Protected Honey Face Templates

Edlira Martiri[1], Bian Yang[2], Christoph Busch[3]
{edlira.martiri; bian.yang; christoph.busch}@hig.no
[1,2]Norwegian Biometrics Laboratory, Gjøvik University College, 2815 Gjøvik, Norway
[3]Hochschule Darmstadt - CASED, Haardtring 100, 64295 Darmstadt, Germany

Abstract. Most existing biometric template protection schemes (BTPS) do not provide as strong security as cryptographic tools; and furthermore, they are rarely able to detect during a verification process whether a probe template has been leaked from the database or not (i.e., being used by an imposter or a genuine user). By using the "honeywords" idea, which was proposed to detect the cracking of hashed password database, we show in this paper how to enable the detectability of biometric template database leakage.

We add an extra layer of protection since biometric features cannot be renewed. The biometric system design implies that protection mechanisms must satisfy the irreversibility property and to this respect we apply different correlation tests to show the non-distinguishability between genuine and honey templates. In this paper we implement the idea of a honey template protection scheme on faces and evaluate the security and accuracy performance.

1 Introduction

From a security perspective, the protection of templates stored in the database of a biometric system is one of the main challenges. An adversary can try to carry out a modification of their contents or even an unauthorized transfer of templates from the database towards another system. He can then generate a pre-image of the template by hill-climbing or brute force attack and since the adversary can create a fake physical characteristic from the biometric template this leads to physical masquerade attacks. All these attacks may occur as in a biometric standalone system, such as an automated border control system, as well as in remote biometric authentication. To prevent the leakage of biometric information, in addition to a better control of database access, other techniques should be implemented to prevent attacks and, even better, warn if such a leak has occurred.

While masquerade attacks are possible to cope with by better anti-spoofing technologies, they are not very possible to be completely prevented. To discourage both the physical and the digital masquerade attacks, we do this by empowering the system with detectability of the leakage of protected templates. Juels in [JR13] proposes the idea of "honeywords" used on passwords and we extend this idea to the biometric templates. In this paper we elaborate further the architecture design for a biometric system using BTPS-based honey templates explained in [YM15] and apply this idea on faces. We also evaluate the security, biometric recognition performance and irreversibility of honey face templates. These are detailed in Section 3 and 4 whereas in section 2 is given the background information and a review of honey objects.

2 Biometric templates as honey objects

Honey objects are used in various aspects of system security to deceive internal threats or external intruders (be they people or machines) against unauthorized data access. An example are honeypots mostly used for the detection of outer intruders, which are network machines used to distract adversaries from other more important machines, and honey farms (a network of honeypots) [Ho15] enabling deep research into server-side attacks. Subsequently, honeytokens [Sp03] are mostly implemented against internal threats and another development we find at system level are honeyclients [Na09], the complementary of honeypots, designed to mimic the behavior of a web browser. On the data level we find solutions such as the honeywords and honeydocuments. The honeyword method [JR13] hides the password of a user between k hash values of random passwords, and honeydocuments [Bo09] is again a trap-based mechanism which uses decoy documents. All these mechanisms serve as a safeguard against adversaries who try to get unauthorized data access. In [Ju14] honey objects must comply to two main properties: (1) *indistinguishability*, honey and real objects must be hard to distinguish from each other (e.g. a real password from a generated password or a database entry of a real patient in a health system from a fake one); (2) *secrecy*, the real object must be secret and camouflaged among the honey objects. In the case of honeywords, if an intruder by some means gets access to the user' set of passwords, he can use/guess only one of them. The system intercepts that a honeyword is used, it will consider this as information leakage and proceed with further steps (set off an alarm and/or update the passwords set).

