Genetic Algorithm Attack on Minutiae-Based Fingerprint Authentication and Protected Template Fingerprint Systems

Andras Rozsa
University of Colorado
Colorado Springs, CO, USA
arozsa@uccs.edu

Albert E. Glock, Jr.
University of Colorado
Colorado Springs, CO, USA
aglock@uccs.edu

Terrance E. Boult
University of Colorado
Colorado Springs, CO, USA
tboult@vast.uccs.edu

Abstract

This paper describes a new generic attack against minutiae-based fingerprint authentication systems. The goal of the attack is to construct a fingerprint minutiae template that matches a fixed but unknown reference template. The effectiveness of our attacking system is experimentally demonstrated against multiple fingerprint authentication systems. The paper discusses this attack on two leading privacy-enhanced template schemes and shows it can easily recover high matching score templates. A more general and novel aspect of our work is showing that despite high scores of the attack, the resulting templates do not match the original fingerprint and therefore the underlying data is still privacy protected. We conjecture that the ambiguity caused by collisions from projections/hashing during the privacy-enhanced template production provides for a multitude of minima, which trap attacks in a high-score but non-authentic region.

1. Introduction

Biometric authentication systems using physiological or behavioral traits are becoming more and more popular compared to traditional systems [1]. Physiological traits are characteristics of a person's body; examples include, but not limited to fingerprints, irises, palm veins or faces. Behavioral traits are related to some pattern of person's behavior such as typing rhythm or voice. The popularity of biometric systems originated from the convenience provided to their users – there are no passwords to remember or tokens to possess – the users only need to use their biometric traits for authentication.

Authentication systems using biometric traits work as follows. First, users need to be enrolled into the system by capturing their biometric traits and storing them in a form of a reference template. Enrolled users can authenticate themselves by providing fresh biometric traits which are captured by the system and transformed into a sample template. The stored reference template is then compared against the sample template. If the two templates match – their similarity is high enough – then the two templates originated from the same user and the user is deemed authenticated. Each system employs an algorithm called the matcher to compare the reference and sample templates and estimate their similarity by calculating their matching score. A match means that the score is higher than a predefined threshold value. If the fixed threshold is too high, then some biometric samples originating from the legitimate user will be falsely rejected and the user needs to provide a new sample. False rejections cause user frustration and reduction in productivity due to repeated verifications. On the other hand, with a low threshold, the system will falsely accept biometric samples coming from illegitimate users, which is a security issue as users can be impersonated.

Fingerprint authentication systems are one of the most widely used biometric authentication systems today due to the reduced size of fingerprints and a widespread social acceptance of such technologies [2]. In spite of the advantages provided by fingerprint authentication systems, they are still vulnerable to attacks [3] and there are many security concerns related to biometric systems [4]. The balance between security and privacy is a growing and important research area [5].

In this paper we introduce a novel approach to attacking minutiae-based fingerprint authentication systems, and show that score-based attacks may compromise security but do not compromise privacy of leading privacy-enhanced algorithms. The rest of the paper is organized as follows: in Section 2 we discuss background, Section 3 surveys related work, and Section 4 describes genetic algorithms and our attack. In Section 5 we introduce the targeted systems and dataset, Section 6 reports our results and, finally Section 7 concludes.

2. Background

This section introduces minutiae as the cornerstone of fingerprint authentication and then provides an overview of possible attacks on fingerprint authentication systems in general.

2.1. Minutiae

Fingerprints are patterns of ridges and valleys on the surface of fingertips. Since each individual has unique fingerprints, they can be used for personal identification

and authentication. The uniqueness of a fingerprint is determined by local ridge characteristics and their relationships.

Minutiae are special ridge characteristics: they are local discontinuities of ridges in the fingerprint pattern [6]. In 1986, ANSI proposed four minutia-types for minutiae classification: ridge ending, ridge bifurcation, compound, and all other cases are identified as undetermined [7]. Overall, more than 100 local ridge characteristics have been identified [8] – they are not distributed evenly and some of them are hard to observe.

After identifying minutiae in a fingerprint image, each minutia m can be defined as a triplet $m = \{x_m, y_m, \theta_m\}$, where x_m and y_m define the minutia location in the image and θ_m is the minutia orientation in the range of $[0, 2\pi)$. The extracted minutiae are used to construct a fingerprint minutiae template, usually consistent with the ISO/IEC 19794-2 [9] standard.

2.2. Attacking fingerprint authentication systems

Fingerprint authentication systems are still vulnerable to attacks. In [3] Ratha et. al surveyed 8 different points of attacks against general biometric verification systems. These possible attack points are depicted in Figure 1.

