On biometric key generation from handwritten signatures

Dirk Scheuermann, Bastian Wolfgruber, Olaf Henniger

{dirk.scheuermann | olaf.henniger}@sit.fraunhofer.de
Bastian_Wolfgruber@gmx.de

Abstract: This paper investigates the extraction of a reproducible bit string referred to as biometric key from biometric data. This is difficult due to the natural variability of biometric data. If the biometric-key generation process were sufficiently resistant against attacks, a biometric key may be used e.g. as basis for the generation of application-specific user passwords. Handwritten signatures or handwritten signs have been chosen because of their high level of user acceptance and because the exchange of compromised keys is possible by using other phrases to write. The experimental results show an insufficient reproducibility of the biometric keys if no auxiliary data is used for smoothing out the natural variability of the presented data.

1 Introduction

Many existing security solutions are based on knowledge-based user authentication using PINs (Personal Identification Numbers) or passwords. Many users are overwhelmed by the task of memorizing PINs and passwords for a growing number of applications. "Single sign-on" systems and password safes provide solutions for the administration of PINs and passwords. The PasswordSitter [SIT06], for instance, is a kind of password safe; however, it does not store individual passwords, but computes them each time anew from a master password, which is not stored either in the system. The users have to memorize strong master passwords. When they forget or reveal the master passwords, they should renew the password for every individual application as soon as possible.

Biometric methods may be an alternative to knowledge-based user authentication methods because biometric characteristics are more strongly bound to a person than PINs and passwords are and cannot easily be forgotten or passed on to other people, be it intentionally or unintentionally. Like PINs and passwords, biometric reference data must be stored securely and be protected against unauthorized use. However, many users are troubled by the risks associated with storing biometric reference data in computer systems: Once compromised, biometric reference data can only a limited number of times be replaced by new biometric reference data of the same person. Furthermore, biometric data often contain information beyond what is needed for authentication (e.g. information about body conditions and diseases), which one would like to keep private. This has been shown also for signature dynamics [HGS06].

Due to the natural variability of biometric data (data captured via a biometric sensor from the same person are never completely the same), a biometric reference cannot serve as a

direct replacement for a password or for a cryptographic key. There exist key regeneration systems (such as the fuzzy commitment scheme [JW99] or the fuzzy vault scheme [JS02]) where a key is bound with the biometric data of a user and can be regenerated only by providing matching biometric data from the same user. These methods work with error-correcting codes to smooth out the differences when the encoded key is bound with slightly different biometric data belonging to the same user. However, these concepts require the secure storage of the security-sensitive key in encoded form. A solution without the need for storing any security-sensitive data on the user's side would be advantageous.

Biometric data contain highly redundant information. It may be possible to consistently extract a relatively small number of bits out of the information contained in biometric data. This paper investigates the extraction of a reproducible bit string (referred to as biometric key) from biometric data, in particular from handwritten signatures. Theoretical work on the information content of such biometric keys has already been performed [Pla09].

The remainder of this paper is organized as follows: Section 2 describes the use case considered. Section 3 lists requirements that the biometric key generation system must satisfy in order to be useful in this use case. Section 4 describes our specific approach based on handwritten signatures with all its considered aspects. Section 5 presents some experimental results obtained with the described approach. Section 6 summarizes the results and gives an outlook.

2 Use case

The goal is to generate application passwords on demand based on a biometric key and some other parameters that do not necessarily have to be kept secret. The biometric key should be generated directly from presented biometric characteristics without comparison with a stored biometric reference (see Figure 1). If necessary, some auxiliary data may be used to always reconstruct the same biometric key. The biometric key then serves as a "master password" from which application-specific passwords are derived using further data like application names and password rules.

