Simulated annealing attack on certain fingerprint authentication systems

Andreas Pashalidis

KU Leuven
ESAT/SCD-COSIC & iMinds
Kasteelpark Arenberg 10
3001 Heverlee, Belgium
andreas.pashalidis@esat.kuleuven.be

Abstract: This paper describes a simple and generic attack against minutiae-based fingerprint authentication systems. The aim of the attack is to construct a fingerprint minutiae template, compliant to ISO/IEC standards, that matches a fixed but unknown target fingerprint. Our attack is expected to be most effective against systems that employ *vicinity-based* matching algorithms, i.e. systems that divide fingerprints into multiple regions and then compute similarity over these regions. The effectiveness of our attack is experimentally demonstrated against the recently proposed 'Protected Minutiae Cylinder Code' (PMCC) scheme.

1 Introduction

Biometric authentication systems work as follows. Initially, users are enrolled in the system. A user's enrolment involves capturing a biometric trait, such as his fingerprint, and storing the captured data in the form of a 'reference' template. An enrolled user who wishes to authenticate himself does so by providing a fresh biometric sample. This is again captured by the system, and transformed into a 'sample' template. This sample template is then compared against the stored reference template. If the two templates 'match', then the user is deemed authenticated.

In order to decide whether or not two templates match, every biometric authentication system employs an algorithm called the 'matcher'.[1] The matcher takes as input two biometric templates and outputs a similarity score. During authentication, this score is compared against a fixed threshold value; if a given comparison yields a similarity above the threshold, then the two templates are deemed to originate from the same user. The threshold value determines the accuracy of the biometric authentication system: if the threshold is too high, then too many biometric samples that should be accepted by the system will actually be rejected. If, on the other hand, the threshold is too low, then too many

[1] The terms 'comparison', '(dis)similarity measurement', and '(dis)similary estimation' are sometimes used in the biometrics literature instead of 'matching'; see [ISO05b] for a recent attempt to harmonize the use of these and related terms.

samples that should be rejected will actually be accepted. The False Match Rate (FMR) and False Non-Match Rate (FNMR) are two well-known measures, which we use in this paper, that depict the dependency of system accuracy on the threshold value. These measures enable an informed choice for this value, also having in mind the context in which the system is or will be deployed.

Unfortunately, access to the matcher's output enables an adversary to launch 'hill climbing' attacks against the system. In such an attack, the adversary first provides a synthesized biometric sample to the system and observes the matcher's output. It then iteratively modifies this sample depending on how the matcher's output reacts to the modifications: modifications which yield worse similarity are discarded while modifications leading to better similarity score are further modified. Using this approach, the adversary may end up with an artificially constructed biometric sample that passes the threshold and can therefore impersonate a particular user. Using template inversion techniques (see, for example, [FJ11] and Chapter 2 of [Nag12]) the adversary may be even able to reconstruct a 'complete' matching biometric imitation such as a 'gummy finger' [MMYH02]. In this paper, we describe a 'simulated annealing' attack, which uses a slightly more sophisticated technique than hill climbing. Our attack focuses on minutiae-based fingerprint systems that use *vicinity-based* matchers.

Since biometric data is considered to be sensitive personal data, it is important to safeguard reference templates. To this end, special biometric template protection schemes have been developed (see, for example, [Sim12, Nag12]). These schemes encode biometric templates in a way that is irreversible. Among other things, they aim to ensure that, even if an adversary gets access to stored protected biometric templates, this data cannot be abused to reconstruct original biometric data. An adversary, however, with access to protected biometric templates, can launch hill climbing and similar attacks *even without access to the matcher's output*; assuming that the matcher logic is not secret, the adversary simply uses its own implementation of the matcher. In this case, the attack can be launched in an offline fashion. Such offline attacks are particularly undesirable because their occurrence cannot be detected. In this paper, we demonstrate the effectiveness of our attack against the 'Protected Minutiae Cylinder Code' (PMCC) scheme [FMC12], which is a template protection scheme for fingerprints. That is, we demonstrate our attack in a setting where it can be launched in an offline manner.

