Pre-print of article that appeared at WIFS 2010.

The published article can be accessed from:
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5711435

Beyond PKI: The Biocryptographic Key Infrastructure

W. Scheirer *#1, B. Bishop *2, T. Boult *#3

Department of Computer Science, University of Colorado at Colorado Springs
Engineering Building Room 199, 1420 Austin Bluffs Parkway, Colorado Springs, CO 80933-7150 USA
[1] wjs3@vast.uccs.edu
[3] tboult@vast.uccs.edu

* Securics, Inc.
1867 Austin Bluffs Parkway, Suite 200, Colorado Springs, CO 80918 USA
[2] bill.bishop@levault.net

***Abstract*—Public Key Infrastructure is a widely deployed security technology for handling key distribution and validation in computer security. Despite PKI's popularity as a security solution, Phishing and other Man-in-the-Middle related network attacks are accomplished with ease. The major problems with PKI come down to trust, and largely, how much faith we must place in cryptographic keys alone to establish authenticity and identity. In this paper, we look at a novel biometric solution that mitigates this problem at both the user and certificate authority levels. More importantly, we examine the trouble with the placement of unprotected biometric features directly into PKI, and propose the integration of a secure, revocable biometric template protection technology that supports transactional key release. A detailed explanation of this new *Biocryptographic Key Infrastructure* is provided, including composition, enrollment, authentication, and revocation details.**

I. Introduction

Public Key Infrastructure (PKI) [1] [2] has been a popular, yet often maligned technology since its widespread adoption in the 1990s. PKI is the infrastructure for handling the complete management of digital certificates (x.509 compliant), which contain a piece of trusted information - a public key. PKI attempts to solve an important problem in key management - namely, how can Alice verify that Bob's public key is really Bob's? Addressing this problem remains a paramount concern, as the Internet has experienced an explosion of successful Phishing and other Man-in-the-Middle attacks in recent years. Users of networks, both those well-informed and those blissfully ignorant of security protocols, routinely ignore security provisions put into place by PKI to guard against such attacks. Sadly, providers of information security services are also to blame for using PKI as a catch-all security solution and ignoring its limitations.

The problems with PKI [3] are well-known, and have remained mostly unsolved thus far. The overarching criticism stems back to the notion of trust in a PKI system - why would we place any trust in a system with entities signifying their identity only with keys? A very real and recent attack[1] presented at the 2008 Chaos Communications Conference highlights the ease with which a rogue certificate authority can be established, with an MD5 hash collision attack against the digital signatures used for certificate validation. With all trust being placed in expected messages, presumedly derived from *legitimate* keys, there is no way to tell the difference between a Man-in-the-Middle and a legitimate site if a useful collision has been located. While MD5 was directly responsible for the attack in this instance, the entire infrastructure will always be susceptible to trust related attacks if any cryptographic component is flawed. Can we only trust an entity based on expected numbers?

By adding a second factor, we can mitigate the trust problems inherent in PKI. Biometrics, those methods of uniquely recognizing humans based on physiognomy or behavior have become ubiquitous in many areas of technology and society, having matured to the point of general acceptance as valid and useful security tools. For PKI, the addition of biometric data has a very attractive feature - if a user or certificate authority presents a key and biometric during some action, we have more confidence that this action is legitimate (but this does not absolutely prove that the owner of the key and biometric actually performed the action - stolen keys and spoofing attacks are very real). With biometrics, we have improved *non-repudiation*. A series of related concerns follow the trust problem: the security of the verifying computer, certificate authority establishment, and general certificate issue. With the proper protocols including a biometric component, we can address each of these.