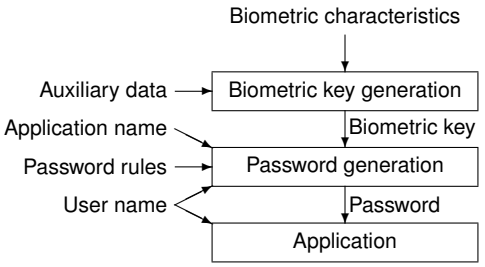

Figure 1: Block diagram

3 Requirements

In order to be useful for the protection of application passwords, a biometric key generation system should satisfy a couple of requirements. Many of these requirements are related to the aspect of security. However, there are further requirements, e.g. related to usability, that have an impact on the choice of biometric method if several biometric methods provide the same level of security.

Security requirements

- *Distinguishability of biometric keys of different persons:* The biometric keys generated from any two persons must be different.
- *Reproducibility of the biometric key of the same person:* For the same person always (or at least almost always over a long period of time) the same biometric key must be computed.
- *Resistance against "brute force" attacks:* The biometric keys must have sufficient length to withstand "brute force" attacks (where an attacker systematically tries possible values one after the other).
- *Resistance against spoofing:* It must be difficult to fake or imitate the biometric characteristics e.g. using gummy fingers or handwritten signature forgeries.
- *Revocability:* Changing the biometric key in case of compromise must be possible. For instance, in case of biometric key generation from keystroke dynamics [MRW02] and from voice [MRLW01] revocability is achieved by changing the typed or spoken password, respectively.
- *No storage of sensitive data:* The privacy of biometric data may best be protected when they are not stored at all. Therefore, no biometric data shall be stored. Auxiliary data, if stored, shall not allow to reconstruct biometric data.

Further requirements

- *Availability of the biometric sensor:* The biometric sensor used for capturing the biometric characteristics should be suitable for integration into devices used for log-in procedures, i.e. PCs, notebooks, or smart phones.
- *Short run time:* The biometric key generation algorithm must be fast because users will not accept much longer processing times than for conventional user authentication when using biometric key generation.
- *Vendor independence:* The biometric key generation algorithm should be independent of specific vendor solutions, i.e. it should be interoperable with systems offered by different vendors for capturing and extracting certain types of biometric data.

- *Scalability:* Depending on the targeted applications, different password lengths may be desired by the user or required by corresponding application policies. The length of a biometric key will be related to the amount of information contained in the biometric data. Therefore, the size of the biometric data being captured and processed should be easily scalable by the choice of the user.

4 Biometric key generation approach

4.1 Choice of handwritten signatures/signs

A major advantage of handwritten signatures over other biometric features is their high level of user acceptance. Handwritten signatures have long been accepted in many places as a means for authenticating persons. By choosing different signature lengths (e.g. first name and last name, last name only, paraph, etc.), the length of the biometric key is easily scalable. In case of compromise, an exchange of the generated biometric key is possible by using other phrases to write or changing the style of writing.

The field of application of signatures/signs may be large since capture devices cannot only be connected to PCs, but are already integrated in many devices (e.g. PDAs or tablet PCs) used for login. Furthermore, there are standardized data formats for signature/sign time series data [ISO07]. Normalized time series data may either be used directly for comparison or for deriving further feature data. Altogether, the dynamics of handwritten signatures/signs seem to be a promising starting point for generating biometric keys.

4.2 General design

As the biometric key generation would be easier to use without auxiliary data, this study takes up the challenge of generating a biometric key without the use of any auxiliary data (cf. Figure 1). Related work on biometric key generation, including that related to handwritten signatures [VSM02, TY09], makes use of auxiliary data.

In general, biometric key generation methods include the following two stages [MRLW01]:

1. *Feature extraction:* First, a set of features is extracted from presented biometric characteristics. Such features must be chosen that are sufficiently similar if extracted from the biometric characteristics of the same user and that are sufficiently different if extracted from the biometric characteristics of different users.

2. *Key generation from the extracted features:* Then, a key is generated from the features set. Keys generated from sufficiently similar features sets must be identical.