We expect our attack to be successful only against fingerprint systems that employ vicinity-based matchers. A vicinity-based matcher roughly works as follows. It first divides all templates into multiple regions. Each (or some) region of a sample template is then 'paired' with some region of the reference template on the basis of how similar they are. Subsequently, the matcher computes similarities over the resulting region pairs, and, finally, merges the different similarity scores into a single, consolidated score.[2] The PMCC scheme employs such a matcher.

The remainder of this paper is organised as follows. The next section surveys some related work, Section 3 describes our attack, and Section 4 describes our experiments. Section 5

[2]The reason we call such matchers 'vicinity-based', rather than 'region-based', is because we are only aware of schemes where region boundaries are defined with respect to detected minutiae points; that is, a region is defined to be a central minutia's vicinity.

reports our results and, finally, Section 6 concludes.

2 Related work

Ratha, Conell and Bolle describe a generic attack model for biometric authentication systems and examine the entropy of minutiae templates with respect to matching algorithms that require a minimum number of minutia points to 'match up' [BS01, RCB01]. They explain that, if sufficiently many minutiae points are present in the template, then synthesizing a matching template by exhaustively searching through the space of all possible templates, becomes increasingly infeasible. The search strategy employed in this paper ignores large parts of the search space; instead, it divides the template area into a number of smaller regions, and navigates through the space by operating on these these regions individually. Moreover, it is not our goal to reconstruct the original minutiae; the goal is to synthesize a minutiae template that contains a sufficient number of small regions, each with a configuration of minutiae that is 'good enough' for the matching algorithm. As a result, the probabilistic analysis of [BS01] does not apply to our approach, and to vicinity-based matchers in general.

Hill climbing techniques have been used to attack various biometric systems. Uludag and Jain, for example, present such an attack [UJ04]. Their attack works against a minutiae-based fingerprint authentication system where matching scores are based on the number of minutiae with a similar location and orientation [JHPB97]. While conceptually similar, the attack in this paper differs in the three following aspects. Firstly, it uses simulated annealing [KGV83], which is a form of hill climbing that is able to escape from local optima. Secondly, it explores the search space in a way that is optimised for vicinity-based matchers; moreover it does not explore the entire search space and its goal is not reconstruct the original minutiae template. Thirdly, the attack is applied against PMCC, i.e. a template protection technique where the adversary is explicitly allowed to access protected templates. This means that the attack can be launched in an offline manner and all safeguards proposed in [UJ04] are not applicable.

We also note that, in order to discourage hill climbing attacks in general, Soutar proposed to use only coarse similarity scores, for example by using large quantization intervals [Sou]. While this countermeasure would make our attack less efficient, it only works if the adversary does not have access to reference templates (or in the case where the matching algorithm is secret).

3 Attack description

The logic of our attack is depicted in the algorithms shown in Figures 1 and 2. Table 1 explains the meaning of the different input parameters. The algorithm of Figure 1 outputs a synthesized 'gummy' template compliant to [ISO05a] that is meant to imitate a fixed 'target' fingerprint; the attack is successful if this gummy template passes the matcher's

threshold in an authentication attempt for the user to whom the target fingerprint belongs.

How exactly the target template is fixed is outside our scope. The algorithm in Figure 1 simply invokes $\mathsf{match}(T)$, which is assumed to return the system matcher's output, where T is a fingerprint sample compliant to [ISO05a] as generated by our algorithm, and where the reference biometric is the target fingerprint. Note that the target/reference fingerprint does not need to be available in a format compliant to [ISO05a]; our attack merely requires that the matcher handles fresh samples in this or an equivalent format.

The attack algorithm starts by composing an initial 'current' template (T) which consists of multiple, $\mathsf{ha} \times \mathsf{va}$ to be precise, vicinities. Each of these vicinities is populated with a number of minutiae that have, within the vicinity, random locations and orientations. The number of minutiae per vicinity may differ per vicinity and is randomly chosen between $\mathsf{vp}_{\mathsf{min}}$ and $\mathsf{vp}_{\mathsf{max}}$. The algorithm then constructs a candidate template (T') by replacing one of the vicinities of the initial template by a fresh, randomly generated, vicinity. If the candidate template yields a better matching score than the current template, then the current template is replaced by the candidate template, and the procedure is repeated for REP times. Sometimes, the algorithm also replaces the current template with a candidate that yields a worse similary score. This is done according to a 'cooling schedule' which influences the way in which the search avoids getting trapped in local optima. Since we expect that no single cooling schedule will work efficiently against all systems, we do not define this schedule in detail here; we describe the particular schedule we used in our experiments in Section 4. In the end, the algorithm outputs the template which yielded the best similarity throughout the entire search (T_{best}).