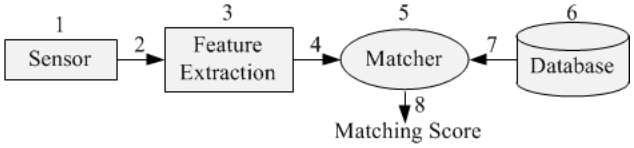

Figure 1: Attack points of a biometric verification system

Type 1 attacks aim directly at the entry point of the system – the sensor. From the attacker's perspective this is the best point to attack since no knowledge about the biometric authentication system and/or access privileges is necessary. After collecting a fingerprint – from a surface for example – an artificial, dummy finger can be created within few hours and used for the attack [10][11].

Type 2 attacks aim at the communication channel between the sensor and the feature extractor. The attack can be as simple as replaying a previously intercepted fingerprint image, trying several images hoping that there is a similarity or reconstructing synthetic fingerprint images from a set of minutiae. While challenging, *fingerprint reconstruction* has been demonstrated in [12] and [13].

Type 4, 7 and 8 attacks also aim at communication channels. Replaying previously intercepted content might work. Otherwise, a priori information about the exchange format is needed to construct synthetic content. Possible defense techniques against communication channel attacks can be simply protecting the content by using encryption, and some challenge-response based system can prevent content coming from attackers to be handled as legitimate.

The targets of type 3 and 5 attacks are the feature extractor and the matcher. These attacks might be performed as Trojan Horse attacks to change their behavior, so protecting the integrity of those components is critical.

Finally, type 6 attacks aim at the template database storing the reference templates. Templates can be deleted or replaced and the attacker can also try to reverse-engineer reference templates to recover the minutiae template. If the attempt is successful, type 4 attacks can be launched. By reconstructing the fingerprint image from the recovered minutiae template type 2 attacks are also possible. Furthermore, by creating a dummy finger from the reconstructed fingerprint image, type 1 attacks can be launched as well.

3. Related work

Our attack system launches type 4 attacks where the goal is to synthesize minutiae template files resulting in a match against given reference template. Consequently, we focus our discussion of related work on this type of attack.

After examining a generic attack model against biometric authentication systems, Ratha et al. illustrate that in case sufficiently many minutiae points are present in the reference template, then synthetizing a matching target minutiae template by exhaustively searching the space of all possible templates is infeasible [14]. In other words, brute-force attacks are not possible against reference templates containing "enough" minutiae points. In general, the feature extractor can identify and extract 40-100 minutia points from a good quality fingerprint image and that number is sufficiently high.

Hill climbing techniques have been designed and implemented to attack different biometric systems (e.g. face- and fingerprint-based systems).

Uludag and Jain present such an attack against minutiae-based fingerprint authentication systems [15]. Their attacking system inputs synthetic minutiae sets to the matcher trying to gain access to the system in place of a genuine user. By using the matching scores returned by the attacked system and the characteristics of the minutiae sets, the attacking system tries to generate a minutiae set to produce a sufficiently high matching score for positive authentication. The attack system follows 5 steps described below.

Step 1 (Initial guessing): The attacking system generates predefined number of synthetic templates containing randomly generated minutiae points. The number of synthetic templates and the number of contained minutiae are configurable. In their implementation, 100 templates are created with 25 random minutiae points.

Step 2 (Try initial guesses): Their system attacks a user-account with synthetic templates generated in Step 1 and accumulates the corresponding matching scores.

Step 3 (Pick the best initial guess): The attacking system identifies the best guess by selecting the synthetic template with the highest matching score.

Step 4 (Try modification set): Their system modifies the best guess identified in Step 3 by applying one of the following four operations: perturbing an existing minutia, adding a new minutia, replacing an existing minutia or deleting an existing minutia. If any of these modified templates produce a higher matching score than the previous best, then that template is declared as the best.

Step 5 (Obtaining result): In case the best is a match against the reference template – its score is higher than the threshold, the attack stops as it is successful. Otherwise, Step 4 needs to be repeated.

When Uludag and Jain designed their attacking system, they made the assumption that the attributes (size and resolution) of the images used to create the original reference templates are known. This can be considered as a valid assumption as sensor manufacturers announce those values. As the resolution determines the minimum inter-ridge distance – 9 pixels for 500 dpi – they applied a rectangular grid with 9 x 9 pixel cells to prevent their system from creating minutiae points too close to each other. Also for minutia orientation, they quantized the $[0,2\pi)$ range into 16 equally spaced intervals. These design decisions drastically reduced the number of possible minutiae points in the search space and their attack system is claimed to be quite effective. However, no code is available for direct comparison.

Pashalidis published a conceptually similar attack, which differs in three aspects [16]. First, he uses a simulated annealing attack, an advanced form of hill climbing technique, which is able to escape from local optima. Second, his attacking system is optimized to attack vicinity-based matchers, it does not explore the whole search space and its goal is not to recover the original minutiae template. Instead, it tries to create one template with a sufficiently high matching score against the reference template. Third, the attack is applied against PMCC [18] – a template protection technique, where the attacker is explicitly allowed to access the protected reference template, so an offline attack can be launched. The description of the attack published by Pashalidis summarized below.