But to solve these problems correctly, we cannot simply use standard biometric templates (the data representation of the collected biometric feature) embedded within x.509 certificates, because a *revocation* of raw biometric data can only happen for a very limited number of times (we have 1 face, 2 irises, 10 fingers). Standard templates, while being an abstract representation of the original biometric features, are still effectively invertible [4]. Moreover, providing unprotected biometric data to even "trusted" entities is risky at best. To understand why, we must consider what we term the *Biometric Dilemma*. In essence, as the use of biometrics increases, so does the chance for compromise. If a malicious attacker, Mallory, wishes to impersonate Alice at an area of high

[1] http://www.win.tue.nl/hashclash/rogue-ca/

security, she can obtain the exact biometric data she needs from a different, much lower security area. How well might Alice's gym be protecting her biometric data that she uses to access her locker? Low-hanging fruit is plentiful, and can often be obtained legitimately. In 2001, the state of Colorado tried to sell its DMV face and fingerprint databases[2] to anyone who wanted to buy them. The resulting protests moved the data back off the market, but the state still offers them to any requesting law enforcement agency.

Previous work on the integration of biometrics into PKI has not considered the biometric dilemma, favoring a simplistic unsecured application of biometrics. Proposed standards for PKI with biometrics go back to the mid 1990s [5]. More recently, x.509 certificates augmented with BioAPI[3] have been suggested [6], providing the templates and matching capability needed to use the biometric data. As part of a much larger defense-in-depth approach to authentication, [7] also recommends augmenting x.509 certificates with biometric data. None of these previous standards recommendations and research works propose anything that can be considered a secure handling of the biometric data. All store and match unprotected templates, and have no facility for biometric template revocation and re-issue.

In response to the threat of permanent biometric feature compromise, very recent research [8] has emerged from both the pattern recognition and cryptography communities to address the problem of *biometric template security*. Solutions to this problem seek to create a transformation of original features that can be revoked and reissued if a compromise is detected, in much the same manner as a traditional password or PIN. For unattended network authentication, the risk of spoofing is greatly reduced by secure templates. Unique templates can be generated for different domains and applications, making a template harvested by an attacker at one domain useless when applied to a different domain. This addresses the biometric dilemma described above. Even more interesting for trusted data transfer is that certain classes of protected template schemes support key release upon successful matching. *Key-binding* biometric cryptosystems bind key data with the biometric data. *Key-generating* biometric cryptosystems derive the key data from the biometric data. Both classes support a key release that may be used for cryptographic applications, including standard symmetric key cryptography, where key storage is problematic.

The rest of this paper introduces the details for the Biocryptographic Key Infrastructure. In Section II, the fundamental biometric requirements are defined, including the properties necessary for protecting the biometric data, secure key release, and revocation support. In Section III our full infrastructure is described, including a description of the overall composition, the enrollment process for both biometric certificate authorities and users, authentication protocols, and revocation and re-issue procedures. We conclude in Section IV.

[2]http://www.i2i.org/articles/8-2001.PDF

[3]http://www.bioapi.org/

II. Fundamental Biometric Requirements

Many different secure template technologies exist, but not all are appropriate for use in a PKI-like framework. To be useful for PKI, we suggest that a secure template technology should possess the following properties:

1) Cryptographically strong protection of the underlying biometric features.
2) The ability to revoke and re-issue the template.
3) Nested re-encoding, allowing a hierarchy of templates to be generated from a single base template.
4) Support for public templates that cannot be used to match other public templates, and private templates that are generated dynamically from a biometric sample during matching and immediately discarded following.
5) Key-binding capability without the need of intervention by the person associated with the template.

Throughout the rest of this paper, we will use *revocable biotokens* [9] as a case study for the BKI described herein, though any secure template technology supporting the five aforementioned properties could be used. To date, only revocable biotokens support all five. We briefly introduce the fundamentals for revocable biotokens in the remainder of this section as an illustration of the biometric requirements.

