoK of a discrete logarithm*, *PoK of a DLREP*, and *PoK of a RSAREP*. Definitions for each of the PoKs follow:

PoK of a discrete logarithm [84]: On input (the description of) a cyclic group G of order n, a generator g of G, and an element $y \in G$, the prover

P proves knowledge to the verifier V of x, the discrete logarithm of y to the base g.

PoK of a DLREP [84]: On input the description of a cyclic group G of order n and elements $y, g_1, \ldots, g_m$ of G, the prover P proves knowledge of a $(G, g_1, \ldots, g_m)$-DL representation $(x_1, \ldots, x_m)$ of y.

PoK of a RSAREP [84]: On input $n, e, g_1, \ldots, g_m$ where (n, e) is an RSA public key and $g_1, \ldots, g_m$ are randomly chosen elements of a $\mathbb{Z}_n^*$, the prover proves knowledge of a $(\mathbb{Z}_n^*, e, g_1, \ldots, g_m)$-RSA representation $(x_1, \ldots, x_m, x)$ of $y \in \mathbb{Z}_n^*$.

The proofs of knowledge are shown in Brands' papers [18, 19]. In Brands' PoK of a DLREP, a prover, Alice, proves to Bob that she in fact knows the secret portion, $(x_1, \ldots, x_m, \alpha)$, of the digital credential, $y = g_1^{x_1} \ldots g_m^{x_m} h_0^{\alpha}$, she has been granted by a certificate authority. See **Figure 4.1** for the details on how Alice proves to Bob that her credential has an attribute with a value y_1. In the PoK of a RSAREP, a prover, Alice, proves knowledge of the secret portion, $(x_1, \ldots, x_m, x)$, of $y = g_1^{x_1} \ldots g_m^{x_m} x^e$, by showing the value y_1 to a verifier Bob. The details are very similar to **Figure 4.1**, and are shown in **Figure 4.2**.

These PoKs in conjunction with theorems from [18, 19, 26] form the basis of the mathematical background required to realize Persiano and Visconti's scheme. Detailed descriptions and proofs of the Theorems can be found in [84]. The four basic theorems follow:

Theorem 4.2.1 *Let G be a cyclic group of order n, where n is the product of two safe primes. There exists a zero-knowledge proof of knowledge for*

Alice		Bob
		$m' = nonce \| \ldots$
		$\mathrm{F} = \text{“}x_1 = y_1\text{”}$
	$\xleftarrow{\quad m' \quad}$	
$w_2, \ldots, w_{m+1} \in_R \mathbb{Z}_n$		
$a = g_2^{w_2} \ldots g_m^{w_m} h_0^{w_{m+1}}$		
$c = \mathcal{H}(y, a, F, m')$		
$r_2 = cx_2 + w_2 \pmod{n}$		
$\vdots$		
$r_m = cx_m + w_m \pmod{n}$		
$r_{m+1} = c\alpha + w_{m+1} \pmod{n}$		
	$\xrightarrow{y, sign(y), y_1, (a, r_2, \ldots, r_{m+1})}$	
		$c = \mathcal{H}(y, a, F, m')$
		$y^c a \stackrel{?}{=} g_1^{y_1 c} g_2^{r_2} \ldots g_m^{r_m} h_0^{r_{m+1}}$

Figure 4.1: Brands' DLREP Proof that $x_1 = y_1$ [19]

the polynomial-time relation $((G, h_1, h_2, e, y), (x_1, x_2, x))$, where h_1, h_2 are two elements of G, e is co-prime with $\phi(n)$, $y \in G$, (x_1, x_2) is a (G, h_1, h_2)-DL representation of y and $x \in \mathbb{Z}_n^*$ is the e-th root of $x_1 \in \mathbb{Z}_n^*$ [26].

Theorem 4.2.2 *Let G be a finite cyclic group of order n and Φ be a linear Boolean formula, then there exists a zero-knowledge proof of knowledge for the polynomial time relation $((G, u, g_1, \ldots, g_m, \Phi), (x_1, \ldots, x_m))$ such that $(x_1, \ldots, x_m)$ is a $(G, g_1, \ldots, g_m)$-DL representation of $u \in G$ and $\Phi(x_1, \ldots, x_m) = 1$ [18, 19].*

Alice		Bob
		$m' = nonce \| \ldots$
		$\mathrm{F} = \text{“}x_1 = y_1\text{”}$
	$\xleftarrow{m'}$	
$w_2, \ldots, w_{m+1} \in_R \mathbb{Z}_n$		
$a = g_2^{w_2} \ldots g_m^{w_m} w_{m+1}^e$		
$c = \mathcal{H}(y, a, F, m')$		
$r_2 = cx_2 + w_2 \pmod{n}$		
$\vdots$		
$r_m = cx_m + w_m \pmod{n}$		
$r_{m+1} = x_{m+1}^c w_{m+1}$		
	$\xrightarrow{y, sign(y), y_1, (a, r_2, \ldots, r_{m+1})}$	
		$c = \mathcal{H}(y, a, F, m')$
		$y^c a \stackrel{?}{=} g_1^{y_1 c} g_2^{r_2} \ldots g_m^{r_m} r_{m+1}^e$

Figure 4.2: Brands' RSAREP Proof that $x_1 = y_1$ [18]

Theorem 4.2.3 *Let Φ be a linear Boolean formula, (n, e) be an RSA public key, then there exists a zero-knowledge proof of knowledge for the polynomial time relation $((\mathbb{Z}_n^*, e, u, g_1, \ldots, g_m, \Phi), (x_1, \ldots, x_m, x))$ and it holds that $(x_1, \ldots, x_m, x)$ is a $(\mathbb{Z}_n^*, e, g_1, \ldots, g_m)$-RSA representation of $u \in \mathbb{Z}_n^*$ and $\Phi(x_1, \ldots, x_m) = 1$ [18, 19].*

In [84], a PoK of a discrete logarithm for an element y of the form $y = h_1^{g^{x_1}} h_2^{x_2} \in G$ is given (G, h_1, h_2 are described in Theorem 4.2.1). In the proof the prover, P, proves to the verifier, V, knowledge of the discrete logarithm

x_1 to the base g of the h_1-part of the (G, h_1, h_2)-DL representation of y to the bases h_1, h_2 by performing the following challenge/response proof of knowledge [84].

1. P computes $t = h_1^{g^{r_1}} h_2^{r_2}$ with randomly chosen r_1, r_2 and sends t to V;
2. V sends a bit b to P;
3. If $b = 0$ then P computes $s_1 = r_1$ and $s_2 = r_2$, otherwise P computes $s_1 = r_1 - x_1$ and $s_2 = r_2 - x_2 g^{s_1}$ and sends s_1, s_2 to V;
4. V accepts if $t = h_1^{g^{s_1}} h_2^{s_2}$ and $b = 0$, or $t = y^{g^{s_1}} h_2^{s_2}$ otherwise.

Theorem 4.2.4 *Given a cyclic group G of order n where n is the product of two safe primes, two elements h_1, h_2 of G and an element $g \in \mathbb{Z}_n^*$, the previous PoK is an zero-knowledge proof of knowledge of the discrete logarithm x_1 to the base g of the h_1-part of a (G, h_1, h_2)-DL representation of (g^{x_1}, x_2) of an element $y \in G$ [84].*

4.3 Detailed Description of our Proposed Privacy-Preserving Digital Identity System

In order to achieve the privacy-preserving digital identity system described in section 3.4 four basic protocols are required:

SetUp : This protocol is run by each IdP in the system. During *SetUp* the IdP chooses its public parameters, *Pub*, as well as its private parameters, *Priv*. At the completion of *SetUP*, the IdP makes *Pub* publicly available and is ready to issue credentials.

Issue : The issuing protocol is executed by the user, an IdP, and the user's IdM. During the issuing protocol, the user supplies the IdP with their attributes and, in conjunction with their IdM, they supply a commitment on the user's biometric reading. At the completion of this phase, the user is supplied with private credential material as well as the IdP's public parameters.

Upload : The upload protocol is executed by the user and their IdM. During the upload protocol, the user, based on the policy they chose to store their credential with, uploads the necessary credential material to their IdM along with *Pub*.

Show : The showing protocol is executed by the user, their IdM and an RP. During the showing protocol, the RP supplies the user with a linear Boolean formula, Φ, that the user must satisfy in order to be authenticated. The user, in conjunction with their IdM, chooses the appropriate digital credential to use which satisfies Φ. Once an appropriate digital credential is selected, a sequence of zero-knowledge PoKs are executed to prove that the user holds a digital credential that can satisfy Φ. The PoKs are executed by either the user or the IdM, depending on the policy the selected digital credential is stored with.

4.3.1 The *SetUp* Protocol

Of the four high level protocols required to realize the privacy-preserving digital identity system, both the *SetUp* and *Issue* protocol are the only protocols independent of the policy the user chooses to store their credentials with. The *SetUp* protocol described here is very similar to the *SetUp* protocol

of [84]. Only a few modifications are required in order to add new parameters so that biometric commitments can be calculated. The *SetUp* algorithm takes as a parameter a bitstring of length k, the security parameter.

It should be noted that throughout the description of all the protocols the user will only have two attributes, x_1, x_2, in their digital credential, where x_2 is the user's biometric commitment. We note however, that the number of mathematical operations scales linearly as more attributes are encoded in a user's digital credential.

Protocol 4.3.1 SetUp(1^k)

1. *Randomly pick two k-bit primes* $p_1 = 2q_1 + 1$, $p_2 = 2q_2 + 1$ *such that* $gcd(3, \phi(p_1p_2)) = 1$ *and* q_1, q_2 *are primes. Set* $n = p_1p_2$;

2. *Randomly pick* $e \in \mathbb{Z}_n^*$ *such that* $gcd(e, \phi(n)) = 1$ *and* $gcd(3, e) = 1$;

3. *Compute* $d \in \mathbb{Z}_n^*$ *such that* $3d \equiv 1(mod\phi(n))$;

4. *Select element* $g, c \in \mathbb{Z}_n^*$ *of large order;*

5. *Randomly pick elements* v_1, v_2, v_3, v_4, v_5, *and set* $g_1 \equiv g^{v_1}, g_2 \equiv g^{v_2}, g_3 \equiv g^{v_3}, g_c \equiv g^{v_4}$, *and* $h_c \equiv g^{v_5}$ *(mod n);*

6. *Compute the signature* $s \equiv g^d$ *(mod n);*

7. *Compute a cyclic group G of order n in which computing the discrete logarithm is believed to be infeasible (e.g., G can be computed as a subgroup of* $\mathbb{Z}_q^*$ *for a prime q such that* $n|(q-1)$ *[84]), along with six elements* $h_1, \ldots, h_6 \neq 1$ *of G.*

8. *Set* Pub = $(n,e,g,s,c,g_1,g_2,g_3,g_c,h_c,G,h_1,\ldots,h_6)$ *and*
   Priv = $(q_1,q_2,d,v_1,\ldots,v_5)$, *and output* Pub.

Once the *SetUp* protocol is complete the IdP is ready to issue credentials. During the *Issue* protocol, the user will receive a digital credential on their two attributes $0 \leq x_1, x_2 < e$, where x_2 is a commitment on the user's biometric reading. The biometric commitment will be constructed using Adams' biometric commitment scheme [2].

4.3.2 The *Issue* Protocol

The *Issue* protocol is also independent of the policy the digital credential is stored with. Again, the *Issue* protocol here is very similar to the protocol described in [84]. Only minor modifications were made to calculate the biometric commitment. The *Issue* protocol follows:

Protocol 4.3.2 Issue$(x_1$,Pub$)$

1. *The user supplies their biometric reading to their IdM. Their IdM, through an algorithm described in Chapter 5, extracts a biometric key,* b_I, *from the user's biometric reading. A random value* $r_I \in \mathbb{Z}_n$ *is generated by the IdM and the biometric commitment* $c_I \equiv x_2 \equiv g_c^{b_I} h_c^{r_I} \pmod{n}$ *is calculated;*

2. *The value for* x_2 *is submitted to the user by the IdM;*

3. *The IdP verifies the user's attributes,* x_1, *and* x_2 *out of band through some means governed by the IdP's policy;*

4. *The IdP randomly chooses x_3, such that $0 \leq x_3 < e$, x_3 is co-prime with e and not a multiple of 3, as well as $x \in_R \mathbb{Z}_n^*$;*

5. *The IdP sets $a \equiv g_1^{x_1} g_2^{x_2} g_3^{x_3} x^e \pmod{n}$, $b \equiv c \pmod{n}$, and computes $v \equiv (a+b)^d \pmod{n}$;*

6. *The IdP sends the tuple (x_1, x_2, x_3, x, v) to the user;*

7. *The user verifies that the tuple received is properly constructed by verifying that $v^e \equiv (a+b) \pmod{n}$;*

8. *As pointed out in [84], the user and the IdP must engage in a zero-knowledge PoK in which the IdP proves that n is the product of two safe primes and that g is a large-order element of $\mathbb{Z}_n^*$, and that g_1, g_2, g_3, x are elements of the group generated by g. These proofs must be executed once per issuing protocol; details of these proofs can be found in [84].*

If the *Partial Trust* or *No Trust* policy is used, there is a potential privacy issue with the IdM learning the value of r_I. If the IdM knew r_I it could potentially send the identifying string, x_2 to the issuing IdP, and thus in collusion with the IdP de-anonymize the user. To address this potential problem we rely on assurance. If the assurance levels of the Liberty Alliance Foundation are used, then trust can be used to prevent the IdP and the IdM from colluding. A highly sensitive credential, stored with a *Partial Trust* or *No Trust* policy, is assumed to be issued by an IdP with a relatively high assurance level. Under this assumption the IdP has a vested interest to never reply to a query from an IdM asking them to de-anonymize the user with attribute $x_2 = g_c^{b_I} h_c^{r_I}$.

4.3.3 The *Upload* Protocol

Both the *Upload* and *Show* protocol are dependent on the policy chosen by the user. The *Upload* protocol was not originally part of the credential scheme of [84], it has been added to achieve our digital identity management system. The protocol proceeds as follows. If the *Complete Trust* policy is used the user simply uploads their tuple, (x_1, x_2, x_3, x, v), to their IdM and the *Show* protocol is carried out between the IdM and the IdP with only the user's biometric reading as input from the user. Using the the *No Trust* policy, the user simply keeps their tuple, (x_1, x_2, x_3, x, v), secret. During the *Show* protocol, the IdM simply calculates a fresh commitment, c_F, on a fresh biometric reading, b_F, and supplies c_F, and r_2 to the RP. If the *Partial Trust* policy is used, the user and their IdM interact in a more involved *Upload* protocol, the details follow:

Protocol 4.3.3 Upload$(x_1, x_2, x_3, x, v,$Policy$)$

1. *If* Policy = Complete Trust *then the user uploads* (x_1, x_2, x_3, x, v) *to the IdM.*

2. *If* Policy = Partial Trust *then*

   (a) *The user randomly selects a value* $k \in_R \mathbb{Z}_n$.

   (b) *The user calculates* $x_1' = x_1 + k$, $x_3' = x_3 + k$, $g_k = g_1^k g_2^k g_3^k$, $g_1' = g_1^{-k}$, *and* $g_2' = g_2^{-k}$.

   (c) *The user uploads* $a, b, v, x_1', x_3', x, g_k, g_1'$, *and* g_2' *to the IdM.*

4.3.4 The *Show* Protocol

The *Show* protocol is also dependent on the policy chosen by the user. The *Show* protocol requires five separate PoKs. The parties involved in each of the PoKs vary depending on which policy has been selected for the credential that is being disclosed. We have modified the origianl *Show* protocol of [84] to add the PoK for the biometric commitment.

Protocol 4.3.4 Show$(x_1, x_2, x_3, x, v$,Policy,Φ,Pub$)$

1. *The user/IdM sets* $a \equiv g_1^{x_1} g_2^{x_2} g_3^{x_3} x^e \pmod n$, $b \equiv c \pmod n$, $m \equiv a + b \pmod n$ *(if required);*

2. *The user/IdM picks* $y \in_R \mathbb{Z}_e$ *such that* $0 \leq y < e$;

3. *When Policy* =Partial Trust, *the user uploads* y *to the IdM.*

4. *The user/IdM sets* $\hat{m} \equiv g^y m \pmod n$, $\hat{v} \equiv s^y v \pmod n$, $\hat{a} \equiv g^y a \pmod n$ *and* $\hat{b} \equiv g^y b \pmod n$ *so that* $\hat{m} - \hat{a} - \hat{b} \equiv 0 \pmod n$; *The factors* g^y *and* s^y *are used to blind the true values of* a, b, m, *and* v, *and thus allow the user's credentials to be shown multiple times. By simply generating a new random* y *value, new values for* $\hat{a}, \hat{b}, \hat{m}$ *and* $\hat{v}$ *are calculated and shown to the RP in order to satisfy all the PoKs required;*

5. *The user/IdM computes commitments to* $\hat{m}, \hat{a}, \hat{b}$ *by generating* r_1, r_2, *and* $r_3 \in_R \mathbb{Z}_n$ *and computing the following values:*

   (a) $Com(\hat{m}) = h_1^{\hat{m}} h_2^{r_1}$;

   (b) $Com(\hat{a}) = h_3^{\hat{a}} h_4^{r_2}$;

   (c) $Com(\hat{b}) = h_5^{\hat{b}} h_6^{r_3}$;

6. *The user/IdM sends Com($\hat{m}$), Com($\hat{a}$), Com($\hat{b}$) and $\hat{a}$ to the RP;*

7. *The RP calculates α =Com($\hat{m}$)Com($\hat{a}$)Com($\hat{b}$).*

8. ***The user/IdM and RP engage in five PoKs:***

   (a) *The user sends their biometric reading to their IdM and the IdM constructs a fresh commitment, c_F, from the user's biometric reading, b_F, by selecting a random value, $\bar{r}_I \in_R \mathbb{Z}_n$ [2]. The user/IdM interacts with the RP and proves that the fresh commitment, c_F, **is a different commitment on the same value as** c_I. The zero-knowledge PoK used for this proof is shown in Theorem 2 of [2].*

   (b) *The user/IdM proves knowledge of a $(\mathbb{Z}_n^*, e, g_1, g_2, g_3)$-RSA representation (u_g, u_1, u_2, u_3, u) of $\hat{a}$ such that $\Phi(u_1, u_2) = 1$ using Theorem 4.2.3 and (y, x_1, x_2, x_3, x) as the witness.*

   (c) *The user/IdM proves knowledge of the $(u_1, u_2, u_3, u_4, u_5, u_6)$, a $(G, h_1, \ldots, h_6)$-DL representation of α such that $(u_1 - u_3 - u_5 = 0 \wedge (u_3 = \hat{a}))$ using Theorem 4.2.2 and $(\hat{m}, r_1, \hat{a}, r_2, \hat{b}, r_3)$ as the witness.*

   (d) *The user/IdM proves knowledge of the $(\mathbb{Z}_n^*, 3)$-root of the h_1-part of the (G, h_1, h_2)-DL representation of Com($\hat{m}$) using Theorem 4.2.1 and the fact that $\hat{v}^3 \equiv \hat{m} \pmod{n}$ because $3d \equiv 1 \pmod{\phi(n)}$.*

   (e) *The user/IdM proves knowledge of the discrete logarithm with base g of the h_5-part of the (G, h_5, h_6)-DL representation of Com($\hat{b}$)$^{c^{-1}}$ using Theorem 4.2.4, the fact that $\hat{b} \equiv g^y b \equiv g^y c \pmod{n}$, and $y \pmod{n}$ as the witness.*

It is important to note that the random value y can never be shown to the issuing IdP. The value y blinds the user's values, a, b, m, and v, so as to prevent linkage between showings of the user's credential. If the value of y is obtained by the issuing IdP, and the RP supplies the IdP with values for $\hat{a}, \hat{b}, \hat{m}$, and $\hat{v}$, the IdP can de-anonymize the user. It is therefore of paramount importance that the issuing IdP never sees the value y and the corresponding blinded versions of a, b, m, and v.

4.3.5 Showing Protocol for *No Trust* and *Complete Trust*

The PoK for the showing protocol for credentials stored with a *No Trust* and a *Complete Trust* policy are very similar. The main difference is which party interacts with the RP. When a *No Trust* policy is used, the user interacts directly with the RP, while the IdM interacts with the RP when a *Complete Trust* policy is used. The fresh biometric commitments, however, are always computed by the IdM. Once the biometric commitment is calculated and sent to the RP, the PoK can take place.