Step 1: An initial template is generated which consists of multiple vicinities. The exact number of vicinities is horizontally and vertically configurable (e.g. 4 and 4: the image-size is divided into 16 vicinities). Each vicinity is populated with a number of random minutiae (with random location and orientation). The number of minutiae is randomly chosen from a configurable range (e.g. [2, 4]).

Step 2: The matching score of the initial template is calculated.

Step 3: A new candidate template is constructed by replacing one of the vicinities by one new and randomly generated vicinity. If the candidate template produces a higher matching score than the original, the candidate template is kept and the procedure is repeated for a predefined, configurable number of iterations. Compared to the hill climbing technique simulated annealing sometimes replaces the current template with another candidate having a worse matching score. The replacement is done by a configurable probability and its goal is to avoid the search getting trapped in a local optimum.

Although the attack of [16] does not apply a minimum inter-ridge distance, based on the results of the conducted experiments, it also appears to be effective. Unfortunately, no code is available for direct comparison.

As presented hill climbing or simulated annealing can be used to obtain templates with increased scores. While sufficiently large optimization problems can become difficult or even infeasible for local search algorithms such as hill climbing techniques, genetic algorithms can still perform well [19]. This advance of genetic algorithms derives from the implicit parallelism within populations, fitness-proportionate reproduction, and a high crossover rate [20].

While both of these prior works shows how to obtain attack templates with moderate to high score, that alone does not answer if the templates can compromise privacy. This paper develops a new attack tool and then shows that while attacks may impact security of some templates they do not directly attack the privacy. Our tool is free to qualified researchers to test their own template technology, but for security reasons is not being released publically.

4. Approach

After giving a general overview of genetic algorithms, this section provides an overall summary of the designed attacking system. The major goal of the design was to keep the system as simple and flexible as possible, so it can be easily adapted to attack any minutiae-based fingerprint authentication system by applying small modifications such as configuration changes.

4.1. Genetic algorithms

A Genetic Algorithm (GA) is simply a robust searching and optimization technique based on ideas originating from genetic and evolutionary theory [21] and biological genetic development principles first developed by John Holland [22]. GAs are generally described in terms of the processes observed in natural selection, genetics and

evolution.

The genetic algorithm or genetic process starts with an initial set of solutions – called a *population*. Each individual in the population – called a *chromosome* – represents a solution to the problem. The chromosomes of the initial population are usually created randomly. The chromosomes evolve through iterations – called *generations* – as the GA modifies the chromosomes by genetic operators such as selection, crossover and mutation. Also, they are evaluated by using some measure of fitness to assess their applicability as solutions of the given problem, which is referred to as the *fitness value*. A new chromosome – called an *offspring* – is usually created by selecting its parents from the current population, "merging" or "breeding" them by applying some crossover operation and/or mutating the offspring.

The three genetic operators are the keys to the GA. First, to produce better and better chromosomes their parents need to be chosen with care – fitter parents need to be selected with higher probabilities. Second, a crossover operation performs a kind of mixing – the chromosomes of parents are combined together to create the offspring. Finally, mutation randomly changes part(s) of the chromosome to introduce something "new" to the population, which may have beneficial results. Crossover in GA is responsible for exploring local maxima in the hypothesis space, while mutation helps to break out from spurious solutions which are far from the global optimum, so the GA is able search for solutions globally.

Chromosomes are usually represented as binary-strings. Crossover recombines the representations of the selected parents and mutation is as simple as inverting bit(s). Proper selection of the mutation probability is critical to the success of genetic algorithms. A low mutation rate would not introduce genetic diversity – the GA could not escape from local optima – on the other hand, too high mutation probability would transform GA into a random search algorithm.

4.2. Attacking system

The primary goal of our attacking system is to synthetize minutiae templates with sufficiently high matching scores against a reference template by applying a genetic algorithm. Carefully defining the search space is important for any optimization technique, as the number of possible solutions can determine the overall performance of the algorithm.

Similar to the attack presented by Uludag and Jain [15], our attack system applies a rectangular grid with a configurable cell size. Each cell of the grid can contain at maximum one minutia. As the resolution of the fingerprint image determines the minimum inter-ridge distance, the grid prevents our system from creating minutiae unnecessarily close to one another. For example, the minutiae templates released with Minutia Cylinder-Code Software Development Kit (SDK) Version 1.4 [23] have been extracted from fingerprint images with the size of 351 x 492 pixels and 500 dpi resolution. The resolution defines the minimum inter-ridge distance as 9 pixels, so the grid can be configured to consist of 9 x 9 pixel cells, having 39 columns and 54 rows. The spatial location of minutiae can be represented by 12 bits – 6 bits per row and column respectively. Considering minutia orientation to be quantized into 16 equally spaced intervals, so defined by 4 bits, a minutia can be represented by a 16-bit binary string. Our attacking system takes another step to further reduce the number of possible solutions. The perimeter of the fingerprint images usually contains fewer minutiae than the center area due to the shape of the images, so our system can be configured to down-weight much of the perimeter by setting a centralized "target area" within the grid, where all the generated minutiae are placed.

