

Contribution on Biometric Threat Models to AHGBEA

Date: 24 January 2006

Contributors:

- Matthew Young, Purdue University (POC)
- Shimon Modi, Purdue University
- Cathy Tilton, Daon
- Alessandro Triglia, OSS Nokalva

## 1 Threats, Vulnerabilities, and Models

Biometrics possess the powerful potential to provide added security for a variety of applications. Already, biometrics have been deployed to protect personal computers, ATMs, credit card transactions, electronic transactions, airports, nuclear facilities, and international borders.

Yet, while biometrics may improve security in a plethora of environments and serve a medley of purposes, biometric systems, like any other security system, have vulnerabilities. As the 2005 Las Vegas Defcon conference illustrates, for instance, the increasingly high profile use of biometrics for security purposes has provoked new interest in researching and exploring methods of spoofing biometric systems.

### 1.1 *Biometric Attacks*

This section addresses biometric device and system vulnerabilities. The considerations detailed herein apply both to specific modalities, such as fingerprint and iris recognition, as well as to generic biometric systems. [IBG]

Attacks on biometric devices and systems can be grouped into three categories:

- First, attacks at the input level;
- Second, attacks at the processing and transmission level;
- Third, attacks on the backend/storage level.

#### 1.1.1 **Input Level Attacks**

This report begins by discussing input-level attacks, specifically vulnerabilities at the point of sample acquisition and initial processing. The primary input-level attacks are spoofing and bypassing. The spoofing part of this discussion is located in section 6.2.

While spoofing is the most frequently-cited input-level vulnerability, other input-level vulnerabilities may be just as problematic, such as “overloading.” “Overloading” is an attempt to defeat or circumvent a system by damaging the input device or overwhelming it in the attempt to

generate errors. For example, the rapid flashing of bright lights against optical fingerprint sensors or facial recognition capture devices can disrupt their proper functioning. And silicon sensors can be easily damaged by short circuiting them or dousing them with water.

Because many biometric systems rely on sensitive equipment that can be overloaded relatively easily, users may have opportunities to induce device or system failure. Systems must be designed such that, if overwhelmed, basic functions must not fail. And when biometric devices can no longer serve their intended function, fallback processes must be defined and enforced. A person who causes a biometric system to fail may be doing so knowing that, as a consequence, an unguarded door may be used as a temporary alternative means of entry. Security systems must account for the potential functional failure of biometric systems and devices by means of adequate backup measures.

### **1.1.2 Processing and Transmission Level Attacks**

Though input-level attacks are an obvious illustration of biometric system vulnerability, attacks at the processing and transmission level also deserve close attention.

As many biometric systems transmit image or template data to local or remote workstations for processing, it is also imperative that this transmission be secure, lest the transmission be intercepted, read, or altered. Most biometric systems encrypt data in transit, but not all applications and devices lend themselves to encryption. Security techniques such as encryption are often seen as deployer-specific aspects of system design. While certain standards do treat encryption techniques, notably the X9.84 standard utilized by financial services institutions, standards such as BioAPI are encryption-agnostic.

Deployers need to assess the degree to which image or template data might be exposed in transit or during storage, and they need to define applicable system security techniques and best practices. Taken as a whole, anti-spoofing measures, encryption of data in transmission, and applying appropriate fallback techniques are all critical aspects of biometric system security. These techniques can be further enhanced through the introduction of multi-factor authentication and randomization.

Multi-factor authentication can take two primary forms: the use of multiple biometrics or the use of biometrics in conjunction with smart cards and PINs. Both methods reduce the likelihood of an imposter being authenticated. Spoofing also becomes more time consuming and challenging when multiple body physiological or behavioral characteristics need to be copied and imitated. Impostors for whom a biometric matches an enrolled user are unlikely also to match with respect to a secondary biometric.

Adding randomization to the equation also adds security. Verification data, for example, could be randomized, such as asking for three fingerprints one day and a different combination of two fingerprints the next day. Additionally, where time provides, designers of biometric technologies and systems should explore random or cued challenges. That is, even if a person correctly authenticates once, the system might still challenge the user to re-authenticate to help increase its confidence that the biometric data submitted is genuine.

