A Biometric Identity Based Signature Scheme

Andrew Burnett Adam Duffy Tom Dowling

Crypto Group
Computer Science Dept.
NUI Maynooth
Co. Kildare, Ireland
e-mail: cryptogrp@cs.may.ie

Abstract

We describe an identity based signature scheme that uses biometric information to construct the public key. Such a scheme would be beneficial in a legal dispute over whether a contract had been signed or not by a user. A biometric reading provided by the alleged signer would be enough to verify the signature. We make use of Fuzzy extractors [7] to generate a key string from a biometric measurement. We use this biometric based key string and an elliptic curve point embedding technique [13] to create the public key and corresponding private key. We then make use of a pairing based signature scheme [5] to perform signing and verification with these keys. We describe a possible attack on this system and suggest ways to combat it. Finally we describe how such a biometric signature scheme can be developed by reusing existing components in our Java Identity Based Encryption implementation. The design allows traditional as well as biometric identity based signatures. [8].

1 Introduction

In this paper we present a biometric identity based signature scheme (BIO-IBS). Traditional public key cryptosystems use very long integers, typically 2048 bits, as public keys. These systems rely on digital certificates to connect an identity like a person or a machine to a public key. Identity based systems have the advantage that a public key is the identity, usually an arbitrary string like an email address. In our case we use a biometric measurement of an individual. Using biometrics does however create a problem with variation due to biometric identities changing over time. We discuss how to overcome this problem below. One of the key uses of signature schemes is in the area of non-repudiation of documents. Our scheme is particularly useful in this area as biometric measurements such as fingerprints are long established evidential tools [14].

Consider the following situation: A user signs a contract using BIO-IBS and later a dispute develops about the signature on the contract. The user only needs to present their biometric measurement to an arbitrator to determine the validity signature. As the biometric measurement is used as a public key here there is no need to worry about the biometric measurement being compromised [15]. Also a trained arbitrator can detect attempts to deny signing such as using a film containing another user's print on a verifier's finger [14]. The paper is organized as follows. In section 2 we briefly outline the basics of elliptic curves over finite fields. In section 3 we give an overview of the process of turning biometric data into key strings. Section 4 will discuss how the key string is converted into a key pair for use in the signature scheme. Section 5 will give an overview of the BLS identity based signature scheme using the key pair generated from the biometric data. Section 6 outlines a possible attack on the system and suggests countermeasures. Section 7 will outline design issues involved in incorporating BIO-IBS into our existing Java Identity Based encryption system. Finally, we will discuss conclusions and future work.

2 Elliptic curve background

We use the symbol $\oplus$ to denote bitwise exclusive or, XOR. We define the finite field $\mathbf{F}_p = \{0, 1, 2, 3, \ldots, p-2, p-1\}$. We define the finite extension field $\mathbf{F}_p^2 = \{a + ib\}$ where $a, b \in \mathbf{F}_p$ and $i = \sqrt{-1}$. The inverse of an integer, a, in the finite field $\mathbf{F}_p$ is denoted by a^{-1} and defined by $a * a^{-1} = 1 \bmod p$. The concept of division in finite fields is equivalent to multiplication by an inverse i.e. $a \div b \bmod p \equiv a * b^{-1} \bmod p$.

The basic units for elliptic curve arithmetic are points (x, y) on an elliptic curve, E, over a finite field, $\mathbf{F}_p$, denoted $E(\mathbf{F}_p)$, of the form

$$y^2 = x^3 + ax + b \text{ with } x, y, a, b \in \mathbf{F}_p \ .$$

We define abstract concepts of addition, $P + Q$, and scalar multiplication by an integer, sQ, on the points of $E(\mathbf{F}_p)$. We also define a special point at infinity, ∞. These operations combine to make $E(\mathbf{F}_p)$ a finite Abelian group. Details of how these concepts are implemented appear in [1, 13]. The *order* of a point P is defined to be the smallest integer n such that $nP = \infty$. We let $E(\mathbf{F}_p)[q]$ be the subgroup of $E(\mathbf{F}_p)$ consisting of points of order q. We let $\mu(q) = \{a \in \mathbf{F}_p^2 \mid a^q = 1\}$.
Fundamental to identity based systems is the concept of a bilinear mapping. An example of such a mapping is the Tate pairing,

$$\tau_q : E(\mathbf{F}_p)[q] \times E(\mathbf{F}_p^2)/qE(\mathbf{F}_p^2) \rightarrow \mu(q).$$

A computationally efficient approach to evaluating the Tate pairing appears in [17]. A very useful property of the Tate pairing is bilinearity,

$$\tau_q(xP, yQ) = \tau_q(P, Q)^{xy} \text{ for any points } P, Q \text{ and for any integers } x, y.$$

Identity Based systems make heavy use of these operations and mappings.