In general, biometric data cannot be encrypted reliably, because of the unstable nature of the data, which can vary as a function of environment, age, and acquisition circumstances. However, many biometric modalities yield features that can be split into stable and unstable (or residual) components, allowing the reliable encryption of the stable component, which can then be matched in encoded space, with additional residual matching adding accuracy. Using this knowledge, and the concept of public key cryptography, we can develop the re-encoding methodology for revocable biotokens. The re-encoding property is essential for supporting a viable transactional framework - tokens with unique data must be generated quickly and automatically to support cryptographic transactions (such as session key exchange). The *bipartite biotoken* form of a revocable biotoken supports data-binding (key-binding) at the transactional level. Bipartite biotoken generation from a stored biotoken allows the required data release when only matching against tokens generated from original biometric features during the transaction.

Assuming the biometric produces a value v that is *transformed* via scaling and translation to $v' = (v - t) * s$, the resulting v' is split into the overall stable component q, and the the residual component r. The amount of stable & unstable data is a function of the biometric modality being considered. In the base scheme, for a user j, their residual $r_j(v')$ is left un-encoded. For the initial transformation $w_{j,1}(v', P)$ of q, a public key P is required. For nested re-encodings, w_j is re-encoded using some transformation function T (which may be a hash function, or another application of public key cryptography) creating a unique new transformation for each key that

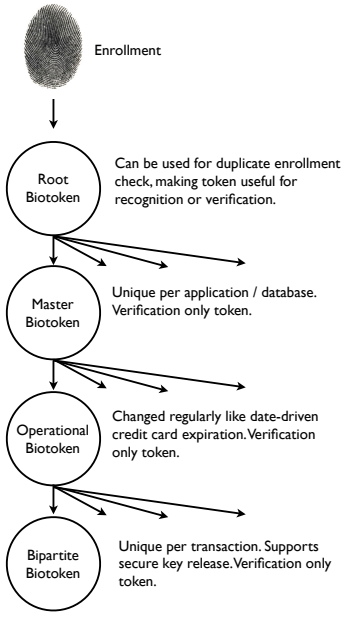

Fig. 1. The biotoken issue/re-issue tree. Biotokens can be re-encoded, starting from the root token generated at enrollment time, through subsequent applications of PK encryption (supporting automatic revocation and re-issue) or a hash function.

is applied: $w_{j,1}(v', P),\ w_{j,2}(w_{j,1}, T_2), \ldots, w_{j,n}(w_{j,n-1}, T_n)$

Using public key cryptography, the nesting process can be securely invertible if the private keys all the way back to the first stage of encoding are available. Partially inverting the nesting facilitates revocation and automatic re-issue of the biotoken, which is an attractive feature for the BKI system. A tree containing our standard hierarchy of biotokens is shown in Figure 1. The public keys used for encoding here are strictly for this biometric process, and are different from the keys contained in the user's certificate. With this nesting, we can define three properties for the bipartite biotoken:

1) Let B be a secure biotoken, as described in [9]. A bipartite biotoken B_p is a transformation $bb_{j,k}$ of user j's kth instance of B. This transformation supports matching in encoded space of any bipartite biotoken instance $B_{p,k}$ with any secure biotoken instance B_k for the biometric features of a user j and a common series of transforms P, T_2, …, T_k.

2) The transformation $bb_{j,k}$ must allow the embedding of some data d into B_p, represented as: $bb_{j,k}(w_{j,k}, T_k, d)$.

3) The matching of B_k and $B_{p,k}$ must release d if successful, or a random string r if not successful.

The primary benefit of BKI is the ability to store public biotokens that any user in a particular infrastructure can retrieve and use to generate a bipartite biotoken to send some secret back to the owner of the biotoken, with the assurance that the certificate containing the biotoken is valid. The security of such a scheme to publicly distribute biotokens derived from biometrics is of course a concern. It was shown in [9] that revocable biotokens are cryptographically secure. Further, [9] presents a test of over 500 *Million* impostor trials, with no false matches. Thus, we have confidence that revocable biotokens can be used in a public setting.

