A Security Analysis of RF Biometric Fingerprint Scanners

Kit Mun Chan, Ana Pop, Shadi Safarkhah, Gurpreet Virdi

***Abstract*— The security policy of RF biometric scanners is concerned with origin integrity. To achieve this, there are three layers of security that a user must pass through: the scanning layer, the processing layer and the storing layer. At the scanning layer, the biometric tool can be fooled by an attacker using a gummy finger – a fake mold of a fingerprint that is recognized by the scanner as an authentic fingerprint. At the processing layer, the biometric data, i.e. the fingerprint, can be intercepted as it is computed by the complementation function, and recovered for later use by an attacker. Finally, at the storing layer, the template used to compare the complementary data can be accessed and corrupted so that the user will no longer be able to login, or it could be recovered remotely by an attacker if it is stored on the computer's hard drive. We conclude that the APC BioPod's security policy can be breached because of vulnerabilities at the three layers.**

***Index Terms*— biometrics, origin integrity, fingerprinting, RF scanning, APC BioPod.**

I. INTRODUCTION

Security measurements are a major factor in computers. Whether the computer is used for commercial, government, military or even personal use, there is a great need for security. To prevent unauthorized access to computers, several security measurements have been introduced. Biometrics is now among the most popular and reliable methods of achieving a more secure system and is currently used as a highly secure way for identification and personal verification.

The number of security breaches and transaction fraud is increasing every day and as a result, a highly secure identification and personal verification technology has a great and much needed demand. Biometrics is a recognition system based on each individual's behavioral and physiological characteristics.

Fingerprints are unique and no two individuals have the exact same friction ridge. In the case of a damaged finger, the fingerprint will be restored without change after it has healed. Fingerprints do not change significantly with age.

Dr. Henry Faulds published a paper in 1880 regarding fingerprinting and how it can be used in police department. Fingerprinting devices are now used as authentication methods to log in to systems. Before using a fingerprinting device, the user must pass three layers of security. The first stage is the scanning stage, which involves the user allowing his fingerprints to make contact with the scanner. The second stage is the processing stage, which computes the complementary information given the fingerprint and compares the data with the template. The third stage is the storing stage, which stores the template.

In this report, we outline details of how each stage works, as well as describe how each stage was attacked in order to try to break the security policies of the device. Also, we explain which principles of secure design this device has broken based on our experience with it.

II. SCANNING LAYER

A. How the APC BioPod and the TruePrint® Fingerprint Scanning Technology Works

The APC BioPod fingerprint scanning device implements the TruePrint® fingerprint scanning technology. This technology utilizes semiconductor-based sensors that use RF signals to detect the fingerprint ridge and valley pattern underneath the skin's surface.

The sensing area of the APC BioPod consists of tiny RF sensor plates inserted near the surface of a semiconductor and above the continuous conductive plane of the semiconductor as shown in Figure 1.

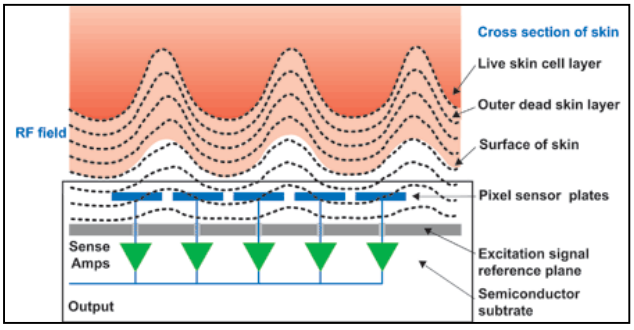

Figure 1. How TruePrint® fingerprint scanning technology works [1]

Each sensor plate is connected to an amplifier that converts the potential sensed on the plates into voltages, which represent the fingerprint pattern. The layer of live skin cells just beneath the surface of our skin is highly conductive. When an RF signal is applied between the conductive layer of the semiconductor and the finger, the amplitude of the generated electric fields resemble the shape of the layer of live skin cells which, in this case, is the ridge and valley pattern of the fingerprint.

The amplitude of the generated electric fields are then sensed by the RF sensor plates and converted into voltages by an amplifier.