Cued challenges could also be paired with certain behaviors causing alarm – such as an uncommon stillness, lack of movement, or change during the acquisition of biometric data. Technologies can still bear further development and enhancement for monitoring and sensing micro-movement. Or perhaps aggressive challenges could be utilized in conjunction with measurements of intelligent response time. For example, voice verification biometric systems could measure the time it takes for a prospective entrant to read back a randomly generated pass phrase in order to try to fight playback attacks pieced together from various recordings. If the response time exceeds a minimum threshold or varies significantly from an average time captured over a series of template submissions at enrollment, the biometric system could issue a challenge and require recitation of a new pass phrase.

Finally, in conjunction with multi-factor authentication and randomization, vendors and researchers should explore taking advantage of internal or subcutaneous characteristics. By focusing on biometric aspects that are difficult to observe, capture, and duplicate covertly, security can thus be enhanced.

However, regardless of how well one tries to secure a biometric system, failures will inevitably occur. It is therefore critical that attention not only be paid to preventing breaches, but also to handling breaches that have occurred. A recently-publicized technique to mitigate the impact of certain system breaches is the concept of cancelable biometrics. IBM's cancelable biometrics solution uses algorithms to distort an image proffered and records the distortion into its generated templates. The original image is never stored anywhere. The idea is that if a thief steals the template with the distortion on it, that particular distortion can be eliminated from the list of access-approved users, and the legitimate user can resubmit their original biometric data to generate a new distorted template. As long as the algorithms that generate the distortions are carefully protected and ideally varied from company to company or even system to system, this solution may be highly conducive to containment and resolution of a breach.

The solution, however, is not foolproof. If the original image is captured, it could theoretically be re-enrolled to generate a new, distorted template. Nevertheless, the creation of cancelable biometrics is a step in the right direction. If the biometrics community continues openly and aggressively to identify its weaknesses and to pursue methods of strengthening them, the entire international community will all benefit tremendously.

## ***1.2 Threat Modeling***

### **1.2.1 Vulnerable points of a biometric system**

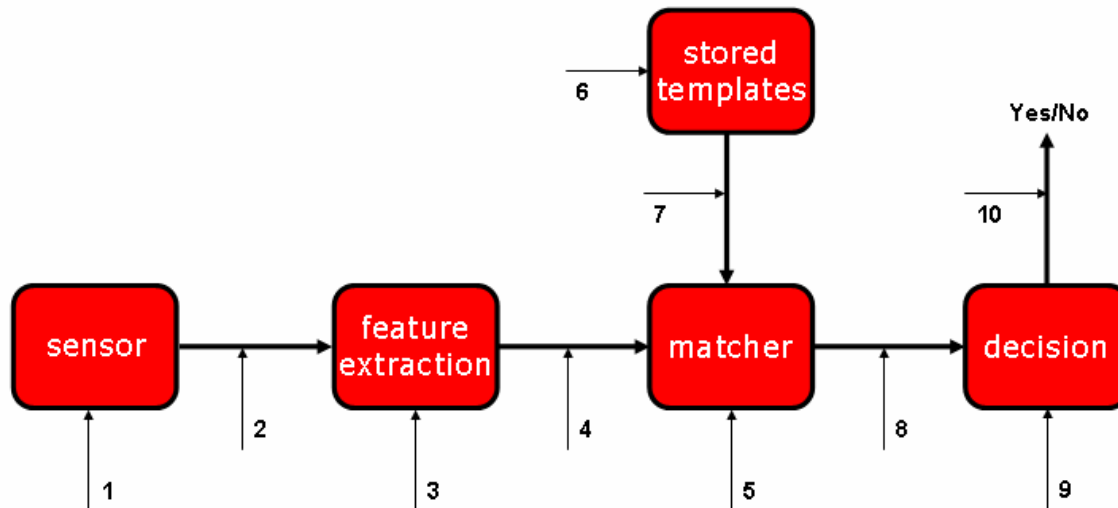
A generic biometric system can be cast in the framework of a pattern recognition system. The stages of such a generic system are shown in the Figure X below.

The first stage involves biometric signal acquisition from the user (e.g., the inkless fingerprint scan). The acquired signal typically varies significantly from presentation to presentation; hence, pure pixel-based matching techniques do not work reliably. For this reason, the second signal

processing stage attempts to construct a more invariant representation of this basic input signal (e.g., in terms of fingerprint minutiae). The invariant representation is often a spatial domain characteristic or a transform (frequency) domain characteristic, depending on the particular biometric; this is generally referred to as a template.