III. A Biocryptographic Key Infrastructure

A. Composition and Enrollment

Biometric Certificate Authorities (BCAs) are certificate authorities that support both public keys and revocable biotokens, and are biometrically verified by higher-level authorities, in a process described in detail below. As in PKI, a central root authority exists to authorize all BCAs below it. Enrollment and key management follows from each BCA up to the root. Auth Stations exist at the outermost regions of the infrastructure, and are the places where users submit their biometric samples to generate enrollment biotokens or biotokens for a particular session. Report Engines can also be deployed throughout the infrastructure to propagate registration and transaction reports to other authorities.

In order to support the biotoken, we add some additional fields to the base x.509 v3 certificate via its extensions provision, similar to [6]. We can use certificates in both an online and offline setting. If we are operating in an offline setting, such as a standalone computer or private network, we are not able to connect to BCAs on outside networks, including the root. In order to indicate the operating mode to the underlying BKI software, the certificate contains an "Online Only" and "Standalone Only" flag. For the subject's biotoken, we first note the type of biotoken included. Recall from Figure 1 that a tree of different biotokens exists for a particular subject, with the possibility of Root Biotokens, Master Biotokens, and Operational Biotokens being included in a certificate. Following the "Biotoken type" flag, the biotoken itself is included.

We need BCAs to trust each other, and we need to be able to place some trust in our end-users. To do this, we need an enrollment process where we require that someone biometrically register with the root BCA, which can search for this person in the existing records. To introduce an increased level of trust with biometrics, the standard Certificate Signing Request (CSR) [2] is augmented as per Figure 2. The changes take advantage of the open nature of registration information detail for new text fields, and the open extensions in the certificate template, as defined by [2].

Specifically, BKI requires that a representative of an organization making a request generate an enrollment biotoken, which is passed up to the root authority for a *duplicate enrollment check* (has this person been flagged as a trouble maker? or are they impersonating someone else?). The enrollment biotoken is always generated as a Root Biotoken (Figure 1) using the root authority's public key to ensure consistent matching behavior between all enrollees. The enrollment token is stored at the root for use in all future enrollment checks. While this does not protect the privacy of the organizational representative at the database level, it does maintain the integrity of the BCA establishment, and still protects the security of the representative's biometric data. The same process follows for end users, except the enrollment

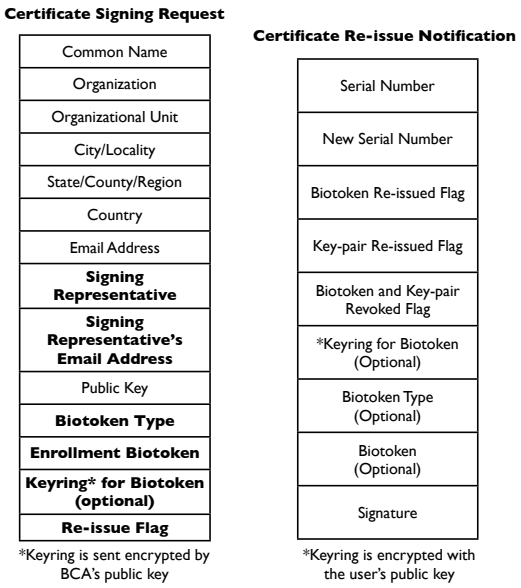

Fig. 2. A modification of the typical CSR message, including biotoken enrollment information, on the left, and the newly defined CRN message, for certificate revocation and re-issue, on the right

token need not be passed all the way back to the root from the user's Auth Station; local BCAs can manage it.

B. Authentication Framework

Extending the standard protocols defined in Section 24.9 of [10], we can support authentication with stronger non-repudiation via the BKI. Presume Alice has established a certification path to Bob and Bob's certificate, containing his public key and biotoken.