B. Security of the TruePrint® Fingerprint Scanning Technology

The APC BioPod fingerprint scanner is not more secure compared to similar devices using capacitive or optical scanners. Although the scanning technology utilizes the conductive layer of live skin cells beneath the skin's surface, a gummy finger can still successfully fool the scanner, despite reports that it is less susceptible to such attacks. This is because the conductance of a properly made gummy finger is almost similar to that of a real live finger. Figure 2 shows a comparison of the values of electric resistance for a live, gummy, and silicone finger.

	Moisture	Electric Resistance
Live Finger	18%	16 Mohms/cm
Gummy Finger	23%	20 Mohms/cm
Silicone Finger	impossible to measure	impossible to measure

Figure 2:. Moisture content and electric resistance values of a live, gummy, and silicone finger. [3]

Based on the values of electric resistance in Figure 2, the conductance values of a live and gummy finger can then be calculated as:

	Conductance (1/Electric Resistance)
Live Finger	62.5×10^{-9} Siemens.cm
Gummy Finger	50.0×10^{-9} Siemens.cm

Our group carried out tests on the APC BioPod fingerprint scanner with a gummy finger and has been successful during several attempts in breaking the system.

C. Tests done with Gummy Fingers on the BioPod

Our group perfected a recipe to make a gummy finger that can fool the BioPod scanner into believing that the finger is real. We placed a solution of 90% powder gelatin and 10% water in a bowl that was placed inside a frying pan filled with water heated at 60°C. The fingerprint was first embedded in a silly putty mold, and then the gelatin mixture was poured on it to form a thin layer of hot gelatin. Figure 3 shows the setup on the left, and the final gummy finger on the right. It is important to ensure that the gelatin layer is as thin as possible so that the gummy finger would be very flexible. The flexibility of the gummy finger is crucial in determining whether the BioPod scanner is able to scan it properly. Initially, we had made some rather rigid gummy fingers, and the BioPod scanner was not able to detect them at all.

Figure 3. Gummy finger mold and gummy finger

III. PROCESSING LAYER

A. Description of how the Processing Layer Works

The processing layer uses one of various algorithms to translate the authentication information (fingerprint) with a complementation function (algorithm) into complementary information (data) that will be then compared to a template resident on the computer's hard drive. This function is of the utmost importance because it must be accurate enough to distinguish different ridge depths and their divisions (called minutiae) and accurately convert these physical measurements into numeric values. The processing stage is composed of two different parts: enrollment and verification. Enrollment creates the template, while verification compares the scan with the template.

B. Algorithm used by the Processing Layer

The algorithm uses correlation filters, which classify fingerprints based on localized high frequency features. A two-dimensional input image array in the space domain is transformed by a function into the Fourier transform in the spatial frequency domain. That is, $f(x)$ and $F(u)$ --> FT. During the enrollment session, an image, $f_0(x)$ is obtained and a filter function $H(u)$ is derived from this image. The correlation function, $c(x)$, is taken as

$$c(x) = \int_{-\infty}^{\infty} f_1(v) f_0^*(x+v)\,dv$$

where $f_1(v)$ is the changed version of the input and $f_0^*(x+v)$ is the complex conjugate of the original verification [5].

The biometric template is the filter function $H(u)$. This filter function creates a peak at the output of the system called a *correlation peak,* which approaches a delta function. The correlation peak is identified in a correlator system and its position is used to track one part of the fingerprint. A measure of how close $f_1(x)$

and $f_0(x)$ are can be derived from a scalar from the correlation plane [5].

The filter function must be able to consistently authenticate the same user (decrease the false acceptance rate, FAR) while accounting for minor imperfections in the valid user's skin (decrease the false rejection rate, FRR).

Figure 4 shows the enrollment stage of the scan.