Table 1: Input parameters

w	Template width, in pixels
h	Template height, in pixels
ha	Number of vicinities, horizontal
va	Number of vicinities, vertical
$\mathsf{vp}_{\mathsf{min}}$	Minimum number of minutiae per vicinity
$\mathsf{vp}_{\mathsf{max}}$	Maximum number of minutiae per vicinity
$\mathsf{r}_{\mathsf{min}}$	Minimum vicinity radius, in pixels
$\mathsf{r}_{\mathsf{max}}$	Maximum vicinity radius, in pixels
REP	Number of search steps until gummy template is output

4 Experimental setup

In order to demonstrate the effectiveness of our approach, we conducted a number of experiments which were driven by the FVC2006 fingerprint database DB2_A [JFGR07]. This database is a collection of 1680 bitmap images, each of size 400×560 pixels, that depict the fingerprints of 140 fingers, with 12 different impressions per finger. In or-

ConstructGummyTemplate (input: $\mathsf{w}, \mathsf{h}, \mathsf{ha}, \mathsf{va}, \mathsf{vp_{min}}, \mathsf{vp_{max}}, \mathsf{r_{min}}, \mathsf{r_{max}}, \mathsf{REP}$)

1. Generate an initial template T by dividing the area of $\mathsf{w} \times \mathsf{h}$ pixels into $\mathsf{ha} \times \mathsf{va}$ non-overlapping rectangles, each of size $(\mathsf{w}/\mathsf{ha}) \times (\mathsf{h}/\mathsf{va})$ pixels. For each rectangle:
   (a) Compute $(\mathsf{c_x}, \mathsf{c_y})$ as the coordinates of the center of the rectangle.
   (b) Run **RandomVicinity**$(\mathsf{c_x}, \mathsf{c_y}, \mathsf{vp_{min}}, \mathsf{vp_{max}}, \mathsf{r_{min}}, \mathsf{r_{max}})$ and add the resulting vicinity to T.
2. Set $\mathsf{sc} = \mathsf{match}(T)$.
3. Set $\mathsf{sc_{best}} = \mathsf{sc}$ and $T_{\mathsf{best}} \leftarrow T$.
4. Repeat REP times:
   (a) Set $T' \leftarrow T$.
   (b) Pick a random vicinity V in T', and replace it with a fresh random vicinity as output by **RandomVicinity**$(\mathsf{c_x}, \mathsf{c_y}, \mathsf{vp_{min}}, \mathsf{vp_{max}}, \mathsf{r_{min}}, \mathsf{r_{max}})$, where $(\mathsf{c_x}, \mathsf{c_y})$ are the coordinates of V's center.
   (c) Set $\mathsf{sc}' = \mathsf{match}(T')$.
   (d) If $\mathsf{sc}' > \mathsf{sc_{best}}$, then set $\mathsf{sc_{best}} \leftarrow \mathsf{sc}'$ and $T_{\mathsf{best}} \leftarrow T'$.
   (e) If $\mathsf{sc}' > \mathsf{sc}$, then set $\mathsf{sc} \leftarrow \mathsf{sc}'$ and $T \leftarrow T'$; otherwise check whether or not the cooling schedule permits moving to worse solution candidates. If it does, then set $\mathsf{sc} \leftarrow \mathsf{sc}'$ and $T \leftarrow T'$.
5. Output the minutiae in the vicinities of T_{best}.