1) The one-way protocol::

1) Alice generates a nonce, R_A.
2) Alice constructs a message, $M = (T_A, R_A, I_B, B_{BB}(d))$, where T_A is Alice's timestamp, I_B is Bob's identity, and d is a small piece of arbitrary data. d is embedded into a bipartite biotoken $B_{BB}(d)$ that is generated from Bob's biotoken.
3) Alice sends $(C_A, D_A(M))$ to Bob. (C_A is Alice's certificate; D_A is Alice's private key.)
4) Bob verifies C_A and obtains E_A. He makes sure these keys have not expired. (E_A is Alice's public key).
5) Bob uses E_A to decrypt $D_A(M)$. This verifies both Alice's signature and the integrity of the signed information.
6) Bob checks the I_B in M for accuracy.
7) Bob checks the T_A in M and confirms that the message is current.
8) Bob submits a biometric sample to a sensor; a local biotoken B_{BL} is then generated from the sample. B_{BL} is then matched against $B_{BB}(d)$, releasing d.
9) As an option, Bob can check R_A in M against a database of old random numbers to ensure the message is not an old one being replayed.

This protocol also works by encrypting d with Bob's public key - but with the biometric version, Bob does not need to have his private key handy. Further security is provided if Alice has access to a private BCA that holds Bob's certificate, which would make Bob's biotoken a shared secret. Thus, a successful Man-in-the-Middle would need to know not only Alice's private key, but Bob's secret stored biotoken as well.

2) The two-way protocol::

10) Bob generates another nonce, R_B.
11) Bob constructs a message $M' = (T_B, R_B, I_A, B_{AB}(d))$, where T_B is Bob's timestamp, I_A is the identity of Alice, and d is the same data as in step 2. d is embedded into a bipartite biotoken $B_{AB}(d)$ that is generated from Alice's biotoken, obtained from C_A.
12) Bob sends $D_B(M')$ to Alice.
13) Alice uses E_B to decrypt $D_B(M')$. This verifies both Bob's signature and the integrity of the signed information.
14) Alice checks the I_A in M' for accuracy.
15) Alice checks the T_B in M' and confirms that the message is current.
16) Alice submits a biometric sample to a sensor; a local biotoken B_{AL} is then generated from the sample. B_{AL} is then matched against $B_{AB}(d)$, releasing d. If this d matches the d sent in the first transmission, Alice can be assured that Bob's biometric was used to unlock $B_{BB}(d)$.
17) As an option, Alice can check R_B in M' to ensure the message is not an old one being replayed.

Now Alice has further assurance Bob is actually Bob, and not an impostor. But Bob still has no assurance of Alice's identity beyond her certificate. This can be solved by a three-way protocol, where in addition to the original d, Bob also sends a d' in the same token. Alice can verify d, and send d' back to Bob for validation.

3) The three-way protocol::

18) Alice takes the recovered d' from step 16, and sends $D_A(d')$ back to Bob.
19) Bob uses E_A to decrypt $D_A(d')$, unlocking d'. Bob can be assured that Alice's biometric was used to unlock $B_{AB}(d)$ in step 16.

C. Revocation and Reissue

Unlike standard PKI, we simply cannot just revoke a certificate, generate a new random key and re-issue - we must deal with the biometric re-issue as well. When we describe compromise in this section, we mean a compromise of the biotoken itself, and not the original biometric features.

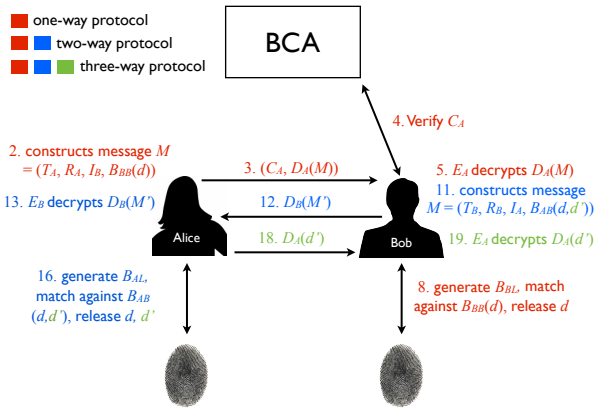

Fig. 3. The data-transfer steps for the one-way, two-way, and three-way protocols described in Section III-B. It is assumed that Alice has Bob's certificate C_B at the beginning of the one-way protocol.