We randomly generate the initial population according to uniform distribution. Each chromosome represents a minutiae template containing configurable number of minutiae. Each minutia is defined by a randomly generated x, y and θ triplet. If the location of the newly generated minutia is already taken in the configured target area, then we ignore it and generate another random minutia. The size of the initial population is a configuration parameter of our attack system.

The populations of all further generations are created by following the genetic process. The three genetic operators are responsible for performing a global search in the solution space. Selection needs to imitate natural selection to make the genetic algorithm mimic the evolutionary process. By simply selecting the few best chromosomes for breeding, the genetic algorithm can easily produce inbred populations lacking diversity. To prevent such inbreeding, our genetic algorithm applies tournament selection, i.e. when the genetic algorithm needs to find a parent for breeding it randomly picks a predefined number of chromosomes from the population and then selects the best/fittest of them. The number of randomly selected chromosomes is called tournament size and needs to be chosen carefully. With a relatively large tournament size compared to the population size, the fittest chromosomes of the population can completely prevent the rest from being selected for breeding, so an inbred population can be produced as an unwanted result. As described later, our genetic algorithm uses a crossover operator breeding two parents. Since tournament selection is used to select one parent, it is possible to use different tournament sizes for parent1 (mother) and parent2 (father). Another aspect of selection needs to be considered is elitism or elitist selection. In practice, elitism simply means allowing the best/fittest chromosome(s) to be selected for breeding in multiple,

consecutive generations. This process further increases the probabilities that very few of the fittest chromosomes breed throughout multiple generations and it is indeed very useful to prevent them from not being selected at all, while the chances of producing an inbred population are still low. Of course, the size of the elite and number of consecutive generations a chromosome of the elite can breed must be carefully configured.

The second genetic operator that requires careful consideration is crossover. After selecting the parents, crossover operation is responsible for forming offspring chromosomes by using the genome of the parents. As each chromosome represents a possible solution to the problem, by mixing the parents the genetic algorithm tries to create a new, fitter chromosome than its parents. There are several different crossover operators regarding how the chunks need to be selected to form the new offspring. Our system applies minutiae-based uniform crossover: the operator uses each minutia of the parents with 50-50% probability in offspring1 or offspring2. Considering the parent chromosomes consist of n different minutiae, there are 2^n different offspring chromosomes to be created, so even if the same parents are selected for breeding multiple times, the diversity of the offspring chromosomes can be maintained.

While crossover is responsible for exploring a local hill by combining the existing chromosomes, mutation – the third genetic operator – randomly changes them by flipping bit(s) to break out from there. If the introduced change has beneficial results and helps to create fitter chromosomes, the genetic algorithm can search for possible solutions globally in the whole search space. The implemented mutation operator used by our genetic algorithm relies on a new technique called a Sequential Mutation Method (SMM) [24]. SMM has been designed to improve genetic algorithms' overall performance by speeding up its convergence. The basic idea behind SMM is to focus mutation only on a small section of the chromosomes. In each generation one gene is selected from each chromosome, and the mutation is applied on the bits of that gene. The bits of the gene to be mutated are selected randomly. Our solution mutates a configurable number of bits within one gene – the binary representation of one minutia – in each generation. The gene selected for mutation is determined by the generation of the actual population. In the first generation the first gene is mutated, in the second generation the second gene and so on. Focusing the mutation on a small section (i.e. 16 bits) of the whole chromosome increases the probability that the outcome of the mutation operation will be beneficial.

After forming an offspring chromosome by following the genetic process of selection, crossover and mutation, our genetic algorithm is ready to create the minutiae template based on the binary string. The outcome of crossover can be a binary representation containing multiple minutiae with the same spatial location and mutation can also produce minutiae with spatial location out of the target area. Since our attack system is designed to synthetize minutiae templates containing a predefined, configurable number of minutiae, our system ignores the possible duplicates or out-of-scope minutiae points and uses new, randomly generated ones instead.

5. Targeted systems and dataset

Our attack system has been designed to attack minutiae-based fingerprint authentication systems by synthetizing minutiae templates with sufficiently high matching scores by using the described genetic algorithm. In this section we provide a short overview about the systems we have attacked and introduce the dataset that we used for performance evaluation.