Figure 1: Simulated annealing attack

RandomVicinity (input: $\mathsf{c_x}, \mathsf{c_y}, \mathsf{vp_{min}}, \mathsf{vp_{max}}, \mathsf{r_{min}}, \mathsf{r_{max}}$)

1. Pick a random radius $\mathsf{r} \in \{\mathsf{r_{min}}, \mathsf{r_{min}} + 1, \ldots, \mathsf{r_{max}}\}$ and a random vicinity population $\mathsf{vp} \in \{\mathsf{vp_{min}}, \mathsf{vp_{min}} + 1, \ldots, \mathsf{vp_{max}}\}$.
2. Start with an empty vicinity V.
3. For all values of $i \in \{1, 2, \ldots, \mathsf{vp}\}$:
   (a) Pick two random offsets $x', y' \in \{1, 2, \ldots, \mathsf{r}\}$, a random angle $\theta \in [0, 2\pi]$, and two random signs $s_x, s_y \in \{-1, 1\}$.
   (b) Generate minutia $m_i = (\mathsf{c_x} + s_x x', \mathsf{c_y} + s_y y', \theta)$ and add it to V.
4. Output $V = \{m_1, m_2, \ldots, m_{\mathsf{vp}}\}$.

Figure 2: Generating a random vicinity

der to prepare our experimental environment, we extracted minutiae templates compliant to [ISO05a] from these images using the `fjfxSample` command line utility[3]. Since this utility only accepts 'portable gray map images', we first converted the images into this format using the `bmptopnm` utility. Unfortunately, `fjfxSample` was unable to extract minutiae from one of the images (fifth finger, first impression); hence we ended up with 1679 templates.

Then we derived protected fingerprint templates for each of these 1679 templates using the PMCC scheme [FMC12]. For this, we used the PMCC implementation made available by University of Bologna's BioLab[4], in particular version 1.3 of the Minutia Cylinder-Code Software Development Kit that is available on their website. In fact, we derived two datasets, each consisting of 1679 PMCC templates. For the first dataset we used parameter value $K = 64$ and for the second dataset we used parameter value $K = 128$. Hence, we call the two datasets 'K64' and 'K128'.

The parameter K influences the length of PMCC templates; higher values lead to lengthier PMCC templates and better system accuracy [FMC12]. The reason we chose these two values is because (a) we expect the templates derived under these parametrisations to be more resistant to our generic attack than templates derived with smaller values for K, and (b) the results in [FMC12] show that these values yield better accuracy than any other examined values.

After completing the above preparations, we started our main experiments in which we aimed to separately attack each of the 1679 templates in both the K64 and the K128 dataset, using the algorithm from Figure 1. We aimed to conduct three experiment sets for each of the K64 and the K128 dataset, namely one experiment set for each of the repetition values $\mathsf{REP} = 300, 500, 1000$. That is, we aimed to conduct six experiment sets in total, with each experiment set generating a collection of 1679 'gummy' fingerprint templates. Unfortunately, due to the non-optimised nature of PMCC code and its unstable integration of our attack code, we were unable to complete our experiments. However, for all target templates processed so far, our algorithm has generated a gummy template, and the results shown in the next section are based on the gummy templates that were generated at the time of writing; more precisely, for $K = 64$, and for $\mathsf{REP} = (300, 500, 1000), (501, 508, 217)$ gummy templates were generated respectively while, for $K = 128$, the corresponding numbers are $(416, 476, 0)$.

The other attack parameter values we used in our experiments were fixed: $\mathsf{w} = 400$, $\mathsf{h} = 560$, $\mathsf{ha} = \mathsf{va} = 4$, $\mathsf{vp}_{\mathsf{min}} = \mathsf{vp}_{\mathsf{max}} = 4$, and $\mathsf{r}_{\mathsf{min}} = \mathsf{r}_{\mathsf{max}} = 40$. The cooling schedule we employed caused the algorithm to move to worse candidates with probability $4r/(3\mathsf{REP})$, where r is the number of the current repetition, for the first $\mathsf{REP}/2$ repetitions. For the remaining $\mathsf{REP}/2$ repetitions, the cooling schedule caused the algorithm to behave like a standard hill climbing algorithm, i.e. no longer moving to worse candidates. The above parameter values and cooling schedule were chosen based on manual observations during early experimentation. We did not systematically optimise over the parameter space in an automated fashion.

[3] `http://www.digitalpersona.com/fingerjetfx/`
[4] `http://biolab.csr.unibo.it`

It should be noted that, when enrolling and matching PMCC templates, a number of further parameters, specific to the PMCC scheme, have to be fixed as well. In our experiments, we used the default parameter values, as documented in [FMC12] and provided in the configuration files with which the PMCC software is shipped.