The honeywords method provides us a systematic way to counter the masquerade attack against protected biometric templates. It resorts to probability (i.e. information-theoretic security) instead of computational complexity based security to cope with the crackable-hash assumption. In the biometric context, most databases are facing the same challenges. In hash based biometric template protection scheme, such as fuzzy commitment [JW99], and secure sketch [SLM07], if the hash is cracked, then the adversary can estimate the pre-image of the biometric features. And for feature-transformation based BTPSs (in [RCB01] and [TGN06]), the masquerade attack is even more straightforward. This is because the protected templates, PTs, are compared directly with a distance threshold and the attacker can find a PT's pre-image (biometric feature) with normally less effort than the case finding a pre-image of a hash value. As a result, for every enrolled user in the database, we need to provide a protection mechanism which needs to be applied on all the sweet templates (sugar and honey) and satisfies the abovementioned properties. Firstly, templates must be constructed in such a way that an adversary is not able to distinguish a sugar from a honey one, even if he: breaks the protection mechanisms; uses automatic tools such as classifiers; or tries to visually capture differences of honey and sugar templates pre-images to differentiate them. Secondly, the sugar template must be placed in a random position in the user database entry, or user data file, among the honey templates and this specific index must be known only to the honeychecker. We note that the aim of our approach on biometric templates, as well as the honeywords method, is not to lure the intruders with fake data, but to provide a means to alert the system that an internal or external adversary had access to the users' data and used them back: in other words that there have been system attack, information leakage, and user impersonation (masquerade attack).

2. 1. Honey objects database design

The design of a honey objects database is shown in Figure 1 and it can be applied to passwords, biometric templates and other objects which have similar properties with them, in terms of usage and storage. As in [YM15] for the i^{th} user, the sugar object S_i will be created from the user data and *K-1* honey objects H_{ij} (j = 1, 2, …, K-1) will be generated. These objects will pass through a protection mechanism (like hashing in passwords or one of the BTPS in biometric systems) having as a result a set of K protected objects: one SP_i from the sugar object and *K-1* protected honey objects HP_{ij} (j=1, 2, …, K-1).
To hide SP_i among the other honey objects, its memory address or index is randomly allocated and to the other objects are assigned the remaining addresses or indices. This process is handled by the Order Randomization block, which uses the auxiliary data AD_i to generate the SP_i's index and the indices of the protected honey objects. We define the protected objects as PI_{ij} (j = 1, 2, …, K) and the set PO_i=$\{AD_i, PI_{i1}, PI_{i2}, …, PI_{iK}\}$ of user i, containing the auxiliary data and the randomized protected objects. This set is stored in the database whereas the index L_i is stored in the Honey Checker Database.

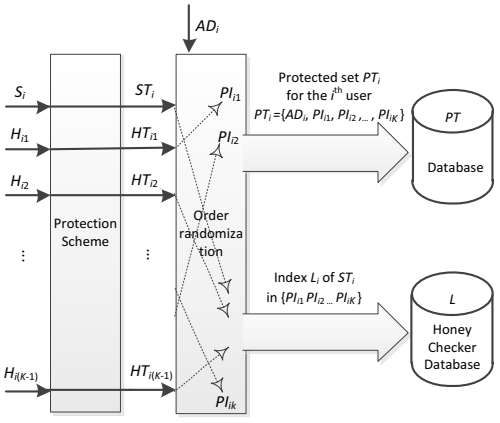
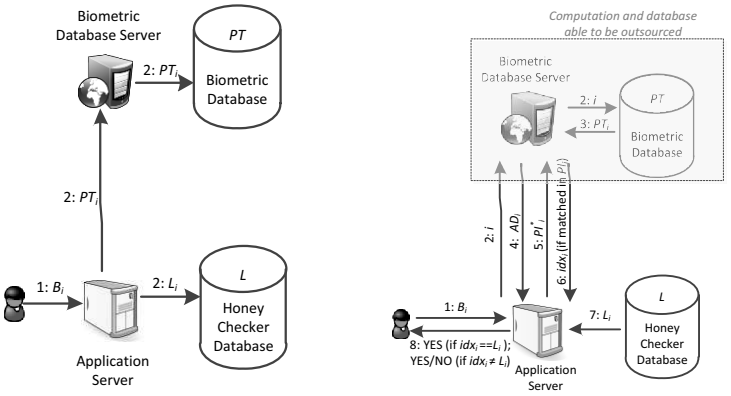