During enrollment of a subject in a biometric authentication system, an invariant template is stored in a database that represents the particular individual. To authenticate the user against a given ID, the corresponding template is retrieved from the database and matched against the template derived from a newly acquired input signal. The matcher arrives at a decision based on the closeness of these two templates while taking into account geometry, lighting, and other signal acquisition variables.

Note that password-based authentication systems can also be set in this framework. The keyboard becomes the input device. The password encryptor can be viewed as the feature extractor and the comparator as the matcher. The template database is equivalent to the encrypted password database. Ten (10) potential points of attack were identified in the figure below.



**Figure X**

1. Presenting fake biometrics at the sensor: In this mode of attack, a possible reproduction of the biometric feature is presented as input to the system. Examples include a fake finger, a copy of a signature, or a face mask.
2. Resubmitting previously stored digitized biometrics signals: In this mode of attack, a recorded signal is replayed to the system, bypassing the sensor. Examples include the presentation of an old copy of a fingerprint image or the presentation of a previously recorded audio signal.
3. Overriding the feature extraction process: The feature extractor is attacked using a Trojan horse, so that it produces feature sets pre-selected by the intruder.

4. Tampering with the biometric feature representation: The features extracted from the input signal are replaced with a different, fraudulent feature set (assuming the representation method is known). Often the two stages of feature extraction and matcher are inseparable and this mode of attack is extremely difficult. However, if minutiae are transmitted to a remote matcher (say, over the Internet) this threat is very real. One could “snoop” on the TCP/IP (Transmission Control Protocol/Internet Protocol) stack and alter certain packets.
5. Corrupting the matcher: The matcher is attacked and corrupted so that it produces pre-selected match scores.
6. Tampering with stored templates: The database of stored templates could be either local or remote. The data might be distributed over several servers or in tokens carried by users. Here the attacker could try to modify one or more templates in the database, which could result either in authorizing a fraudulent individual or denying service to the persons associated with the corrupted template. A smartcard-based authentication system, where the template is stored in the smartcard and presented to the authentication system, is particularly vulnerable to this type of attack.
7. Attacking the channel between the stored templates and the matcher: The stored templates are sent to the matcher through a communication channel. The data traveling through this channel could be intercepted and modified.
8. Overriding the score sent to the decision process: If the communication channel is hacked at this point, a different score could be sent to the decision process, causing a varied result.
9. Overriding the decision process: If the decision process can be altered by an attacker the threshold score could be changed to allow lower scores to produce a final match decision of “Yes”
10. Overriding the final decision: If the final match decision can be overridden by the hacker, then the authentication system has been disabled. Even if the actual pattern recognition framework has excellent performance characteristics, it has been rendered useless by the simple exercise of overriding the match result.