5 Results

Figures 3 and 4 show our main results. The figures show the FNMR and the FMR rates we obtained when executing the 'FVC protocol' over the K64 and K128 datasets using the PMCC matcher. More precisely, the FNMR curves were obtained from 9229 'genuine' comparisons: each PMCC template in the dataset was compared to all other templates of the same finger, but without comparing any pair of templates more than once. The FMR curve, on the other hand, was obtained from 9730 'impostor' comparisons: the first PMCC template of each finger was compared to the first sample of all remaining fingers in the dataset, but, again, without comparing any pair of templates more than once.

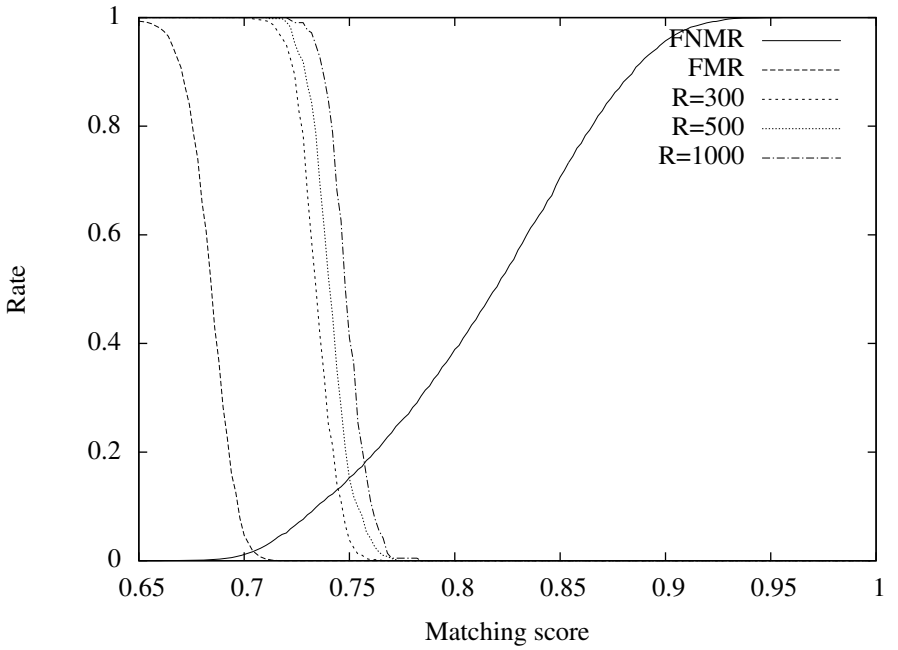

Figure 3: FNMR and FMR curves for the K64 dataset

The figures also show three more FMR curves; each of these curves was obtained by comparing each of the 'gummy' fingerprint minutiae templates generated by our attack to the corresponding enrolled PMCC template. The three curves show how the amount of