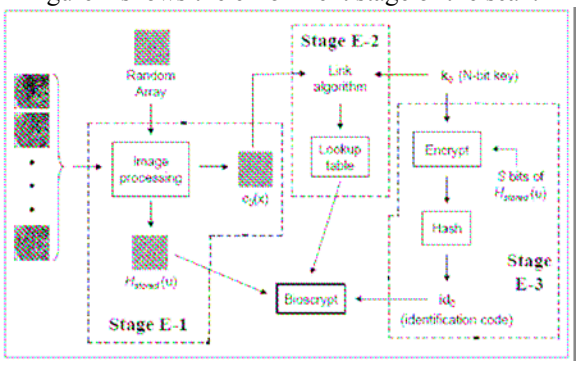

Figure 4. Enrollment of the fingerprint [5]

Stage E-1 combines the input fingerprint with a random phase array to create two output arrays: $H_{stored}(u)$ and $c_0(x)$. $H_{stored}(u)$ is a modified version of $H(u)$ that stores only the phase component of $H(u)$. Limiting the storage accomplishes security of the fingerprint. However, the magnitude is also needed to get the best possible results against the template. It is generated each time the verification procedure is executed using a transitory filter [5]. Figure 5 shows in detail stage E-1.

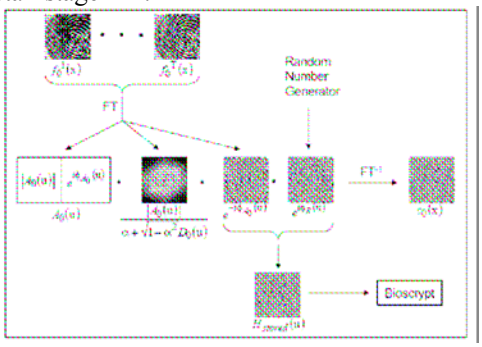

Figure 5. Stage E-1 in enrollment [5]

The three inputs are the fingerprint, a randomly generated phase-only array, and an N-bit cryptographic key, which are all independent from each other. The outputs of this stage are

$$c_0(x) = FT^{-1}\{A_0(u) \bullet |H_0(u)| \bullet H_{stored}(u)\}$$

$$H_{stored}(u) = e^{-i\phi_{A_0}(u)} e^{i\phi_R(u)}$$

Stage E-2 uses a link algorithm to link a cryptographic key k_0 to the pattern $c_0(x)$ [5]. Figure 6 shows the details of this stage.

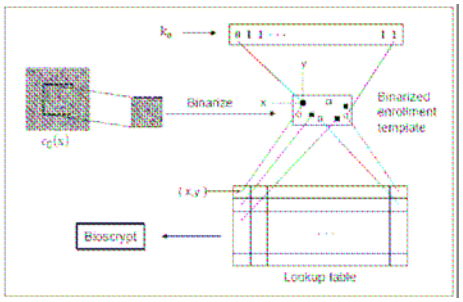

Figure 6. Stage E-2 in enrollment [5]

Each key bit must be represented by choosing the central 64x64 parts of $c_o(x)$ to undergo binarization and selecting L values to represent each bit. The reason for choosing only a certain part of $c_o(x)$ is so that the verification process will allow for overlap invariance. The binarization process concatenates the real and imaginary components of the extraction to form the template. The template is modified such that an imaginary element a+bi that appears at (x, y) in $c_o(x)$ will become a real valued element by being placed at position (x+64, y). The binarization process converts the complex-valued 64x64 array into a real-valued 128x64 array. Binarization is completed by converting each element d in the array into a 1 or 0, effectively forming a 128x64 binarized enrollment template. The lookup table is created from the binarized template and k_0. Depending on what the value of the n^{th} bit of k_0 is, L locations of that same bit value are chosen from the binarized template and stored as the n^{th} column in a lookup table. There are 128 columns in the lookup table, with each location representing one key bit [5].

Stage E-3 uses standard encryption and hashing algorithms to derive an identification code id_0. To do this, k_0 is used as an encryption key for S bits from the $H_{stored}(u)$ with the Triple-DES algorithm. A one-way hash function, SHA-1, is applied on the encrypted text to give id_0. The lookup table, id_0, and $H_{stored}(u)$ are stored on the computer's hard drive.

The goal of the verification stage is to retrieve the N-bit key specific to a user and is shown in Figure 7.

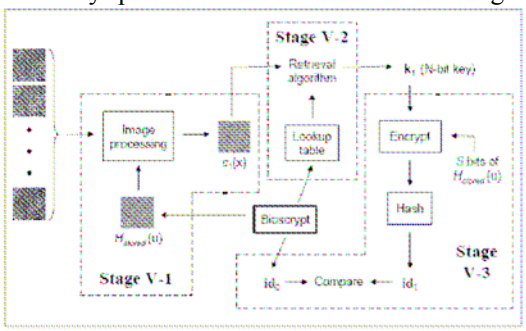

Figure 7. Verification of the fingerprint [5]

Stage V-1 is shown in Figure 8 and corresponds to the input of a fingerprint into the system.

