A Survey of Template Protection Schemes and What One Might do With Them

Walter Scheirer & Terrance Boult
“Biometrics: Practical Issues in Privacy and Security”
IJCB 2011
Security Basics
Template Protection as a Solution

- Protect the Privacy and Security of the Biometric Features
- Revoke and re-issue biometric templates like a password or credit card #
- Match in an encoded space
- Prevent linking across databases (solve the biometric dilemma)
- Prevent the doppelganger attack (multi-factors)

“Getting this right has been much more challenging than we first thought.” – Fabian Monrose
Lots of stuff out there!

- Biometric Encryption
- Non-invertible Transforms
- BioHashing
- Robust Hashing
- Fuzzy Vaults
- Fuzzy Commitment
- Fuzzy Extractors
- Revocable Biotokens
- Hybrid Combinations

How do they work?
How well do they work?
How secure are they?
General Categories*

- Straight feature protection
- Key-generating
- Key-binding

Straight Feature Protection

- Simply protect the original biometric features using some transformation that allows matching in encoded space.

```
Enrollment

Transform

Secure Enrollment Template

Match?

Dynamic Verification Template

Verification

Transform
```
Key-binding

• Biometric cryptosystem that binds key data with the biometric data
Key-generating

- Biometric cryptosystem that derives a key from the biometric data
Attacks Against Secure Template Protection Technologies

• Basic Brute Force
• Correlation Attack*
• Known Key Attack*
• Substitution Attacks*

• Decodability Attack
• Doppelganger Attack
• Hill Climbing

Basic Brute Force

- Attacker tries every possible bit combination till they guess the correct original feature data or key
  - Need a way to test each bit combo

<table>
<thead>
<tr>
<th>Bit Combos</th>
</tr>
</thead>
<tbody>
<tr>
<td>00001...</td>
</tr>
<tr>
<td>00010...</td>
</tr>
<tr>
<td>00011...</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>11111...</td>
</tr>
</tbody>
</table>
Correlation Attack

Enrollment

$X_1 \rightarrow F_1(\kappa_1)$

$X_2 \rightarrow F_2(\kappa_2)$

$X_n \rightarrow F_n(\kappa_n)$

Stolen $X$ and $\kappa_1..\kappa_n$
Known Key Attack

Key $\kappa$ is used by owner and obtained by attacker

$Stolen \: X$ attacker

$F(\kappa)$

database
Substitution Attacks

“How difficult will it be to break into a folder containing biometric signatures and replace them with an attacker's biometric signature so that the attacker can get in with his/her own signature easily?”

Decodability Attack

- Exploit available information to link across databases*
- Assume a template $W$ contains helper data $H$ and biometric data $X$
  
  \[-W_1 = H_1 \oplus X_1; W_2 = H_2 \oplus X_2\]
- If $W_1 \oplus W_2$ is decodable, the two templates are probably derived from the same person

---

The Doppelganger Threat

• If the FAR is 1 in $X$, then an attacker can try more than $X$ different prints

• Lots of public data available!
  – Fingerprint: NIST DB 14, NIST DB 29, FVC 2002, FVC 2004 ...
  – Face: MBGC, FRGC, FVT, FERET...
  – Think of this as a biometric dictionary attack
Information Theoretical Security Analysis vs. Practical Matching Security

• A disconnect exists between information theoretical security models and matching accuracy
  – Both are important!
• Information leakage is bounded by matching accuracy
  – If a false match to a template releases the correct key, the system leaks 100% of the key information
    • ECC often overcorrects, which drives up the FAR
Hill Climbing

• Requires less than brute-force effort to recover an embedded secret
• Provides an estimate of the enrollment image

In an iterative fashion, modifications are made to the input, and those that increase the match score are retained.
Prevalent Template
Protection Schemes
Fuzzy Vaults

- Not specific to biometric data, but typically applied to minutiae based fingerprint matchers as a key binding biometric cryptosystem