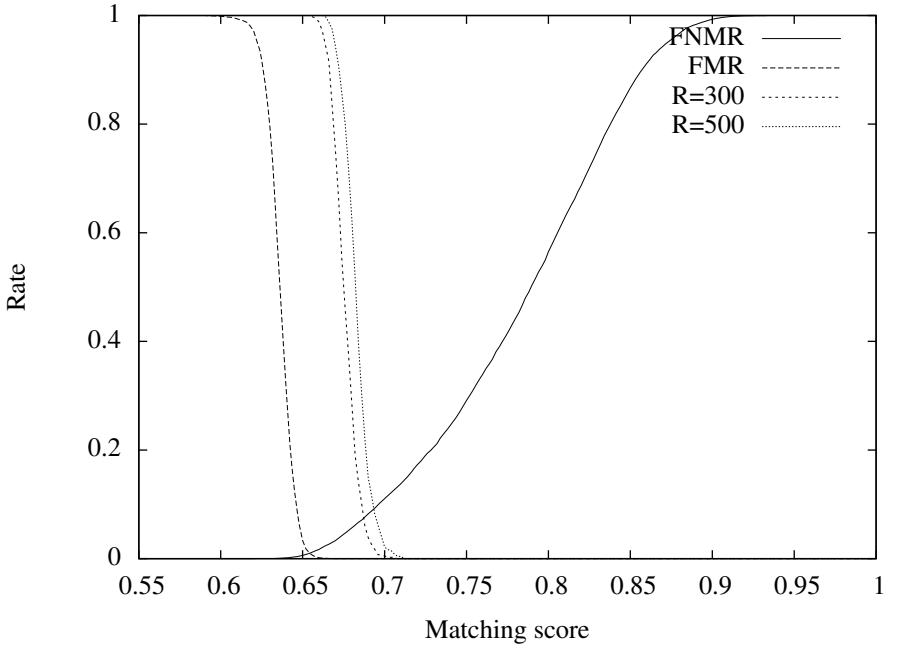

Figure 4: FNMR and FMR curves for the K128 dataset

computational effort, i.e. for $\mathsf{REP} = 300, 500, 1000$, affects the effectiveness of the attack.

Tables 2 and 3 show the Equal Error Rate (EER), the FMR_{1000} and the Zero_{FMR} rates as determined from the curves in Figures 3 and 4, as well as the corresponding values reported in [FMC12]. Our slightly higher FVC protocol rates are probably due to the fact that the minutiae extractor used in [FMC12], both for the tuning of PMCC-specific parameters and the subsequent construction of PMCC templates, differs from the extractor used in our experiments.

Impostor Templates	EER	FMR_{1000}	Zero_{FMR}
FVC Protocol on DB2_A reported in [FMC12]	0.32	0.47	1.07
FVC Protocol on DB2_A	1.7	3.6	4.9
Gummy templates, $\mathsf{REP} = 300$	13.2	21.2	21.5
Gummy template, $\mathsf{REP} = 500$	15.2	25.0	25.4
Gummy template, $\mathsf{REP} = 1000$	17.9	29.8	30.3

Table 2: EER, FMR_{1000} and Zero_{FMR} rates over the K64 dataset (percentage values)

From our results it is evident that our attack is efficient and effective; even after a number of iterations as small as 300, a significant proportion of impostor matches is achieved. We

Impostor Templates	EER	FMR_{1000}	Zero_{FMR}
FVC Protocol on DB2_A reported in [FMC12]	0.17	0.23	0.42
FVC Protocol on DB2_A	1.0	1.9	3.3
Gummy templates, $\mathsf{REP} = 300$	7.4	13.0	13.2
Gummy templates, $\mathsf{REP} = 500$	9.1	14.2	14.4

Table 3: EER, FMR_{1000} and Zero_{FMR} rates over the K128 dataset (percentage values)

also observe that, as the number of iterations grows, so does the success degree of the attack. However, the relationship between the increase of computational effort and the attack success does not seem to be linear. Due to various technical constraints related to implementation details, we were unable to perform tests with more realistic amounts of iterations (e.g. $\mathsf{REP} = 10^7$). Experiments of this order of magnitude therefore remain a topic of future research. While there is certainly a point beyond which additional efforts will only yield diminishing returns, our results suggest that, with only moderate additional computational efforts, the success degree of the attack can still be increased significantly.

6 Concluding remarks

One of the limitations of our attack is that it does not take into account constraints imposed by human nature. For example, the attack does not take into account the issue of inter-ridge distance. As a result, some gummy templates may contain minutiae points that are closer to each other than would be possible on a natural fingerprint. This shortcoming can be used to automatically detect an attack template generated by our algorithm. Therefore, a future improvement of our attack would be the incorporation of constraints regarding the issue of inter-ridge distance, and statistical properties of minutiae locations more generally [CM06].

Another interesting future research question is whether or not it is possible to turn our attack, which merely aims to find minutiae templates that are 'equivalent' to a given target template, into an attack that actually reconstructs the original target template. Our intuition on this idea is simple: after running the attack on the same target multiple times, each time with a different random initial template, the intersection of all 'common' minutiae points of the resulting gummy templates may resemble the original minutiae points.