1) Scenario 1: Manual Re-issue: The BCA that issued the certificate must maintain a certificate revocation list (CRL). This list only contains revoked certificates, and not expired certificates. If the user's key has been compromised, or the user's biotoken has been compromised, or the BCA's key has been compromised, or because the BCA no longer wants to certify the user, the user's certificate can be revoked. In this scenario, it is presumed that the BCA has not retained any keying information necessary to invert the biotoken it stores.

To begin the revocation process with re-enrollment, the BCA places the certificate in question on its CRL, and notifies the owner with a Certificate Re-issue Notification (Figure 2) (CRN) via the contact information provided in the CSR. This CRN is a new notice introduced in this work. If the owner is allowed to re-issue, they generate a new public-private key pair, and a new biotoken at the Auth Station. This information is sent back to the BCA in the form of a new CSR. If this CSR is accepted, a new certificate is issued.

In an alternate, yet valid, scenario for manual re-issue, re-enrollment is not required. If the user's biotoken, or biotoken and key pair, has been compromised, and the BCA possesses a stored biotoken that has not been compromised, and is the same base token that was used to generate the compromised biotoken, the owner can re-issue by varying the keys used for encoding on their end, while not needing to submit another biometric sample. To begin this revocation process, the BCA places the certificate in question on its CRL, and notifies the owner with a CRN via the contact information provided in the CSR. This CRN contains the owner's base biotoken. The owner will generate new keys for biotoken re-encoding, and use them to generate a new biotoken. This new biotoken, and optionally a new public key, is sent back to the BCA in a new CSR.

While two scenarios for automatic re-issue are discussed below, if a public key and biotoken are compromised for a particular certificate, manual re-issue with re-enrollment will always be forced. Manual re-issue with re-enrollment is also forced if the BCA's key has been compromised, where trust can no longer be placed in the existing data stored at the BCA.

2) Scenario 2: Automatic Re-issue of Biotoken: In cases where the BCA detects a compromise (especially in its own infrastructure) of a stored biotoken, it is very desirable to revoke and re-issue certificates in some automated fashion. To support this, the BCA must possess the necessary keys to invert the token, and subsequently generate a new token based on the base information. This base information *need not be* the original biometric features. Referring back to the biotoken issue/re-issue tree of Figure 1, any level of token can be generated by an Auth Station, and transmitted on to the BCA. Thus, if the biotoken exists at the 2nd - nth level of encoding, any BCA performing the inversion will not be able to recover the original biometric features.

The initial enrollment process is modified in this scenario to transmit the keying information used to create the enrollment biotoken to the BCA. The CSR contains an optional field (shown in Figure 2) to include a keyring with all of the necessary keys / passwords / identifiers used to encrypt the stable (that is, some encoding $w_{j,n}(w_{j,n-1}, T_n)$, where $n > 1$, if the original biometric features are to be protected) portion of the biotoken, during the transform. The requesting entity will include this keyring, encrypted by the BCA's public key, in its CSR. The BCA will store this encrypted keyring for later use if revocation and re-issue becomes necessary.

If the user's biotoken has been compromised, the user's certificate can be revoked and re-issued automatically. To begin the revocation process, the BCA places the certificate in question on its CRL, and notifies the owner via the contact information provided in the CSR. If the owner is allowed to re-issue, the BCA will take it upon itself to invert the biotoken back a level (to $w_{j,n-1}$, where $n > 1$), generate a new set of keying information, and re-encode the biotoken (producing $w'_{j,n}$). A new certificate is then created with the new biotoken, and the original public key. The BCA then sends the owner of the certificate a CRN, which indicates the serial number of the revoked certificate, the serial number of the re-issued certificate, and the new keyring for the new biotoken (encrypted with the user's public key). This message is signed by the BCA. Automatic re-issue may happen transparently to the user, with the underlying BKI software taking note of the CRN, and updating the keying information for biotoken generation at the user's Auth Station.