3 Generating Key data from Biometrics

Using biometric data as a basis for cryptographic keys is problematic as biometric measurement is not perfectly reproducible. Recent work by Dodis [7] demonstrates how such data can be used to generate strong keys for any kind of cryptographic application. They use the notion of a fuzzy extractor to describe the process of extracting a random string U from a biometric input b, in such a way that a certain amount of error is allowed for. If the input changes slightly to b' then the extracted U will be the same. To enable the recovery of U from b' the fuzzy extractor also outputs a public string V. The extractor is structured in such a way that the public value V does not leak any information about U. Dodis [7] describe three metrics to measure the variation in the biometric reading: Hamming Distance, Set Difference and Edit Distance. They then detail the construction of fuzzy extractors using these metrics. Hamming Distance is defined to be the number of bit positions that differ between b and b' and is probably the most natural and straightforward metric to work with, although the other metrics may be more efficient for particular biometrics and applications.

The fuzzy extractor construction using the Hamming Distance metric is based on previous work on a fuzzy commitment scheme in [11]. We now give a simplified outline of how such an extractor is constructed. A comprehensive account appears in [7, 11] more on the other metrics can be found in [7, 12, 6].

First we give the formal definition of a fuzzy extractor. Let $\mathcal{M}$ be a finite dimensional metric space consisting of biometric data points, with a distance function $\texttt{dis}$: $\mathcal{M} \times \mathcal{M} \rightarrow \mathbb{Z}^+$, which calculates the distance between two points based on the metric chosen. Also, let l be the number of bits of the extracted output string U and t be the error threshold (i.e. for two points $b, b' \in \mathcal{M}$ to be classed as the same $\texttt{dis}(b, b') \leq t$). An $(\mathcal{M}, l, t)$-fuzzy extractor is constructed using two functions $\texttt{Gen}$ and $\texttt{Rep}$. $\texttt{Gen}$ is a probabilistic generation procedure, which on input $b \in \mathcal{M}$ outputs an "extracted" string $U \in \{0,1\}^l$ and public string V. $\texttt{Rep}$ is a deterministic reproduction procedure allowing recovery of U from the corresponding public string V and any b' sufficiently close to b. To clarify

$$\forall\, b, b' \in \mathcal{M} \text{ with } \texttt{dis}(b, b') \leq t, \text{ if } \texttt{Gen}(b) \rightarrow \langle U, V \rangle, \text{ then } \texttt{Rep}(b', V) \rightarrow U.$$

We now outline the construction of a fuzzy extractor for the space $\mathcal{M} = \{0,1\}^n$ under the Hamming Distance metric. We define C to be a binary error-correcting code of k-bit binary string codewords with an encoding function $C_e : \mathcal{M} \rightarrow \{0,1\}^k$ and a decoding function $C_d : \{0,1\}^k \rightarrow \mathcal{M}$. Then $\texttt{Gen}(b)$ produces a random $U \in \{0,1\}^l$ and $V = b \oplus C_e(U)$. The equivalent $\texttt{Rep}(b', V)$ function returns $C_d(V \oplus b')$ which is equal to U if and only if $\texttt{dis}(b, b') \leq t$.

Both of the values output from the $\texttt{Gen}$ function U and V are passed to the key generation phase, which is dealt with in the next section.

4 Key Pair Generation

This section details how the biometric data is used in the generation of a key pair for use in an IBS scheme such as the BLS Short Signature Scheme [5]. First we show how the biometric is embedded onto a point on the elliptic curve. Then, we show how that point is used as part of the key pair generation for the signature scheme.

The "extracted" string, U, has to be embedded onto a point P on the elliptic curve $E(F_p)$. This requires the use of a hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ where $\mathbb{G}_1$ is a subgroup of the points on an elliptic curve. Rather than hash directly onto $\mathbb{G}_1^*$ we use a standard hash function, H, i.e. Secure Hash Algorithm 1 (SHA-1), to hash to a set $A \subseteq \{0,1\}^*$. Then we use a deterministic encoding function, g, to map A onto $\mathbb{G}_1^*$ so that $H_1(U) = g(H(U))$. See [3] for more details.