Encoding

Fuzzy Vaults

Decoding
## Performance Numbers

<table>
<thead>
<tr>
<th></th>
<th>112 Bits</th>
<th></th>
<th>128 Bits</th>
<th></th>
<th>160 Bits</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GAR</td>
<td>FAR</td>
<td>GAR</td>
<td>FAR</td>
<td>GAR</td>
<td>FAR</td>
</tr>
<tr>
<td>F.P. Fuzzy Vaults$^1$</td>
<td>89</td>
<td>0.13</td>
<td>89</td>
<td>0.01</td>
<td>84</td>
<td>0</td>
</tr>
<tr>
<td>F.P. FV, Mosaic with 2 Queries$^1$</td>
<td>96</td>
<td>0.24</td>
<td>95</td>
<td>0.04</td>
<td>89</td>
<td>0</td>
</tr>
<tr>
<td>Password Vault$^2$</td>
<td>88</td>
<td>?</td>
<td>86</td>
<td>?</td>
<td>79</td>
<td>?</td>
</tr>
</tbody>
</table>

Fuzzy Vaults: Security Problems

- Chaff Point Identification\(^1\)
- Improved Brute Force Attack\(^2\)
- Correlation Attack, Known Key Attack, Correlation Attacks
- Hill Climbing
  - May be theoretically possible
    - Security proof assumes data held in the vault is random; not the case with biometrics
    - Chaff is placed carefully so as not to conflict with legitimate points; strays from randomness assumption

Fuzzy Vaults: Correlation Attack

- Without a matching sample, the polynomial reconstruction problem is infeasible to solve.
- What if we have *two or more* BFV instances?
  - Take the intersection of the abscissa \( (x) \) values for the BFV instances.
  - The result is the original template data.
  - Some chaff points are likely to match - but the error correcting code is designed for this possibility.
Fuzzy Vaults: Known Key Attack

• From $\kappa$, the polynomial $p$ is directly reconstructed
• $R$ may be directly enumerated to separate the template data, in the form $(A, p(A))$, from the chaff
• Again, the error correcting code will help us if some chaff matches
Fuzzy Vaults: Substitution Attacks

- Most of the vault is chaff. Matching uses only a small fraction of real data hidden in it.
- Overwrite chaff lines with attacker’s template data, encoding $X_A$ and $\kappa_A$
- Resulting template has both the user’s and attacker’s data.
- Insidious attack - attacker encodes their data with the user’s key $\kappa_U$
Response To Vulnerabilities in Fuzzy Vaults

- Password Hardened Fuzzy Vault*

*Karthik Nandakumar, Abhishek Nagar and Anil K. Jain, “Hardening Fuzzy Vault Using Pasword”, in Proc. of ICB 2007 (and image credit)
Response to Vulnerabilities in Fuzzy Vaults

- Fuzzy Commitment to “encrypt” polynomial evaluations¹
- Carefully chosen chaff²
- Incorporate local ridge information of minutiae (also incorporates a password)³
- Distance preserving hash functions⁴

Fuzzy Commitment

• Another well known key binding approach*
• Enrollment
  – Commit a codeword $C$ (acts as the key) of an error correcting code using a fixed length biometric feature vector $X$ as a witness
  – Store a hash $h$ of $C$ as “helper data”
  – Fuzzy Commitment: $X \oplus C, h(C)$

Fuzzy Commitment

• Verification
  – User presents a biometric, producing feature vector $X'$
  – $X'$ is then used to unlock the codeword
    • $(X \oplus C) \oplus X' = C' = C \oplus e$
    • Hamming distance $d_H$ indicates the number of errors corrupting $C$
      – $\epsilon = d_H(X, X') = ||e||$
    • An ECC Decoder can correct errors, yielding an extracted candidate key $K$
    • A successful match occurs when $h(K) = h(C)$
Illustration of Fuzzy Commitment

Grid of small dots: word space $\{0,1\}^n$
Bigger dots: codewords from $C$ with the error correcting capability of the circles with radius $t_c$