In the five PoKs, the player *Prover* will be used to denote either the user, or the IdM, depending on the policy the credential to be shown is stored with. If the credential is stored with a *No Trust* policy *Prover* will be the user, while if the *Complete Trust* policy is used, *Prover* will denote the user's IdM.

If a *No Trust* policy is used, the user performs steps one through five of the *Show* protocol, as well as the PoKs in steps 8.b through 8.e, whereas if a *Complete Trust* policy is used, the user's IdM performs steps one through five as well as the PoKs in steps 8.b through 8.e (see Protocol 4.3.4). The

biometric PoK performed in step 8.a however, is always done in conjunction with both the user and their IdM. The steps of the biometric showing protocol are shown in **Figure 4.3**. In **Figure 4.3** it is assumed that the fresh commitment $c_F = g_c^{b_F} h_c^{\bar{r}_I} \pmod{n}$ has already been calculated by the IdM from a new random value $\bar{r}_I \in_R \mathbb{Z}_n$, and a fresh biometric reading b_F.

Prover		RP
$w_y, w_1, w_2, w_3 \in_R \mathbb{Z}_e$		
$w_e \in_R \mathbb{Z}_n^*$		
$a' = g^{w_y} g_1^{w_1} g_3^{w_3} w_e^{\frac{e}{c_F}}$		
$d = h_c^{w_2}$		
	$\xrightarrow{a', d, c_F}$	
		$z \in_R \mathbb{Z}_n$
	$\xleftarrow{z}$	
$c' = h_c^z$		
$r_y = \frac{c'd}{c_F} y + w_y$		
$r_1 = \frac{c'd}{c_F} x_1 + w_1$		
$r_2 = h_c^{(r_I - \bar{r}_I) + z + w_2}$		
$r_3 = \frac{c'd}{c_F} x_3 + w_3$		
$r_e = x^{c'd} w_e$		
	$\xrightarrow{r_y, r_1, r_2, r_3, r_e}$	
		$c' = h_c^z$
		$\hat{a}^{\frac{c'd}{c_F}} a' \stackrel{?}{=} g^{r_y} g_1^{r_1} g_2^{r_2} g_3^{r_3} r_e^{\frac{e}{c_F}}$

Figure 4.3: *No Trust/Complete Trust* Biometric Protocol

Before any of the PoKs can take place, however, the blinded versions of a, b, m, and v must also be calculated along with the commitments, $Com(\hat{a})$, $Com(\hat{b})$, and $Com(\hat{m})$. Once calculated, the values $\hat{a}, c_F$ as well as $Com(\hat{m})$, $Com(\hat{a})$, $Com(\hat{b})$ must be sent to the RP.

The biometric protocol works by ensuring that c_I and c_F are commitments on the same value, i.e. $b_I = b_F$. In order for both c_I and c_F to be commitments on the same biometric value, b_I be must equal to b_F. To check that this is true the RP must verify that $[\hat{a}g_k']^{\frac{c'd}{c_F}} a' = g^{r_y} g_1^{r_1} g_2^{r_2} g_3^{r_3} r_e^{\frac{e}{c_F}}$. The proof of the equality check performed by the RP is shown in **Proof 4.3.1**.

Proof 4.3.1

$$
\begin{aligned}
L.S. &= \hat{a}^{\frac{c'd}{c_F}} a' \\
&= [g^y g_1^{x_1} g_2^{x_2} g_3^{x_3} x^e]^{\frac{c'd}{c_F}} (g^{w_y} g_1^{w_1} g_3^{w_3} w_e^{\frac{e}{c_F}}) \\
&= g^{\frac{c'd}{c_F} y + w_y} g_1^{\frac{c'd}{c_F} x_1 + w_1} g_2^{\frac{c'd}{c_F} x_2} g_3^{\frac{c'd}{c_F} x_3 + w_3} x^{\frac{c'd}{c_F} e} w_e^{\frac{e}{c_F}} \\
&= g^{\frac{c'd}{c_F} y + w_y} g_1^{\frac{c'd}{c_F} x_1 + w_1} g_2^{\frac{c'd}{c_F} x_2} g_3^{\frac{c'd}{c_F} x_3 + w_3} [x^{c'd} w_e]^{\frac{e}{c_F}} \\
&= g^{r_y} g_1^{r_1} g_2^{\frac{c'd}{c_F} x_2} g_3^{r_3} r_e^{\frac{e}{c_F}} \\
&= R.S. \text{ if } b_I = b_F, \text{ because then} \\
\frac{c'd}{c_F} x_2 &= \frac{h_c^z h_c^{w_2} g_c^{b_I} h_c^{r_I}}{g_c^{b_F} h_c^{\vec{r_I}}} \\
&= g_c^{(b_I - b_F)} h_c^{(r_I - \vec{r_I}) + z + w_2} \\
&= g_c^0 h_c^{(r_I - \vec{r_I}) + z + w_2} \\
&= r_2
\end{aligned}
$$

In PoK 8.b, the *Prover* proves to the RP that the credential the user is holding satisfies some linear Boolean formula Φ (obtained from the RP). Many examples of PoKs can be found in [18, 19]; however, for simplicity we

will only show the PoK required for the function $\Phi(\text{“}x_1 = y_1\text{”}) = 1$, but stress that this PoK can easily be extended to satisfy more complicated functions of Φ. **Figure 4.4** shows the steps involved in satisfying $\Phi(\text{“}x_1 = y_1\text{”}) = 1$.

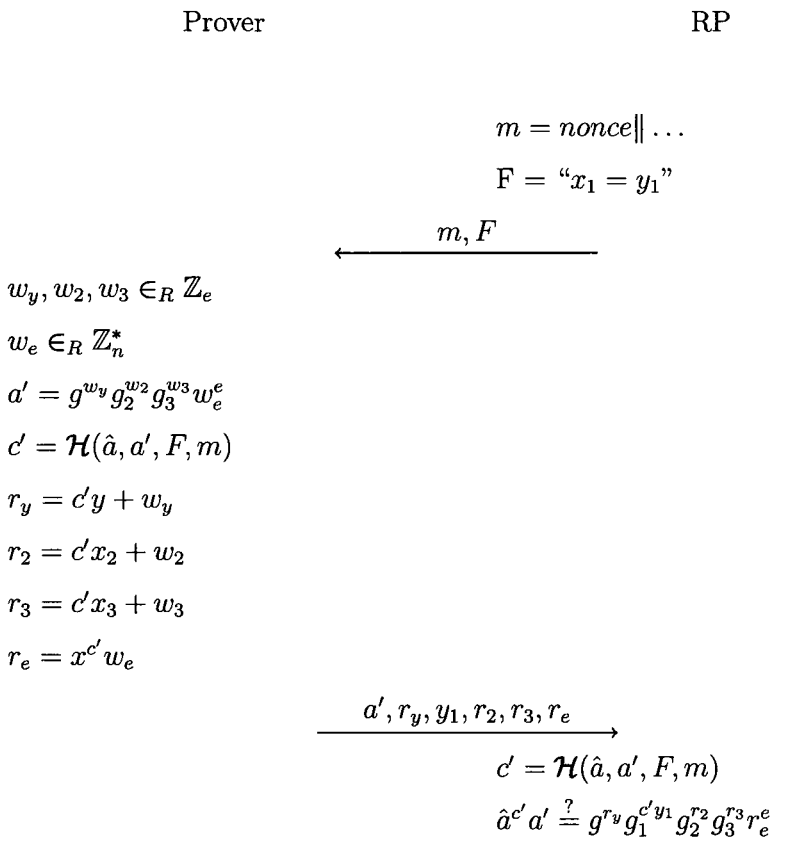

Figure 4.4: *No Trust/Complete Trust* Showing Protocol

In order to determine if the *Prover* can satisfy the function $\Phi(\text{“}x_1 = y_1\text{”}) = 1$, the RP must check to see if the claimed value y_1 is in fact equal to the true value x_1. The proof is shown in **Proof 4.3.2**.

Proof 4.3.2

$$\begin{aligned}
L.S. &= \hat{a}^{c'} a' \\
&= [g^y g_1^{x_1} g_2^{x_2} g_3^{x_3} x^e]^{c'} (g^{w_y} g_2^{w_2} g_3^{w_3} w_e^e) \\
&= g^{c'y+w_y} g_1^{c'x_1} g_2^{c'x_2+w_2} g_3^{c'x_3+w_3} x^{c'e} w_e^e \\
&= g^{c'y+w_y} g_1^{c'x_1} g_2^{c'x_2+w_2} g_3^{c'x_3+w_3} [x^{c'} w_e]^e \\
&= g^{r_y} g_1^{c'x_1} g_2^{r_2} g_3^{r_3} r_e^e \\
&= R.S.\ if \\
c'x_1 &= c'y_1 \\
x_1 &= y_1
\end{aligned}$$

In PoK 8.c, the *Prover* proves to the RP that the commitments, $\hat{a}, \hat{b}$, and $\hat{m}$ have been properly constructed. The PoK in 8.c is a DLREP proof of the linear Boolean function $u_1 - u_3 - u_5 = 0 \wedge u_3 = \hat{a}$ with $(\hat{m}, r_1, \hat{a}, r_2, \hat{b}, r_3)$ as the witness. The protocol is shown in **Figure 4.5**.

In order to determine if the *Prover* properly constructed the commitments, the RP must check to see if the supplied value $\alpha^{c'} a' = h_1^{r'_5 + c'\hat{a}} h_2^{r'_2} h_3^{c'\hat{a}} h_4^{r'_4} h_5^{r'_5} h_6^{r'_6}$, by ultimately verifying that $c'\hat{m} + w_5 = r'_5 + c'\hat{a}$, which is the equivalent of checking that $c'\hat{m} + w_5 - r'_5 - c'\hat{a} = 0$:

$$\begin{aligned}
c'\hat{m} + w_5 - r'_5 - c'\hat{a} &= c'\hat{m} + w_5 - (c'\hat{b} + w_5) - c'\hat{a} \\
&= c'(\hat{a} + \hat{b}) - c'\hat{b} - c'\hat{a} \\
&= 0
\end{aligned}$$

The proof can be seen in **Proof 4.3.3**.

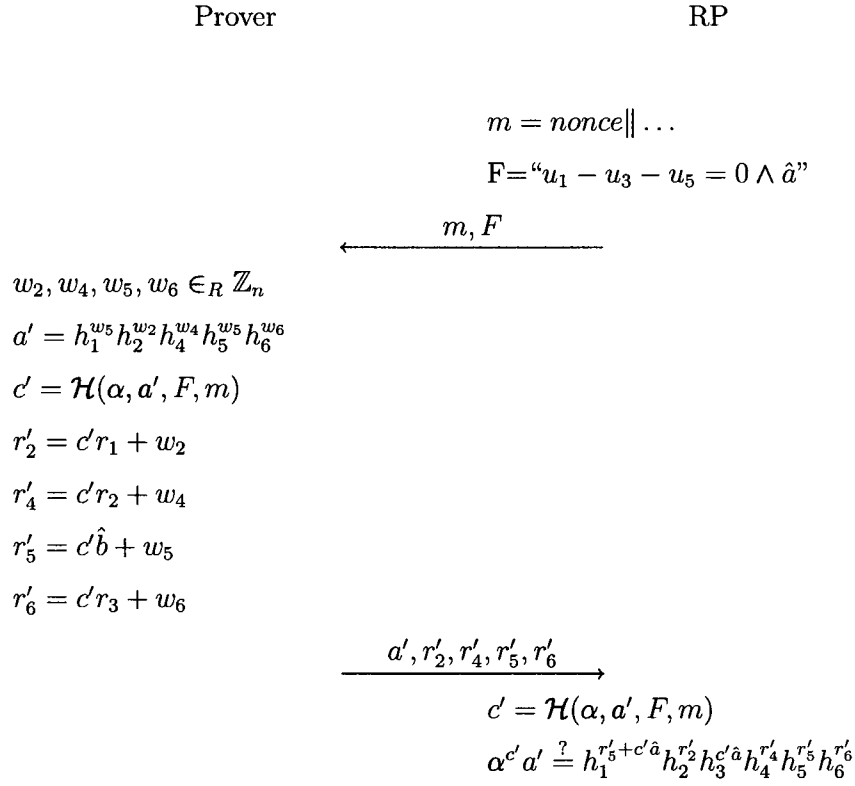

Figure 4.5: *No Trust/Complete Trust* Commitment Protocol

Proof 4.3.3

$$\begin{aligned}
L.S. &= \alpha^{c'} a' \\
&= (h_1^{\hat{m}} h_2^{r_1} h_3^{\hat{a}} h_4^{r_2} h_5^{\hat{b}} h_6^{r_3})^{c'} (h_1^{w_5} h_2^{w_2} h_4^{w_4} h_5^{w_5} h_6^{w_6}) \\
&= h_1^{c'\hat{m}+w_5} h_2^{c'r_1+w_2} h_3^{c'\hat{a}} h_4^{c'r_2+w4} h_5^{c'\hat{b}+w_5} h_6^{c'r_3+w_6} \\
&= h_1^{c'\hat{m}+w_5} h_2^{r'_2} h_3^{c'\hat{a}} h_4^{r'_4} h_5^{r'_5} h_6^{r'_6} \\
&= R.S. \text{ since} \\
c'\hat{m} + w_5 &= c'(\hat{a} + \hat{b}) + w_5 \\
&= c'\hat{a} + (c'\hat{b} + w_5) \\
&= c'\hat{a} + r'_5
\end{aligned}$$

In PoK 8.d, the *Prover* proves to the RP knowledge that the third root of the commitment $\hat{m}$ is the value $\hat{v}$. As a reminder $3d \equiv 1 \pmod{\phi(n)}$ and $\hat{v} \equiv \hat{m}^d \pmod{n} \Rightarrow \hat{v}^3 \equiv \hat{m} \pmod{n}$. The protocol is shown **Figure 4.6**. The proof is simply a construction proof that follows directly from Theorem 4.2.4, with g replaced with $\hat{v}$, x_1 with 3, and x_2 with r_1. The RP requires knowledge of $\hat{v}$ to confirm the construction, but this leaks no additional information as the value $\hat{v}$ is simply a blinded signature on the value $\hat{m}$ and the RP has no knowledge of d, or y. The proof can be seen in **Proof 4.3.4**.

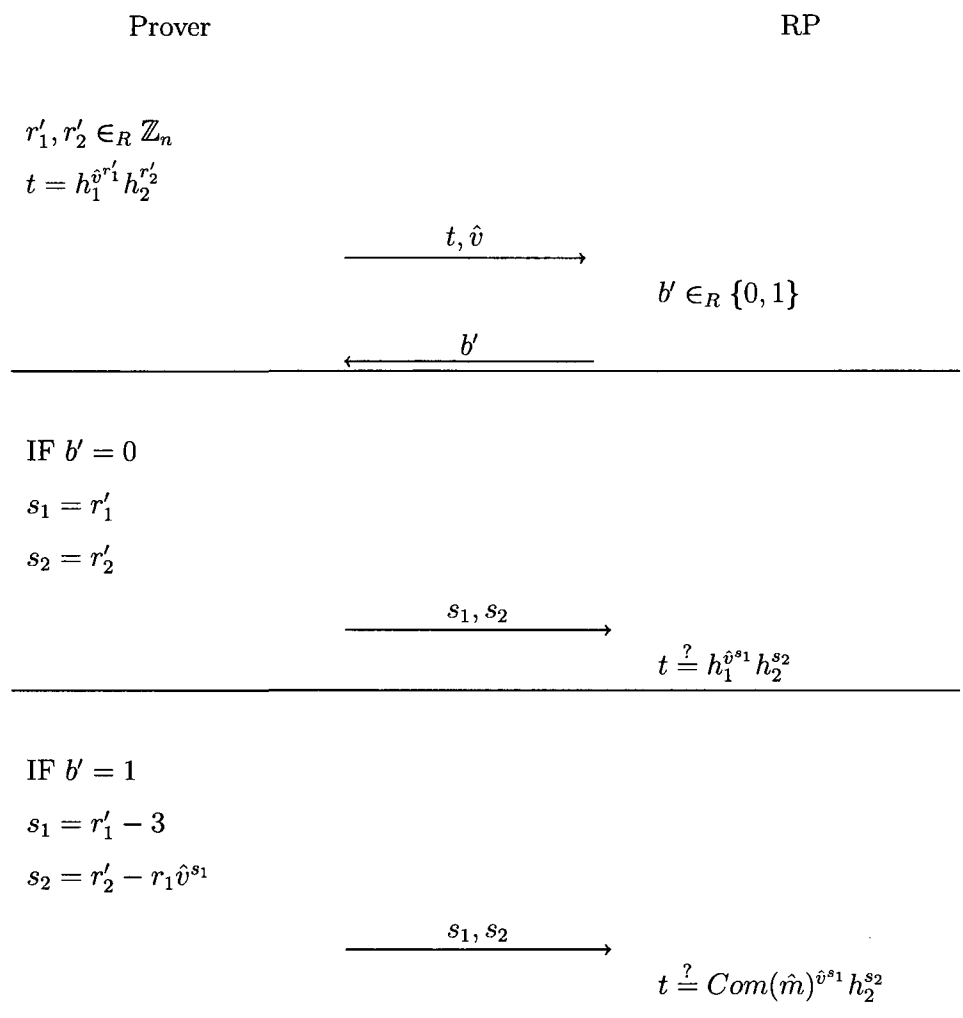

Figure 4.6: *No Trust/Complete Trust* $\hat{m}$ Commitment Protocol

Proof 4.3.4

$$
\begin{aligned}
&\textit{Case } b' = 0 \\
&\qquad R.S. &&= h_1^{\hat{v}^{s_1}} h_2^{s_2} \\
&\qquad &&= h_1^{\hat{v}^{r'_1}} h_2^{r'_2} \\
&\qquad &&= t \\
&\qquad &&= L.S. \\
&\textit{Case } b' = 1 \\
&\qquad R.S. &&= Com(\hat{m})^{\hat{v}^{s_1}} h_2^{s_2} \\
&\qquad &&= (h_1^{\hat{m}} h_2^{r_1})^{\hat{v}^{s_1}} h_2^{r'_2 - r_1 \hat{v}^{r'_1 - 3}} \\
&\qquad &&= h_1^{\hat{m}\hat{v}^{r'_1 - 3}} h_2^{r_1 \hat{v}^{r'_1 - 3}} h_2^{r'_2 - r_1 \hat{v}^{r'_1 - 3}} \\
&\qquad &&= h_1^{\hat{v}^3 \hat{v}^{r'_1 - 3}} h_2^{r'_2} \\
&\qquad &&= h_1^{\hat{v}^{r'_1}} h_2^{r'_2} \\
&\qquad &&= t \\
&\qquad &&= L.S.
\end{aligned}
$$

In PoK 8.e, the *Prover* proves to the RP that he knows the value of y without revealing the value y. This can be achieved by remembering that $b \equiv c \pmod{n} \Rightarrow \hat{b} \equiv g^y b \equiv g^y c \pmod{n}$. The RP uses this fact to check the construction of the value t. The protocol can be seen in **Figure 4.7**.