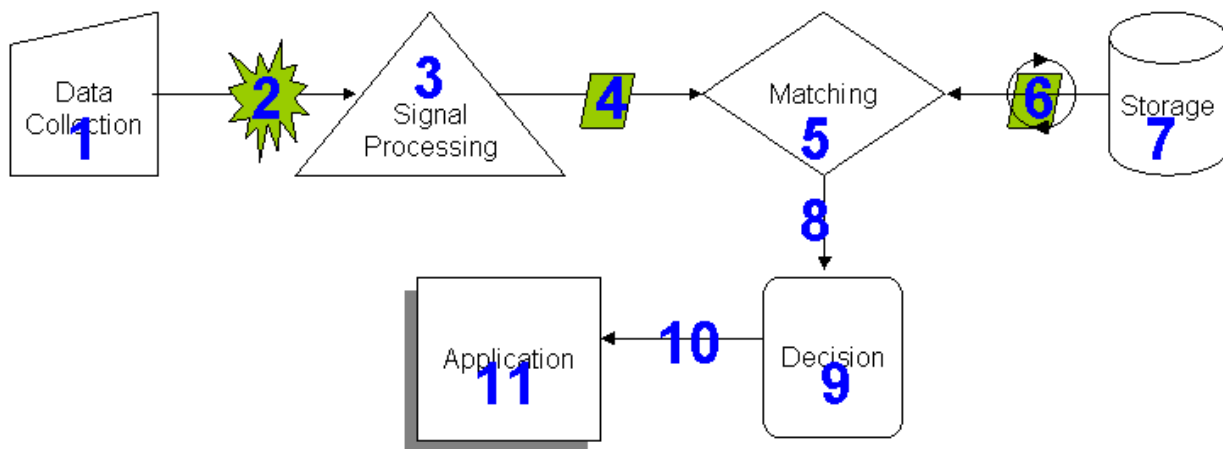
### **1.2.2 Employing Countermeasures**

There exist several security techniques to thwart attacks at these various points. For instance, finger conductivity or fingerprint pulse at the sensor can stop simple attacks at point 1. Encrypted communication channels can eliminate at least remote attacks at point 4. However, even if the hacker cannot penetrate the feature extraction module, the system is still vulnerable. The simplest way to stop attacks at points 5, 6, and 7 is to have the matcher and the database reside at a secure location. Of course, even this cannot prevent attacks in which there is collusion. Use of cryptography prevents attacks at transmission and storage points.

The threats outlined in the figure above are similar to the threats to password-based authentication systems. For instance, all the channel attacks are similar. One difference is that there is no “fake password” equivalent to the fake biometric attack at point 1 (although, perhaps if the password was in some standard dictionary it could be deemed “fake”). Furthermore, in a password- or token-based authentication system, no attempt is made to thwart replay attacks (since there is no expected variation of the “signal” from one presentation to another). However, in an automated biometric-based authentication system, one can check the liveness of the entity originating the input signal.

Clearly there are benefits and threats to using biometric technologies for e-Authentication. When compared to conventional authentication mechanisms such as PINS, Passwords, and Tokens; biometrics are stronger in some points and weaker in others. Based on this information, tables later in this document have been developed to show where biometric use is appropriate based on the assurance levels set forth in OMB 04-04 and NIST 800-63.

### 1.2.3 Threats & Countermeasures



Location	Threats	Countermeasures
1 Data Collection	Spoofing	Liveness detection
	Use of un-trusted device	Mutually authenticate device to server
2 Raw data transmission	Eavesdropping attack	Transmit data over encrypted path Mutually authenticate/use symmetric key
	Replay attack	Digitally sign data Utilize TTL tag
	Man in the middle attack	Bind biometric to PKI certificate
	Brute force attack	Time out/lock out policies

3 Signal Processing	Insertion of imposter data	Use strong tested algorithms
4 Processed data transmission	Eavesdropping attack	Transmit data over encrypted path Mutually authenticate/use symmetric key
	Replay attack	Digitally sign data Utilize TTL tag
	Man in the middle attack	Bind biometric to PKI certificate
	Brute force attack	Time out/lock out policies
5 Matching	Insertion of imposter data	Use strong tested algorithms
6 Template retrieval	Eavesdropping attack	Transmit data over encrypted path Mutually authenticate/use symmetric key
	Replay attack	Digitally sign data Utilize TTL tag
	Man in the middle attack	Bind biometric to PKI certificate
	Brute force attack	Time out/lock out policies
7 Storage	Database compromise	Hardened server Store encrypted templates Store template on smartcards or other device.
8 Value of matching score	Hill climbing attack	Incremental feedback
9 Decision	Hill climbing attack	Incremental feedback
10 Communication to application	Eavesdropping attack	Transmit data over encrypted path Mutually authenticate/use symmetric key
11 Application	Malicious code	Conform to standards (BioAPI, CBEFF)

### Possible Threats Specific to Biometrics for e-Authentication and Possible Countermeasures

Threats	Countermeasures
Biometrics cannot be changed when compromised	There have been several papers written in this area and continued research is occurring. The prevalent theory for thwarting this attack is to incorporate

	some kind of “wrapper” that can be changed if a biometric is compromised.
Biometrics are not secrets	This is an opinion, but can be assuaged when used in multi-factor implementations.
Biometrics do not possess and adequate degree of randomness	Again, this can be solved with some kind of “wrapper” that utilizes cryptography.
Spoofing attacks	Sensors are much less likely to be infiltrated by this kind of attack and vendors continue to make this more and more difficult by implementing various liveness detection and other anti-spoofing mechanisms.
Hill climbing attacks	This attack can be thwarted through quantization

We may want to address the role of the following security mechanisms:

- Encryption
- Signing
- Nonce's
- Timestamps
- Attribute certs
- Mutual authentication
- Trusted path / secure messaging
- Certified devices
- MOC
- Challenge/response
- Protocols

Consider role of a “Biometric CSP” for non-token based biometric implementations within [remote e-authentication architectures](#). ~~a remote e-Auth architecture.~~

### **1.3 Architectures analyzed by data transfer:**

#### **Biometric data being transferred:**

No matter what architecture is pursued, there is always going to be transfer of biometric data for remote electronic authentication. For most systems, there will be two distinct pieces of data which are being transferred are listed below. The principles of confidentiality and integrity should be applied to this data from its creation through its lifetime.