Figure 1. Honey objects database architecture design

In figure 2 we recall from [YM15] the architecture design of a biometric system using honey templates. During enrollment the Application Server converts the plain biometric feature B_i of user i to a set of protected templates. This set is defined in the same way as we did with the protected objects PO, i.e. $PT_i = \{AD_i, PI_{i1}, PI_{i2}, …, PI_{iK}\}$. PT_i. It will be stored in the Biometric Database and the index L_i in the Honey Checker Database. During verification the user i will show his biometric characteristic and the Application Server will retrieve the PI^*_i from the Biometric Database and send it to the Biometric Database Server. After comparing PI^*_i with all the PI_{ij} (j = 1, 2, …, K) of user i, it will send the best-matched template's index to the Application Server. If the index idx_i matches L_i the user will grant access to the system, otherwise an alarm will be set off and specific rules will follow, according to the defined security policies of the system. In the latter case, if the user personal identifiers match but the templates' indices do not, the system will consider this attempt as information leakage.



(a) Enrollment (b) Verification

Figure 2. Architecture for a honey templates based biometric system [YM15].

3 Honey Face Templates

The architecture of a honey templates based biometric system in figure 2 can be applied to different biometric characteristic. Our first attempt is in the construction and protection of honey face templates. The main challenge, as we mentioned in section 2, for a honey based system is to generate honey objects, in our case synthetic face templates, which cannot be distinguished from real templates. The protection mechanism we have adopted on the faces' feature vectors, is the plain-feature-defined sub-set selection borrowed by the idea in [Ya10]. Recognition performance degradation can be anticipated from the adoption of BTPS as discussed in [Si12]. This degradation caused by adding honey templates can be easily seen as well in our case. To make a quick proof-of-concept evaluation of the proposed honey template based biometric system, we created two small-scale face databases denoted as DB_{aux} and DB_{tst} representing an auxiliary database (for purposes of PCA training and the construction of AD_i used by BTPS) and a testing database, respectively. DB_{tst} is formed by 40 faces with 10 samples each as a sub-set of the ORL face database [ORL15]. To be fair in performance evaluation, DB_{aux} is formed by 40 faces from public websites with 1 sample for each face. All these 40 samples in DB_{aux} were cropped and normalized in their size to the same specification of those ORL samples (*i.e.*, 92×112 pixels, 256 gray scales).

(a) (b)

Figure 3: (a) Samples in DB_{aux} which is used for eigen faces training and used as AD_i by BTPS. (b) Samples in DB_{tst} which is used for recognition performance

Suppose we take the first $N = 20$ PCA coefficients $\boldsymbol{c}_{tst} = (c_{tst1}, c_{tst2}, \ldots, c_{tst20})$ (weights of eigenfaces that are trained from DB_{aux} to decompose each of test sample in DB_{tst}) of a test sample as the sugar biometric feature vector $SB = \boldsymbol{c}_{tst}$. Define an indicative binary vector $\boldsymbol{v} = (v_1, v_2, \ldots, v_{20})$ as

$$v_j = \begin{cases} 0 & if\ c_{tst_j} \geq 0 \\ 1 & if\ c_{tst_j} < 0 \end{cases} \quad (1)$$

According to the indicative binary vectors we can randomly group DB_{aux} into two non-overlapped sub-sets DB_{1aux} and DB_{2aux} with 20 samples each. Then we generate 20 PCA coefficient feature vectors, denoted as $\boldsymbol{c1}_{auxq} = (c1_{auxq1}, c1_{auxq2}, \ldots, c1_{auxq20})$ and $\boldsymbol{c2}_{auxq} = (c2_{auxq1}, c2_{auxq2}, \ldots, c2_{auxq20})$, from each of the two sub-sets, respectively, with $1 \leq q \leq 20$. From the two groups of feature vectors $\boldsymbol{c1}_{auxq}$ and $\boldsymbol{c2}_{auxq}$ ($1 \leq q \leq 20$), a 20-dimensional mask vector $\boldsymbol{m} = (m_1, m_2, \ldots, m_{20})$ is constructed as