Various algorithms exist to embed onto a point on the curve. We use the algorithm by Koblitz [13, §6.2], as we were using the curve of the form $y^2 = x^3 + x$. Boneh and Franklin [3, §4.3] detail a simpler method for embedding points on curves of the form $y^2 = x^3 + 1$ which is suitable for Weil Pairing.

With this, we can generate $P_b = g(H(U))$ and $P_s = xP_b$ where x is a randomly generated secret key in F_p^*. The variable P_b is the point corresponding to the biometric input. The key pair consists of the private key x and the public key P_s.

The techniques given here can be applied to key pair generation for other IBS schemes including Blind Signature Scheme [2], Multisignature Scheme [2], Aggregate Signature [4], Bilinear Verifiably Encrypted Signature [4], ID-Based Blind Signature Scheme [18] and ID-Based Signature from Pairing [10] among others. A summary of each of these schemes can be found in the survey by Dutta et al [9].

5 Incorporating into an Identity Based Signature scheme

An overview of the BLS Short Signature Scheme is given in this section. This scheme was proposed by Boneh, Lynn and Shacham [5] and consists of three stages. The first stage is the Key Generation stage. This stage was discussed above and should result in a point P_b and a key pair (P_s, x).

The second stage is the Signing stage. Given the secret key $x \in F_p^*$ and a message $m \in \{0,1\}^*$ the signature can be computed by $\sigma = xg(H(m))$. The security of the scheme depends on keeping x a secret.

The third and final stage is the Verification stage. The verifier uses V and biometric input b' to recalculate P_b by reproducing $U = \texttt{Rep}(b', V)$ and calculating $P_b = g(H(U))$. The verifier has the signature σ, the message m and the public key P_s. Note that the signer's secret key x is not needed for verification. The signature is verified if

$$\tau_q(P_b, \sigma) = \tau_q(P_s, g(H(m)))$$

since, using the bilinearity of the Tate pairing,

$$\begin{aligned} \tau_q(P_b, \sigma) &= \tau_q(P_b, xg(H(m))) \\ &= \tau_q(P_b, g(H(m)))^x \\ &= \tau_q(xP_b, g(H(m))) \\ &= \tau_q(P_s, g(H(m))) \end{aligned}$$

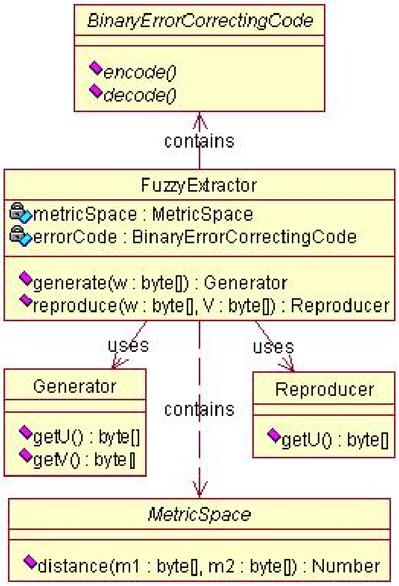
BinaryErrorCorrectingCode
encode()
decode()
contains
FuzzyExtractor
metricSpace : MetricSpace
errorCode : BinaryErrorCorrectingCode
generate(w : byte[]) : Generator
reproduce(w : byte[], V : byte[]) : Reproducer
uses
uses
Generator
getU() : byte[]
getV() : byte[]
contains
Reproducer
getU() : byte[]
MetricSpace
distance(m1 : byte[], m2 : byte[]) : Number

Figure 1: Fuzzy Extractor Class Model

6 A Possible Attack

Here we describe a possible attack to the scheme and identify ways of preventing it.

It is possible for an attacker to imitate a signer by obtaining a copy of their biometric data. For example, see [16] for methods of duplicating fingerprints. After obtaining a copy of the signer's biometric data, the attacker can sign a forged message that will appear genuine on verification by the signer.

To prevent this attack, genuine messages can be signed in the presence of a trusted witness. Alternatively, signers can utilise a digital certificate obtained from a trusted certificate authority. The digital certificate will contain the public key P_s as well as some information about the signer.