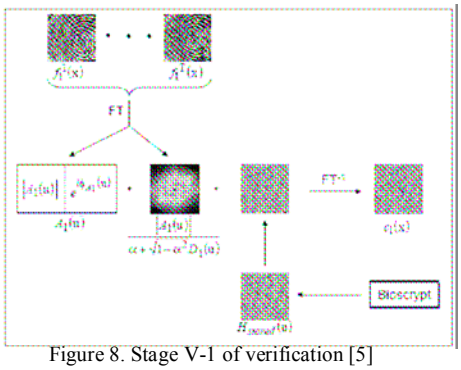

Figure 8. Stage V-1 of verification [5]

The input to this stage is the fingerprint. A series of Fourier transforms on different parts of the fingerprint are used in conjunction with $H_{stored}(u)$ to give

$$c_1(x) = FT^{-1}\{A_1(u) \bullet |H_1(u)| \bullet H_{stored}(u)\}$$

This value is fed into the second stage of verification, which retrieves the key as shown in Figure 9 [5].

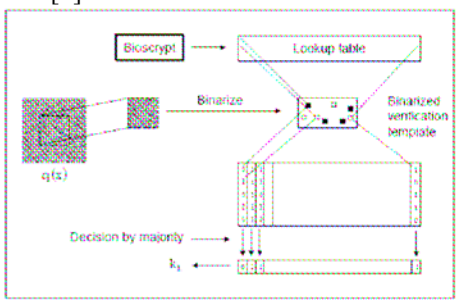

Figure 9. Stage V-2 of verification [5]

The same procedure as for stage E-2 applies with respect to the retrieval of the center 64x64 part of the print and the conversion to a real-valued 128x64 array. The lookup table is used to extract bits and sum L of them in the nth column of the table corresponding to the nth bit of k_1. Stage V-3 is invoked now, which calculates the identification code id_1 and compares it with the stored id_0. If they match then the user is authenticated. If they don't match, then stage V-2 is invoked again with a pixel offset by 1 from the center. Only if 16 pixel offsets had to be executed, will there be a message stating that the verification failed [5].

C. Tests Done to Change the Algorithm Output

Our group looked at the files generated by the BioPod software on the account that was set up. One specific .dll file called `fngrdll.dll` was identified as that holding the algorithm. It contained a certain pattern in the bits, but it was unfortunately unreadable and we were unable to change the algorithm.

IV. STORING LAYER

A. Algorithms used by the APC BioPod for Storing

XTEA

XTEA is a block cipher. It is a 64 bit block Feistel network with a 128 bit key. Feistel networks are product ciphers and consist of the following basic operations(repeated):

- Bit-shuffling (P-boxes)
- Simple non-linear functions (S Box)
- Linear mixing using XOR

The basic algorithm is as follows:

- Split the plaintext block into two equal pieces, (L_0, R_0)
- For each round $i = 1, 2, \ldots, n$, compute
  $L_i = R_{i-1}$
  $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
  where f is the round function and K_i is the sub-key. Then the ciphertext is (L_n, R_n).
- Regardless of the function f, decryption is accomplished via
  $R_{i-1} = L_i$
  $L_{i-1} = R_i \oplus f(L_i, K_i)$

A sample of XTEA is shown in Figure 10 [7].

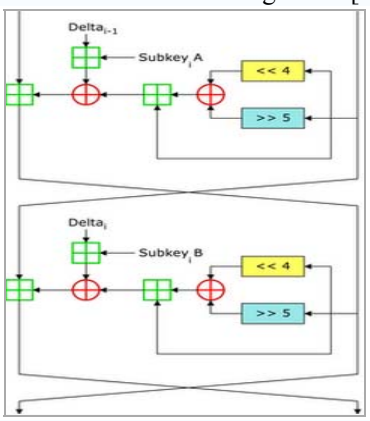

Figure 10: Two Fiestel Rounds ("one" cycle" of XTEA) [7]

RSA

RSA is the public key encryption method that relies on the difficulty of determining the number of numbers relatively prime to a large number(n). Plaintext is encrypted in blocks m (m < n).