Performance Numbers

<table>
<thead>
<tr>
<th></th>
<th>FVC/CASIA/XM2VTS</th>
<th>WVU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>37%</td>
<td>91%</td>
</tr>
<tr>
<td>Face</td>
<td>30%</td>
<td>2%</td>
</tr>
<tr>
<td>Finger</td>
<td>33%</td>
<td>12%</td>
</tr>
</tbody>
</table>

Comparison of GAR at 53 bits of security

Performance Numbers

- 3-layer coding scheme\(^1\): ERR of 6.5% for 1032 bit key on FVC2000 DB2
- Multibiometric Fusion\(^2\):

<table>
<thead>
<tr>
<th></th>
<th>FVC/CASIA/XM2VTS</th>
<th>WVU</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND Rule</td>
<td>27%</td>
<td>89%</td>
</tr>
<tr>
<td>“Multibiometric Cryptosystem”</td>
<td>75%</td>
<td>99%</td>
</tr>
</tbody>
</table>

Comparison of GAR at 53 bits of security

- Bringer et al. 2008\(^3\) for 2028 bit keys:
  - ICE: FRR 5.62%, FAR < 10\(^{-5}\)
  - CASIA: FRR 6.65%, FAR 0%
  - FVC 2000: FRR 2.73%, FAR 5.53%

---

1. X. Shao et al., “A 3-layer Coding Scheme for Biometry Template Protection Based on Spectral Minutiae”, ICASSP, 2011.
Fuzzy Commitment: Security Problem

- **Decodability Attack**
  - Codewords: \( C_1, C_2 \)
  - Biometric Data: \( X_1, X_2 \)
  - \( W_1 = C_1 \oplus X_1; W_2 = C_2 \oplus X_2 \)
  - \( W_1 \oplus W_2 = (C_1 \oplus C_2) \oplus (X_1 \oplus X_2) = C_3 \oplus (X_1 \oplus X_2) \)
  - If \( (X_1 \oplus X_2) \) is small, the result of the XOR will be close to another codeword (decodes)

---

Response to Vulnerabilities in Fuzzy Commitment*

- Incorporate random bit permutation process
- Prior to the XOR operation of the biometric data $X$ with the code word $C$, randomize $X$ with a bit permutation matrix $M_r$
- The new template: $W = C \oplus M_rX$
- $M_r$ is not considered a secret

Fuzzy Extractors

- Key generating biometric cryptosystem*
- Attractive proposition, but difficult due to intra-user variability
- Goal: Extract a uniformly random string $R$ from its input $w$ in a noise-tolerant way
  - If the input changes to some $w'$, but remains close, the string $R$ can still be reproduced exactly

Secure Sketch*

- “Helper Data” for Fuzzy Extractors
- A secure sketch produces public information about its input $w$ that does not reveal $w$, and yet allows exact recovery of $w$ given another value that is close to $w$.

Fuzzy Extractors

- A secure sketch SS producing a string $s$ bound with a random number $x$ forms the basis of the helper string $P$
- Recovery procedure allows matching with a “close” string $w'$
- Extractor returns a string $R$, the key, when approximate input matching is successful
- $P$ assists in the reproduction of $R$

Sketching Procedure

Recovery Procedure

$r$ is some randomness
Security Analysis: Fuzzy Extractors

• Security analysis of the fuzzy extractor scheme made in terms of the \textit{min-entropy}

• An adversary’s best strategy is to guess the most likely value
  – Predictability of a random variable
  – Min-entropy is the “worst case” entropy

• Information theoretical balance between stability and suitable randomness

*Analysis is not made with consideration to FAR/GAR!*
Practical Concerns

• At the present, fuzzy extractors exist in the realm of theory
• Fuzzy extractors may suffer from practical constraints during error-prone data collection; difficulty for key generation
  – Unclear whether known constructions can correct the errors typically generated by humans
  – Require biometric inputs with high min-entropy, but haven’t discussed feature selection

What’s so difficult about all of these “fuzzy” techniques?

• In essence, if biometric features are not aligned properly, these schemes fail to work

• Solution* for fingerprint fuzzy vaults: helper data
  – accurately aligns the template and query minutiae, but does not reveal any information about the minutiae points - larger templates

What’s so difficult about all of these “fuzzy” techniques?