The attack described here is possible in traditional IBS schemes and not just the BIO-IBS scheme proposed here. However, the use of biometric data increases the effort required by an attacker than that required for traditional IBS schemes.

7 Extension of an existing IBE API to accomodate Biometric Signatures

The design issues of developing a pairing based biometric signing scheme are discussed in this section. We first introduce the design of the fuzzy extractor classes. Next we present the design of classes for key pair generation. Following that, the design for an implementation of the IBS BLS scheme is given.

The core technology, an implementation of the Tate pairing, was previously developed for use in an Identity Based Encryption system [8]. The flexible design accomodates both perfectly and not perfectly reproducible identities and, where appropriate, adheres to the Java Cryptographic Architecture (JCA). The design follows a pluggable architecture allowing for use of both alternative and enhanced implementations.

The fuzzy extractor class (`FuzzyExtractor` in Figure 1) performs two functions, the generation function `Gen` and the reproduction function `Rep`. Since the generation function returns two strings U and V, those strings are encapsulated in the `Generator` class. The reproduction function result is encapsulated in the `Reproducer` class. This allows for alternative representations of the results from the fuzzy extractor.

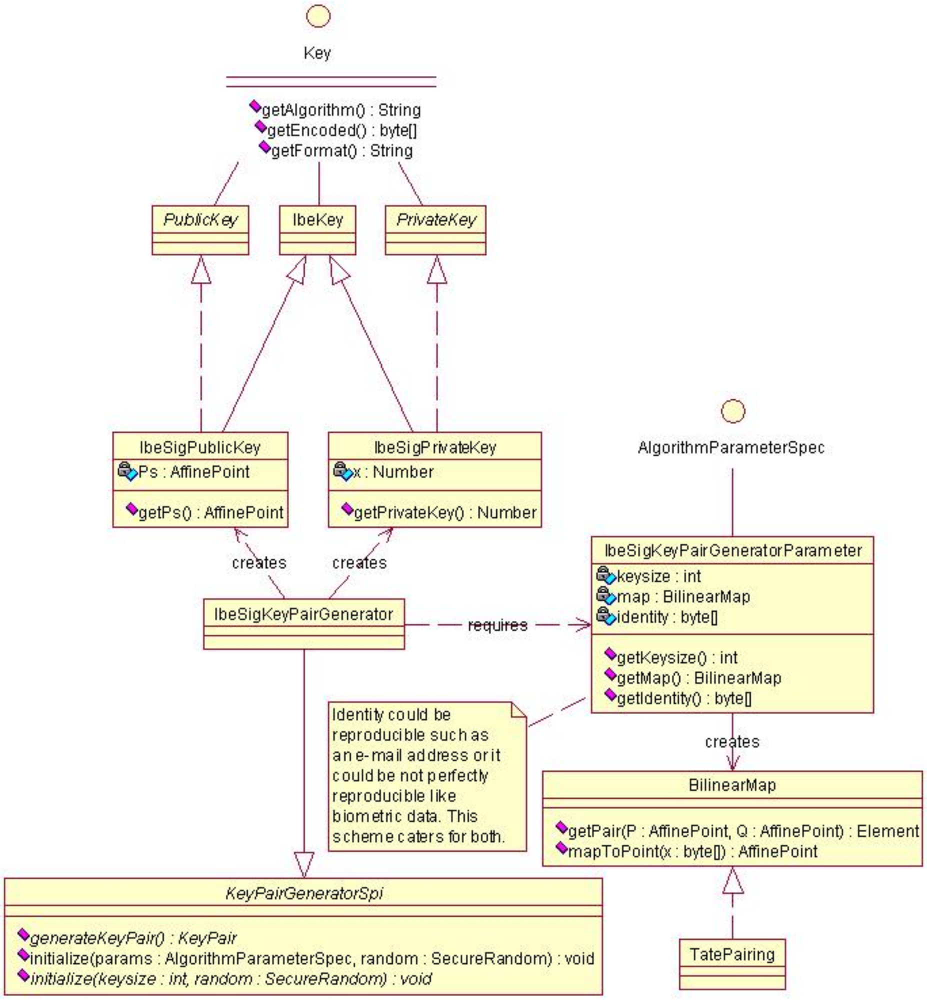
Key
getAlgorithm() : String
getEncoded() : byte[]
getFormat() : String
PublicKey
IbeKey
PrivateKey
IbeSigPublicKey
Ps : AffinePoint
getPs() : AffinePoint
IbeSigPrivateKey
x : Number
getPrivateKey() : Number
AlgorithmParameterSpec
creates
creates
IbeSigKeyPairGenerator
requires
IbeSigKeyPairGeneratorParameter
keysize : int
map : BilinearMap
identity : byte[]
getKeysize() : int
getMap() : BilinearMap
getIdentity() : byte[]
Identity could be reproducible such as an e-mail address or it could be not perfectly reproducible like biometric data. This scheme caters for both.
creates
BilinearMap
getPair(P : AffinePoint, Q : AffinePoint) : Element
mapToPoint(x : byte[]) : AffinePoint
KeyPairGeneratorSpi
generateKeyPair() : KeyPair
initialize(params : AlgorithmParameterSpec, random : SecureRandom) : void
initialize(keysize : int, random : SecureRandom) : void
TatePairing