5.1. Targeted systems

The following three minutiae-based fingerprint authentication systems have been attacked by our system: Minutia Cylinder-Code (MCC) [17], Protected Minutia Cylinder-Code (PMCC) [18] and Biotope [25][26]. While there are many privacy-enhanced systems, we focus on the two demonstrating the best performance in the FVC-Ongoing [27] competition on template protected fingerprint systems.

Since we designed our attack system to launch type 4 attacks against any minutiae-based system, no internal knowledge about the targeted systems is necessary. The attack system is able to launch offline attacks against MCC and PMCC systems by using Minutia Cylinder-Code Software Development Kit (SDK) Version 1.4 [23]. Also, for Biotope a custom interface was provided by the authors of [25][26] for the experimental attacks. A brief overview is provided here to highlight the basic differences of the targeted systems.

MCC creates a descriptor – a cylinder – for each minutiae point to describe its environment with respect to other minutiae points. Each cylinder maps a portion of the fingerprint image and from the cylinders of the MCC reference template the original minutiae template can be restored.

PMCC is based on MCC and its primary goal is to protect PMCC templates from being restored to their original minutiae templates. Each cylinder undergoes to a non-invertible transformation and is then represented by a fixed length bit-vector – the length of the bit-vector is configurable, values are 16, 32, 64 or 128 bits. Longer bit-vectors provide more accuracy, but less privacy [18]. The shorter the bit-vectors are, the more cylinders are mapped to the same bit-vector.

Biotope [25][26] couples biometrics with hashing or

cryptography to create biotokens. Basic biotopes produce match scores and – as of 2015 – currently have the highest reported accuracy in FVC-Ongoing [27] for the template protected category. Bipartite biotopes [25] authentication differs from MCC or PMCC since it is not based on matching scores but on key release. The system defines an authentication attempt as successful, if the target template makes the stored bipartite Biotope reference template release the "secret" from it. We will test on both basic biotopes and bipartite biotopes. While in practice the bipartite Biotope does not produce a matching score, the attack herein presumes we get the actual matching score as someone could implement the basic biotope algorithm based on the papers to estimate that score. We use custom code from the authors of [25][26] to produce scores.

5.2. Dataset

In our experiments we used the set of minutiae template files released with Minutia Cylinder-Code SDK Version 1.4 [23] as reference templates to attack. Those minutiae template files have been extracted from fingerprint images of Fingerprint Verification Contest (FVC) 2006 [28] DB2-B database by using an internal minutia extraction algorithm. The dataset contains 120 minutiae template files extracted from 12 fingerprint images (12 impressions) of 10 fingers.

6. Experiments

The ultimate goal of our genetic algorithm is to effectively synthetize minutia templates with high matching scores against reference templates for any minutiae-based fingerprint authentication systems.

By running some experimental attacks and evaluating their results we needed to find a close-to-optimal configuration for our genetic algorithm to work efficiently. The population size has been defined to contain 50 chromosomes in each generation. Tournament selection with asymmetric tournament sizes for the two parents provided the best results regarding to diversity and fitness value convergence: tournament size was configured to be 4 for parent1 and 2 for parent2. All offspring chromosomes go through mutation – a sequential mutation method in which exactly 1-bit of each applicable minutia gets mutated. Elitism has been configured to give the best three, younger than 5-generation old chromosomes another chance to be selected for breeding. This is our base-configuration.

There are only two parameters that vary between the attacks of different systems: the size of the grid's target area and the number of minutiae our system synthetizes in each minutiae template. By attacking a given reference template with synthetized minutiae templates representing various sizes of the target area containing different number of minutiae, and then by comparing the average matching scores of the randomly generated initial populations and the first few generations the best configuration can be identified. A smaller target area with fewer minutiae reduces the search space and also produces shorter chromosomes, so the goal of this process is to find the smallest possible target area containing as few minutiae as possible.

Since the dataset we have used is not too large, defining FAR (False Acceptance Rate) and FRR (False Rejection Rate) curves might not represent a realistic threshold value we could aim at in our experimental attacks. Furthermore, our goal was to design a system that can create synthetic minutiae templates with efficiently high matching scores, so we have targeted the best genuine impression's matching scores of the dataset while attacking a given reference template of certain authentication system. This is a more challenging task, but reaching the level of the best genuine impressions ensures that the attacks cannot be avoided by slightly increasing the threshold value.

To highlight the results of our concluded experiments we display the distributions of the matching scores for each targeted system. Genuine attempts represent the true matches within the dataset, i.e. an impression is matched against the other impressions of the same finger. Since the used dataset contains 12 impressions for each 10 fingers, the number of genuine attempts is 10 x 11 x 12 = 1320. The impostor attempts are the false matches meaning that each impression of one finger is matched against each impression of all other fingers. Since we avoid duplicates (i.e. finger1 vs. finger2 and finger2 vs. finger1) the number of the impostor attempts is 12 x 12 x 9 x 10 / 2 = 6480. Finally, as our attack system creates one synthetized minutiae template for each impression of the dataset, we have 10 x 12 = 120 synthetized attempts.