• Fuzzy Commitment requires a fixed length feature vector representation of a biometric modality
  – Minutia-based representation will not work

One approach*: Fourier-Mellin Transform; invariant to translation, scaling and rotations become translations

Revocable Biotokens

- We want two different things:
  - Robust distance/matching
  - Security/Revocability

→ Break data into two parts:
  - **Stable** and **Unstable**

- Stable part is encryptedhashed to provide security/privacy and revocability - straight feature protection
- Two parts together provide robust distance measure, which we can prove will not decrease accuracy
Revocable Biotokens*

– Assume a biometric produces a value $v$ that is transformed via scaling and translation
  • $v' = (v - t) \times s$
– Split $v'$ into stable component $q$ and residual component $r$
– For user $j$, leave the residual un-encoded (base scheme)
  • $r_j(v')$
– Encrypt $q$ with public key $P$
  • $w_{j,1}(v', P)$

Brute Force Attack to revert biotoken back to original features: $2^{108}$ for insider, $2^{120}$ without access to all keys/data

Revocable Biotoken Performance*

Nesting Property

• $w_j$ is re-encoded using a transformation function $T$
  
  1$^{\text{st}}$ encoding: $w_{j,1}(v', P)$
  
  2$^{\text{nd}}$ encoding: $w_{j,2}(w_{j,1}, T_2)$
  
  $n$th encoding: $w_{j,n}(w_{j,n-1}, T_n)$

• The nesting process is formally invertible via the keys, but cryptographically secure
The Goal: Transactions
Does “nesting” apply to other secure template technologies?

- Fuzzy Vaults have already been “cracked,” but...

- Any nesting of a fuzzy vault (with or without passwords) would have to be able to identify and then modify the data and the embedded key, which means the nesting system effectively knows the “secrets” and hence can compromise the security and privacy protection of the data.

Remote Server

\[ F_1(\kappa_1), \quad F_2(\kappa_2), \quad \ldots, \quad F_n(\kappa_n) \]
Does “nesting” apply to other secure template technologies?

• Fuzzy commitment is reasonably secure
  – But its base formulation does not possess a nesting property

• The feature data $X$ is always needed when changing keys

\[ W_1 = C_1 \oplus X; \]
\[ W_2 = C_2 \oplus X; \]
…
\[ W_n = C_n \oplus X \]
Does “nesting” apply to other secure template technologies?

- Fuzzy extractors* theoretically provide secure template protection.
  - But they do not possess a nesting property

Lemma 5.1*

Suppose we compose an $(m, \tilde{m}, t)$-secure sketch, $(SS, REC)$ for a space $M$ and a universal hash function $EXT : M \rightarrow \{0,1\}^l$ as follows: In $Gen$, choose a random $i$ and let $P = (SS(w), i)$ and $R = Ext(w; i)$; let $Rep(w', (s, i)) = Ext(Rec(w', s), i)$. The result is an $(m, l, t, \varepsilon)$-fuzzy extractor with $l = \tilde{m} + 2 - 2\log(1/\varepsilon)$.

One needs the original biometric data $w$ and a random $i$ to create a new instance of a fuzzy extractor!

Bipartite Biotokens

– Let $B$ be a revocable biotoken. A bipartite biotoken* $B_p$ is a transformation $bb_{j,k}$ of user $j$’s $k^{th}$ instance of $B$. Any bipartite biotoken $B_{p,k}$ can match any revocable biotoken $B_k$ for the same user.