$$m_j = \frac{s_j}{20} \sum_{q=1}^{20} |(1 - v_q)\, c1_{auxqj} + v_q c2_{auxqj}| \quad (2)$$

where s_j is defined as a sign value (+1 or -1) in eq. (3) depending on A_j defined in eq.(4):

$$s_j = \begin{cases} 1 & if\ A_j \geq 0 \\ -1 & if\ A_j < 0 \end{cases} \quad (3)$$

$$A_j = \sum_{q=1}^{20} ((1 - v_q)\, c1_{auxqj} + v_q c2_{auxqj}) \quad (4)$$

Now with the plain feature vector $\boldsymbol{c}_{tst}$ and the mask vector $\boldsymbol{m}$, we can generate the protected sugar template $ST = (st_1, st_2, \ldots, st_{20})$ as

$$st_j = \mathrm{rem}(\mathrm{rem}(w_1 m_j, 2\sigma) + w_2 c_{tstj}, w_3 \sigma) \quad (5)$$

where σ is an estimated standard deviation of all feature vectors in DB_{tst}, and the three weights $w_1, w_2,$ and w_3 can be tuned to achieve the best trade-off between security and recognition performance. The values of the weights w_i with i in {1, 2, 3} will be discussed later during the experimental evaluation. "*rem*" is a modular operation but keep the value's sign. In a brief, the above BTPS relates the mask feature vector generation to the plain feature vector's components' signs and thus obtains varied mask feature vectors for each different plain feature vector. For each sugar protected template, we try generating 15 honey protected templates to make the total number of templates K = 16. The generation process of a honey template is exactly the same as that of a sugar template as described in Eq.(1-5) except that the plain feature c_{tst} is replaced by a random feature vector with each component's dynamic range [-0.5, +0.5].

4 Security and irreversibility evaluations

In our BTPS design, AD_i is assumed to be public and length N of the PCA feature vector (*i.e.*, how many eigenfaces are used) decides the security level since $\boldsymbol{m}$ is constructed by

N selected feature vectors from $\boldsymbol{c1}_{aux}$ and $\boldsymbol{c2}_{aux}$. In our quick proof-of-concept experiment, *N* is set to be 20 and therefore the complexity of identifying the correct 20 feature vectors selected is 2^{20}. In practice, we can extend the security depth by increasing *N*, or keep the DB_{aux} as a secret parameter.

Whether an adversary can distinguish the generated sugar protected templates from those honey protected templates is the key criterion for the proposed honey template generation method. We further check if a machine learning algorithm can be used to classify the protected templates and identify the *ST* from other *HT*s. SVM was trained by half (200 test *PI*s) generated protected templates with ground-truth labels (sugar or honey) and evaluated by the other half (the other 200 test *PI*s) generated protected templates in our experiments and the classification results are given in Table 1 with various $w_1, w_2,$ and w_3 settings. It indicates that the proposed honey template generation method can well hide the sugar template among those honey ones – though the FMR and the FNMR are less than 50%, they should be already large enough to discourage the adversary to launch a masquerade attack if *N* is large enough. In Table 1, the values (1, 1, 1) of (w_1, w_2, w_3) imply that we have not taken into consideration these parameters.

Table 1: Classification of protected templates by SVM for different values of w_1, w_2, w_3.

w_1	w_2	w_3	FMR	FNMR
1	1	1	0.3433	0.4050
1	1	36	0.2287	0.4000
1	1	72	0.2380	0.4200
1	2	6	0.1470	0.4350
1	2	36	0.1553	0.3800
1	2	72	0.2417	0.4100
2	2	6	0.1150	0.4750
2	4	6	0.1537	0.3800
36	6	6	0.1573	0.3500
72	36	6	0.1767	0.3950

The need for a deep understanding of the security properties of our BTP scheme is crucial and has to be well characterized. In [Ma06] it is shown that all template protection schemes including fuzzy encryption, biometric salting and cancelable biometrics offer limited protection against different attacks. For instance, fuzzy encryption is vulnerable against linkage attacks, biometric encryption to Hill-Climbing attack and helper data scheme is limited especially by the correlation of features. An analysis of security and privacy protection can be seen in [Zh09] where it is implied that if algorithms such as ICA, PCA or LDA are used, the resulting features are more uncorrelated.
Biometric systems design implies that template protection mechanism must be such that the template should not imply the raw data of the extracted features. This is known as irreversibility [ISO11] and it is defined as the difficulty of determining exactly or with tolerance margin from a protected template the biometric sample(s) or features used during enrolment to generate that template [Si12].