The proof is again a construction proof which follows direction from Theorem 4.2.4 by replacing g with c, x_1 with y, and x_2 with r_3. The RP requires no additional information to perform this check as c is publicly available from the issuing IdP. The value of y however, is highly sensitive; the RP must never learn the value of y. In the PoK the RP simply learns the fact that the *Prover* has knowledge of the value y without it being revealed. The proof

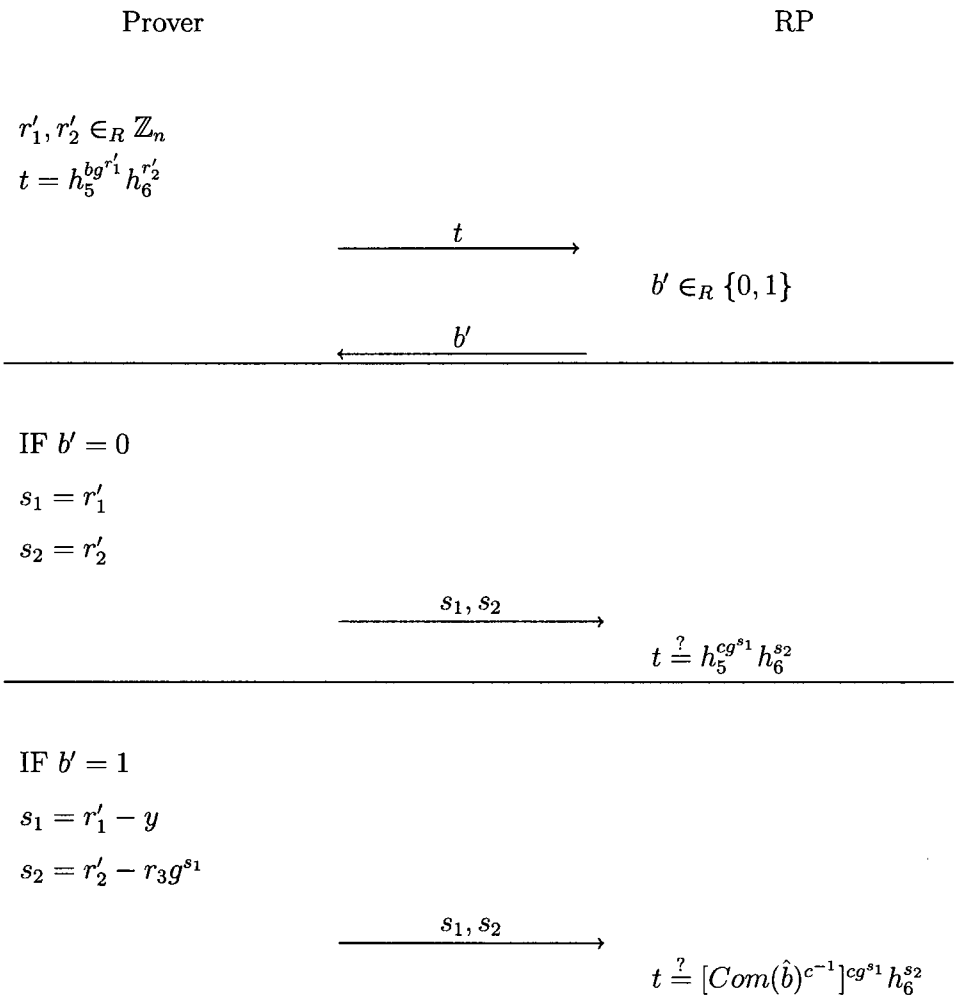

Figure 4.7: *No Trust/Complete Trust* $\hat{b}$ Commitment Protocol

can be seen in **Proof 4.3.5**.

Proof 4.3.5

Case $b' = 0$

$$\begin{aligned} R.S. &= h_5^{cg^{s_1}} h_6^{s_2} \\ &= h_5^{cg^{r'_1}} h_6^{r'_2} \\ &= t \ (since\ b \equiv c \ (mod\ n)) \\ &= L.S. \end{aligned}$$

Case $b' = 1$

$$\begin{aligned} R.S. &= [Com(\hat{b})^{c^{-1}}]^{cg^{s_1}} h_6^{s_2} \\ &= [h_5^{\hat{b}} h_6^{r_3}]^{g^{r'_1 - y}} h_6^{r'_2 - r_3 g^{r'_1 - y}} \\ &= h_5^{g^y b g^{r'_1 - y}} h_6^{r_3 g^{r'_1 - y} + r'_2 - r_3 g^{r'_1 - y}} \\ &= h_5^{bg^{r'_1}} h_6^{r'_2} \\ &= t \\ &= L.S. \end{aligned}$$

By using the *No Trust* policy the user leaks no information to their IdM about their credential. Using this policy, the IdM learns *none* of the private values of the user's credential (x_1, x_3, x, v) other than the biometric commitment, x_2. The downside to the *No Trust* policy however, is the fact that all the PoKs require the user's participation and therefore requires the user to perform intense calculations as well as store their private credential material. However, using the *No Trust* policy prevents the user's IdM from both masquerading as the user and harvesting their attributes.

The opposite of the *No Trust* policy is the *Complete Trust* policy. Storing credentials with the *Complete Trust* policy requires the user to store all their private credential material at their IdM, thus allowing the IdM to both masquerade as the user and harvest all of their attributes. The major upside to the *Complete Trust* policy however, is that it allows the user to access their credentials from any Internet enabled device (without migrating their credential material). By storing all of their private information at the IdM, users are able to access their credentials from anywhere. Although both the *No Trust* and *Complete Trust* policies have advantages and disadvantages, a more robust policy is required to allow users to off-load intense computations to their IdM, while still limiting their IdM's ability to masquerade as the user and harvest their attributes.

4.3.6 Showing Protocol for *Partial Trust*

In between the *No Trust* and *Complete Trust* policies is the *Partial Trust* policy. Using the *Upload* protocol, users are able to store blinded versions of their private credential material at their IdM in an effort to off-load all the intense calculations required during the five PoKs. Using this policy the IdM learns only the value of the initial biometric commitment $x_2 = g_c^{b_I} h_c^{r_I}$, as well as the random value x; however, the IdM learns none of the attribute values of the user's credential (in our example this is only the value x_1).

The showing protocol for the *Partial Trust* policy requires all three parties to be involved: the user, the user's IdM and the RP. The user is responsible for calculating the blinding factor y as well as formulating any responses to challenges that involve the value for y. The PoK for 8.e also requires the user

to participate when knowledge of the value y must be proven. Other than participating in a minor role in the PoKs the user is generally idle during the PoKs.

The fresh biometric commitment c_F, generated from a biometric reading (*BioReading*), along with commitments on a, b, m, and v must be calculated before the PoKs can take place. The commitments Com($\hat{a}$), Com($\hat{b}$), Com($\hat{m}$), and Com($\hat{v}$) are calculated by the IdM on receiving a random value $y \in_R \mathbb{Z}_n$ from the user. The initial setup is shown in **Figure 4.8**.

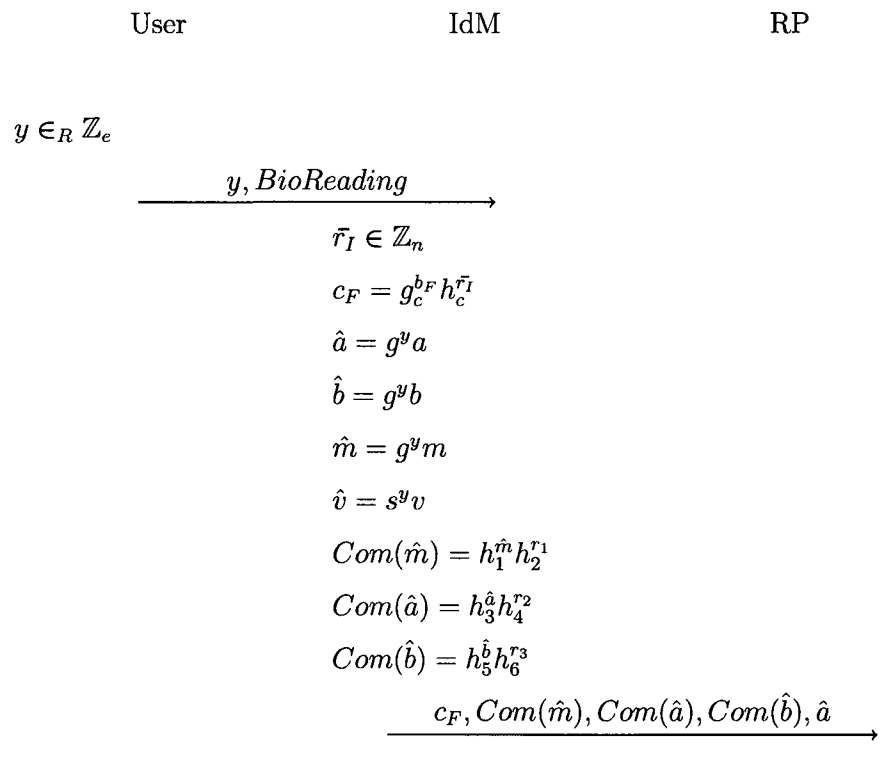

Figure 4.8: *Partial Trust* Initial Setup

Once initialization has taken place the PoKs can occur. The biometric com-

mitment PoK, PoK 8.a, requires very little involvement from the user. The user is only responsible for calculating the response value, r_y, for the value y. This is meant to keep the IdM honest: by having the user generate the value y and answer challenges with the response r_y, the IdM is forced to construct all commitments correctly. The protocol for the biometric commitment is shown in **Figure 4.9**. The proof $[\hat{a}g_k']^{\frac{c'd}{c_F}}a' \stackrel{?}{=} g^{r_y}g_1^{r_1}g_2^{r_2}g_3^{r_3}r_e^{\frac{e}{c_F}}$ is shown in **Proof 4.3.6**. Again, the proof succeeds if c_F and c_I are commitments on the same biometric reading (i.e. $b_I = b_F$).

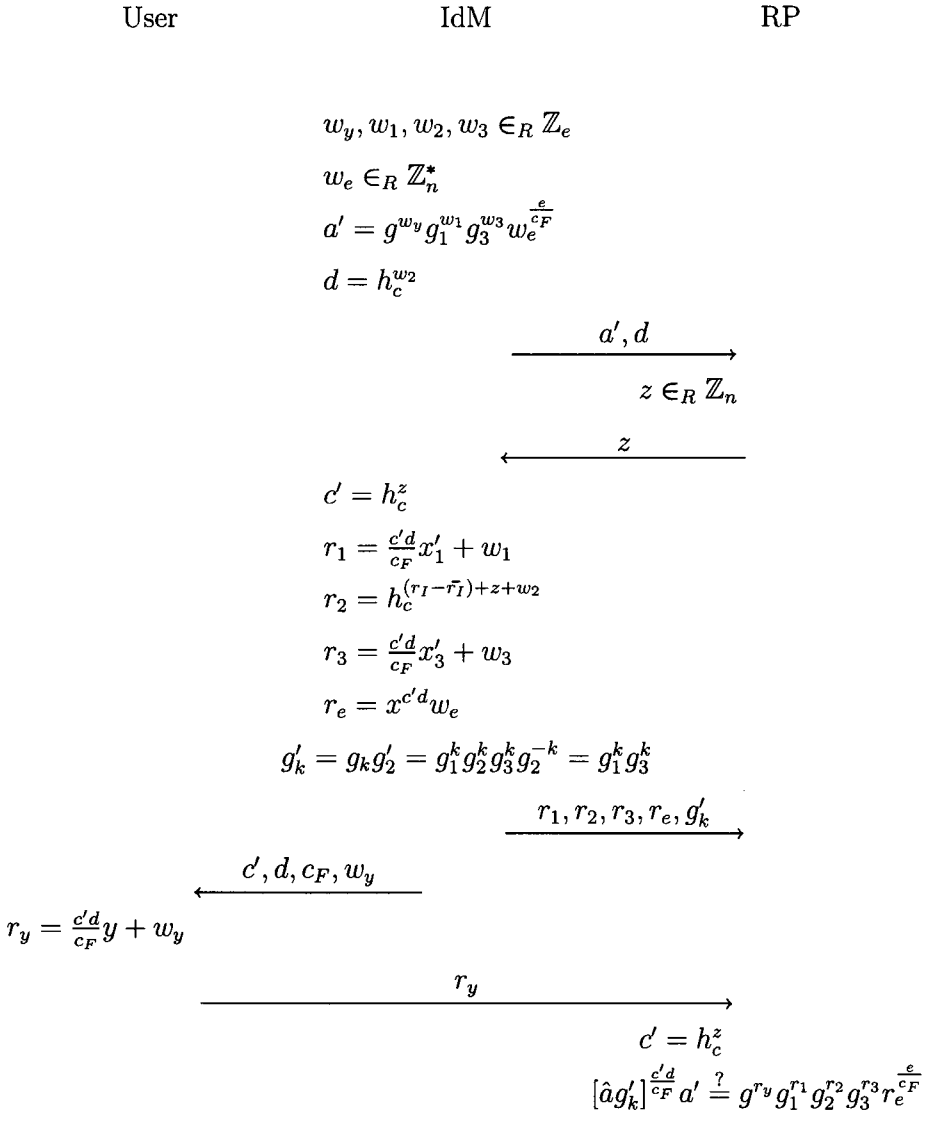

Figure 4.9: *Partial Trust* Biometric Protocol

Proof 4.3.6

$$
\begin{aligned}
L.S. &= [\hat{a}g_k']^{\frac{c'd}{c_F}} a' \\
&= [(g^y g_1^{x_1} g_2^{x_2} g_3^{x_3} x^e)(g_1^k g_3^k)]^{\frac{c'd}{c_F}} (g^{w_y} g_1^{w_1} g_3^{w_3} w_e^{\frac{e}{c_F}}) \\
&= g^{\frac{c'd}{c_F}y+w_y} g_1^{\frac{c'd}{c_F}(x_1+k)+w_1} g_2^{\frac{c'd}{c_F}x_2} g_3^{\frac{c'd}{c_F}(x_3+k)+w_3} x^{\frac{c'd}{c_F}e} w_e^{\frac{e}{c_F}} \\
&= g^{\frac{c'd}{c_F}y+w_y} g_1^{\frac{c'd}{c_F}x_1'+w_1} g_2^{\frac{c'd}{c_F}x_2} g_3^{\frac{c'd}{c_F}x_3'+w_3} [x^{c'd} w_e]^{\frac{e}{c_F}} \\
&= g^{r_y} g_1^{r_1} g_2^{\frac{c'd}{c_F}x_2} g_3^{r_3} r_e^{\frac{e}{c_F}} \\
&= R.S. \text{ if } b_I = b_F, \text{ because then} \\
\frac{c'd}{c_F}x_2 &= \frac{h_c^z h_c^{w_2} g_c^{b_I} h_c^{r_I}}{g_c^{b_F} h_c^{\bar{r}_I}} \\
&= g_c^{(b_I-b_F)} h_c^{(r_I-\bar{r}_I)+z+w_2} \\
&= g_c^0 h_c^{(r_I-\bar{r}_I)+z+w_2} \\
&= r_2
\end{aligned}
$$

The showing PoK, 8.b, requires a little more input from the user. Since the value of k is never revealed to the IdM, the blinded version of x_2, $x_2' = x_2 + k$, can only be calculated by the user. Furthermore, the value x_2' can never be revealed to the IdM. The RP is trusted to not send the value $x_2' = x_2 + k$ to the IdM. If x_2' is sent to the IdM, the IdM can simply calculate the value for $k = x_2' - x_2$, and thus extract the private value for $x_1 = x_1' - k$ and $x_3 = x_3' - k$, thus allowing the IdM to masquerade as the user as well as harvest their information. The PoK for the showing protocol is shown in **Figure 4.10** for the function $\Phi(\text{“}x_1 = y_1\text{”}) = 1$. The proof of the showing protocol is shown in **Proof 4.3.7**. This proof differs from the **Proof 4.3.2** in that the blinded versions of x_2 and x_3 are used instead of the raw values.

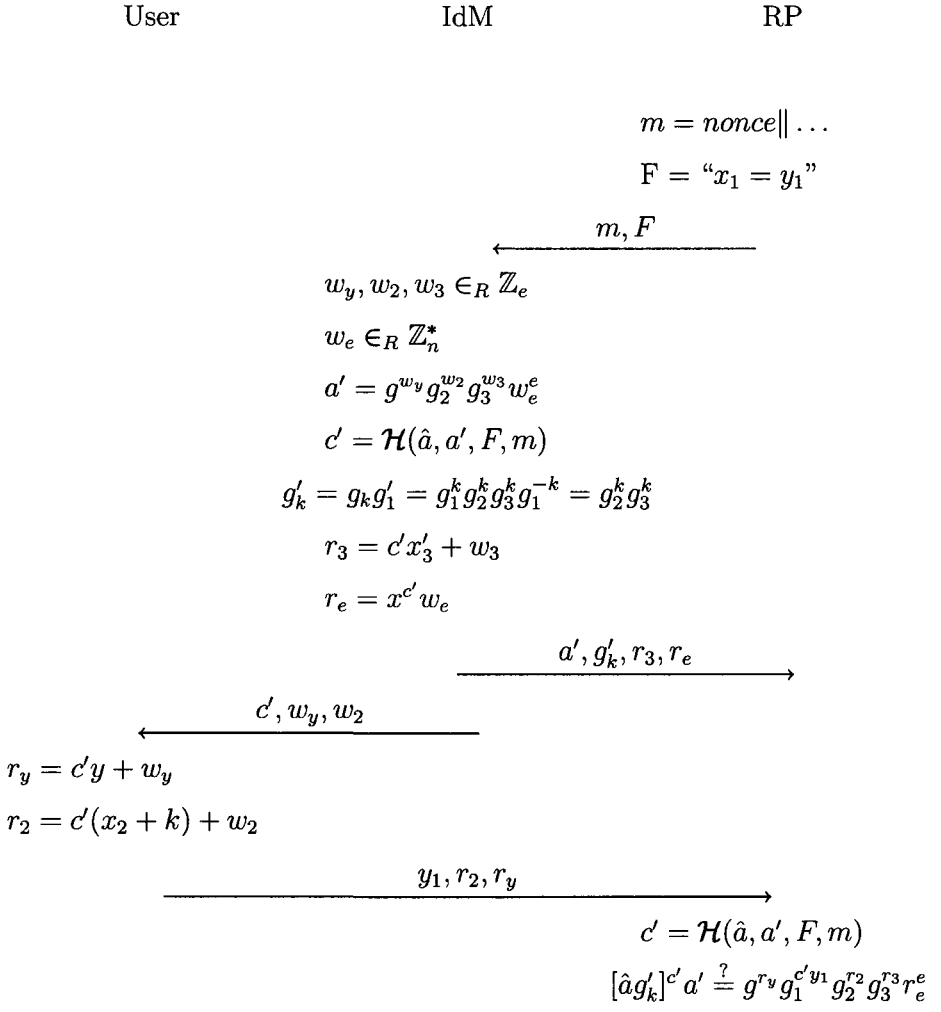

Figure 4.10: *Partial Trust* Showing Protocol

Proof 4.3.7

$$\begin{aligned}
L.S. &= [\hat{a}g_k']^{c'}a' \\
&= [(g^y g_1^{x_1} g_2^{x_2} g_3^{x_3} x^e)(g_2^k g_3^k)]^{c'}(g^{w_y} g_2^{w_2} g_3^{w_3} w_e^e) \\
&= g^{c'y+w_y} g_1^{c'x_1} g_2^{c'(x_2+k)+w_2} g_3^{c'(x_3+k)+w_3} x^{c'e} w_e^e \\
&= g^{c'y+w_y} g_1^{c'x_1} g_2^{c'x_2'+w_2} g_3^{c'x_3'+w_3} [x^{c'} w_e]^e \\
&= g^{r_y} g_1^{c'x_1} g_2^{r_2} g_3^{r_3} r_e^e \\
&= R.S.\ if \\
c'x_1 &= c'y_1 \\
x_1 &= y_1
\end{aligned}$$

The PoK 8.c requires no input whatsoever from the user. The IdM has all the necessary information in order to convince the RP that $\alpha = Com(\hat{m})Com(\hat{a})Com(\hat{b})$ has the proper construction. The witness to the PoK is the tuple $(\hat{m},r_1,\hat{a},r_2,\hat{b},r_3)$, all of which the IdM has access to. The protocol is shown in **Figure 4.5**, and the proof is shown in **Proof 4.3.3**. The PoK 8.d again requires no input whatsoever from the user. The IdM has all the necessary information in order to convince the RP that $\hat{v}$ is the third root of $\hat{m}$. The protocol is shown in **Figure 4.6**, and the proof is shown in **Proof 4.3.4**.

The PoK 8.e does require input from the user. The PoK 8.e is a challenge/response protocol which requires the *Prover* to demonstrate knowledge of the blinding factor y without revealing it. The protocol is similar to that of **Figure 4.7** and is shown in **Figure 4.11**. The user is required to participate when the challenge bit $b' = 1$. The user is responsible for constructing the response and sending it to the RP. The proof for PoK 8.e is the same

whether the policy is *Partial Trust*, *Complete Trust* or *No Trust* and is shown in **Proof 4.3.5**.