Figure 2: Key Generation Class Model

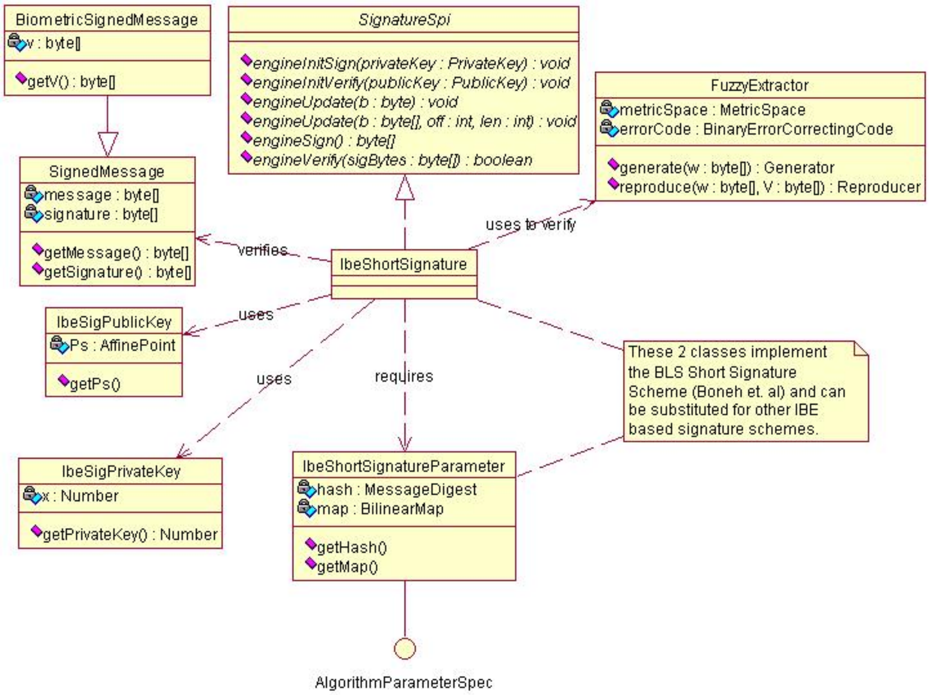
BiometricSignedMessage
v : byte[]
getV() : byte[]
SignatureSpi
engineInitSign(privateKey : PrivateKey) : void
engineInitVerify(publicKey : PublicKey) : void
engineUpdate(b : byte) : void
engineUpdate(b : byte[], off : int, len : int) : void
engineSign() : byte[]
engineVerify(sigBytes : byte[]) : boolean
FuzzyExtractor
metricSpace : MetricSpace
errorCode : BinaryErrorCorrectingCode
generate(w : byte[]) : Generator
reproduce(w : byte[], V : byte[]) : Reproducer
SignedMessage
message : byte[]
signature : byte[]
getMessage() : byte[]
getSignature() : byte[]
verifies
uses to verify
IbeShortSignature
IbeSigPublicKey
Ps : AffinePoint
getPs()
uses
uses
requires
These 2 classes implement the BLS Short Signature Scheme (Boneh et. al) and can be substituted for other IBE based signature schemes.
IbeSigPrivateKey
x : Number
getPrivateKey() : Number
IbeShortSignatureParameter
hash : MessageDigest
map : BilinearMap
getHash()
getMap()
AlgorithmParameterSpec

Figure 3: Biometric Identity Based Signature Scheme Class Model

A binary error correcting code and a metric space make up the attributes of the fuzzy extractor. These attributes are represented by the interfaces `BinaryErrorCorrectingCode` and `MetricSpace` respectively, allowing for alternative correcting codes and metric spaces to be used. We extend the `MetricSpace` class with a `HammingDistanceMetricSpace` that implements the Hamming Distance metric.