#### **Sample Data:**

The presented sample data which is used to create a biometric template of the user for use in future transactions. This can also be the sample which is presented in subsequent authentication attempts.

#### **Biometric Template:**

The processed data which is then stored and compared each time the user makes a biometric authentication attempt.

**Authentication determination information:**

As mentioned above, biometrics play only one role of identity systems, in most cases, the determination of the matching algorithm will have to be delivered to some other component of the system for use in granting privileges to the user.

**Data Transfer:**

The following diagram shows biometric data transfer for the individual architectures.

**NOTE:** The diagram only depicts data that **MUST** be transferred between the two devices in consideration for each architecture. This diagram does not address any middleware that might be in between the two entities listed. In some cases, some of the same components listed as possible storage and matching locations will be in the middle of In this diagram, ‘S’ indicates Sample and ‘T’ indicates Template data. The data being transferred will be of significant interest when addressing the threats of each individual architecture later in the report.

Store / Match	Server	Client	Device	Token
Server	Device S → Server	Server ← T Client	Server ← T Device	Server ← T Token
Client	Client ← T Server	Device S → Client	Client ← T Device	Client ← T Token
Device	Device ← T Server	Device ← T Client	Device S → Device	Device ← T Token
Token	Token ← T Server	Token ← T Client	Token ← T Device	Device S → Token

#### ***1.4 Architectures analyzed by threat and level:***

Based upon these parameters, the following architectures have been identified as the six most feasible architectures. These are shown below and also outlined in further detail as they relate to the assurance and security levels addressed in both OMB M04-04 and SP800-63.

Store Match	Server	Client	Device	Token
Server	Level 1 Level 2 Level 3 Level 4	Not <del>Being Pursued</del> <u>Included in this Report</u>	Not <del>Being Pursued</del> <u>Included in this Report</u>	Level 1 Level 2 Level 3 Level 4
Client	Not <del>Being Pursued</del> <u>Included in this Report</u>	Level 1 Level 2 Level 3 Level 4	Not <del>Being Pursued</del> <u>Included in this Report</u>	Not <del>Being Pursued</del> <u>Included in this Report</u>
Device	Not <del>Being Pursued</del> <u>Included in this Report</u>	Not <del>Being Pursued</del> <u>Included in this Report</u>	Level 1 Level 2 Level 3 Level 4	Level 1 Level 2 Level 3 Level 4
Token	Not <del>Being Pursued</del> <u>Included in this Report</u>	Not <del>Being Pursued</del> <u>Included in this Report</u>	Not <del>Being Pursued</del> <u>Included in this Report</u>	Level 1 Level 2 Level 3 Level 4

Store Match	Server	Client	Device	Token
Server	Level 1 Level 2 Level 3 Level 4	Not Being Pursued	Not Being Pursued	Level 1 Level 2 Level 3 Level 4
Client	Second Priority	Level 1 Level 2 Level 3 Level 4	Not Being Pursued	Second Priority
Device	Second Priority	Second Priority	Level 1 Level 2 Level 3 Level 4	Level 1 Level 2 Level 3 Level 4
Token	Second Priority	Not Being Pursued	Not Being Pursued	Level 1 Level 2 Level 3 Level 4

NOTE: The intent here will be to first analyze all threats and associated countermeasures and categorize them as applying to a given security level (i.e., what threats should you be concerned with at each level – if there are 20 threats identified, for example, perhaps there are 4 that are added at each level). Then analyze each selected architecture in terms of threats/vulnerabilities (perhaps identifying only those uniquely present or missing from the generic – for example, attack on templates in a server DB would not apply to a store/match on token scenario). From this, it is hoped that we can determine which architectures are suitable at each level.

Eventually, we should have a diagram for each architecture.

#### 1.4.1 Store on Server (A)

**Match on Server:** This architecture has some extreme advantages and disadvantages. Centralized storage allows for easier management; however it also creates a single point of failure and attack.