6.1. Minutia Cylinder-Code (MCC)

Surprisingly, placing only 12 minutiae in the highly reduced, centralized area of the grid produced the fittest populations. The target area was defined to be only 16% of the entire grid (40-40% of the grid's width and height). The reference templates of the whole dataset have been attacked with these settings with the genetic process limited to 200 generations.

Figure 2 summarizes the overall result of the attacks against the MCC system by displaying the distribution of all matching scores defined by the dataset (genuine and impostor attempts) and also highlighting all matching scores of the synthetic templates produced by our attack system. Due to the larger number of impostors in the dataset, and the fact that their matching scores are limited to a small range ([0.01, 0.025]), the chart is focused on the curves of genuine and synthetized attempts by limiting the

number of occurrences (vertical axis).

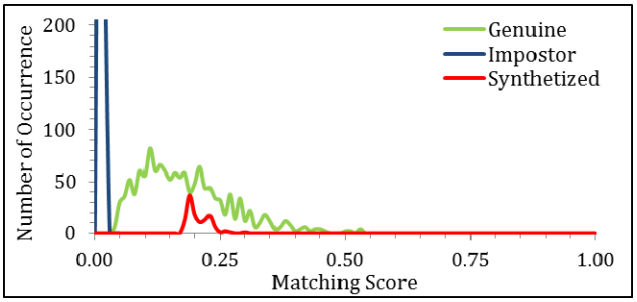

Figure 2: Distribution of matching scores (MCC system)

As the results show, MCC templates can be compromised by even very small synthetized minutiae templates. Our attack cannot be avoided by simply increasing the threshold to a tolerable FRR, as the majority of the genuine attempts produce lower scores than the synthetized templates. Compromising an MCC template also violates the user's privacy: the original biometric trait can be restored and reused by the attacker, and with that the user also becomes traceable through different systems – wherever that biometric trait is used.

6.2. Protected Minutia Cylinder-Code (PMCC)

Since PMCC templates using 128-bit long bit-vectors are vulnerable regarding privacy [18], we targeted PMCC systems applying the shorter bit-vectors. We configured our genetic algorithm to use only 30.25% (55-55% of the grid's width and height) of the entire grid while synthetizing minutiae templates with 25 minutiae in each. This configuration was used against PMCC systems using 16, 32 or 64 bits long bit-vectors as well.

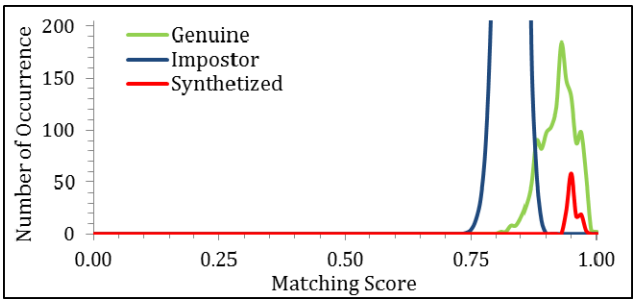

Figure 3: Distribution of matching scores (PMCC with k=16)

As Figure 3 shows, PMCC system with the shortest bit-vectors (16-bit) is the least accurate as there is a relatively large range where the curves of genuine and impostor attempts overlap each other. The shorter the bit-vectors are, the more cylinders are mapped to the same bit-vector. Due to the large number of such collisions, our genetic algorithm was set to run through only 100 generations, yet produced scores higher than the majority of true matches.

Figure 4 shows the overall distribution of matching scores for the PMCC system using 32-bit long bit-vectors.

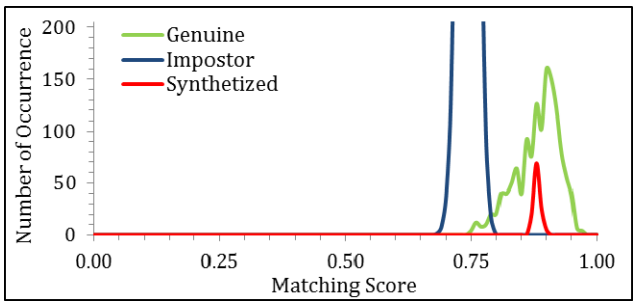

Figure 4: Distribution of matching scores (PMCC with k=32)

As expected, this system is more accurate than the previous. Due to fewer collisions our genetic algorithm was configured to apply 200 generations after which it was producing scores about the average matching score.