– $bb_{j,k}$ must allow the embedding of some data $d$ into $B_p$
  • $bb_{j,k}(w_{j,k}, T_k, d)$

– If $B_{p,k}$ and $B_k$ match, $d$ is released

## Experimental Results

<table>
<thead>
<tr>
<th>FVC02 DB 2</th>
<th>112 Bits</th>
<th>128 Bits</th>
<th>160 Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GAR</td>
<td>FAR</td>
<td>GAR</td>
</tr>
<tr>
<td>F.P. Fuzzy Vaults&lt;sup&gt;1&lt;/sup&gt;</td>
<td>89</td>
<td>0.13</td>
<td>89</td>
</tr>
<tr>
<td>F.P. FV, Mosaic with 2 Queries&lt;sup&gt;1&lt;/sup&gt;</td>
<td>96</td>
<td>0.24</td>
<td>95</td>
</tr>
<tr>
<td>Password Vault&lt;sup&gt;2&lt;/sup&gt;</td>
<td>88</td>
<td>?</td>
<td>86</td>
</tr>
<tr>
<td>Bipartite Biotokens</td>
<td>97</td>
<td>0</td>
<td>97</td>
</tr>
</tbody>
</table>

### Comparison with Fuzzy Faults

<table>
<thead>
<tr>
<th>FVC02 DB #</th>
<th>192 Bits</th>
<th>256 Bits</th>
<th>512 Bits</th>
<th>1024 Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GAR</td>
<td>ECC</td>
<td>GAR</td>
<td>ECC</td>
</tr>
<tr>
<td>1</td>
<td>97</td>
<td>5</td>
<td>94</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>97</td>
<td>2</td>
<td>97</td>
<td>2</td>
</tr>
</tbody>
</table>

### Larger Key Sizes


The Big Test

• The Doppleganger Attack
  – If the FAR is 1 in $X$, then an attacker can try more than $X$ attempts

• A very large impostor test
  – Mixed combinations of FVC 2002, FVC 2004, NIST DB 14 and NIST DB 29
  – 6 bytes of ECC, 128 bit, 256 bit, and 512 bit keys, 8000 byte probe/gallery biotokens

Zero False Accepts from processing over 1 Billion impostor trials to date!
Protocols and Applications
Security in the “Cloud”

• The model isn’t new: an updated version of timesharing from the 1960s…
  – Many popular services have always been in the cloud

  • Gmail
  • Facebook
  • Paypal
  • Dropbox

• What is different from the PC model: the trust boundary shifts one step further away
Risks of the Cloud

“If you entrust your data to others, they can let you down or outright betray you.”

- Misplaced, stolen or sold data
- Less privacy protection in practice and under the law
- Vendor defines how much control a user has over their own data

Risks of the Cloud

• Is it dangerous to move biometric data to the cloud?
  – Maybe Not

• The key issue*: another level of trust
  – When a computer is on your network, you control the security mechanisms
  – There should be some facility for the owner to protect and control their data

Biometric Solution?

- By adding a second factor, we can mitigate the inherent trust problem with the cloud model

What about Biometrics?
- Improved non-repudiation
- Strong verification for actors in a transaction
- Strong verification for PKI-like functionality
  - certificate authority establishment, and general certificate issue

Address the trouble with Biometrics using Template Protection
Biocryptographic Key Infrastructure

• Solution to both traditional and biometric data management in the cloud
• Analogous to PKI, but incorporates biometrics and template protection to establish identity beyond certificates

Public Key Infrastructure enables asymmetric secure machine-to-machine communication, but it does not solve Identity Issues. We need asymmetric verification.
Public Key Infrastructure

- PKI is the infrastructure for handling the complete management of digital certificates (x.509 compliant)
  - Certificates contain trusted information: a public key
Benefit of a BKI

- Ability to store public biotokens in digital certificates
  - Any entity in the infrastructure can send secret data that only the owner of the biotoken can unlock
Requirements for a Biocryptographic Key Infrastructure

1. Cryptographically strong protection of the underlying biometric features
2. Ability to revoke and re-issue templates
3. Nested re-encoding, allowing a hierarchy of templates to be generated from a single base template
4. Support for public templates
5. Key-binding capability without the need of intervention by the person associated with the template
Can a BKI be supported by other technologies besides revocable biotokens?