4. 1. Irreversibility evaluation tests and results

In this section we will show the honey-based BTP on faces shows lack of irreversibility. To do this we provide the tests below showing the correlation values between the feature elements and the protected sugar templates; between the excerpt of the feature vectors used to create the honey templates and the honey templates themselves; and between sugar templates and the correspondent honey templates in their protected and unprotected form.

Correlation test between plain feature vectors and protected templates

Correlation tests are done to measure how strong the relationship between two vectors or variables is. Our first test to this regard is between the plain feature vectors c_{tst} and the protected templates ST of user *i*. The general mathematical formula we have used is:

$$corr(A, B) = \frac{A * B^T}{\sqrt{var\ (A) * var\ (B)}} \quad (6)$$

where *var (x)* is the variance of the vector *x*, and it is interpreted as the covariance of *x* with itself. The covariance *cov(x, y)* between vectors *x* and *y* shows how much the two vectors differ from each other.

The correlation values interval is between 1 and -1. If the correlation is zero or near to zero, means that there is no correlation between the vectors. If the correlation value tends to be 1, this means that there exists a direct correlation between the two vectors and if it tends to be -1 then the two vectors are inversely related (they go in opposite directions).

In our terms, the first test will be applied between the PCA coefficients $\boldsymbol{c}_{tst} = (c_{tst1}, c_{tst2}, \ldots, c_{tst20})$ (weights of eigenfaces that are trained from DB_{aux} to decompose each of test sample in DB_{tst}) and the sugar protected templates $ST = (st_1, st_2, \ldots, st_{20})$ which is expressed in eq. (5). We have evaluated this test for different values of the tuning parameters triplet (w_1, w_2, w_3). The results are presented in Table 2 where we can see that there is zero correlation between the two vectors. This means that there can be no information leakage from the protected sugar template.

Correlation test between selected or unselected feature vectors and protected templates

As we mentioned in section *3.1* the sugar template is constructed by selecting randomly a part of the set of the plain feature vectors. In order to argue the no correlation between the final protected sugar template and these two separate and non-overlapping subsets we performed two correlation tests between the selected vectors used to construct the sugar templates and the protected sugar templates. In this construction, the selected feature vectors are expressed in equation (4), and the unselected portion of the set of feature vectors can be expressed as:

$$A^*_j = \sum_{q=1}^{20} (v_q\ c1_{\mathrm{aux}qj} + (1 - v_q) c2_{\mathrm{aux}qj}) \quad (7)$$

The results of the two abovementioned tests are presented in Table 2.

Table 2: Correlation values between {plain feature vectors, selected, and unselected vectors} and the protected sugar template

No.	A	B	Corr (A, B)
1	Sugar feature vectors (SB)	Protected Sugar Template (ST)	0
2	Selected vectors for Sugar Template (A_j)	Protected Sugar Template (ST)	0,06423
3	Unselected vectors for Sugar Template (A^*_j)	Protected Sugar Template (ST)	-0.06416

The problem of irreversibility between templates must be considered also for the honey protected template case. We can see that between the selected vectors of the sugar feature vectors and the corresponding honey-templates the correlation mean value is very low. Nearly the same correlation situation can be seen between the protected honey templates and the counterpart of the selected vectors in the set of the plain feature vectors. The results are presented in Table 3.