We envision our attack to become one of the evaluation tools for vicinity-based fingerprint authentication systems. There is certainly a lot of further evaluation work to be done, both with respect to different parametrisations of PMCC, but also with respect to other schemes (e.g. [YBBG10, CFM10, BD10]). Finally, we would like to stress that, if a scheme appears robust against our attack, this does not imply that it is secure in any general sense. We envision our attack to be merely useful as an evaluation tool in the first stages of the development of minutiae-based fingerprint authentication systems; naturally, attacks that are specific to any particular scheme will perform better than the generic one described in this paper.

Acknowledgements

The author would like to thank Matteo Ferrara for his extensive help with setting up and integrating the PMCC toolkit for the experiments, as well as his feedback on earlier versions of this paper. The author would also like to thank Roel Peeters for insightful discussions. This work was supported by the Flemish Government, IWT SBO SPION, FWO G.0360.11N Location Privacy, and by the Research Council KU Leuven: GOA TENSE; and by the European Commission through the FIDELITY project (contract number 284862).

References

[BD10] Julien Bringer and Vincent Despiegel. Binary Feature Vector Fingerprint Representation From Minutiae Vicinities. In *Proceedings of the Fourth IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS)*, pages 1–6. IEEE, 2010.

[BS01] Josef Bigün and Fabrizio Smeraldi, editors. *An Analysis of Minutiae Matching Strength*, volume 2091 of *Lecture Notes in Computer Science*. Springer, 2001.

[CFM10] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2128–2141, December 2010.

[CM06] Jiansheng Chen and Yiu-Sang Moon. A statistical study on the fingerprint minutiae distribution. In *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, volume 2, pages II–II. IEEE, 2006.

[FJ11] Jianjiang Feng and Anil K. Jain. Fingerprint Reconstruction: From Minutiae to Phase. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):209–223, 2011.

[FMC12] Matteo Ferrara, Davide Maltoni, and Raffaele Cappelli. Non-invertible Minutia Cylinder-Code Representation. *IEEE Transactions on Information Forensics and Security*, 7(6):1727–1737, December 2012.

[ISO05a] ISO/IEC. *ISO/IEC 19794-2: Information technology – Biometric data interchange formats – Part 2: Finger minutiae data*, 2005.

[ISO05b] ISO/IEC. *ISO/IEC 2382-37:2012: Information technology – Vocabulary – Part 37: Biometrics*, 2005.

[JFGR07] D. Torre-Toledano J. Fierrez, J. Ortega-Garcia and J. Gonzalez-Rodriguez. BioSec baseline corpus: A multimodal biometric database. *Pattern Recognition*, 40(4):1389–1392, April 2007.

[JHPB97] Anil K. Jain, Lin Hong, Sharath Pankanti, and Ruud M. Bolle. An Identity-Authentication System Using Fingerprints. *Proceedings of the IEEE*, 85(9):1365–1388, September 1997.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[MMYH02] Tsutomu Matsumoto, Hiroyuki Matsumoto, Koji Yamada, and Satoshi Hoshino. Impact of artificial gummy fingers on fingerprint systems. In *Electronic Imaging 2002*, pages 275–289. International Society for Optics and Photonics, 2002.

[Nag12] Abhishek Nagar. *Biometric Template Security*. PhD thesis, Michigan State University, 2012.

[RCB01] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001.

[Sim12] Koen Simoens. *Analysis of Fuzzy Encryption Schemes for the Protection of Biometric Data*. PhD thesis, KU Leuven, 2012.

[Sou] C. Soutar. *Biometric System Security*. White Paper, `http://www.bioscrypt.com`.

[UJ04] Umut Uludag and Anil K. Jain. Attacks on biometric systems: a case study in fingerprints. In Edward J. Delp and Ping Wah Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents VI, San Jose, California, USA, January 18-22, 2004, Proceedings*, volume 5306 of *Proceedings of SPIE*, pages 622–633. SPIE, 2004.

[YBBG10] Bian Yang, Christoph Busch, Patrick Bours, and Davrondzhon Gafurov. Robust minutiae hash for fingerprint template protection. In Nasir D. Memon, Jana Dittmann, Adnan M. Alattar, and Edward J. Delp, editors, *Media Forensics and Security II, part of the IS&T-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 18–20, 2010, Proceedings*, volume 7541 of *SPIE Proceedings*. SPIE, 2010.