3) Scenario 3: Automatic Re-issue of key-pair: Similar to Scenario 2, it is very desirable to revoke and re-issue certificates in some automated fashion when the public/private key-pair becomes compromised. To support this, the BCA can use a bipartite biotoken generated from the uncompromised biotoken stored in the user's certificate to convey a secret back to the user.

If the user's key-pair has been compromised, the user's certificate can be revoked and reissued automatically. To begin the revocation process, the BCA places the certificate in question on its CRL, and notifies the owner via the contact information provided in the CSR. If the owner is allowed to re-issue, the BCA will take it upon itself to generate a new key-pair. A new certificate is then created with the new public

key, and the original biotoken. The BCA then embeds the new private key into a bipartite biotoken generated from the user's biotoken. The BCA then sends the owner of the certificate a Certificate Re-issue Notification (CRN), which indicates the serial number of the revoked certificate, the serial number of the re-issued certificate, and the bipartite biotoken containing the embedded private key. This message is signed by the BCA. For automatic re-issue, the user must submit their biometric at the Auth Station to release their new private key from the bipartite biotoken in the CRN.

IV. Conclusions

In this paper, we have taken a look at security issues with both PKI and biometrics, and introduced a Biocryptographic Key Infrastructure incorporating a secure template technology that solves problems with both. In summary, PKI suffers from problems related to the trust that is presumed for all entities in the infrastructure. By incorporating a secure biometric template technology such as revocable biotokens into digital certificate signing requests, we can achieve improved non-repudiation, and thus increase the trust placed in both certificate authorities and users, while addressing the biometric dilemma. Moreover, with a second factor that allows the secure transfer of embedded data, we can support automatic certificate revocation and re-issue. Ultimately, the goal here is to prevent common Phishing and Man-in-the-Middle attacks, which can be accomplished using the protocols we have defined for secure authentication between two parties using keys and biotokens.

Acknowledgment

This work was supported in part by NSF STTR Award Number 0750485 and NSF PFI Award Number 065025.

References

[1] C. Adams and S. Farrell, "Internet X.509 Public Key Infrastructure Certificate Management Protocols," RFC 2510 (Proposed Standard), Mar. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2510.txt

[2] J. Schaad, " Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)," RFC 4211 (Proposed Standard), Sep. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc4211.txt

[3] C. Ellison and B. Schneier, "Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure," *Computer Security Journal*, vol. 16, no. 1, pp. 1–7, 2000.

[4] J. Feng and A. Jain, "FM Model Based Fingerprint Reconstruction from Minutiae Template," in *Proc. of IEEE/IAPR Int. Conf. on Biometrics*, 2009, pp. 544–553.

[5] L. Reinert and S. Luther, "User Authentication Techniques Using Using Public Key Certificates, National Security Agency, Central Security Service," Dec. 1997.

[6] G. Martinez-Silva, F. Henriquez, N. Cortes, and L. Ertaul, "On the Generation of X.509v3 Certificates with Biometric Information," in *Proc. of the 2007 International Conference on Security and Management (SAM '07)*, 2007.

[7] E. Dawson, J. Lopez, J. Montenegro, and E. Okamoto, "BAAI: Biometric Authentication and Authorization Infrastructure," in *Proc. of the International Conference on Information Technology: Research and Education (ITRE 2003)*, 2003, pp. 274–278.

[8] A. Jain, K. Nandakumar, and A. Nagar, "Biometric Template Security," *EURASIP Journal on Advances in Signal Processing*, 2008.

[9] W. Scheirer and T. Boult, "Bipartite Biotokens: Definition, Implementation, and Analysis," in *Proc. of IEEE/IAPR Int. Conf. on Biometrics*, 2009, pp. 775–785.

[10] B. Schneier, *Applied Cryptography, Second Edition*. John Wiley & Sons, Inc., 1996.