**Sample Transferred:**

From the remote sensor to the server

**Template Transferred:**

Internal on the server from database to matching algorithm

**Authentication Determination Transferred:**

**If the matching function is performed on a centralized server, there is a good chance the information about the authentication determination will not need to travel outside of the trusted environment.**

**Specific Threats:**

1. Database compromise
2. Denial of Service attack

**Specific Countermeasures:**

1. Hardened server
2. Store encrypted templates

**Assurance:**

**Level 1:** YES

**Level 2:** YES

**Level 3:** YES: As long as there is multi-factor authentication

**Level 4:** YES: As long as a hard crypto token is used

## 1.4.2 Store on Client (B)

**Match on Client:**

This architecture would be a simple way to use biometrics for website log-ins and transactions. Further more, if the client is truly trusted, it would promote a starting point for Single Sign On solutions.

**Sample Transferred:**

From the remote sensor to the client

**Template Transferred:**

Internal on the client from database to matching algorithm

**Authentication Determination Transferred:**

**If the matching function is performed on a remote client, there is a good chance the information about the authentication determination will need to still travel over an un-trusted network to reach its final destination for eventual use in the security system.**

**Specific Threats:**

1. Replay attack on the client
2. Hill climbing attempt

**Specific Countermeasures:**

1. Use TTL tag
2. Implement incremental feedback to the user

**Assurance:**

**Level 1:** YES

**Level 2:** YES

**Level 3:** YES: As long as there is multi-factor authentication

**Level 4:** YES: As long as a hard crypto token is used

### 1.4.3 Store on Device (C)

**Match on Device:**

This architecture would be ideal for remote physical access devices that are being monitored and communicating over the internet. Using the device as the computing platform creates a greater degree of independence.

**Sample Transferred:**

From the remote sensor to the device.

**Template Transferred:**

Internal on the device from database to matching algorithm.

**Authentication Determination Transferred:**

**If the matching function is performed on a remote device, there is a good chance the information about the authentication determination will need to still travel over an un-trusted network to reach its final destination for eventual use in the security system.**

**Specific Threats:**

1. Spoofing
2. Hill climbing attack

**Specific Countermeasures:**

1. Live ness detection
2. Implement incremental feedback to the user

**Assurance:**

**Level 1:** YES

**Level 2:** YES

**Level 3:** YES: As long as there is multi-factor authentication

**Level 4:** YES: As long as a hard crypto token is used

### 1.4.4 Store on Token

**Match on Server (D):**

Currently this is the NSA preferred method for two reasons: there is no centralized storage as a single point of attack. The server is in charge of signing the tokens before they are deployed, this provides for easier management and revocation. This architecture does contain the requirement to transfer both the stored template and presented sample each time an authentication attempt is made.

**Sample Transferred:**

From the remote sensor to the server

**Template Transferred:**

From the token to the server

**Authentication Determination Transferred:**

**If the matching function is performed on a centralized server, there is a good chance the information about the authentication determination will not need to travel outside of the trusted environment.**

**Specific Threats:**

1. Eavesdropping attack on either of the two communication channels
2. Insertion of imposter data on either of the two communication channels

**Specific Countermeasures:**

1. Enforce strong data protection during communication
2. Implement means in which the template can be verified as valid when returned to the server.

**Assurance:**

**Level 1:** YES

**Level 2:** YES

**Level 3:** YES: As long as there is multi-factor authentication

**Level 4:** YES: The verifier is a hard crypto token in and of itself.

**Match on Device (EF):** This would allow for a single trusted device that is also a token and biometric reader. The most obvious use of this architecture would be an all encompassing cell phone device.

**Sample Transferred:**

Internal from the sensor on the device to the matching algorithm on the same device.

**Template Transferred:**

Internal from the database to the matching algorithm on the device.

**Authentication Determination Transferred:**

**If the matching function is performed on a remote device; there is a good chance the information about the authentication determination will need to still travel over an un-trusted network to reach its final destination for eventual use in the security system.**

**Specific Threats:**

1. Spoofing
2. Physical attacks to the device

**Specific Countermeasures:**

1. Live ness detection
2. Require tamper resistant devices to prevent disclosure of sensitive information

**Assurance:**

**Level 1:** YES

**Level 2:** YES

**Level 3:** YES: As long as there is multi-factor authentication

**Level 4:** YES: The verifier is a hard crypto token in and of it self.

**Match on Token (FE):** This would be a biometric PIN replacement. This architecture is most similar to the way biometrics is viewed as being acceptable for use by NIST SP800-63. Certified authentication match of the biometric characteristic can “unlock” another form of authentication which is released to the system

**Sample Transferred:**

From the sensor to the matching algorithm on the token.