Encipher $c = m^e \bmod n$

Decipher $m = c^d \bmod n$

{e,n} is the public key and {d,n} is the private key. It is infeasible to determine d given e and n [8].

DPAPI

DPAPI is the API provided by the Windows OS. It uses the password of the user account in association with the code that calls the DPAPI functions in order to derive the encryption key [9]. Behind the scenes,

DPAPI uses the triple DES algorithms and CryptoAPI, which are defined next.

CryptoAPI

This is Microsoft's API using for encrypting data. The APC BioPod uses the CryptoAPI to encrypt data using the RSA key [10].

Fingerprint template

Minutiae are the ends and the branches of the ridges on the finger, stored as a mathematical template called the fingerprint template. The template is irreversible i.e. the fingerprint image cannot be reconstructed from the template [11].

B. How the Storing Process Works

The fingerprint template is generated from the input from the scanner using the finger minutiae and algorithms mentioned before. The template is encrypted with XTEA using a 128 bit key. This is further encrypted using DPAPI which uses the user account information for encryption. In this way, the user account information is associated with the template and these are signed using an RSA private key and the CryptoAPI libraries [12].

C. Security of these Algorithms

The best attack reported on XTEA is a related-key differential attack on 26 out of 64 rounds of XTEA, requiring $2^{20.5}$ chosen plaintexts and a time complexity of $2^{115.15}$ [13]. RSA is based on the problem of factoring very large numbers. RSA keys are generally 1024-2048 bits and might be breakable in the future with quantum computers. RSA might be broken, with difficulty, through timing attacks or Fermat Factorization [14]. The fingerprint template is irreversible.

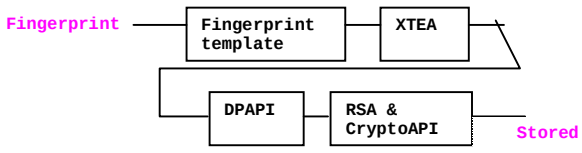

Figure 11. Steps in storing a fingerprint

As shown in Figure 11, the fingerprint goes through a number of powerful security steps before getting stored. Yet we have been successful in breaking the security of the APC BioPod within one month of purchasing it.

D. Tests done on Storage on the APC BioPod

Accessibility - Storage on disk

After going through the encryption algorithms mentioned above, the fingerprint template ends up stored on the hard drive of the computer. In our case the fingerprint template was stored in `c:\Program Files\softex` under the file `sfinger.dat`. This file was easily accessible and was not write protected. We could open the file and easily modify it. This fingerprint was registered in a user account with "limited privileges" and by default was accessible across other limited and administrative accounts.

Linearity – Changes in file followed a linear pattern

When finger1 was registered in the system the `sfinger.dat` file had four encrypted lines. It seemed impossible to break the encryption. But when finger2 was registered in the system, another 4-5 encrypted lines were added in the `sfinger.dat`. Clearly the added lines must have belonged to the newly registered finger2. We compared this file and the orginal one (with only one registered finger) and marked the additional different lines that had been added. To make things simpler, we noticed that these additional lines were all added at the end of the file. We removed these additional lines and the fingerprinting device did not authenticate finger2 anymore but still authenicated finger1. After putting back those four lines, once again the BioPod authenticated finger2. With similar methods, additional fingers could be added and deleted without going through the formal registration.

D. Proposed Methods to Improve the Security of the Device with respect to Storing

The key security vulnerabilities exist because the APC BioPod seems to concentrate on encryption. The main goal of the BioPod is to prevent reversibility or ways to get back the fingerprint from the template. However, most hackers have a sole aim to get access to the device and bypass the fingerprint results. They are not interested in getting the original fingerprint from the template.

We outline below so rudimentary changes that can prevent such vulnerabilities.

Access Control – Deals with accessibility problems

The file `sfinger.dat` should be write protected and should be accessible only to the APC BioPod processes. There should be an access control list associated with the file that contains the identity of the APC BioPod process.

The other alterative to this problem is that the fingerprint template can be stored on the BioPod itself. However, this brings in vulnerabilities of the "result" traveling over wire to the computer.