The PMCC system with 64-bit long bit-vectors is the most accurate targeted PMCC system. Similar to the previous attack the genetic process was limited to 200 generations. While the matching scores of the synthetized templates are lower compared to the genuine attempts, they are still sufficiently high as shown by the distribution of matching scores in Figure 5. By further adjusting our genetic algorithm or simply running through more generations, higher attack scores can be produced. The purpose of applying only 200 generations was simply to limit the time consumption of the attacks.

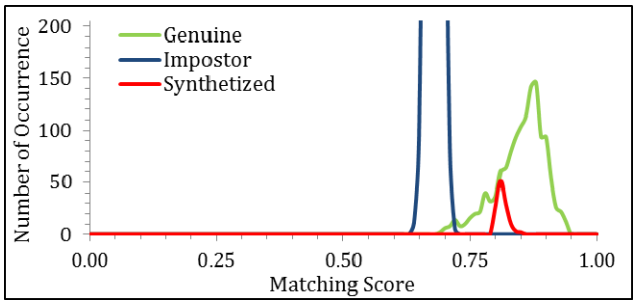

Figure 5: Distribution of matching scores (PMCC with k=64)

Our attacks against PMCC systems can be considered successful regarding the matching scores of the synthetized templates, so the security of these systems has been compromised. The privacy of the attacked PMCC systems has not been compromised. After enrolling the synthetized minutiae templates and the reference templates into MCC system and matching them against each other, the low matching scores produced indicate that the templates do not originate from the same fingerprint. Consistent with our previous discussion of PMCC systems, we did not expect to compromise privacy.

6.3. Biotope

To attack Biotope [25][26] templates, the same GA-configuration was used as against PMCC templates, but it

needed to be modified when the reference templates of the third finger were attacked. There was a huge degradation in the average matching scores of the initial populations compared to other fingers. By increasing the target area size to 49% of the grid (70-70% of grid's height and width) and applying 50 minutiae in the synthetized minutiae templates, this degradation was eliminated. The larger size would also work for other fingers but would have been slower. The Biotope system investigates the number of minutiae contained by the target and reference templates and if there is a significant difference between them, it does not give high matching scores. Less than 50 generations were needed for our genetic algorithm to produce high matching scores against the reference templates (shown by Figure 6).

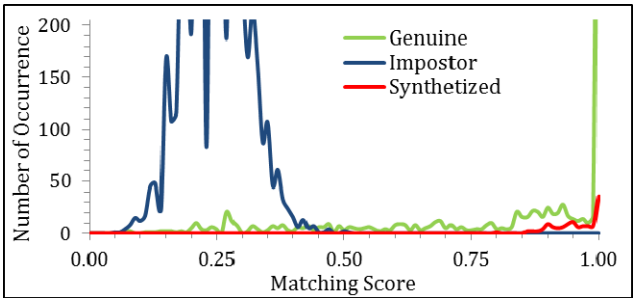

Figure 6: Distribution of matching scores (Biotope)

There are both chaffs and collisions in biotope matching. If using just the matching scores as in [26], the attacks would compromise the system's security for basic biotopes. Since in [25] the authentication-decision in the bipartite form is not based on the matching scores, the curves of the genuine and impostor attempts don't address the security of the system. *While the attacks produced templates with high matching scores, none of them released the bipartite key* when the templates were matched with bipartite biotopes [25]. **Our synthetized templates have produced sufficiently high scores, but neither the security nor the privacy of the bipartite Biotope system was compromised.**

7. Conclusion

We have introduced a new attack system that uses a genetic algorithm to efficiently synthetize minutiae templates that generate high matching scores across the targeted fingerprint authentication systems. As discussed, submitting these templates to different systems has different security and privacy ramifications. MCC templates are vulnerable regarding security and privacy, PMCC templates and basic biotopes maintain privacy while security is compromised, and finally, bipartite Biotope templates provide both security and privacy.

Further design improvements to our attack system could leverage pseudo-random initial populations or initial populations with chromosomes derived from real fingerprints or a hybrid/parallel implementation (fine- or coarse-grained) to maintain higher levels of diversity within the genetic process. Designing and implementing a self-adaptive genetic algorithm might also improve the overall performance of the genetic process and mitigate the need for manual tuning to find close-to-optimal configurations experimentally. Applying larger target areas with more minutiae points and more iteration would allow our genetic algorithm to produce higher matching scores. We conjecture that our attack system will almost always be able to find a false match with higher matching score than a true match. This means that while security might be compromised, the true minutiae set cannot be approximately reconstructed.

Future work will continue testing with other datasets to ensure results are not dataset dependent. The most interesting new result – showing that high-scoring attack templates for either PMCC or Biotope systems do not match the underlying data – is unlikely to be data dependent, but much larger test sets are needed for completeness.

References

[1] A.K. Jain, R. Bolle, and S. Pankanti, (Eds.). Biometrics: Personal Identification in Networked Society, Kluwer Academic Publishers, 1999.