**Template Transferred:**

Internal from the database to the matching algorithm on the token.

**Authentication Determination Transferred:**

**If the matching function is performed on a remote token; there is a good chance the information about the authentication determination will need to still travel over an un-trusted network to reach its final destination for eventual use in the security system.**

**Specific Threats:**

1. Spoofing
2. Physical attacks to the device

**Specific Countermeasures:**

1. Live ness detection
2. Require tamper resistant devices to prevent disclosure of sensitive information

**Assurance:**

**Level 1:** YES

**Level 2:** YES

**Level 3:** YES: As long as there is multi-factor authentication

**Level 4:** YES: The verifier is a hard crypto token in and of it self.

### Architecture to Assurance Level Compliance Matrix

	Assurance Level 1	Assurance Level 2	Assurance Level 3	Assurance Level 4
<b>Architecture A</b>	Yes	Yes	Yes, if used with multi-factor authentication	Yes, if used with hard crypto token
<b>Architecture B</b>	Yes	Yes	Yes, if used with multi-factor authentication	Yes, if used with hard crypto token
<b>Architecture C</b>	Yes	Yes	Yes, if used with multi-factor authentication	Yes, if used with hard crypto token
<b>Architecture D</b>	Yes	Yes	Yes, if used with multi-factor authentication	Yes, if used with hard crypto token
<b>Architecture E</b>	Yes	Yes	Yes, if used with multi-factor authentication	<u>Yes, the verifier is a hard crypto token in and of it self</u> Yes, if used with hard crypto token
<u><b>Architecture F</b></u>	<u>Yes</u>	<u>Yes</u>	<u>Yes, if used with multi-factor authentication</u>	<u>Yes, the verifier is a hard crypto token in and of it self</u>

## 1.5 Considerations

### 1.5.1 Trust

One of the key aspects of consideration is the amount of trust and confidence between the two entities which are interacting to achieve remote e-authentication. The amount of trust in the end to end system will be a determining factor in which assurance levels can be achieved.

#### “Semi-Open”

Both remote and centralized entities are part of the same organization, but the data must be traversed over the internet or some sort of un-trusted network.

#### “Completely Open”

Remote entity has no relationship with the centralized entity from an information technology perspective.

These two architectures can most closely be related to the modern example of VPN technologies. The “semi-open” architecture would be similar to an office to office VPN where both entities are at a high level of mutual trust.

The “completely open” architecture would be similar to an employee connecting remotely via VPN to the main corporate headquarters from an airport internet kiosk. In this case, the organization must initially accept all initial VPN requests because all the possible origins of VPN connection can not be pre-determined. The level of trust in this architecture is lower because it is reliant solely on the claimant provided credentials.

### **1.5.2 Multi-factor authentication**

The verification location for each individual credential being authenticated is important to note when discussing multi-factor authentication.

As indicated in NIST SP800-63, biometrics or passwords can be used to “unlock” or activate a hard token which can then be sent to the system in achieving level 4 authentication.

**Currently as documented:** Allowed

**Environments affected:**

- Store on Server, Match on Client. Assuming the client is not authenticating the hard crypto token but simply passing it to the system along with biometric match determination.
- Store on Client, Match on Client. Assuming the client is not authenticating the hard crypto token but simply passing it to the system along with biometric match determination.
- Store on Device, Match on Device. Assuming the device is not authenticating the hard crypto token but simply passing it to the system along with biometric match determination.

Matching of the sample provided against the stored template on the hard crypto token itself is currently viewed as acceptable multi-factor environment.