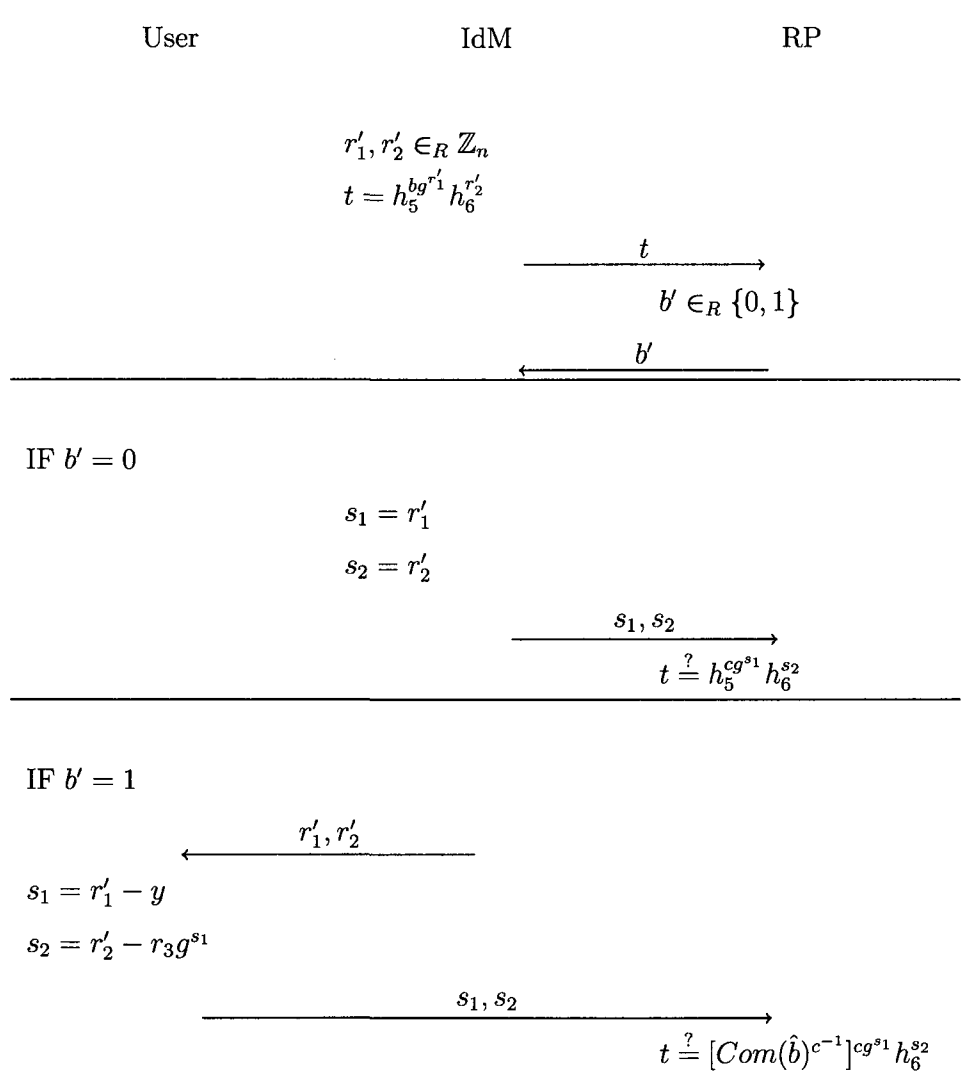

Figure 4.11: *Partial Trust* $\hat{b}$ Commitment Protocol

The *Partial Trust* policy provides an excellent middle-ground between the

No Trust and *Complete Trust* with respect to the number of computations the user is required to perform. The term *Partial Trust* has been used quite intentionally to describe this policy. The IdM, which stores credentials under this policy, is partially trusted because of its knowledge of identifiable information about its users. Although the IdM does not know the explicit values of x_1 and x_3, it does know the values for a, b, m, v, x_2 and x, all of which potentially identify the user. If the IdM was in fact malicious, and it was able to find a colluding issuing IdP that also knew the values of a, b, m, v, x_2, x and by extension x_1 and x_3, the two could potentially work together to de-anonymize the user. It is for this reason that the policy has been termed *Partial Trust*, because although the IdM alone cannot de-anonymize the user, if it colludes with a malicious IdP the user could potentially be de-anonymized. A more complete discussion as to why it is infeasible for the IdM to learn x_1 and x_3 on its own follows in section 4.5.

4.4 Computational Analysis

In order for the *Partial Trust* policy to be effective, it must off-load some of the computations the user is required to perform to their IdM. Each of the PoKs required during the showing protocol are analyzed for the number of random numbers that must be generated as well as the number of additions/subtractions, multiplications/divisions, and modular exponentiations that the user is required to perform.

The results of our analysis of each of the PoKs are shown in **Tables 4.1, 4.2, 4.3, 4.4, and 4.5**, where we assume the credential contains n attributes and during the showing protocol one of the n attributes is shown, e.g. $\Phi(\text{“}x_i =$

y_i") = 1. Counts denoted with a '*' are the number of additions/subtractions or multiplications/divisions required in the worst case, i.e. when $b' = 1$.

Policy	rand	+/-	×/÷	Mod. Exp.
No Trust	3+n	3+n	8+3n	6+n
Partial Trust	0	1	3	0
Complete Trust	0	0	0	0

Table 4.1: Computational Analysis of the Biometric Commitment Protocol

Policy	rand	+/-	×/÷	Mod. Exp.
No Trust	2+n	1+n	3+2n	4+n
Partial Trust	0	3	2	0
Complete Trust	0	0	0	0

Table 4.2: Computational Analysis of the Showing Protocol

Policy	rand	+/-	×/÷	Mod. Exp.
No Trust	4	4	8	5
Partial Trust	0	0	0	0
Complete Trust	0	0	0	0

Table 4.3: Computational Analysis of the Commitment Protocol

The cumulative results are shown in **Table 4.6**, with a graph of the total number of computations by policy shown in **Figure 4.12**. It is clear that the number of computations required by the user is considerably less when the

Policy	rand	+/-	×/÷	Mod. Exp.
No Trust	2	2*	2*	4
Partial Trust	0	0	0	0
Complete Trust	0	0	0	0

Table 4.4: Computational Analysis of the $\hat{m}$ Commitment Protocol

Policy	rand	+/-	×/÷	Mod. Exp.
No Trust	2	2*	2*	4
Partial Trust	0	2*	1*	1
Complete Trust	0	0	0	0

Table 4.5: Computational Analysis of the $\hat{b}$ Commitment Protocol

Partial Trust policy is used. When the *No Trust* policy is used the number of random numbers that must be generated as well as the number of additions/subtractions, multiplications/divisions and modular exponentiations scale linearly with the number of attributes in the credential. Although the number of calculations is linear in the parameter n, a constant number of calculations while using the *Partial Trust* policy will always outperform the computations required with the *No Trust* policy. We acknowledge that not all computations are equally intense for a processor to calculate (e.g. a modular exponentiation is much more computationally intensive than a simple addition); however, our analysis demonstrates that the number of computations required using the *No Trust* policy is $O(n)$, while the number of computations required using the *Partial Trust* policy is $O(1)$.

Policy	rand	+/-	×/÷	Mod. Exp.
No Trust	13+2n	12+2n	23+5n	23+2n
Partial Trust	0	6	6	1
Complete Trust	0	0	0	0

Table 4.6: Computational Analysis Results

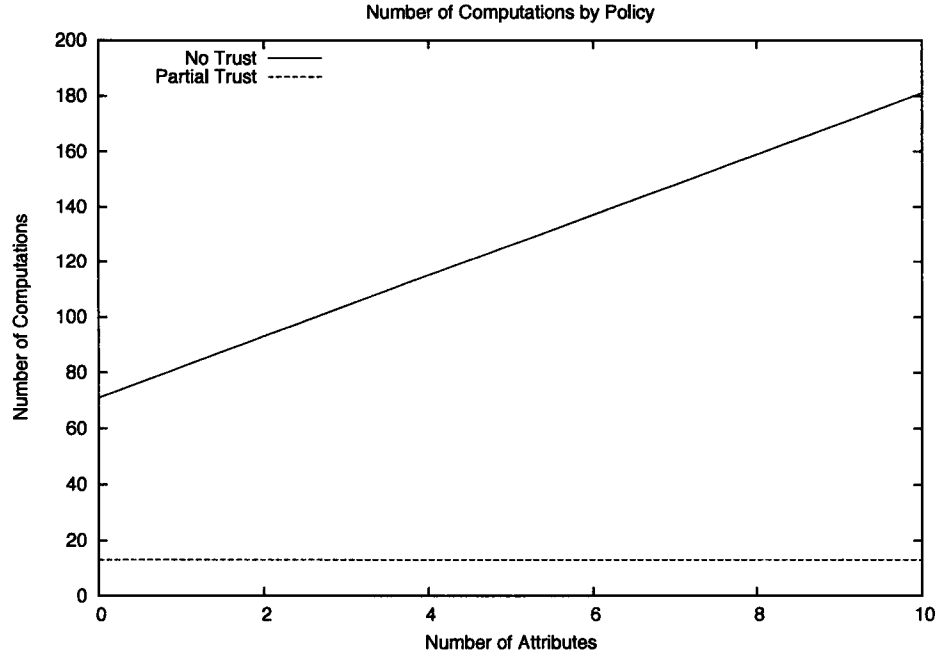

Figure 4.12: Total Number of Computations by Policy

4.5 Security Discussion

A security discussion is presented in this section for each of the trust models. Each of the adversaries in our new scheme will be also be analyzed.

4.5.1 Issuing Identity Provider

No matter which policy is used to store the user's credentials, the issuing IdP is not able to masquerade as the user. Since the issuing IdP does not know the values b_I, nor r_I, it is unable to impersonate the user's IdM in the biometric commitment PoK (as the IdP is unable to create a valid r_2' or prove that c_F and c_I are commitments on the same value). Even though the IdP knows the attribute values x_1, and $x_2 = g_c^{b_I} h_c^{r_I}$ as well as x_3 and x, the issuing IdP cannot determine the values b_I and r_I since it has one equation in two unknowns. Furthermore, even though an issuing IdP does in fact receive the user's attribute values, it is believed that a user will have credentials from many disparate IdPs, thus reducing the effectiveness of a single IdP from harvesting a large quantity of a user's private information spread across many different credentials.

4.5.2 Relying Party

Even though the RP receives any number of private attribute values during the showing protocol, it must do so by advertising which attributes it requires knowledge of through its access control policy, Φ. The formula Φ advertises which attributes the RP requires in order to grant access to the user. Once the user and IdM receive the function Φ they can make an educated decision as to what credential and attributes are appropriate to use and disclose to the RP in order to satisfy Φ. Furthermore, bookkeeping can be done by either the IdM or the user as to which attributes have been shown to which RPs. By keeping track of which attributes have been shown to which RPs, users are able to mitigate the effectiveness of an RP who stores historical information

about its users' transactions. Lastly, since digital credentials prevent linkage when shown multiple times (the multi-show property), colluding RPs are unable to correlate attributes from the same user (provided the user does not reveal a globally unique identifier assigned to them to both parties, e.g. their social insurance number).

It is important to note the difference between the privacy-preserving digital identity system in this section and that of OpenId [54, 98]. In OpenId, the user presents an identifier to an RP and the RP selects which attributes it requires knowledge of (perhaps through some access control policy set up by the IdP). However, in this system the user independently chooses which identity to present to the RP, as well as which credential and thus attributes. This approach is taken from CardSpace [29], which gives the power to the user as to what identity to use when attempting to satisfy an RP's access control policy. By organizing which attributes and credentials are shown to which RPs the user and IdM can mitigate the effectiveness of colluding RPs.

4.5.3 User

Users are prevented from sharing their digital credentials through the use of biometrics. This is the discussion of Chapter 5, where we analyze the effectiveness of an imposter to generate a genuine b_I or b_F value by determining the false accept rate of our proposed biometric algorithm.

4.5.4 Identity Manager

It has been made clear that if the user decides to store their credentials with a *Complete Trust* policy then the IdM is able to both masquerade as the user and harvest all of the private attributes stored with this policy. Under this policy the privacy-preserving digital identity system approaches that of the Shibboleth system.

On the other hand, storing credentials with the *No Trust* policy prevents the IdM from learning any of the user's private attributes, excluding the value for x_2, the initial biometric commitment. Even though the IdM learns the value for x_2 it cannot masquerade as the user since it does not know any of the user's other attributes and therefore cannot satisfy an RP's formula Φ.

When credentials are stored with a *Partial Trust* policy the IdM never learns the private attribute values, or the random value x_3 generated by the issuing IdP. Under this policy the IdM knows a, x_2, and x and thus is able to calculate $\frac{a}{g_2^{x_2} x^e} = g_1^{x_1} g_3^{x_3}$; however, even though the IdM knows the bases g_1 and g_3 the IdM cannot determine x_1 and x_3 due to the fact that calculating discrete logarithms is assumed to be infeasible in $\mathbb{Z}_n^*$. The IdM also knows the value $x_1' = x_1 + k$ and $x_3' = x_3 + k$; however the IdM at no point learns the value for k as the value k is kept secret at all times by the user. Lastly, even though the IdM knows the values for $g_k = g_1^k g_2^k g_3^k$ and $g_i' = g_i^{-k}$ it cannot calculate the value for k due to the discrete logarithm problem.

In this section the details of the proposed privacy-preserving digital identity management system have been presented. Each of the protocols required to realize the system as well as detailed mathematical proofs have been shown as evidence the system is theoretical achievable. In addition to the theoretical

discussion, an empirical study on the number of computations required for each of the PoKs required during the showing protocol for each of the policies was also given. It has been shown that the *Partial Trust* policy effectively off-loads computations from the user while maintaining the user's privacy. Lastly, a security discussion was given for each of the players in system as well for each of the policies. In the next chapter a biometric key generation algorithm will be presented to show that values for b_I and b_F can be practically achieved. We present metrics to show that the biometric values we generate are both of sufficient entropy and are unique for each individual.

Chapter 5

Biometric Key Generation

In this chapter we apply a biometric key generation algorithm to voice biometrics as a means to reliably create the required bitstrings, b_I and b_F. In this study we generate our biometric values from voice biometrics in order to achieve non-transferability in our digital identity management system. The algorithm discussed in this chapter is expected to be run by the IdMs in our system and all data structures, i.e. the self-organizing map, and biometric templates, which will be discussed, are expected to be stored at the IdMs and consist of biometric data from all the users whose identities are managed by each IdM.

Biometric key generation is an active area of research due to the permanence, non-repudiation, and portability of an individual's biometric signal. In general, individuals have shown difficulty in generating strong secrets. This has been exemplified by their tendency to choose insecure passwords, share their password with someone else, write their password down, or completely forget their password [5, 47]. These problems can be mitigated by generating

strong cryptographic keys from a user's biometric signal [9,73–76,106]. This alternative relies on the user's ability to reliably recreate the same biometric signal when prompted to do so. Not only is the user no longer required to memorize their secret, but the cryptographic keys that they generate are much stronger than the text based secrets they traditionally choose. Biometric Key Generators (BKGs) have been designed to measure the inherent entropy in biometric signals. In previous research, BKGs have been applied to many different physical applications. It has been well documented that biometrics can be used as a replacement for password based authentication as well as key management schemes [9,73–76,106]. However, biometrics can also be used in anonymous credential systems [18, 22] as pointed out by Adams in [2]. By using biometrics, Adams shows how it is possible to link a user's digital identity with their physical identity in a privacy-preserving way.

In this chapter, we present the results of applying our modified version of a BKG algorithm [9] traditionally used to generate cryptographic keys from handwriting biometrics to voice biometrics. We show how a biometric key, what we've been referring to as b_I and b_F, can be reliably extracted from a user's biometric reading. We also empirically evaluate the ability of an adversary to recreate a user's biometric key when given access to the user's decrypted template as well as feature statistics from the entire population. The adversary's ability to generate cryptographic keys is evaluated using a probabilistic search algorithm based on the *Guessing Distance* metric [7]. Additionally, we show that cryptographic keys at least 2^{10} times stronger than keys generated from voice biometrics using previous work [73–75] can be generated for 70% of the population. Our novel contribution in this chapter is our algorithm for extracting *reliable features* from voice biometrics. We show that it is possible to extract features that demonstrate a high inter-user

variation and a low intra-user variation across our population.

In this study we analyze voice biometrics due to the prevalence of microphones in the daily lives of individuals and the perceived comfort that individuals demonstrate when using voice-based systems. Additionally, this study focuses on voice biometrics because the corpus of research from the last 40 years in both automated speech recognition (ASR) and speaker verification (SV) is quite vast and easily accessible [59, 85]. Lastly, voice biometrics are a behavioural biometric, as opposed to a physiologic biometric such as face, and fingerprints. Behavioural biometrics can easily be varied by the individual supplying the sample, allowing users to generate many different cryptographic keys by simply modifying the contents of their speech. This property is extremely attractive, especially in digital identity systems where a user possibly has many different identities.

5.1 Background

In this section we review Ballard et al.'s RBT algorithm [9] in detail and review the security requirements of BKGs as outlined in [10]. Additionally, we outline the basic theory behind speech processing and feature extraction, specifically the perceptual linear predictive (PLP) analysis of speech [55]. Lastly, we conclude with the theory behind self-organizing maps (SOMs) [65], an important artificial intelligence data structure used in our implementation.

5.1.1 Randomized biometric templates (RBTs)

The first step in generating a cryptographic key from a biometric signal is to extract features from the signal. The RBT algorithm introduced in this section takes as input a sequence of biometric features and outputs a cryptographic key. In [9], Ballard et al. hypothesized that their RBT algorithm could be applied to biometric modalities other than handwriting as long as proper features could be extracted. In this Thesis we show that the RBT algorithm can be modified to in fact generate strong cryptographic keys from voice biometrics.

RBTs require features that are able to effectively differentiate between individuals in the population (i.e. have a large inter-user variation), yet reliably extract similar values from samples of the same individual (i.e. have a low intra-user variation). Once features have been extracted, the RBT algorithm is able to distill entropy from the features by capturing the inter-user variation inherent in the biometric signal. In order to reliably recreate the same cryptographic key, RBTs use quantization as a means to correct small perturbations in subsequent readings from the same individual. Ballard et al.'s algorithm consists of two main algorithms: an enrollment algorithm, *Enroll*, and a key generation algorithm, *KeyGen*.

Enroll $(\beta_1, \ldots, \beta_l, \pi)$: The *Enroll* algorithm is a four step process and is shown in **Figure 5.1**. The algorithm assigns features to a user, computes the necessary error correction information for each of the features, creates a cryptographic key for the user, and finally encodes their secure template, which is used by *KeyGen* to recreate the newly created key.

Input: The password, $\pi \in \Pi$, and biometric samples $\beta_1, \ldots, \beta_l$.

Input: (Global value): The set of all features Φ

Input: (Global value): Quantization widths $\delta_0, \ldots, \delta_N$.

Output: The key K, and template T..

1: $(\Psi, \widetilde{\Psi}) \leftarrow$ Select$(\beta_1, \ldots, \beta_l)$ // Select biometric features
2: $L \leftarrow$ Permute$(\Psi)||$Permute$(\widetilde{\Psi})$
3: $k_0 \leftarrow H_{pass,0}(\pi), k_1 \leftarrow H_{pass,1}(\pi)$
4: **for** $j = 0$ to $|L| - 1$ **do**
5: $i \leftarrow L[j]$
6: $\mu_i \leftarrow \text{Median}(\phi_i(\beta_1), \ldots, \phi_i(\beta_l))$
7: **if** $\mu_i \geq \frac{\delta_i}{2}$ **then**
8: $\alpha_i \leftarrow \lfloor \mu_i - \frac{\delta_i}{2} \rfloor \bmod \delta_i$
9: **else**
10: $\alpha_i \leftarrow \lfloor \mu_i + \frac{\delta_i}{2} \rfloor$
11: **end if**
12: $x_i \leftarrow \max(0, \lfloor \mu_i - \frac{\delta_i}{2} \rfloor)$ // Quantize feature outputs
13: $\gamma_i \xleftarrow{R} [\alpha_i, \Delta]_{\delta_i}$ // Select random quantization offset
14: $C_j = (E^N_{k_0}(i), E^{\Delta}_{k_1}(\gamma_i))$ // Encrypt values
15: $K_j = i||x_i$
16: **end for**
17: $K \leftarrow H_{key}(\pi||K_0||K_1||\ldots||K_{|\Psi|-1})$ // Derive key
18: $C \leftarrow (C_0||C_1||\ldots||C_{|L|-1})$
19: $v \leftarrow H_{ver}(\pi||K_0||K_1||\ldots||K_{|\Psi|-1})$
20: **return** $K, T = (C, v)$

Figure 5.1: Enroll Algorithm [9]

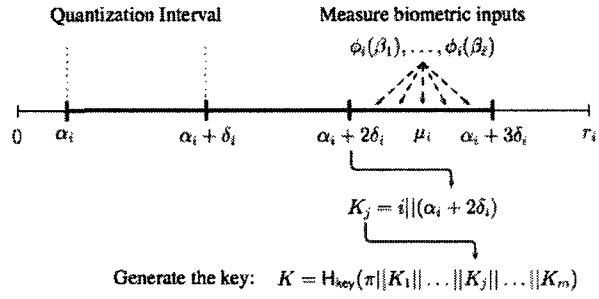

Figure 5.2: Quantizing the feature ϕ_i with quantization width δ_i [9]

When the user presents their biometric samples, $\beta_1, \ldots, \beta_l$, to *Enroll*, the algorithm computes some statistics over the samples and creates a set of *reliable features*, Ψ, consisting of features that the user can reliably recreate. To ensure that all biometric templates have the same number of features the set of *reliable features*, Ψ, is padded with a random set of features, $\tilde{\Psi}$. The set of random features, $\tilde{\Psi}$, however, is not used in the creation of the user's key. Both sets of features are randomly permuted and added to the user's template.