The design of the classes needed for the key pair and key pair generation are shown in Figure 2. These classes can also be used in generating key pairs for other IBE based signature schemes such as Blind Signature Scheme, Multisignature Scheme and so on.

The key pair generation utilises the `mapToPoint` functionality of the Tate pairing implementation `TatePairing`. This is an implementation of the hash function H_1 mentioned earlier.

The design of the classes for signing and verification are given in Figure 3. The `IbeShortSignature` class and related classes may be substituted for alternative IBE based signature schemes with minimal impact on the remaining code.

8 Conclusion

We have presented a biometric identity based signature scheme. We have reused ideas in the areas of string construction from biometric data, key generation, and pairing based signature schemes to form the components of our system. We have discussed the application of such a scheme to non-repudiation of contracts or documents. Finally we outlined how such a biometric signature scheme could be incorporated into an existing identity based encryption software package. Our pluggable architecture provided for easy incorporation of different implementations of algorithms to perform the various component procedures in the system. This will facilitate inclusion of future performance enhancements to existing algorithms or inclusion of new algorithms to the system.

References

[1] I. Blake, G. Seroussi, and N. Smart. *Elliptic Curves in Cryptography.* Cambridge University Press, 1st edition, 1999.

[2] A. Boldyreva. Efficient Threshold Signature Multisignature and Blind Signature Schemes Based on the Gap-Diffie-Hellman Group Signature Scheme. *Proceedings on the 6th International Workshop on Practice and Theory in Public Key Cryptography*, 2003.

[3] D. Boneh and M. Franklin. Identity Based Encryption from the Weil Pairing. *SIAM Journal of Computing, Vol. 32, No. 3*, pages 586–615, 2001.

[4] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and Verifiably Encrypted Signature from Bilinear Maps. *Proceedings from Advances in Cryptology - EuroCrypt*, 2003.

[5] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. *Proceedings from Advances in Cryptology - Asiacrypt*, 2001.

[6] T. Clancy, N. Kiyavash, and D. Lin. Secure Smartcard-Based Fingerprint Authentication. *Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications*, 2003.

[7] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *Proceedings from Advances in Cryptology - EuroCrypt*, 2004.

[8] T. Dowling, A. Duffy, and L. Owens. An Identity Based Encryption System. *Proceedings of the 3rd International Conference on Principles and Practice of Programming in Java*, 2004.

[9] R. Dutta, R. Barua, and P. Sarkar. Pairing-Based Cryptographic Protocols: A Survey. *Cryptology ePrint Archive*, 2004.

[10] F. Hess. Efficient Identity Based Signature Schemes Based on Pairings. *Proceedings from Symposium on Applied Computing*, 2002.

[11] A. Juels and M. Wattenberg. A Fuzzy Commitment Scheme. *Proceedings of the 6th ACM conference on Computer and Communications Security*, pages 28–36, 1999.

[12] A. Juels and M. Wattenberg. A Fuzzy Vault Scheme. *Proceedings of the IEEE International Symposium on Information Theory*, 2002.

[13] N. Koblitz. *A Course in Number Theory and Cryptography.* Springer, 2nd edition, 1994.

[14] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition.* Springer, 2003.

[15] B. Schneier. Inside Risks: The Uses and Abuses of Biometrics. *Communications of the ACM, Vol. 42*, page 136, 1999.

[16] T. van der Putte and J. Keuning. Biometrical Fingerprint Recognition: Don't Get Your Fingers Burned. *Proceedings of the Fourth Working Conference on Smart Card Research and Advanced Applications*, 2000.

[17] L. Washington. *Elliptic Curves: Number Theory and Cryptography.* Chapman and Hall, CRC, 1st edition, 2003.

[18] F. Zhang and K. Kim. ID-Based Blind Signature and Ring Signature from Pairings. *Proceedings from Advances in Cryptology - Asiacrypt*, 2002.