Table 3: Correlation values between selected and unselected vectors and protected honey template

No.	A	B	Corr (A, B)
1	Selected vectors for Sugar Template (A_j)	Protected Honey Template (HT)	0,0506
2	Not-selected vectors for Sugar Template (A^*_j)	Protected Honey Template (HT)	0.05004

The correlation between the selected feature vectors that we have chosen to build the honey-templates and the honey-templates themselves is near to zero. The same holds for the correlation between the unselected feature vectors and the constructed honey-templates. The Equal Error Rates for the sugar templates and the honey templates are measured for different values of the three tuning coefficients w_1, w_2, and w_3.

Table 4. EER for sugar and honey templates for different values of w_1, w_2, w_3.

w_1	w_2	w_3	EER_{Sugar}	EER_{Honey}
1	1	1	0.4078	0.4197
1	1	36	0,3993	0,4080
1	1	72	0,3993	0,4084
1	2	6	0.4249	0.4368
1	2	36	0.4214	0.4353
1	2	72	0.4214	0.4357
2	2	6	0.4365	0.4509
2	4	6	0.4668	0.4771
36	6	6	0.4749	0.4854
72	36	6	**0.4809**	**0.4994**

Table 2 and 3 give only correlation values between the selected / unselected feature vectors and the protected templates. Furthermore, Table 4 shows the EER of classifying the correlations of the two cases (selected and unselected) for both sugar templates and honey

templates. SVM is used to do the classification training and testing tasks. The optimal value of (EER_{Sugar}, EER_{Honey}) is where the triplet (w_1, w_2, w_3) is equal to (72, 36, 6).

4. 2 Performance comparison between protected and unprotected case

Unprotected case

The first comparison is between samples of unprotected sugar templates. As a first score we measure the similarity between sugar feature vectors which belong to the same subject. We have called this $GEN_{(SB,SB)}$. This stands for genuine score of the sugar feature vector. A second score we have calculated is the similarity between the first samples of the sugar feature vectors. We have called this $IMP_{(SB,SB)}$ which stands for imposter score of the sugar feature vector. As a similarity measure we have used the inverse of SSD (Sum of Squared Differences). For these scores we have the following equations:

$$GEN_{(SB,SB)} = \frac{1}{\sum_{i=1,j=1}^{i=40,j=10}(SB_i(j)-SB_i(j+1))^2} \tag{8}$$

$$IMP_{(SB,SB)} = \frac{1}{\sum_{i=1}^{i=40}(SB_i(j=1)-SB_{i+1}(j=1))^2} \tag{9}$$

The accuracy of the unprotected case is the Equal Error Rate (EER-unprotected case) between $GEN_{(SB,SB)}$ and $IMP_{(SB,SB)}$ and the results are presented in Table 5.

Protected case

The accuracy test of the unprotected case is repeated for the protected templates also. We have two other scores: $GEN_{(ST,HT)}$ and $IMP_{(ST,HT)}$ for the EER evaluation. The $GEN_{(ST,HT)}$ comparison score is between genuine protected templates and the correspondent 15 protected honey templates. We called this parameter "protected genuine score". It is calculated as:

$$GEN_{(ST,HT)} = \frac{1}{\sum_{i=1,j=1}^{i=40,j=10}(ST_i(j)-HT_i(j+1))^2} \tag{10}$$

The last comparison score is between first samples of protected templates of the same subject. We called this parameter "protected imposter score".

$$IMP_{(ST,HT)} = \frac{1}{\sum_{i=1}^{i=40}(ST_i(j=1)-HT_{i+1}(j=1))^2} \tag{11}$$

The accuracy of the protected case is calculated as the EER (protected case) between $GEN_{(ST,HT)}$ and $IMP_{(ST,HT)}$ and the results are presented in Table 5. Results for both error rates are measured for different values of the tuning parameters triplet (w_1, w_2, w_3), where the best result is highlighted.