In the next step of the process each of the feature outputs, ϕ_i, are quantized to a single, repeatable value, x_i. The quantization widths are pre-computed on a per feature basis and take into consideration the population statistics for each feature. Each feature, ϕ_i, is assigned a quantization width, δ_i, which is used for error correction. The process of mapping a feature, ϕ_i, to a quantized value x_i is shown **Figure 5.2**.

Once the user's feature outputs have been quantized, each of the x_i values are concatenated together along with their feature index i, and hashed to form the user's biometric key, K, which is not stored in the template. A verification hash function, H_{ver}, on the other hand is used

to compute a hash, v, of the quantized values which **is stored** in the user's template.

The last step in the *Enroll* algorithm computes the user's template T. The goal of RBTs is to create templates that are unverifiable to an adversary both when they are decrypted using the correct password, π, and when they are decrypted using an incorrect password $\pi' \neq \pi$. In order achieve this, the *Enroll* algorithm randomizes both L and the α_i values so that the they are indistinguishable from a random string of bits. The randomized values are then encrypted using the user's low entropy password, π, to create the user's template. Ultimately, a user's template consists of the verification hash, v, and C, the encrypted feature indices and α_i values.

KeyGen (β, π, T) : The KeyGen algorithm simply decrypts a user's template, T, with the user's password, π, and attempts to recreate the user's key, K, with their biometric sample, β. For each $j \in [0, |C| - 1]$, the feature index i and the alpha value, α_i, are extracted. The algorithm recreates the user's key by mapping the user's reading of $\phi_i(\beta)$ to the quantization sub-interval measured using δ_i over the interval $[0, r_i]$. The algorithm iteratively checks to see if the concatenated hash value computed using H_{ver} is equal to the stored hash, v. Once the hashes match, the true key is computed using the hash, H_{key}, and the key is returned by the algorithm. If the key cannot be recreated the value $\perp$ is returned.

5.1.2 Security requirements

The requirements we use to evaluate our implementation of RBTs are based on the requirements outlined for BKGs in [10]. In [10], Ballard et al. outlined three specific requirements required to properly evaluate the security of BKGs. The requirements are:

Biometric Uncertainty (REQ-BUN) : The biometric samples, β_i, are difficult to predict. It is assumed that an adversary is not able to change their voice in such a way that they can produce other users' cryptographic keys. This assumption requires voice signals that have a high entropy across the population and is evaluated by testing the false accept rate of our implementation.

Key Randomness (REQ-KR) : Assuming biometric uncertainty, keys output by the RBT algorithm must appear random to an adversary who has access to the user's template and any other *auxiliary information*. We evaluate this requirement by empirically evaluating the amount of entropy in each of the keys generated by our implementation.

Strong Biometric Privacy (REQ-SBP) : Assuming biometric uncertainty, an adversary is not able to learn any information about a user's biometric sample even when given access to *auxiliary information*, including the user's template, and the key itself.

We use all three of these security requirements to evaluate our implementation in subsequent chapters.

5.1.3 Automatic speech processing

There exists a vast body of research in both automatic speech recognition (ASR) and speaker verification (SV). In this section we present a very basic overview of how speech is digitally encoded and how features are extracted. There currently exists a number of methods for extracting features from voice, including linear prediction analysis (LPC), mel-frequency cepstral analysis (MFC) and perceptual linear prediction (PLP) analysis [55]. A good discussion of the traditional analysis techniques, such as LPC and MFC, can be found in [59, 85]. The following overview is heavily borrowed from [55, 75] due to the clarity and simplicity with which it is documented.

Speech is captured by a microphone by measuring the vibrations of the microphone's diaphragm. These vibrations are converted into an electrical signal and sampled at 12.5kHz by an analog-to-digital converter to produce a sequence of values $A_{time}(1), A_{time}(2), \ldots, A_{time}(k)$ representing the amplitude of the signal. These amplitudes compose what is referred to as the *time domain* representation of the speech signal. Another common way of analyzing a speech signal is to convert its *time domain* representation into its *frequency domain* representation. The signal can easily be converted from the *time domain* to the *frequency domain* by using the discrete Fourier transform (DFT). The DFT decomposes the signal into a series of sine and cosine waves of varying periods, amplitudes and phases. The signal can also be easily converted back to the *time domain* by using the inverse discrete Fourier transform (IDFT).

In this study, PLP analysis is used to extract feature vectors from speech. PLP has an advantage over other feature extraction methods because it ap-

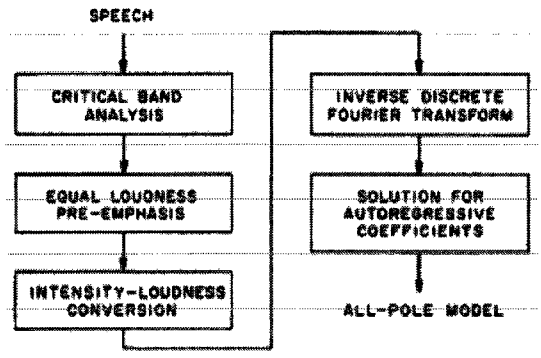

Figure 5.3: Block diagram showing the stages of perceptual linear predictive (PLP) speech analysis [55]

proximates the human auditory system by taking into consideration the psychophysics of hearing. A block diagram showing the stages of PLP analysis is shown in **Figure 5.3**, each stage is briefly explained here.

Spectral analysis : In our experiment each speech sample was segmented into 20ms frames. A new frame was created every 10ms, resulting in 10ms of overlap between frames. PLP analysis functions by weighting the *time domain* representation of each frame by a Hamming window before computing its DFT. Taking the DFT of the *time domain* representation of each frame creates the short-term speech spectrum, $S(w)$, of the frame. From the short-term speech spectrum the short-term power spectrum, $P(w)$, is computed by squaring and summing the real and imaginary parts of $S(w)$.

Critical-band analysis : PLP warps the frequency of the power spectrum according to the Bark scale, converting the power spectrum's frequency from w to Ω:

$$\Omega(w) = 6ln[w/1200\pi + \sqrt{(w/1200\pi)^2 + 1}] \quad (5.1)$$

The resulting power spectrum is then convolved with the critical-band masking curve $\Psi(\Omega)$:

$$\Psi(\Omega) = \begin{cases} 0 & \text{if } \Omega < -1.3, \\ 10^{2.5(\Omega+0.5)} & \text{if } -1.3 \leq \Omega \leq -0.5, \\ 1 & \text{if } -0.5 < \Omega < 0.5, \\ 10^{-1.0(\Omega-0.5)} & \text{if } 0.5 \leq \Omega \leq 2.5, \\ 0 & \text{if } \Omega > 2.5 \end{cases} \quad (5.2)$$

The resulting convolution results in the samples of the critical-band power spectrum $\Theta(\Omega)$.

Equal-loudness pre-emphasis : Taking into account the non-equal sensitivity of human hearing at different frequencies, the sampled signal $\Theta(\Omega)$ is pre-emphasized by the equal-loudness curve $E(w)$ to form $\Xi(\Omega(w)) = E(w)\Theta(\Omega(w))$.

Intensity-loudness conversion : In order to properly approximate the power law of hearing, the cube-root of the amplitude is taken to form $\Phi(w) = \sqrt[3]{\Xi(\Omega)}$.

Autoregressive modeling : During the last stage of PLP analysis the IDFT is taken and an all-pole model of the spectrum is created. Varying values of the order of the model can be calculated. For the purposes of this study an all-pole model of the 12^{th} order was used. The resulting

all-pole model allows for 12 features to be extracted from each frame of speech, resulting in a 12-dimensional, real valued feature vector for each frame of speech.

In addition to the 12 features extracted using PLP analysis, delta coefficients were also extracted, resulting in feature vectors of 24-dimensions for each frame of speech. Delta coefficients calculate the rate of change of the coefficients over the sequence of frames in a speech signal. The use of delta coefficients has been shown to increase the performance of ASR systems [59].

5.1.4 Self-organizing maps

Traditionally, in ASR and SV systems, a model based on a single speaker is created during enrollment. During testing, the model is used to determine the validity of an unknown speaker. Models that are created from samples generated by a single speaker are referred to as *speaker dependent* models, while models that are created from samples from multiple speakers are referred to as *speaker independent* models. Additionally, models that are created from a predefined sequence of speech are referred to as *text dependent* models, whereas models that are created from unstructured speech are referred to as *text independent* models. In this study, storing a *speaker dependent* or *text dependent* model could potentially leak information about a user's biometric signal to an adversary and possibly circumvent *REQ-SBP*. Therefore, the model chosen in this study is a *text independent*, *speaker independent* model that is presumed to leak no biometric information about a single individual. The model, however, is assumed to leak statistical information about the population as a whole.

The data structure chosen to model the population's speech in our study was the self-organizing map (SOM) of Kohonen [65]. Self-organizing maps can effectively be used to project high-dimensional spaces onto a lower, two-dimensional space. In our experiment, feature vectors of 24-dimensions were mapped to a much more manageable two-dimensions through the use of a SOM. In addition to converting high-dimensional data into a two-dimensional space, the SOM also preserves the relative "ordering" of the high-dimensional data in the two-dimensional space. This property of SOMs allows for complex vectors to be analyzed and visualized in a much simpler fashion.

Self-organizing maps can be laid out in a variety of different topologies; however, a grid layout or a hexagon layout, represented by a two-dimensional array of nodes, is typically used. During initialization, each node, at coordinates (x, y) in the grid, is assigned a sequence of values in the 24-dimensional space, referred to as the weight vector of the node, $M_{x,y} = (m_1, m_2, \ldots, m_{24})$. Typically, randomization is used to assign initial weights to the nodes; however, in this study the weights of the nodes were initialized using the eigenvectors of the enrollment data. This method has been shown to be effective in reducing the number of iterations required during training [65].

During training, each of the 24-dimensional enrollment vectors generated from the population's speech samples were used to update the weight vectors of each of the nodes in the SOM. Each weight vector is updated by iteratively determining which node in the SOM is the *best matching unit* (BMU) for each enrollment vector. To determine the BMU, a distance metric must be defined. In our study the Euclidean distance metric was used and is expressed in this Thesis as $d(X, Y)$ and is defined as:

$$d(X,Y) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2} \tag{5.3}$$

Where each of $X = (x_1, x_2, \ldots, x_n)$, and $Y = (y_1, y_2, \ldots, y_n)$ are defined to be vectors with the same number of dimensions, n. Using the Euclidean distance measure, the BMU for each enrollment vector can be calculated. For each enrollment vector, X, the coordinates, (x, y), that best satisfy the minimum equation

$$(x,y) = argmin_{i,j}\{d(X, M_{i,j})\} \tag{5.4}$$

are determined. Once the BMU, $M_{x,y}$, has been found, the weights of both $M_{x,y}$ as well as the "neighbours" of $M_{x,y}$ are updated to approximate the topological ordering of the enrollment data. A large number of iterations, which was determined experimentally, was required before the SOM converged to an acceptable cumulative error.

After training, a SOM can be queried for the BMU of any given input vector. While the input vector is a 24-dimensional vector, the resulting usable value from the SOM is simply the two-dimensional coordinate, (x, y), of the BMU. In **Figure 5.4** a Sammon map [94] of the SOM used in this experiment is shown. The Sammon map effectively presents a multi-dimensional data source in two-dimensions while still preserving the Euclidean distance between the nodes.

An important distinction needs to be made between the SOMs used in this study and the VQ method used in previous work. The SOM data structure

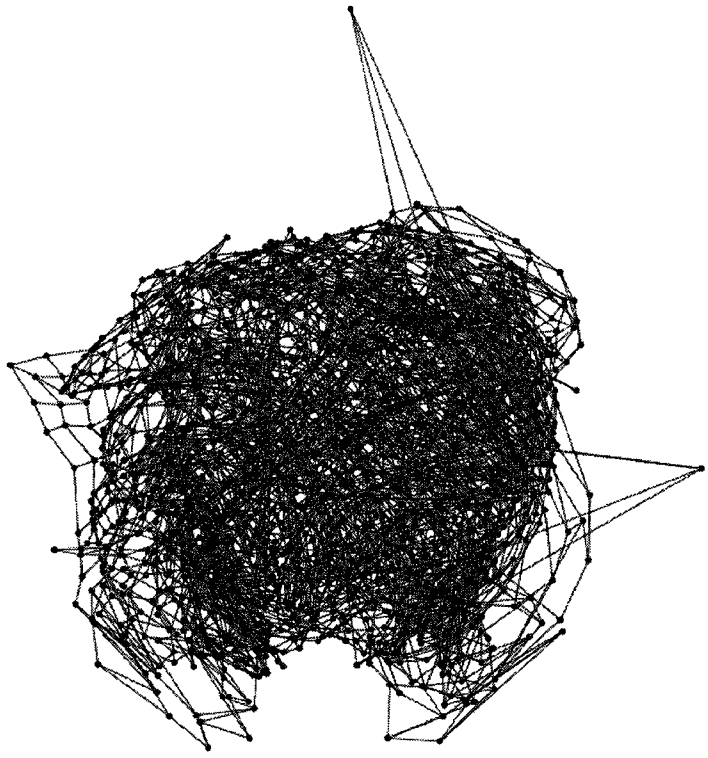

Figure 5.4: Sammon mapping of the 48x48 SOM used in this experiment. This image was generated using the SOM_PAK [66]

was chosen over VQ because of the SOM's ability to preserve the "ordering" of the input vectors. This property was crucial to applying the RBT algorithm to voice biometrics. By preserving the "ordering" of the input vectors we were able to leverage the quantization error correction mechanisms within the RBT algorithm. By combining RBT's error correction mechanism with a new deterministic correction algorithm we were able to achieve very good false rejection rates. Our novel error correction algorithm is described in detail in section 5.3.2.

5.2 Algorithms

In this section we introduce our novel feature extraction algorithm. We also discuss our algorithm for segmenting a sequence of n 24-dimensional feature vectors into a sequence of k (x, y) coordinates. Additionally, we discuss our method for mapping each segment to a set of usable features.

5.2.1 Feature extraction algorithm

Now that we have outlined the background required to understand our study we introduce our novel algorithm for converting speech signals into *reliable features*. As previously mentioned, Ballard et al.'s original implementation of RBTs was applied to handwriting samples. Features were extracted from each handwriting sample and used to created RBTs and cryptographic keys for each of the users in their study. During their study, Ballard et al. extracted a number of different features, e.g. the velocity of the pen during the signature, the start and end coordinates of the signature, etc. Although

effective measures for handwriting biometrics, they have no analogous metric in speech processing.

The steps of our feature extraction algorithm are shown in **Figure 5.5**. Our algorithm takes as input a speech sample, β, and outputs a sequence of features, $\phi_1, \phi_2, \ldots, \phi_{2k}$. Our algorithm starts by capturing a speech sample, β_i, (Step 1) and performs PLP and delta analysis on each frame (Step 2). From the PLP and delta analysis the frames are converted into a sequence of n 24-dimensional feature vectors consisting of PLP coefficients and delta coefficients (Step 3). The BMU, M_{x_i,y_i}, for each vector in the sequence is calculated from our SOM to form the sequence $M_{x_1,y_1}, M_{x_2,y_2}, \ldots, M_{x_n,y_n}$ (Step 4). Each BMUs' coordinates are then extracted to form the sequence of coordinates $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ (Step 5). The sequence of coordinates is then segmented into k segments using our segmentation algorithm (Step 6). Each segment is then mapped to the coordinates, (x_{b_i}, y_{b_i}), which best minimizes the total cumulative distance from each of the coordinates (x_i, y_i) in the segment, to the coordinates (x_{b_i}, y_{b_i}), resulting in a sequence of k coordinates $(x_{b_1}, y_{b_1}), (x_{b_2}, y_{b_2}), \ldots, (x_{b_k}, y_{b_k})$ (Step 7). From the sequence of k coordinates, each of their x_{b_i} and y_{b_i} components are then mapped to the feature values ϕ_i and ϕ_{i+1} respectively to create the sequence of features $\phi_1, \phi_2, \ldots, \phi_{2k}$ (Step 8). The sequence of features $\phi_1, \phi_2, \ldots, \phi_{2k}$ is then fed into the RBT algorithm to generate a user's cryptographic key and biometric template.

A detailed description of how we segment each sequence of coordinates as well as how we map each segment to useable features is presented in the next two sections.

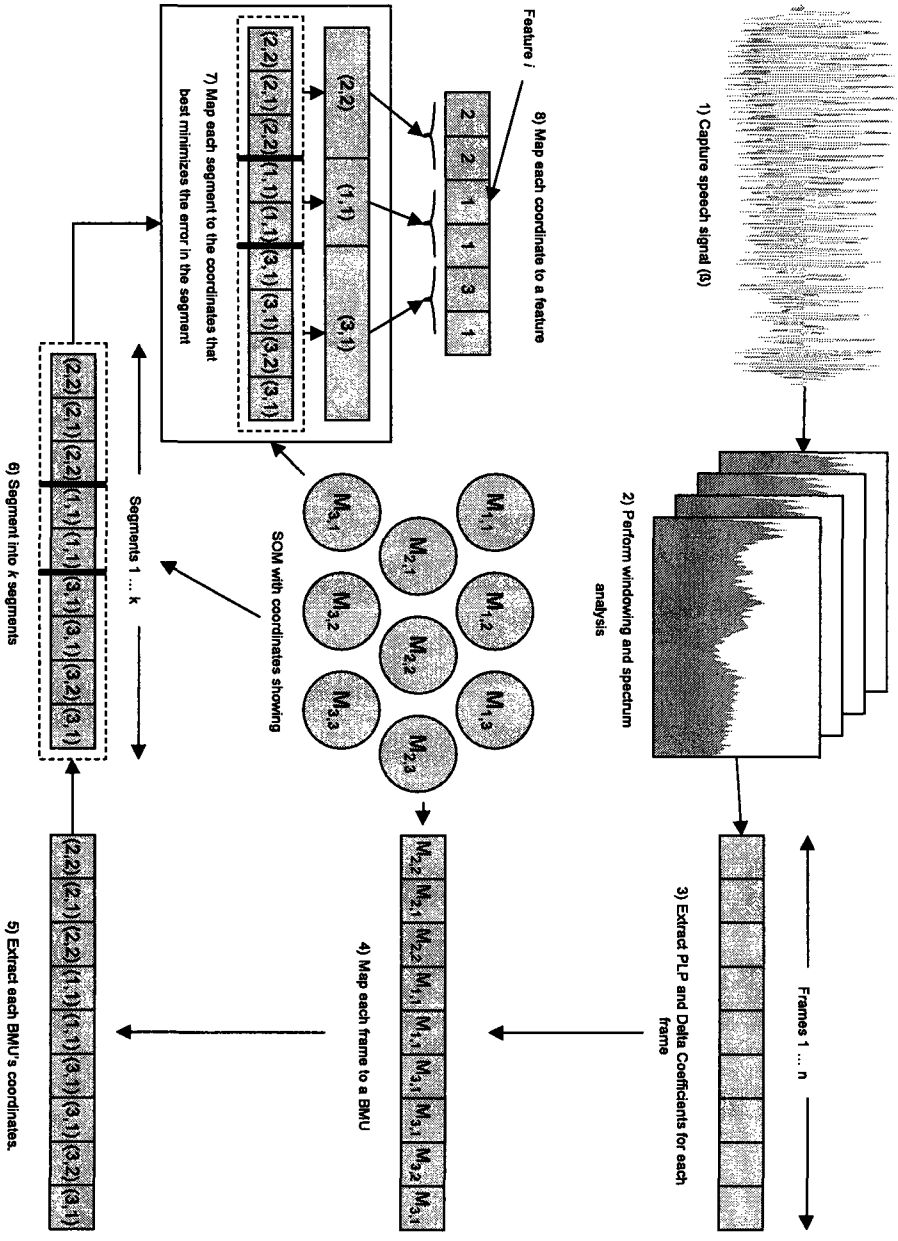

Figure 5.5: Flow diagram of our feature extraction algorithm.