Currently, the comparison and the result are both computed on the computer.

Data Integrity and Audit – Deals with linearity problems

A checksum for the complete file with all the registered fingerprints must be computed using mechanisms like HMAC, which can utilize the blocks and keys used by the secure XTEA algorithm. From time to time, the real process that has access to all the keys can decrypt the entire file (audit check), compute its checksum and compare it with the marked checksum. If it is different, then security breach alarms must be launched and the system should go into recovery mode.

Non-Linear Addition – Deals with linearity problems

When an additional fingerprint template is created, it must not be an independent commodity that can be added and removed in chunks as we showed. Rather, the new fingerprint that is registered must be combined with all the previous fingerprint templates. The combined fingerprint templates of all fingerprints must go through substitution and permutation before being encrypted by XTEA and RSA. This would prevent hackers from detecting that the last few lines in the `sfinger.dat` file belong to the new finger and prevent them from removing or adding such lines of encrypted code linearly and changing the registered fingerprints.

IV. SECURE DESIGN PRINCIPLES

The APC BioPod violates various principles with its design.

It violates the Principle of Least Privilege because it does not provide restrictions on the read/write access for crucial files that the device uses to authenticate users. Our group was able to modify those files without problem and to completely delete any and all fingerprints in the database.

It violates the Principle of Separation of Privilege because it does not have separate mechanisms to handle separate parts of the authentication procedure.

It violates the Principle of Complete Mediation because device requires a user to provide his fingerprint only once for an encrypted folder. Every time he wants to access it after the first time, the device uses caching to authenticate the user instead of asking for the fingerprint again.

However, the device follows the Principle of Psychological Acceptability quite closely as it was very easy to install and use.

The Principle of Open Design is also followed by this device because the device does not attempt to hide the algorithms it uses to convert the fingerprint into data. Rather, it relies on the fact that the fingerprint itself is kept secure and confidential.

V. CONCLUSION

Present day research has focused on breaking optical and capacitive scanners. RF scanners such as the APC BioPod are advertised to be virtually unbreakable, yet our group was able to break it at the scanning stage. Also, most research today concentrates on breaking encryption and not other security layers such as accessibility. We focused, analyzed, and broke these other security layers, which is not frequently done. The BioPod's vulnerabilities lie at its design. We were able to identify several breaches of the principles of secure design by attacking the system at its various levels. Thus, the origin integrity policy was not satisfied by this device. The only layer that remained unbreakable was the processing layer because the device used powerful algorithm design to achieve this.

REFERENCES

[1] "Technology – AuthenTec – Biometric Fingerprint Sensor", http://www.authentec.com/technology.cfm

[2] "White Paper – AuthenTec – Biometric Fingerprint Sensor", http://www.authentec.com/getpage.cfm?sectionID=43

[3] T. Matsumoto, H. Matsumoto, K. Yamada, S. Hoshino, "Impact of Artificial "Gummy" Fingers on Fingerprint Systems", Proc. SPIE Int. Soc. Opt. Eng. 4677, pp. 275 - 289, 2002.

[4] Pablo Hennings et al. "Wavelet packet correlation methods in biometrics", Applied Optics, Volume 44, Issue 5, 637-646, February 2005

[5] Randall K. Nichols, ICSA Guide to Cryptography, Chapter 22, McGraw-Hill (1999)

[7] " XTEA"http://en.wikipedia.org/wiki/XTEA

[8] EECE 412 Notes, UBC, Dr. Konstantin Beznosov

[9] "Windows Data Protection", http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsecure/html/windataprotection-dpapi.asp

[10] "CryptoAPI", http://en.wikipedia.org/wiki/CryptoAPI

[11] "Fingerprint Biometrics" , http://www.identix.com/trends/faqs/minutia.html

[12] "Michael Swanson's Blog: Microsoft Fingerprint Reader" ,http://blogs.msdn.com/mswanson/archive/2004/12/03/274170.aspx

[13] Youngdai Ko et al. "Related key differential attacks on 26 rounds of XTEA and full rounds of GOST", In Proceedings of FSE '04, Lecture Notes in Computer Science, 2004. Springer-Verlag.

[14] "RSA", http://en.wikipedia.org/wiki/RSA