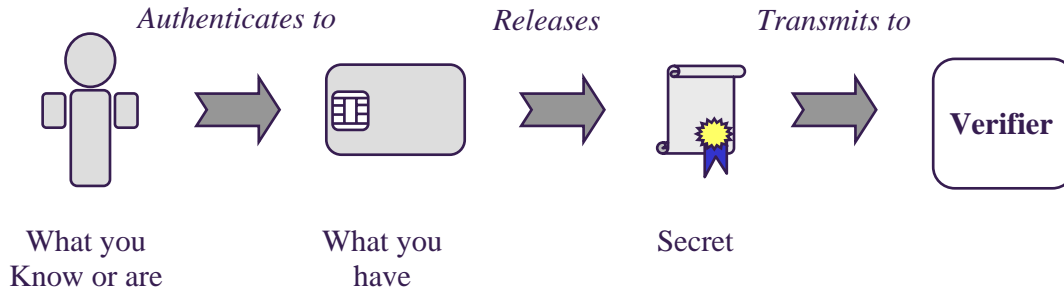
**Currently as documented:** Allowed

**Environments affected:**

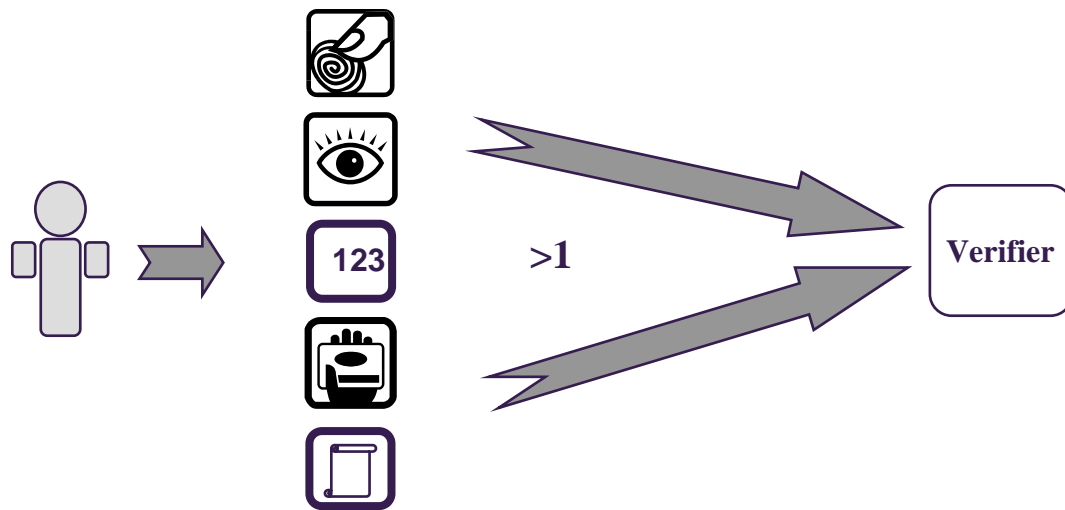
- Store on Token, Match on Server. The verifier is a hard crypto token in and of it self.
- Store on Token, Match on Token. The verifier is a hard crypto token in and of it self.

It should be noted in discussing multi-factor authentication, that there are two methods of implementing this – serial (chained) or parallel (concurrent).

In the chained approach, one factor activates/enables a second factor which is what is presented to the verifier. This is depicted below:



In the concurrent approach, both factors are provided by the user and are independently verified at the verifier, as shown below:



In 800-63, the use of biometrics at Levels 3 & 4 are via the chained method, where the biometric is used to release the cryptographic authentication token (soft or hard cert). A case could be made that this is not as strong as a concurrent approach, as stated in the following:

An authentication protocol must be analyzed from the perspective of the relying party (the Verifier) in an information infrastructure. For an authentication transaction to be "multi-factor", the relying party must be able to consider and validate each form of identity assurance independently. In fact, the introduction to Section 5 of SP 800-63 correctly describes the E-Authentication Model as "When a claimant successfully demonstrates possession and control of a token in an on-line authentication to a verifier through an authentication protocol". While using a PIN or password protected hard token might produce a higher level of trust for the token from a global perspective, it does not represent multi-factor authentication to the Verifier, since it is impossible to independently validate the PIN or password with respect to the token itself, or with respect to the identity being claimed. Nothing in this context construes an irrevocable connection between a user and a claim of identity, nor can it demonstrate the will or intent of the user - an important aspect of non-repudiation in the common law sense. Since the Verifier cannot validate the token and the PIN or password independently, the PIN/password protected

hard token represents only a single authentication factor in the authentication protocol. However, if the PIN/password is validated by the relying party, along with the validation of another token (like a PKI certificate), the authentication process is then truly multi-factor, satisfying section 8.2.4 of SP 800-63 “Authentication requires that the claimant shall prove through a secure authentication protocol that he controls the token.”