5.2.2 Segmenting the frames

Our segmentation algorithm chooses the set of k nodes from our SOM which best describe a sequence of n frames. Let us assume that we have extracted our feature vectors from a given speech sample and have found each feature vector's associated BMU. Let us also assume that we have created a sequence of coordinates from the BMUs, which we represent by $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. Let $R_1 \ldots R_k$ be disjoint, nonempty ranges over the natural numbers $\cup_{i=1}^{k} R_i = [1, n]$, where the i-th segment is $(x_j, y_j), \ldots, (x_{j'}, y_{j'})$ if $R_i = [j, j']$. Our goal in segmenting this sequence of feature vectors is to reduce the sequence of n coordinates into a list of k disjoint ranges over $[1, n]$ by reducing the error, E_i, for each segment:

$$E_i(R_i, (x_{b_i}, y_{b_i})) = \sum_{k=j}^{j'} d((x_k, y_k), (x_{b_i}, y_{b_i})) \tag{5.5}$$

Here (x_{b_i}, y_{b_i}) denotes the coordinates of the node in our SOM found to best minimize **Equation 5.5**.

The goal of our segmentation algorithm is therefore to minimize the total error, E, for all the segments:

$$E = \sum_{k=1}^{k} E(R_i) \tag{5.6}$$

The problem of segmenting a sequence of n two-dimensional values into k segments has been deemed the *k-segmentation* problem. The *k-segmentation* problem is optimally solved by the dynamic programming algorithm of Bell-

man [12]. The main drawback of the traditional dynamic programming approach however, is that Bellman's algorithm, although optimal, has a run-time of $O(n^2k)$. In order to speed up the run-time of Bellman's algorithm we designed a heuristics based algorithm to segment each of our speech samples. Our segmentation algorithm is split into two stages, *Grouping* and *Segmenting*.

Grouping : During the *grouping* stage, coordinates that are "close" to one another are grouped together. The algorithm iteratively places contiguous coordinates into the same group if and only if they are within a certain distance of the component-wise mean of the group. The pseudo-code for the algorithm is shown in **Figure 5.6**. In our experiments a $THRESHOLD$ value of 10 was found to work best.

Segmenting : The next stage of the algorithm segments the n coordinates into k coordinates. Depending on how many groups were created during the *Grouping* stage, groups are either split into two groups if $|R| < k$, or two consecutive groups are merged together if $|R| > k$. Groups were split by searching for a group that when split reduced the total error, E, by the greatest amount. Groups were merged by searching for two consecutive sub-groups that when merged raised the total error by the least amount. In both cases this stage proceeded iteratively either reducing or increasing the cardinality of R until $|R| = k$. Once $|R| = k$, we represented each of the groups in R by the coordinates, (x_{b_i}, y_{b_i}), of the BMU found to best minimize **Equation 5.5**. Each of the coordinates (x_{b_i}, y_{b_i}) were then mapped to features.

After the *grouping* stage, groups in R that had a cardinality of one were

Input: The sequence of coordinates $(x_1, y_1), \ldots, (x_n, y_n)$.

Output: The segmented groups of coordinates, R.

```
1: G ← ()
2: R ← ()
3: for i = 1 to n do
4:   if G is empty then
5:     G.append((x_i, y_i))
6:   else
7:     μ_i ← Mean( G )
8:     if distance( (x_i, y_i), μ_i ) ≤ THRESHOLD then
9:       G.append((x_i, y_i))
10:    else
11:      R.append(G)
12:      G ← ()
13:      G ← (x_i, y_i)
14:    end if
15:  end if
16: end for
17: if G is not empty then
18:   R.append(G)
19: end if
```

Figure 5.6: Grouping algorithm used to segment a sequence of coordinates.

removed prior to the *segmentation* stage of the algorithm. Our heuristics based segmentation algorithm, described in this section, reduced the time it took to segment our speech samples from a number of days to a couple of hours.

5.2.3 Mapping segments to features

Once the frames of speech were segmented into their (x_{b_i}, y_{b_i}) coordinates, the coordinates were used to create the features required by the RBT algorithm. In our study, each of the x and y components that represented a segment of speech were mapped to a feature. The x and y coordinates were mapped to ϕ_i and ϕ_{i+1} respectively, thus resulting in two features being mapped to each segment.

During the creation of our technique, other mappings were also tried. The most promising mapping (other than the one we ultimately used) mapped each node directly to a feature by setting $\phi_i = (x_{b_i}, y_{b_i})$. This required mapping each feature to the two-dimensional coordinates representing each segment. This however, greatly reduced the number of *reliable features* in our study (by half), resulting in lower entropy keys. In this study we report the results based on mapping each segment to two distinct features.

Both the SOM and our segmentation algorithm provided us with a way to efficiently convert a sequence of 24-dimensional feature vectors into a set of usable features. We chose to represent our speech samples as a sequence of two-dimensional coordinates to take advantage of the underlying structure of our SOM. By reducing the dimensionality of our feature vectors from 24 to two, we were better able to create features that relatively made sense.

A segment coordinate $(10, 12)$, for instance, is relatively close to the coordinate $(11, 14)$. By extracting the x-component of both these coordinates we were able to create a feature that exhibits a low intra-user variation (assuming both these coordinates are from the same user's speech samples). A segment coordinate $(10, 12)$ and $(35, 42)$ however, are not relatively close to one another. By extracting the x-component of both these coordinates and comparing them to one another we can discard the x-component as a feature because it exhibits a relatively high intra-user variation (again assuming these coordinates are from the same user's speech samples). This simple property of mapping high dimensional data to two-dimensions, allowed us to extract very strong and reliable features.

The SOM we used to map BMUs to features was configured with a large number of nodes in order to obtain a more precise coordinate value for each BMU. This allowed each feature to be mapped to a larger set of values across the population, ultimately leading to keys with more entropy. Smaller SOMs were tested, however, they mapped a large number of users' feature values to the same value, which ultimately led to cryptographic keys with a lower amount of entropy. By increasing the number of nodes in the SOM we were able to extract more precise inter-user features, resulting in keys that had more entropy. It should be noted that the construction of our SOM did not allow us to extract more entropy than exists in the biometric signal as the amount of entropy that RBTs can extract is upper bounded by the inherent entropy in the biometric signal. However, by mapping features to a larger set of values we were able to allow the RBT algorithm to generate cryptographic keys with relatively higher entropy then when features were mapped to a smaller set of values.

Our algorithm for mapping speech samples to *reliable features* is one of the novel contributions of this Thesis. In our implementation of applying RBTs to voice biometrics our biggest challenge was extracting features that allowed the RBT algorithm to extract a high amount of entropy while minimizing both the FAR and FRR of our implementation. In handwriting, features can easily be extracted by observing the mechanical movements of a user's hand while executing their signature, however, extracting features from voice biometrics required extracting features from the raw input signal of the voice samples in a reliable way.

5.3 Evaluation

In this section we describe the metrics we used to evaluate our implementation and discuss how we modified the original RBT algorithms for our purposes. We also discuss the setup of our experiment and provide empirical results for the important evaluation metrics.

5.3.1 Entropy

Traditionally, the measure *Guessing Entropy* has been used to quantify the entropy in cryptographic keys generated by BKGs. Most notable to us in this study is the entropy reported by Monrose et al. in [73–75]. In their analysis, the authors concluded that the maximum amount of entropy in their keys can be summarized by the equation $guesses = min\{2^m, (|A| + 1)/2\}$, where m is the number of features used to generate a user's key and A is the set of users in their experiment. What this metric says is that given the set of keys

generated by the their BKG it would take *guesses* on average to map a key to a certain individual. There are a number of problems with this metric. One of the problems with this metric is that it assumes keys are generated in a uniform fashion. However, the main problem with *Guessing Entropy* is that it does not account for the fact that some users in the system can generate keys that are easier to guess than others because it attempts to summarize the average number of guesses required for the population as a whole. The metric *Guessing Distance* however, attempts to address this issue [7, 8].

By analyzing empirical data, *Guessing Distance* is able to provide a proper metric for the most likely probabilistic attack an adversary would launch against a BKG when given access to *auxiliary information. Guessing Distance* assumes that an adversary who is attempting to generate a user's key has access to the population statistics for each feature as well as the user's template. Let us assume that a user imposes a probability distribution, U, over the set of feature values, Ω, and the population imposes a probability distribution, P, over the same set of feature values. *Guessing Distance* is able to estimate the number of guesses a logical adversary would make, given the probability distribution, P, and a user's template, T.

Guessing Distance measures the ability of the probability distribution P to predict the most likely elements of U. What we desire is a metric that assigns the same value to two probability distributions, P_1, and P_2 if and only if P_1, and P_2 imply the same guessing strategy. Let us use an example from [7] to expand this further. Let us assume that a user generates a value w_i for some feature, i.e. $U(w_i) = 1$. Let us also assume that the population statistic $P_1(w_i) = 0.9$ and $P_2(w_i) = 0.8$. What we desire is a metric that assigns the same "distance" to U and P_1, and U and P_2. In this example the same

value should be assigned to both distributions since both P_1 and P_2 predict that w_i will be generated by a user with such a high probability. The formal definition of *Guessing Distance* follows:

Definition 5.3.1 *Let* $w* = arg\ max_{w \in \Omega} U(w)$. *Let* $L_P = (w_1, w_2, \ldots, w_n)$ *be the elements of* Ω *ordered such that* $P(w_i) \geq P(w_{i+1})$ *for all* $i \in [1, n-1]$. *Define* t^- *and* t^+ *to be the smallest index and largest index* i *such that* $|P(w_i) - P(w*)| \leq \delta$. *The* Guessing Distance *between* U *and* P *with tolerance* δ *is defined as [7, 8]:*

$$GD_\delta(U, P) = log \frac{t^- + t^+}{2} \tag{5.7}$$

Guessing Distance measures the number of guesses required by an adversary who assumes $U \approx P$. The metric measures how many guesses are made by an adversary before they correctly guess that a specific user generates the value $w*$ for some feature. The *Guessing Distance* estimation algorithm, although originally used to analyze the entropy in keys generated by a handwriting BKG, was designed to be agnostic of the underlying biometric signal being measured. *Guessing Distance* was chosen over *Guessing Entropy* because instead of *summarizing* the security of our implementation it allowed us to model a logical adversary's attack on a given user's biometric template. In this Thesis we evaluate our implementation's ability to satisfy REQ-KR by using the biometric-agnostic *Guessing Distance* metric.

5.3.2 False rejection rates and false acceptance rates

Another group of metrics we use to empirically evaluate our implementation is the group of false rejection rates (FRR) and false acceptance rates (FAR). The FRR was calculated by taking the percentage of authentic samples that were rejected as not being able to generate the correct cryptographic key from the authentic user's template, while the FAR was calculated by taking the percentage of imposter samples that were accepted as having generated the correct biometric key from an authentic user's template. The results obtained from both these measures demonstrate that our system has the potential to be used as a practical system.

Before quantifying the FAR and FRR of our implementation, we first present our implementation of the *KeyGen* algorithm. We modified the original implementation of *KeyGen* to correct a specific type of error commonly found in our implementation. RBTs on their own correct errors through quantization, while this provides a good starting point for error correction it was found that the observed FRR was still very high even when the quantization ranges were quite wide. Deeper inspection of the problem revealed that quite often the same feature in subsequent readings of the same utterance were found to be misaligned. An example of the segmented samples for user $M1$ is shown in **Table 5.1**.

Let us examine the values for segment s_2. We can easily see that the expected value, **(15,14)**, is misaligned and appears in the segment s_3 in *Sample 2*. This type of error was extremely common in our implementation. Fortunately, the errors were quite easy to correct. We modified the *KeyGen* algorithm to calculate the values x_i, x_{i-2}, and x_{i+2} from ϕ_i, ϕ_{i-2}, and ϕ_{i+2} respectively (we

Sample Id	s_1	s_2	s_3	s_4	s_5	s_6
Sample 1	01,40	**14,15**	31,16	18,33	27,00	26,24
Sample 2	11,27	01,41	**15,14**	30,16	27,04	27,25
Sample 3	01,41	**15,13**	31,18	19,33	27,01	29,23
Sample 4	02,41	**14,14**	30,17	18,33	27,00	29,23

Table 5.1: Segmented samples for user $M1$ and utterance zero.

calculate x_{i-2} and x_{i+2} instead of x_{i-1} and x_{i+1} because the values x_{i-2} and x_{i+2} contain the quantized value of the correct x or y component). Our modified version of *KeyGen* works as follows. Let us assume we are generating a cryptographic key from *Sample 2*. Let us also assume that the x-component of s_2 has been encoded in user $M1$'s template. To correct the error found in *Sample 2* we attempt to generate the user's true cryptographic key by creating three keys. Instead of simply generating one cryptographic key using the x-component of s_2, we generated three cryptographic keys each with the values from the x-component of s_1, s_2, and s_3 respectively. In our example, the keys generated from *Sample 2* using the x-component of s_1 and s_2 would result in keys that do not match the template's verification hash. However, the key generated using the x-component of s_3 would result in a key that does match the verification hash in the user's template.

Although our algorithm does correct misaligned features it generates an impractical number of keys for a large template. Let us assume that a template contains 30 *reliable features*, i.e. $|\Psi| = 30$. In this case our algorithm generates upwards of 3^{30} keys per invocation of *KeyGen*! This is clearly not practical. A much smarter and bounded algorithm was required. In order to recreate the user's cryptographic key while achieving a lower FRR in a

practical way we modified the original *KeyGen* algorithm to perform the following steps. For each feature, i, its quantized value, x_i, is calculated. Along with x_i, both x_{i-2} and x_{i+2} are also calculated from both s_{i-1} and s_{i+1} respectively. A set, X_i, is created containing the **distinct** values from the set $\{x_{i-2}, x_i, x_{i+2}\}$. Then for each iteration $j \in [1, n]$ a set of keys is generated containing all possible combinations of the values in each of the sets, X_i, for $i \in [1, j]$, if and only if the number of keys generated is bounded above by τ. If the number of combinations exceeds τ, a different approach is taken. A calculation is made to determine how many possible features can be aligned before the number of combinations exceeds τ. This is determined by iteratively increasing the number of features aligned, d, until the following value exceeds τ:

$$|X_{max}|^d \binom{j}{d} \tag{5.8}$$

where X_{max} represents the set X_i with the highest cardinality. This is obviously a worst case estimate; however, it allows us to calculate how many corrections can be made while staying under the threshold, τ. Theoretically as j increases the value for d obviously cannot grow to a very large number. However, empirically it was found that a large number of features produced a set X_i that contained only one or two elements. Additionally, it was found to be extremely rare that $|X_i| > 2$. This fact was leveraged by our algorithm to create many different keys before determining that *KeyGen* had failed.

Our modified version of *KeyGen* was used to evaluate the FRR of our implementation. A sufficiently small value for τ was used to ensure a timely response from *KeyGen*. However, to show that our scheme is resilient to

imposters, the FAR reported in this study was calculated by assuming that the maximum number of errors that *KeyGen* could correct was unbounded. We report our FAR assuming *KeyGen* was able to try every possible combination of the previous and next feature in the imposter's utterance. In our evaluation of the FRR however, the threshold value τ, was set to two million to coincide with the number of attempts that Monrose et al. made to generate the correct key in their research. By bounding the number of attempts, our algorithm was constrained to ensure that our *KeyGen* implementation performed the same number of attempts (in terms of magnitude) as that of previous work.

It should be noted that the FAR we report is an absolute upper bound, while the FRR we report is the observed FRR of our implementation. The FAR reported assumes an unbounded value for the threshold τ leading to the worst case value for the FAR of our implementation. We report the upper bound for the FAR to show that our modified *KeyGen* algorithm does not negatively affect the security requirement REQ-BUN.

5.3.3 Experimental methodology

Our experiment was setup as follows. We used a pre-existing data set, the TI46 [69] database, to simulate a group of users uttering the sequence of numbers "zero" through "nine" as their passphrase. The TI46 database contained speech samples from each user uttering all the words of the alphabet, the numbers zero through nine, and ten other words. In our study, however, we only used each of the users' 26 utterances of the numbers, resulting in 26 different versions of the passphrase for each user. We partitioned the speech

samples into enrollment samples and test samples in a 4:1 ratio. The enrollment samples were used to create our SOM, as well as each of the user's templates while the test samples were used to test the FRR of our implementation. Additionally, the FAR of our implementation was tested with both the enrollment samples and test samples.

The original TI46 data set consisted of 16 users, eight women and eight men, however, through the use of pitch/time modifications the set of 16 users was grown to 64 [62]. Adobe® Audition® 3 [4] was used to perform the pitch/time modifications. Three different scaling factors were used for the group of men and women and were chosen to ensure the resulting voices still sounded human. The pitch/time modifications performed on our original data set however, were independent of our implementation and are not a required step in our algorithm. To ensure that the we created new users that had human sounding voices we listened to the synthetic speech samples we created to confirm that they sounded human and that they were audibly different from the original speaker.

As part of our algorithm we used traditional signal processing techniques to clean up the samples from the TI46 database. Each speech sample in the data set was amplified and had frames of silence removed from it. Again, Adobe® Audition® 3 was used to perform the signal processing tasks because of its batch scripting capability. Removing silence is a required step in our algorithm. It ensures the speech signals being processed by our feature extraction algorithm contain the most feature rich segments. We found that this step was integral to the success of applying RBTs to voice. Without silence detection a large number of segments were mapped to the same value because no distinguishing features could be extracted from frames containing

no speech.

After processing the speech samples with Adobe® Audition®, the speech samples were further processed individually by the Hidden Markov Model Toolkit (HTK) [109] to extract PLP and delta coefficients. HTK was used to generate each of the 24-dimensional feature vectors that were subsequently used as input to the training phase of our SOM. It should be noted here that while HTK does offer PLP feature extraction it warps the power spectrum according to the Mel scale as opposed to the Bark scale. HTK uses **Equation 5.9** in place of **Equation 5.1**.

$$Mel(f) = 2595log_{10}(1 + \frac{f}{100}) \tag{5.9}$$

where the frequency, f, can be converted to the angular frequency, w, by the equation $f = \frac{w}{2\pi}$. Other practical subtleties exist in HTK's implementation of PLP feature extraction; please see [108]. In addition to using the HTK library to perform both PLP and delta analysis, the HTK library was configured to perform *Cepstral mean normalization* (CMN). CMN is a technique used to compensate for undesirable spectral effects caused by imperfections in audio recording. Although traditionally used when dealing with lengthy samples of speech it was employed in this study.

As previously described, a SOM was used to model the speech samples of our population. All the speech samples were processed by HTK and the enrollment samples were used to train our SOM. The SOM_PAK [66] toolkit was used to train a 48x48 SOM, organized in a hexagon topology. The SOM was initialized with the eigenvectors of the enrollment data to better seed the initial weight vectors of the nodes in the SOM. The SOM was then trained

using a learning rate of 0.1 and a radius of 2 while 10,000,000 iterations were used to allow the nodes to converge to an acceptable cumulative error. A number of varying radii and learning rates were also tested in our study by first creating the SOM with the varying parameters and then testing the cumulative error. The cumulative error was calculated by taking the sum of the cumulative distances between each enrollment vector and its associated BMU. It was found that a lower learning rate and lower radius performed best with our data set.