4. 3. Recognition performance evaluation

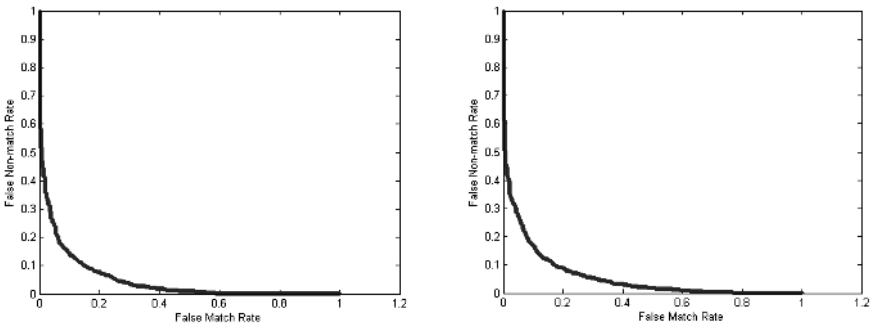
To evaluate the recognition of the proposed BTPS and honey template based biometric

system, we compare the plain template case (without BTPS and without honey templates) and the proposed BTPS and honey template based case. In the plain template case, we use the following testing protocol: for all 40 faces in the testing database DB_{tst}, we cross compare the 10 PCA feature vectors from each same face and this resulted in 1800 genuine comparison scores; and for all 40 faces we use only the PCA feature vector from the first sample of each face to perform cross comparison and this resulted in 780 imposter comparison scores.

Table 5. EER for unprotected and protected case for different values of w_1, w_2, w_3.

w_1	w_2	w_3	EER (unprotected case)	EER (protected case)
1	1	1	0.1223	0.3351
1	1	36	0.1223	0.2791
1	1	72	0.1223	0.2801
1	2	6	0.1223	0.2000
1	2	36	0.1223	0.1735
1	2	72	0.1223	0.1580
1	**6**	**36**	**0.1223**	**0.1285**
2	2	6	0.1223	0.1696
2	4	6	0.1223	0.1975
36	6	6	0.1223	0.2577
72	36	6	0.1223	0.2704

In the proposed BTPS and honey template based case, the highest comparison score out of the 16 comparison scores between the probe's *PI** and all 16 *PI*s in the database is recorded as the final comparison score to verify the probe.



(a) Unprotected templates: EER = 0.1223 (b) BTPS with honey templates: EER = 0.1285

Figure 4. Recognition performance comparison: $w_1 = 1, w_2 = 6, \text{and } w_3 = 36$.

From Figure 4 we can see an example of recognition performance comparison between the plain PCA feature vector case and the proposed BTPS and honey template case with the setting $w_1 = 1, w_2 = 6, \text{and } w_3 = 36$. We also see performance degradation in other parameter settings. Recognition degradation, as in many other BTPS, can be limited in an acceptable range if the parameters are fine tuned.

5 Conclusions and further work

We borrowed the honeywords concept used for detecting leaked passwords in a biometric system and proposed a BTPS based honey template construction method in this paper. The honey biometric protected templates can be used as chaff data to hide the storage address of the sugar (genuine) biometric protected template. Once a honey template is matched by a probe, the system can reasonably conclude with a high probability that the corresponding biometric data entries in the database had been already leaked and a pre-image masquerade attack is launched. This could be very helpful in detecting such data leakage accidents which cannot be achieved by existing biometric template protection schemes.

We proposed in this paper a biometric database construction design and architecture design for a biometric system using such a honey templates idea. It is applicable to verification applications in physical and logical access control, such as ATM, health records, etc. To prove the concept's effectiveness in a practical biometric system, we tested the honey template idea on PCA based face features. Experiments demonstrated the effectiveness of the proposed concepts in both security and recognition performance aspects. An important part of our paper was the security evaluation of our BTPS, focused on the irreversibility of the biometric sugar and honey templates. The low levels of correlations between different feature or template sets show the effectiveness of the scheme.

As a further work we will continue with the system evaluation in terms of unlinkability, as the difficulty of classifying the protected templates over time and across applications [Zh09]. Inuma in [In14] has mathematically proven the relationship between the two notions of irreversibility and linkability in a biometric system. We proved in our paper that if an attacker possesses the set of protected sugar and honey templates, he is not able to recover the feature element and then pretend to be that user. But what if he possesses two sets of protected templates of user *i*, coming from two different applications: is the attacker able to identify that these template sets belong to the same characteristic? This process should be computationally hard and we expect to evaluate the notion of linkability in our future work.