- Fuzzy Extractors support key transfer\(^1\), but not unique transactions
- Kanade et al.\(^2\) proposed a scheme for key-binding without re-enrollment
  - secret key + error correction \(\Theta_{ps} \oplus\) shuffled biometric data \(\Theta_{canc} = \Theta_{lock}\)
  - Vulnerable to the SKI Attack: If an attacker knows \(\Theta_{ps}\), then \(\Theta_{ps} \oplus \Theta_{lock} = \Theta_{canc}\)

---

Biotoken Issue/Re-Issue Tree

- **Enrollment**
  - **Root Biotoken**: Can be used for duplicate enrollment check, making token useful for recognition or verification.
  - **Master Biotoken**: Unique per application/database. Verification only token.
  - **Operational Biotoken**: Changed regularly like date-driven credit card expiration. Verification only token.
  - **Bipartite Biotoken**: Unique per transaction. Supports secure key release. Verification only token.
Digital Cert. Supporting Biotokens

**x.509 v3 digital certificate**

- Version
- Serial Number
  - Algorithm ID
- Issuer
- Validity
  - Not Before Date
  - Not After Date
- Subject
  - Subject Public Key Info
    - Public Key Algorithm
    - Parameters
    - Subject's Public Key
- Issuer Unique Identifier (optional)
- Subject Unique Identifier (optional)

**Biotoken Extensions**

- Certificate Signature Algorithm
- Certificate Signature

- Online Only Flag
- Standalone Only Flag
- Subject's Biotoken
  - Biotoken Type
  - Biotoken
A Biocryptographic Key Infrastructure
Certificate Retrieval Path

Root BCA, authorizes all BCAs below

Certificate signed by BCA_A, signs BCA_B’s certificate

Certificate signed by BCA_A

Alice’s certificate, including her public key and biotoken, is certified

Alice

BCA_A

BCA_C

BCA_D

BCA_E

Bob’s certificate, including his public key and biotoken, is certified

Bob
Three Authentication Protocols

- 1-Way protocol: establishes identity and trust of Receiver
- 2-Way protocol: assures send that Receiver is not impostor
- 3-Way protocol: validates both identities in the transaction
Certificate Revocation

• We must consider certificate and biometric re-issue

• Scenario 1: Manual re-issue
  – Certificate owner generates a new public-private key pair and a new biotoken

• Scenario 2: Automatic re-issue of biotoken
  – BCA retains transformation keys, reverts public biotoken to a lower level, issues new transformation keys and public biotoken

• Scenario 3: Automatic re-issue of key-pair
  – BCA issues new key-pair, transmits secret key to owner via bipartite biotoken
# CRN Message

## Certificate Re-issue Notification

<table>
<thead>
<tr>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial Number</td>
</tr>
<tr>
<td>New Serial Number</td>
</tr>
<tr>
<td>Biotoken Re-issued Flag</td>
</tr>
<tr>
<td>Key-pair Re-issued Flag</td>
</tr>
<tr>
<td>Biotoken and Key-pair Revoked Flag</td>
</tr>
<tr>
<td>*Keyring for Biotoken (Optional)</td>
</tr>
<tr>
<td>Biotoken Type (Optional)</td>
</tr>
<tr>
<td>Biotoken (Optional)</td>
</tr>
<tr>
<td>Signature</td>
</tr>
</tbody>
</table>

*Keyring is encrypted with the user’s public key*
New Applications

• Authenticate to the cloud
• Manage your own data in the cloud

And also:
• Thwart Man-in-the-Middle and Phishing attacks!
• Bio-Kerberos
• Bio-S/Key
• BKI-enabled LDAP
• Biometric Digital Signatures
• Mobile Biometrics

The BKI bring identity to crypto protocols
Commercial Solutions

• GenKey (http://www.priv-id.com)
  – Fuzzy Commitment (?)
• Securics (http://www.securics.com)
  – Revocable Biotokens
• Offering a host of privacy related software products
  – BioHASH SecureID SDK
  – BioHASH Match-on-Card SDK
  – Biometric ID Management System
• Established research group with strong publication record
  – Published work through Philips and the University of Twente
  – “Security with Noisy Data*” is even advertised on their site!

• Multiple products built around Revocable Biotokens and the BKI
• We’ve published the details as Securics, Inc. and the University of Colorado
• Have questions about our technology???
  – Please ask!
Questions?