Once the SOM was created, each sequence of feature vectors was segmented into six segments by the segmentation algorithm discussed in section 5.2.2. Of the six segments, the first and the last segments were not used as *reliable features*. These two segments were discarded because they showed a very high variance in the nodes they matched. This was primarily due to the imperfections in the silence detection tool used during the signal processing stage of our experiment. Since the first and last segments were discarded, however, only four segments were used as possible features for each utterance. Since each passphrase contained the utterances "zero" through "nine", a total of ten utterances were used. Each utterance was segmented into four segments, and each segment was mapped to two features, resulting in 80 possible features per passphrase. It should also be noted that although the first and last segments were discarded as *reliable features*, they were used to align features in our modified *KeyGen* algorithm.

5.3.4 Experimental results

An important algorithm used within the *Enroll* algorithm has not yet been discussed. In **Figure 5.1**, on line 1, the function **Select** is called by *Enroll*. While the details of **Select** are somewhat complicated they must be explained in some detail in order to properly understand the results of our experiment. Further details about the **Select** algorithm can be found in [9]. The **Select algorithm performs two distinct tasks. The first is to calculate** the individual feature quantization widths, δ_i, by taking into account statistics across the entire population. For each feature value, ϕ_i, each user, u, is tested to see how much error tolerance, $d_{u,i}$, they require to replicate ϕ_i reliably. Each of the $d_{u,i}$ values are ordered such that $d_{u,i} \leq d_{u',i}$. The $d_{u,i}$ value at the k^{th} percentile is then taken to be the value for δ_i. After the **Select** algorithm has determined all the δ_i values, it assigns features to users. To do so, each user's $d_{u,i}$ value is tested to see if $d_{u,i} \leq \delta_i$. If their value falls below δ_i the feature i is added to their set of *reliable features*. As expected, as we increased the value for k, more and more features were added to the RBT generated templates. **Figure 5.7** shows the cumulative distribution function (CDF) of the number of features encoded in each user's template, i.e. the number of *reliable features*, $|\Psi|$, for varying levels of k.

In applying RBTs to voice biometrics we found that the original **Select** algorithm generated values for δ_i that were so large that the *Enroll* algorithm quantized a large number of users to the same x_i value. In order to create **stricter quantization widths we modified the Select** algorithm to calculate a smaller value for each user's $d_{u,i}$ value by imposing a threshold. To create each user's template, their $d_{u,i}$ value was determined as follows. For each enrollment sample, β_i, a value for $\phi_i(\beta_i)$ is calculated. The values for $\phi_i(\beta_i)$

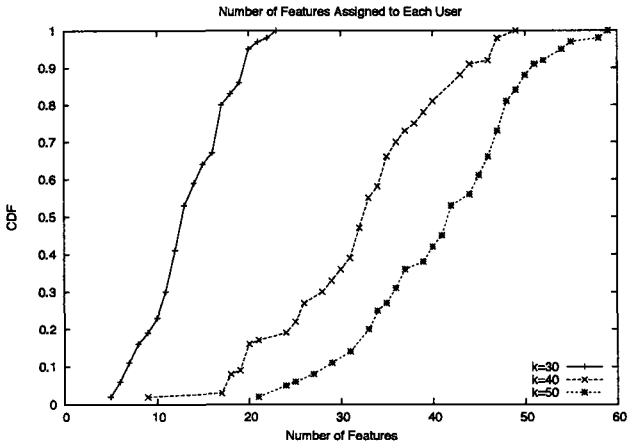

Figure 5.7: CDF of the number of *reliable features* in each user's template, i.e. the size of Ψ, for varying levels of k.

are reordered such that $\phi_i(\beta_i) \leq \phi_i(\beta_{i+1})$, μ_i is set to the mean of the $\phi_i(\beta_i)$'s, and $t = max(\mu_i - \phi_i(\beta_1), \mu_i - \phi_i(\beta_l))$. The value for $d_{u,i}$ is then set to $2t$. It was found however, that in our experiment this algorithm produced extremely large values for $d_{u,i}$ since a number of features were not aligned, causing the range of values for a given feature to be quite broad. To account for this problem a threshold was employed to ensure that values for $\phi_i(\beta_i)$ that were far from μ_i were discarded. In our experiments two thresholds were used. The first threshold we used was the 99% threshold, which allowed $d_{u,i}$ to be calculated while discarding the reading $\phi_i(\beta_i)$ that was furthest away from μ_i. The second threshold that we used was the 90% threshold. At the 90% threshold, the values, $\phi_i(\beta_i)$, that were in the furthest 10^{th} percent away from μ_i were discarded from the calculation of $d_{u,i}$.

In addition to modifying the **Select** algorithm we also had to ensure that features were assigned to user's in both a uniform and independent manner

in order for RBTs to produce cryptographic keys with the most entropy. By assigning features to users whose $d_{u,i}$ value is in the top k^{th} percentile, the RBT algorithm ensures that features are assigned to users uniformly across the population. This however, does not ensure that two features, ϕ_i and ϕ_j, are assigned to users independently of one another. In order to ensure features are assigned to users independently of one another, another algorithm was required, the details of the algorithm follow. Let I_i be the indicator random variable for feature i. For each user $u \in U$, determine if u is able to recreate feature i (i.e. their value for $d_{u,i} \leq \delta_i$). If $d_{u,i} \leq \delta_i$ then set $I_i = 1$, otherwise set $I_i = 0$. Once each feature and each user has been tested the correlation constant ρ_{I_i,I_j} of the indicator random variables I_i and I_j is calculated using the following equation:

$$\rho_{I_i,I_j} = \frac{E((I_i - \mu_{I_i})(I_j - \mu_{I_j}))}{\sigma_{I_i}\sigma_{I_j}} \tag{5.10}$$

Where $-1 \leq \rho_{I_i,I_j} \leq 1$. Traditionally two random variables, X, and Y, are said to be uncorrelated if $|\rho_{X,Y}| < \frac{1}{3}$, correlated if $\frac{1}{3} \leq |\rho_{X,Y}| < \frac{2}{3}$, and strongly correlated if $|\rho_{X,Y}| \geq \frac{2}{3}$. In our implementation, a greedy algorithm was used to discard features that were correlated with a large number of other features, which was determined by checking if $|\rho_{I_i,I_j}| > \tau$, where the value τ, depended on the value k. In order to ensure that features were being assigned to users both independently and uniformly, χ^2-squared tests of both Independence and Homogeneity were performed [91] at the confidence level $\alpha = 0.99$. For lower values of k, a lower value of τ was used. For k values of 30, 40 and 50, τ values of 0.3, 0.6, and 0.6 were used respectively. We ran the the χ^2-squared tests against templates generated with both the 90% and 99% threshold values to ensure we assigned features to user's independently

and uniformly. In both cases we did not reject the Null hypothesis.

Removing features that are strongly correlated with one another was a very important step in our implementation. By construction, each passphrase produced 80 features, however, some features were originally observed to be strongly correlated with one another. Logically, this was expected as both the x and y coordinates of each BMU were used as separate features. However, in evaluating our feature selection algorithm we found that even though **some** x and y coordinates did exhibit a strong correlation to one another the resulting number of *reliable features* generated by our feature selection algorithm was still quite a bit higher than other feature selection algorithms we tried, i.e. mapping each feature $\phi_i = (x, y)$, even after we removed the strongly correlated features. Mapping each coordinate to a separate feature resulted in cryptographic keys with more entropy than keys generated when we mapped each feature to the two-dimensional coordinate of each BMU because of the increased number of *reliable features.*

In our evaluation we empirically evaluated our implementation's ability to satisfy both REQ-BUN and REQ-KR. In order to evaluate REQ-BUN and the accuracy of our proposed feature extraction algorithm, the FAR and FRR of our implementation were calculated. Both metrics were calculated using a repeated leave-out-k cross validation algorithm. Given v samples, we randomly chose $v - k$ samples to create the RBTs and k samples to test the FRR of our implementation. $v - k$ and k were set to be in the ratio 4:1. Additionally, each template created had the maximum number of *reliable features* in it capped at $|\Psi| = 25$, $|\Psi| = 50$, and $|\Psi| = 60$ for $k = 30$, $k = 40$, and $k = 50$ respectively. This ensured that the strongest possible templates were created for each user for varying levels of k.

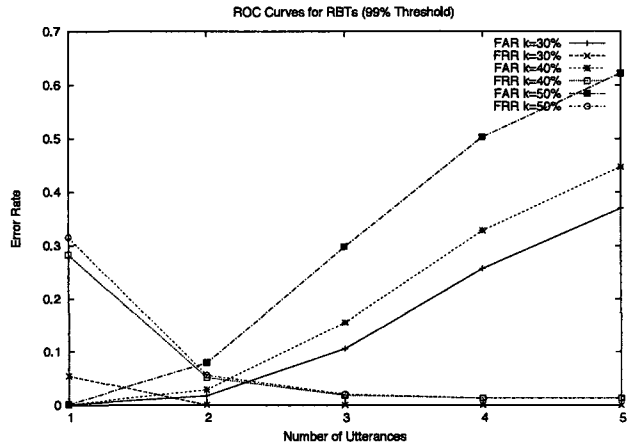

Figure 5.8: ROC for varying levels of k with a 99% threshold.

In our implementation, the FAR was calculated by allowing *KeyGen* to create keys using all possible combinations of the x_{i-2}, x_i, and x_{i+2} values for each feature. The FAR therefore reports the percentage of imposter samples that were able to recreate a legitimate user's key even if a large number of misaligned features needed to be corrected and therefore represents an upper bound. To test the FRR of our implementation, however, the number of corrected features was experimentally calculated to ensure we did not attempt to create more keys than previous work did in their tests. Interestingly, for $k = 30$ all combinations of feature values were able to be tested since the templates for each individual contained only a small number of features. As the value for k was increased however, fewer features could be corrected, eventually only allowing three features to be corrected for some templates when $k = 50$.

As expected, as the threshold value decreased the value for $d_{u,i}$ decreased, causing the quantization width, δ_i, to decrease, thus causing the FRR to

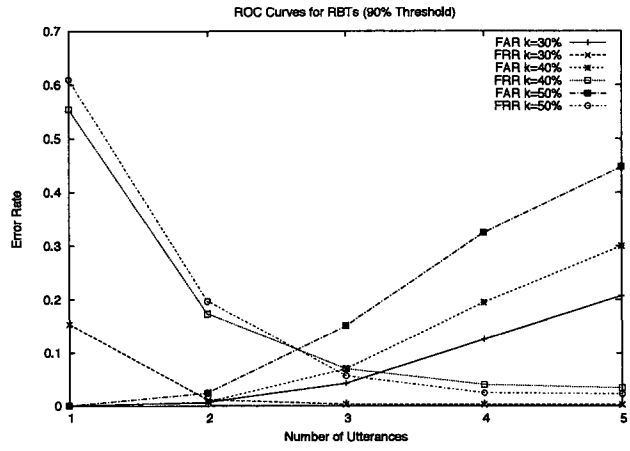

Figure 5.9: ROC for varying levels of k with a 90% threshold.

increase. At higher levels of k, a greater number of features were used to create each of the user's templates, therefore requiring more features to be aligned during *KeyGen*, thus causing the FRR of our implementation to increase. The FRR and FAR of our implementation are shown in **Figure 5.8**, and **Figure 5.9** for the threshold values 99% and 90% respectively. It can be seen that at the 99% and 90% threshold if a user utters their passphrase two and three times respectively a FRR of less than 10% can be achieved. One very interesting observation is that the upper bound FAR at one and two utterances is very low for both the 99% and 90% threshold, thus satisfying the requirement of REQ-BUN.

The most important metric reported in this study is the amount of entropy in the keys generated by our implementation. In order to compare our implementation against a baseline, we implemented the algorithm of Monrose, et al., [74] as a means to compare our implementation to that of prior work. Monrose, et al.'s algorithm was implemented using PLP and delta coeffi-

cients as their features to ensure we made a fair comparison between the two implementations. In our implementation of Monrose, et al.'s algorithm, we were able to achieve FRR and FAR similar to their documented results. Additionally, in order to quantify the entropy in both our keys as well as the keys of Monrose, et al. we implemented the *Guessing Distance* estimation algorithm as detailed in Ballard's dissertation [7]. The results are shown in **Figure 5.10** and **Figure 5.11** for threshold values 99% and 90% respectively, where the baseline is the observed entropy in our implementation of Monrose, et al.'s BKG as documented in [74].

It should be noted that the entropy reported in this analysis assumes an attacker has access to the **decrypted** version of a user's template and therefore does not account for the amount of entropy in the user's password, π. The entropy reported in this study is therefore an absolute lower bound. Using an encrypted template as the input to the entropy calculation would have increased the resulting amount of entropy by the amount of entropy in the user's password. However, we chose to ignore the entropy in π and focus on the entropy generated by the voice biometric in isolation.

In our evaluation of the entropy in the RBT generated cryptographic keys, we constructed the sets Ω used by the *Guessing Distance* calculation out of the quantized values of the observed feature outputs for each feature. In **Figure 5.4**, a high degree of clustering can be seen in certain sections of our SOM. To account for this we assumed our adversary had access to feature sets containing the quantized values, x_i, for each feature as oppose to the raw feature values $\phi_i(\beta)$. By accounting for the amount of clustering in our SOM, the number of elements in Ω was greatly reduced, allowing our adversary to use a more reasonable guessing strategy given the topology of our SOM.

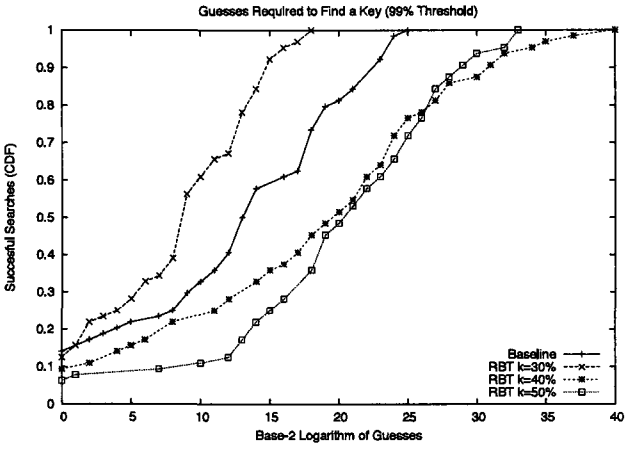

Figure 5.10: CDF of the number of guesses required by Ballard et al.'s search algorithm before finding a user's RBT-derived key at 99% threshold.

The observed distribution of the feature values in Ω was non-uniform, thus allowing our adversary to guess some keys very easily.

In **Figure 5.10** and **Figure 5.11** it can be observed that at $k = 30$, the templates created by the RBT algorithm produced weaker keys than that of previous work. This was due to the fact that at $k = 30$ the templates contained a very low number of *reliable features*, with 50% of the population assigned $|\Psi| = 12$ or fewer features. At higher levels of k however, users were assigned a much larger number of features with 50% of the users being assigned over $|\Psi| = 30$ and $|\Psi| = 40$ features for $k = 40$ and $k = 50$ respectively. The added number of features increased the entropy of the keys generated by the RBT algorithm. At both threshold levels, RBTs outperformed the BKG of Monrose, et al. by assigning keys that required at least 2^{30} guesses to 36% and 13% of the population at the 90% threshold and 99% threshold respectively. The algorithm of Monrose, et al. however, assigned

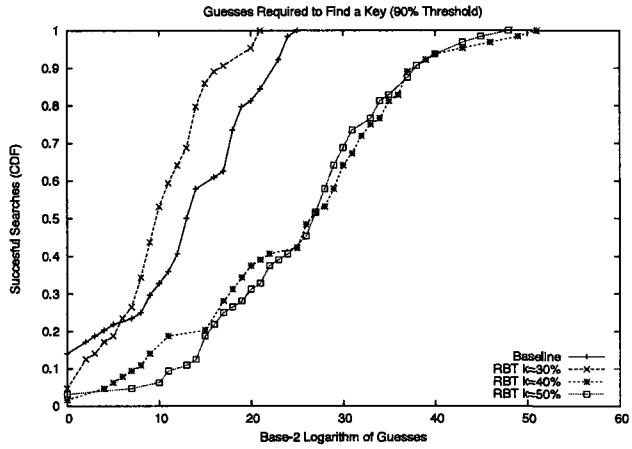

Figure 5.11: CDF of the number of guesses required by Ballard et al.'s search algorithm before finding a user's RBT-derived key at 90% threshold.

keys that required at least 2^{20} guesses to only 19% of the population. Even more encouraging was the fact that for 7% of the population, keys requiring at least 2^{40} guesses were created using the RBT algorithm. The RBT algorithm was also able to generate keys with a maximum entropy of 51 bits at the 90% threshold compared to a maximum entropy of 26 bits for keys generated by Monrose, et al.'s algorithm. By empirically evaluating the entropy in our RBT generated keys, we show that our implementation indeed meets the requirement REQ-KR. We have shown that the keys generated using RBTs are able to achieve much more entropy than the keys generated from the algorithms of prior work.

Although not empirically evaluated we make the argument that RBTs inherently satisfy REQ-SBP. Our feature extraction algorithm along with the RBT algorithm quantize the original biometric signal at many different levels. We argue that being able to go from the global δ_i values and α_i values

in individual templates to the original biometric signal of the template's creator is a very difficult, if not impossible task for a computationally bounded adversary. Not only is each user's biometric sample segmented, and then quantized by our SOM, but many users have their features mapped to the same repeatable value, x_i, which is never stored in their template.

In this section we showed that RBTs can be used to reliably generate a high-entropy bitstring, a biometric key, from voice biometrics. The resulting biometric key can be used as the values b_I and b_F discussed in the previous chapter to achieve non-transferability. By using biometrics, as opposed to username and passwords, users of our digital identity system are prevented from sharing their digital credentials with one another because their biometric is encoded in their digital credential. Our study is a promising first step in applying RBTs to voice biometrics. Although this is a preliminary result, we are confident that it will extend to a larger set of voice samples. We hypothesize that as the number of users in the system grow, more entropy-rich keys can be generated because the feature value distributions should start to approach a more uniform distribution. In the next chapter we conclude and lay out what future work needs to be done in both the area of digital identity management systems as well as biometric key generators.

Chapter 6

Conclusion

In this work we extend a digital credential scheme to support biometric non-transferability and apply our digital credential scheme to online digital identity management. Additionally, we show that current online identity management schemes are insufficient in providing privacy-preserving services to their users. Furthermore, we show that the credential scheme of Persiano and Visconti can be modified to include both non-transferability and proxy support. Non-transferability prevents users from sharing their digital credentials, while proxy support allows users to off-load the intense calculations required by digital credential schemes. Our proxy extension is a required property to allow digital credentials to be used ubiquitously on low-power devices such as handheld PDAs.

In order to achieve non-transferability we performed an experiment by applying a biometric key generation algorithm, randomized biometric templates (RBTs), to voice biometrics. In this work we show that higher entropy keys are achievable by applying a modified version of randomized biometric tem-

plates to voice biometrics. By using our novel feature extraction algorithm we show that it is possible to apply RBTs to voice biometrics in order to achieve higher entropy keys. We also show that an adversary given access to *auxiliary information* is still required to perform an increased number of guesses for a high percentage of the population before guessing a correct key. We compare the entropy in our RBT generated keys to that of previous work and show that our system is able to generate keys with at least 30 and 40 bits of entropy for 36% and 7% of the population respectively, while those of previous work are only able to achieve 20 bits of entropy for 19% of the population. We also show that RBT generated keys are able to achieve a maximum entropy of 51 bits, while keys generated from the algorithms of previous work are only able to achieve a maximum entropy of 26 bits. We also show that acceptable levels of FRR and FAR are achievable and describe, in the next section, how both the amount of entropy and the FRR can be improved by gathering more feature rich utterances from a more diverse population of users.

6.1 Future Work

6.1.1 Non-Transferability

Now that we have outlined the theoretical mathematics behind achieving a non-transferable digital credential scheme that supports both selective disclosure and unlinkable transactions it should be implemented and tested. There are a number of tests that should be run once our system is realized. The first test is an Engineering test that requires testing our proposed system on

low-power devices to determine how efficiently the protocols run. It is the hope that the *Partial Trust* policy, which off-loads the intense computations of the digital credential scheme to the IdM, can be used to effectively reduce the number of computations that must be performed on low-power devices.