We can finally conclude that while use of honey templates in biometrics is still a new direction to explore, we believe both the BTPS method and the honey template construction method have wide room to improve in security and recognition performance aspects in the future. We hope this work can provoke thoughts and discussions in this field.

6 Acknowledgement

This research work was partially funded by the European 7th Framework Programme project FIDELITY and Competiveness and Innovation Framework Programme project PIDaaS. All information is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. The European Commission has no liability in respect of this document, which is merely representing the authors' view.

References

[Bo09] Bowen, B.M.; Hershkop, S.; Keromytis, A. D.; Stolfo, S. J.: Baiting inside attackers using decoy documents. In Security and Privacy in Communication Networks, pages 51–70, 2009.

[Ho15] Honeypots. http://www.honeypots.org/, accessed: May 2015.

[In14] Inuma, M.: A relation between irreversibility and unlinkability for biometric template protection algorithms. In Josai Mathematical Monographs, vol. 7, 2014, pp. 55-65.

[ISO11] ISO/IEC 24745 Information technology - Security techniques - Biometric information protection, 2011.

[JR13] Juels, A.; Rivest, R.: Honeywords: making password-cracking detectable. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (ACM-CCS'13), 2013, pp. 145-160.

[Ju14] Juels, A.: A bodyguard of lies: the use of honey objects in information security. In Proceedings of the 19th ACM symposium on Access control models and technologies. ACM, 2014.

[JW99] Juels, A; Wattenberg, M.: A fuzzy commitment scheme. In CCS '99, ACM, 1999, pp. 28–36.

[Ma06] Martinez-Diaz, M.; Fierrez-Aguillar, J.; Alonso-Fernandez, F.; Ortega-Garcia, J; Siguenza, A.J.: Hill-climbing and brute-force attacks on biometric systems: A case study in Match-on-card fingerprint verification. In Proceedings of IEEE Intl. Carnahan Conf. on Security Technology, ICCST, Lexington, USA, 2006, pp. 151-159.

[Na09] Nazario, J.: Phoneyc: A virtual client honeypot. In Proceedings of the 2nd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more. USENIX Association, 2009.

[ORL15] The Database of Faces (formerly "The ORL Database of Faces"). http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html, accessed: January, 2015.

[RCB01] Ratha, N.; Connell, J.; Bolle, R.M: Enhancing security and privacy in biometrics-based authentication systems. In IBM Systems Journal, 40(3), 2001, pp. 614–634.'

[Si12] Simoens, K.; Yang, B.; Zhou, X.; Beato, F.; Busch, C.; Newton, E.M.; Preneel, B.: Criteria towards metrics for benchmarking template protection algorithms. In Proc. of 5th IAPR International Conference on Biometrics (ICB), 498-505, 2012.

[SLM07] Sutcu,Y.; Li, Q.; Memon, N.: Protecting biometric templates with sketch: theory and practice. In IEEE Transaction on Information Forensics and Security, 2(3), 2007, pp. 503-512.

[Sp03] Spitzner, L.: Honeytokens: The other honeypot. In Symantec Security Focus, July 2003.

[TGN06] Teoh, A; Goh, A; Ngo, D.: Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs. In IEEE Trans. Pattern Anal. Mach. Intell., 28(12), 2006, pp.1892–1901.

[Ya10] Yang, B.; Hartung, D.; Simoens, K.; Busch, C.: Dynamic random projection for biometric template protection. In Proceedings of the 4th IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS), 2010.

[YM15] Yang, B. and Martiri, E.: Using Honey Templates to Augment Hash Based Biometric Template Protection. In Proc. of the 1st International Workshop on Secure Identity Management in the Cloud Environment (SIMICE) at the 39th Annual International Computers, Software & Applications Conference (COMPSAC), 2015.

[Zh09] Zhou, X.; Wolthusen, S.; Busch, C.; Kuijper. A.: Feature correlation attack on biometric privacy protection schemes. In Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IEEE, 2009, pp. 1061-1065.