[2] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. Handbook of Fingerprint Recognition, Springer, 2003.

[3] N.K. Ratha, J.H. Connell, and R.M. Bolle. An analysis of minutiae matching strength, Proc. AVBPA 2001, Third International Conference on Audio- and Video-Based Biometric Person Authentication, pp. 223-228, 2001.

[4] A. K. Jain, A. Ross, and U. Uludag. Biometric template security: Challenges and solutions, Proc. of 13th European Signal Processing Conference (EUSIPCO), Antalya, Turkey, 2005.

[5] Campisi, Patrizio. Security and Privacy in Biometrics. Springer, 2013.

[6] D. Maio and D. Maltoni. Direct gray-scale minutiae detection in fingerprints, IEEE Trans. Pattern Anal. Machine Intell., vol. 19, no. 1, pp. 27–40, 1997.

[7] American National Standards Institute. Fingerprint Identification-Data Format for Information Interchange. New York, 1986.

[8] A. Moenssens. Fingerprint Techniques. London: Chilton Book Company, 1971.

[9] Information Technology—Biometric Data Interchange Formats—Part2: Finger Minutiae Data, ISO/IEC 19794-2:2005, 2005.

[10] T. Putte and J. Keuning. Biometrical fingerprint recognition: don't get your fingers burned, Proc. IFIP TC8/WG8.8, Fourth Working Conf. Smart Card Research and Adv. App., pp. 289-303, 2000.

[11] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino. Impact of Artificial Gummy Fingers on Fingerprint

Systems, Proc. of SPIE, Optical Security and Counterfeit Deterrence Techniques IV, vol. 4677, pp. 275-289, 2002.
[12] A. Ross, J. Shah, and A.K. Jain. From Template to Image: Reconstructing Fingerprints from Minutiae Points, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 29, no. 4, pp. 544-560, Apr. 2007.
[13] R. Cappelli, A. Lumini, D. Maio and D. Maltoni. Fingerprint Image Reconstruction from Standard Templates, IEEE Transactions on Pattern Analysis Machine Intelligence, vol.29, no.9, pp.1489-1503, September 2007.
[14] 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.
[15] U. Uludag and A. K. Jain. Attacks on biometric systems: A case study in fingerprints, Proc. SPIE-EI 2004, Security, Steganography and Watermarking of Multimedia Contents VI, San Jose, CA, pp. 622–633, 2004.
[16] A. Pashalidis. Simulated annealing attack on certain fingerprint authentication systems, In Proceedings of the Special Interest Group on Biometrics and Electronic Signatures, Lecture Notes in Informatics (LNI), A. Brömme, and C. Busch (eds.), Bonner Köllen Verlag, pp. 63-74, 2013.
[17] 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.
[18] 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.
[19] Prügel-Bennett, A. When a genetic algorithm outperforms hill-climbing. Theoretical Computer Science, 320(1), 135-153, 2004.
[20] M. Mitchell, J. Holland, S. Forrest, When will a genetic algorithm outperform hill climbing? in: J. Cowan, G. Tesauro, J. Alspector (Eds.), Advances in Neural Information Processing Systems, Morgan Kauffman, San Francisco, CA, 1994, pp. 51–58.
[21] D. E. Goldberg. Genetic Algorithms in Search, Optimization & Machine Learning, Addison Wesley, 1989.
[22] J. H. Holland, Adaptation in Natural and Artificial System, University of Michigan Press, 1975.
[23] Minutia Cylinder-Code SDK. Biometric System Laboratory, DISI – University of Bologna. 2014. Available: http://biolab.csr.unibo.it
[24] M. Baradaran Nia, Y. Alipouri. Speeding Up the Genetic Algorithm Convergence Using Sequential Mutation and Circular Gene Methods. 9th International Conference on Intelligent Systems Design and Applications, pp. 31-36, 2009.
[25] W. Scheirer and T. Boult. Bipartite Biotokens: Definition, Implementation, and Analysis, in Proc. of IEEE/IAPR Int. Conf. on Biometrics, 2009, pp. 775–785.
[26] T. E. Boult, W. J. Scheirer, and R.Woodworth. Revocable fingerprint biotokens: Accuracy and security analysis, in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.
[27] B. Dorizzi, R. Cappelli, M. Ferrara, D. Maio, D. Maltoni, N. Houmani, S. Garcia-Salicetti and A. Mayoue, "Fingerprint and On-Line Signature Verification Competitions at ICB 2009", in proceedings International Conference on Biometrics (ICB), Alghero, Italy, pp.725-732, June 2009
[28] R. Cappelli, M. Ferrara, A. Franco and D. Maltoni. Fingerprint verification competition 2006, Biometric Technology Today, vol.15, no.7-8, pp.7-9, August 2007.