Additionally, a user trial is required. Although we have designed a policy-based digital identity management system that includes non-transferability and proxy support we must determine how effectively it is used in the real-world. A thorough test would extend one of the open digital identity management protocols, such as OpenId, to support our non-transferable digital credential scheme and have sample users attempt to use it to determine if our system is usable in the real-world.

Lastly, this research should be seriously considered for adoption by online digital identity management protocols. What we present here greatly enhances a user's privacy with respect to their transactions online and gives them ultimate control over how their information is disclosed. In addition to our system protecting the privacy of users, it also provides assurance to relying parties that the users they are authenticating are the users themselves through biometric authentication.

6.1.2 Biometric Key Generators

The results from our biometrics experiment are definitely a step forward in creating cryptographic keys from voice biometrics. However, they are a preliminary result and there are a number of areas that require further research before a practical scheme can be achieved. In particular, a number of opportunities exist to potentially increase the entropy of the generated cryp-

tographic keys as well as decrease the FRR and FAR of our implementation.

In order to achieve more entropy there are two primary ways forward. The first is to gather speech samples from more users. As more diverse samples are gathered, the feature values extracted using RBTs will hopefully start to approach a more uniform distribution, thus reducing the advantage an adversary has in guessing the values for x_i. By choosing a diverse array of users, from varying backgrounds and both sexes it is believed that better feature value distributions can be achieved. In that same vein, more feature rich speech samples are required to increase the number of *reliable features*. In our test, users simply uttered the numerical values "zero" through "nine" as their passphrase. In order to generate keys with more entropy, more features are required and thus more feature-rich utterances are needed. A more comprehensive test would have a diverse array of users, preferably in the hundreds, being prompted to say feature-rich words in order to achieve a more practical amount of entropy.

Achieving better FRR and FAR is also critical to the adoption of generating cryptographic keys from voice biometrics. To achieve better results a more flexible model allowing for the alignment of features is required. What is promising is that if we assume more and more features can be aligned, extremely good FRR results can be achieved. **Figure 6.1** shows the results at the 90% threshold and $k = 50$ when the number of errors corrected is set to 2,000,000, 20,000,000, 200,000,000, 2,000,000,000 and ultimately unlimited. What is promising about **Figure 6.1** is that it is theoretically possible to achieve an extremely good FRR $\approx 0\%$ with two utterances at $k = 50$ at the 90% threshold. A FRR of 11% with one utterance could also be acceptable if very high rates of entropy can be achieved.

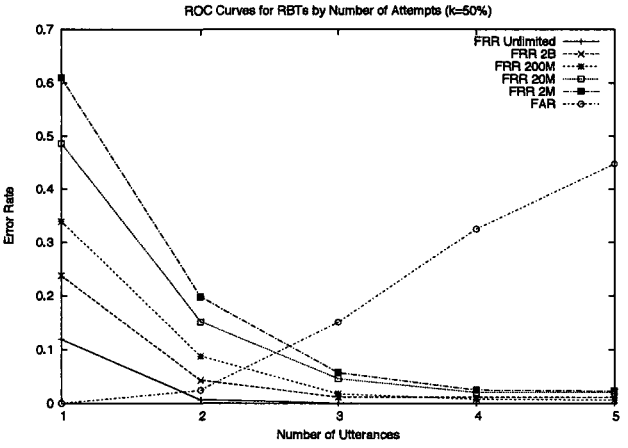

Figure 6.1: ROC for $k = 50$, 90% threshold and varying levels of errors corrected.

Bibliography

[1] `http://www.cellular-news.com/story/33811.php`, September 2008.

[2] C. Adams. Achieving non-transferability in credential systems using hidden biometrics. *Security and Communication Networks (to appear)*, 2009.

[3] A. Adler. Vulnerabilities in biometric encryption systems. *Audio and Video Based Biometric Person Authentication*, 3546:1100–1109, 2005.

[4] Adobe. Adobe audition 3. http://www.adobe.com/products/audition/, 2009.

[5] A. Alvare. How crackers crack passwords or what passwords to avoid. *Proceedings of the Second USENIX Security Workshop*, pages 103–112, 1990.

[6] G. Ateniese, J. Camenish, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. *Advances in Cryptology Crypto 2000*, 1880:255–270, 2000.

[7] L. Ballard. *Robust Techniques for Evaluating Biometric Cryptographic Key Generators (PhD thesis)*. The Johns Hopkins University, 2008.

[8] L. Ballard, S. Kamara, F. Monrose, and M. Reiter. On the requirements of biometric key generators. *Technical report - Whiting School of Engineering - The Johns Hopkins University*, 2007.

[9] L. Ballard, S. Kamara, F. Monrose, and M. K. Reiter. Towards practical biometric key generation with randomized biometric templates. *Proceedings of the 15th ACM Conference on Computer and Communications Security*, pages 235–244, 2008.

[10] L. Ballard, S. Kamara, and M. K. Reiter. The practical subtleties of biometric key generation. *Proceedings of the 17th annual USENIX Security Symposium*, pages 29–41, 2006.

[11] L. Ballard, F. Monrose, and D. Lopresti. Biometric authentication revisited: Understanding the impact of wolves in sheeps clothing. *Proceedings of the 15TH USENIX Security Symposium*, pages 29–41, 2006.

[12] R. Bellman. On the approximation of curves by line segments using dynamic programming. *Communications of the ACM*, 4(6):284, 1961.

[13] M. Blanton. Online subscriptions with anonymous access. *ASIAN ACM Symposium on Information, Computer and Communications Security*, pages 217–227, 2008.

[14] G. Bleumer. Biometric yet privacy protecting person authentication. *Information Hiding*, 1525:99–110, 1998.

[15] T. Boult. Robust distance measures for face-recognition supporting revocable biometric tokens. *International Conference on Automatic Face and Gesture Recognition*, pages 560–566, 2006.

[16] T. E. Boult and R. Woodworth. Privacy and security enhancements in biometrics. *Advances in Biometrics*, pages 423–445, 2007.

[17] X. Boyen. Reusable cryptographic fuzzy extractors. *11th ACM Conference on Computing and Communications Security*, pages 82–91, 2004.

[18] S. Brands. ***Rethinking Public Key Infrastructures and Digital Certificates.*** MIT Press, 2000.

[19] S. Brands. A technical overview of digital credentials. Credentica.com, 2002.

[20] J. Bringer, H. Chabanne, and B. Kindarji. The best of both worlds: Applying secure sketches to cancelable biometrics. *Science of Computer Programming*, 74(1-2):43–51, 2008.

[21] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich. How to win the clone wars: Efficient periodic n-times anonymous authentication. *Proceedings of the 13th ACM conference on Computer and communications security*, pages 201–210, 2006.

[22] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. *Advances in Cryptology Eurocrypt 2001*, 2045:93–118, 2001.

[23] J. Camenisch and A. Lysyanskaya. An identity escrow scheme with appointed verifiers. *Advances in Cryptology Crypto 2001*, 2139:388–407, 2001.

[24] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. *Advances in Cryptology Crypto 2004*, 3152:56–72, 2004.

[25] J. Camenisch, D. Sommer, and R. Zimmerman. A general certification framework with applications to privacy-enhancing certificate infrastructures. *Security and Privacy in Dynamic Environments*, 201:25–37, 2006.

[26] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. *Advances in Cryptology Crypto 97*, 1294, 1997.

[27] J. Camenisch and E. van Herreweghen. Design and implementation of the idemix anonymous credential system. *Conference on Computer and Communications Security*, pages 21–30, 2002.

[28] S. Carmody, M. Erdos, K. Hazelton, W. Hoehn, B. Morgan, T. Scavo, and D. Wasley. Shibboleth architecture. `http://shibboleth.internet2.edu/docs/internet2-mace-shibboleth-arch-protocols-200509.pdf`, September 2005.

[29] D. Chappell. Introducing windows cardspace. `http://msdn.microsoft.com/en-us/library/aa480189(loband).aspx`, April 2006.

[30] D. Chaum. Blind signatures for untraceable payments. *Advances in Cryptology*, pages 199–203, 1982.

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

[32] D. Chaum. Showing credentials without identification: Transferring signatures between unconditional unlinkable pseudonyms. *Proceedings*

of the international conference on cryptology on Advances in Cryptology, pages 246–264, 1990.

[33] D. Chaum and J. Evertse. A secure and privacy-protecting protocol for transmitting personal information between organizations. *Advances in Cryptology Crypto '86*, 263:118–167, 1987.

[34] D. Chaum and T. Pedersen. Wallet databases with observers. *Advances in Cryptology Crypto 92*, 740:89–105, 1993.

[35] D. Chaum and E. van Heyst. Group signatures. *Advances in Cryptology Eurocrypt 91*, 547:257–265, 1991.

[36] L. Chen. Access with pseudonyms. *Cryptography: Policy and Algorithms*, pages 232–243, 1996.

[37] L. Chen, A. N. Escalante B., H. Lohr, M. Manulis, and A. Sadeghi. A privacy-protecting multi-coupon scheme with stronger protection against splitting. *Financial Cryptography and Data Security*, 4886:29–44, 2008.

[38] R. Cramer and V. Shoup. Signature schemes based on the strong rsa assumption. *ACM Transactions on Information and System Security*, 3(3):161–185, 2000.

[39] R. J. F. Cramer and T. P. Pedersen. Improved privacy in wallets with observers. *Advances in Cryptology Eurocrypt 93*, 765:329–343, 1994.

[40] Russ Cutler. Liberty identity assurance framework. `http://www.projectliberty.org/liberty/content/download/4315/28869/file/liberty-identity-assurance-framework-v1.1.pdf`, 2008.

[41] I. B. Damgard. Payment systems and credential mechanisms with provable security against abuse by individuals. *Advances in Cryptology Crypto '88*, 403:328–335, 1990.

[42] G. I. Davida, Y. Frankel, and B. J. Matt. On enabling secure applications through off-line biometric identification. *IEEE Symposium on Security and Privacy*, pages 148–157, 1998.

[43] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *Advances in Cryptology EuroCrypt 2004*, 3027:523–540, 2004.

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

[45] H. Feng and C. C. Wah. Private key generation from on-line handwritten signatures. *Information Management and Computer Security*, 10(4):159–164, 2002.

[46] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. *Advances in Cryptology Crypto 86*, 263:186–194, 1987.

[47] D. Fieldmeier and P. Karn. Unix password security - ten years later. *Proceedings of Advances in Cryptology - CRYPTO*, pages 44–63, 1990.

[48] OpenID Foundation. `http://openid.net/get-an-openid/what-is-openid/`, 2009.

[49] A. Goh and C. L. Ngo. Computation of cryptographic keys from face biometrics. *Communications and Multimedia Security*, 2828:1–13, 2003.

[50] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[51] J. Golic and M. Baltatu. Entropy analysis and new constructions of biometric key generation systems. *IEEE Transactions on Information Theory*, 54(5):2026–2040, 2008.

[52] Miniwatts Marketing Group. `http://www.internetworldstats.com/stats.htm`, 2009.

[53] F. Hao, R. Anderson, and J. Daugman. Combining cryptography with biometrics effectively. *IEEE Transactions on Computers*, 55(9):1081–1088, 2006.

[54] D. Hardt, J. Bufu, Sxip Identity, J. Hoyt, and JanRain. Openid attribute exchange 1.0 - final. `http://openid.net/specs/openid-attribute-exchange-1_0.html`, December 2007.

[55] H. Hermansky. Perceptual linear predictive (plp) analysis of speech. *Journal of Acoustic Society of America*, 87(4):1738–1752, 1990.

[56] T. Ignatenko and F. Willems. On privacy in secure biometric authentication systems. *IEEE Conference on Acoustics, Speech, and Signal Processing*, 2:121–124, 2007.

[57] R. Impagliazzo and S. M. More. Anonymous credentials with biometrically-enforced non-transferability. *Proceedings of the 2003 ACM workshop on Privacy in the electronic society*, pages 60–71, 2003.

[58] A. T. B. Jin, D. N. C. Ling, and A. Goh. Biohashing: two factor authentication featuring fingerprint data and tokenized random number. *Pattern Recognition*, 37(11):2245–2255, 2004.

[59] J. R. Deller Jr., J. H. L. Hansen, and J. G. Proakis. *Discrte-Time Processing of Speech Signals.* Macmillan Publishing Co., 1993.

[60] A. Juels and M. Sudan. A fuzzy vault scheme. *Design, Codes and Cryptography*, 38(2):237–257, 2006.

[61] A. Juels and M. Wattenberg. A fuzzy commitment scheme. *6th ACM Conference on Computer and Communications Security*, pages 28–36, 1999.

[62] M. Kahrs and K. Brandenburg. *Applications of Digital Signal Processing to Audio and Acoustics.* Kluwer Academic Publishers, 1998.

[63] J. Kang, D. Nyang, and K. Lee. Two factor face authentication scheme with cancelable feature. *Advances in Biometric Persons Authentication*, 3781:67–76, 2005.

[64] J. Kilian and E. Petrank. Identity escrow. *Advances in Cryptology Crypto 98*, 1462:169–185, 1998.

[65] T. Kohonen. *Self-organizing maps Third Edition.* Springer, 2001.

[66] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. Som_pak: The self-organizing map program package. *Technical Report A31, Helsinki University of Technology*, 1996.

[67] H. Lee, C. Lee, J-Y. Choi, J. Kim, and J. Kim. Changeable face representations suitable for human recognition. *Advances in Biometrics*, 4642:557–565, 2007.

[68] Q. Li, Y. Sutcu, and N. Memon. Secure sketch for biometric templates. *Advances in Cryptology AsiaCrypt 2006*, 4284:99–113, 2006.

[69] M. Liberman. *TI 46-Word.* Linguistic Data Consortium, 1993.

[70] A. Lipsman. comscore study reveals the number of online banking customers has grown 27 percent in the past year; fueled, in part, by adoption of online bill payment and growth within the high-yield savings marketplace. `http://www.docstoc.com/docs/3876742/comScore-Study-Reveals-The-Number-Of-Online-Banking-Customers-Has`, April 2006.

[71] A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. *Selected Areas of Cryptography*, 1758:184–1999, 1999.

[72] A. J. Menezes, P. C. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1997.

[73] F. Monrose, M. K. Reiter, Q. Li, D. P. Lopresti, and C. Shih. Toward speech-generated cryptographic keys on resource constrained devices. *Proceedings of the 11th USENIX Security Symposium*, pages 283–296, 2002.

[74] F. Monrose, M. K. Reiter, Q. Li, and S. Wetzel. Cryptographic key generation from voice. *IEEE Symposium on Security and Privacy*, pages 202–213, 2001.

[75] F. Monrose, M. K. Reiter, Q. Li, and S. Wetzel. Using voice to generate cryptographic keys. *ODYSSEY-2001*, pages 237–242, 2001.

[76] F. Monrose, M. K. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. *International Journal of Information Security*, 1(2):69–83, 2002.

[77] R. L. Morgan, S. Cantor, S. Carmody, W. Hoehn, and K. Klingenstein. Federated security: The shibboleth approach. `http://www.educause.edu/EDUCAUSE+Quarterly/EDUCAUSEQuarterlyMagazineVolum/FederatedSecurityTheShibboleth/157315`, 2004.

[78] K. Nandakumar, A. K. Jain, and S. Pankanti. Fingerprint-based fuzzy vault: Implementation and performance. *IEEE Transactions on Information Forensics and Security*, 2(4):744–757, 2007.

[79] M. Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991.

[80] T. Natsuno, P. Hoschka, and B. Prasad. `http://www.w3.org/2004/Talks/w3c10-WebOnEverything/?n=16`, 2004.

[81] L. Nguyen and R. Safavi-Naini. Dynamic k-times anonymous authentication. *Applied Cryptography and Network Security*, 3531:318–333, 2005.

[82] J. Owyang. A collection of social network stats for 2009. `http://www.web-strategist.com/blog/2009/01/11/a-collection-of-soical-network-stats-for-2009/`, January 2009.

[83] P. Persiano and I. Visconti. An anonymous credential system and a privacy-aware pki. *Information Security and Privacy*, 2727:27–38, 2003.

[84] P. Persiano and I. Visconti. An efficient and usable multi-show non-transferable anonymous credential system. *Financial Cryptography*, 3110:196–211, 2004.

[85] L. Rabiner and B-H. Juang. *Fundamentals of Speech Recognition.* Prentice Hall, 1993.

[86] N. Ragouzis, J. Hughes, R. Philpott, E. Maler, P. Madsen, and T. Scavo. Security assertion markup language (saml) v2.0 technical overview. `http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf`, March 2008.

[87] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):561–572, 2007.

[88] N. K. Ratha, J. H. Connell, and R. M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001.

[89] L. Reardon. The radicati group, inc. releases email statistics report, 2009–2013. `http://www.radicati.com/wp/wp-content/uploads/2009/05/e-mail-statistics-report-2009-pr.pdf`, May 2009.

[90] D. Recordon and D. Reed. Openid 2.0: A platform for user-centric identity management. *Proceedings of the 2nd ACM Workshop on Digital Identity Management*, pages 11–16, 2006.

[91] J. A. Rice. *Mathematical Statistics and Data Analysis.* Wadsworth, Inc, 1988.

[92] R. L. Rivest, A. Shamir, and L. Adelman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120– 126, 1978.

[93] J. Rompel. One-way functions are necessary and sufficient for secure signatures. *ACM Symposium on Theory of Computing*, pages 387–394, 1990.

[94] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18(5):401–409, 1969.

[95] M. Savvides, B.V.K.V. Kumar, and P.K. Khosla. Cancelable biometric filters for face recognition. *International Conference on Pattern Recognition*, 3:922–925, 2004.

[96] D. Shapiro, V. Thareja, and C. Adams. A privacy-preserving 3rd-party proxy for transactions that use digital credentials. *Proceedings of the 20th Annual IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 970–973, April 2007.

[97] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. K. V. Kumar. Biometric encryption. *ICSA Guide to Cryptography*, 1999.

[98] specs@openid.net. Openid authentication 2.0 - final. `http://openid.net/specs/openid-authentication-2_0.html`, December 2007.

[99] Y. Sutcu, H. T. Sencar, and N. Memon. A secure biometric authentication scheme based on robust hashing. *Proceedings on the 7th Workshop on Multimedia and Security*, pages 111–116, 2005.

[100] A. B. J. Teoh and C. T. Yuang. Cancelable biometrics realization with multispace random projections. *IEEE Transactions on Systems, Man, and Cybernetics*, 37(5):1096–1106, 2007.

[101] I. Teranishi, J. Furukawa, and K. Sako. k-times anonmyous authentication. *Advances in Cryptology AsiaCrypt 2004*, 2239:308–322, 2004.

[102] S. Tulyakov, F. Farooq, and V. Govindaraju. Symmetric hash functions for fingerprint minutiae. *Pattern Recognition and Image Analysis*, 3687:30–38, 2005.

[103] U. Uludag, S. Pankanti, S. Prabhakar, and A. K. Jain. Biometric cryptosystems: Issues and challenges. *Proceedings of the IEEE*, 92(4):948–960, 2004.

[104] E. Verheul. Self-blindable credential certificates from the weil pairing. *Advances in Cryptology AsiaCrypt 2001*, 2248:533–551, 2001.

[105] C. Vielhauer and R. Steinmetz. Handwriting: Feature correlation analysis for biometric hashes. *EURASIP Journal of Applied Signal Processing*, 2004:542–558, 2004.

[106] C. Vielhauer, R. Steinmetz, and A. Mayerhofer. Biometric hash based on statistical features of online signatures. *16th International Conference on Pattern Recognition*, 1:123–126, 2002.

[107] Y. Yang, F. Bao, and R.H. Deng. Security analysis and fix of an anonymous credential system. *Information Security and Privacy*, 3574:537–547, 2005.

[108] S. J. Young. The htk book. http://htk.eng.cam.ac.uk/prot-docs/htk_book.shtml, 2009.

[109] S. J. Young. *HTK: Hidden Markov Model Toolkit V3.4*. Cambridge University Engineering Department, 2009.

[110] G. Zheng, W. Li, and C. Zhan. Cryptographic key generation from biometric data using lattice mapping. *18th International Conference on Pattern Recognition*, 4:513–516, 2006.

[111] M. Zuckerberg. Now connecting 250 million people. `http://blog.facebook.com/blog.php?post=106860717130`, July 2009.