Features of handwritten on-line signatures can be classified into statistical, spatial, temporal, and spectral features. For this study, different signature features have been tried and

compared with respect to their reproducibility. The chosen approach is based on signature/sign time series data per [ISO07] containing x and y coordinates and time information (either time information per sample point or based on a uniform sampling rate). The availability of pen-up and pen-down information from the capture device is required. This may be achieved in different ways, e.g. using the optional tip-switch state channel or the pen-tip force channel or by analysis of technical events such as mouse events.

The signature is first divided into segments in a canonical way by considering different parts of the signature separated by lifting up the pen. Afterwards, each segment is divided into more abstract subsegments. These subsegments are classified into four different curve types. Different considerations how to identify curve types like “line”, “right-hand bend”, or “left-hand bend” finally lead to a specific algorithm.

4.3 Design details

4.3.1 Signature normalization

The first preprocessing step consists of normalizing the vertical size of the signature/sign and translating the signature/sign to non-negative x and y coordinates with a minimum value of 0 (with the aid of the maximum and minimum values of x and y). This means that the signature/sign is placed into a bounding box just open to the right side to allow different lengths of signatures/signs (see Figure 2).

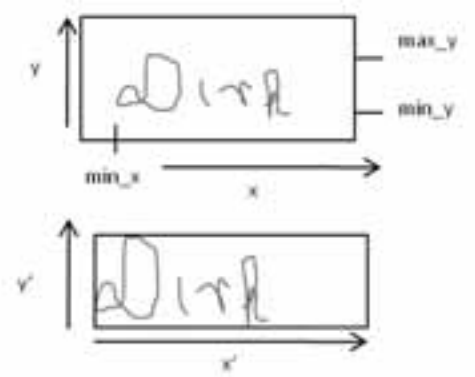

Figure 2: Normalization process applied to a signature/sign

4.3.2 Segmentation

The normalized signature/sign is divided into segments with the aid of the pen-up and pen-down information. An example of a segmented signature/sign is given in Figure 3.

Figure 3: Segmentation of a signature/sign

4.3.3 Normalization of individual segments

The segments are individually normalized in preparation for further analysis. For each segment, the starting point is moved into the origin and the whole segment is rotated such that the endpoint is located on the positive x axis.

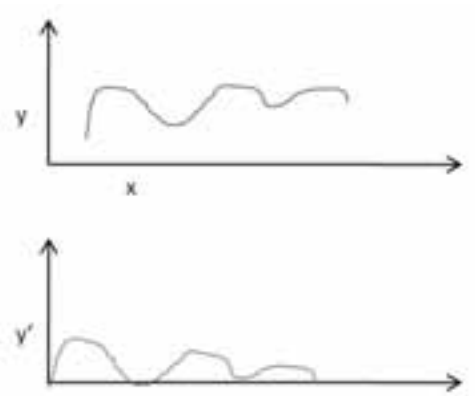

Figure 4: Normalization process applied to segments

This normalization step (illustrated in Figure 4) provides a abstract transformation not preserving the natural signature/sign image. It serves as a preparation for later consideration of the subsegments as uniquely aligned functional graphs, i.e. the y values as a function of the x values.

4.3.4 Determination of subsegments

Be N_{seg} the total number of segments and n_s the number of pixels of segment s. We denote the x and y components of the segments normalized according to Section 4.3.3 by

$$(xs_s^k, ys_s^k), s = 0, \ldots, N_{seg} - 1, k = 0, \ldots, n_s - 1.$$

This more complicated notation is used to clearly distinguish these values from the original x and y values of the captured signature. We now analyze these data for sequences of strictly monotonic increasing and monotonic decreasing sequences of x values. For this purpose, we start from the starting point (xs_s^0, ys_s^0) of the whole segment and determine

starting points (xs_s^k, ys_s^k) of new subsegments in the following way: Be (xs_s^l, ys_s^l) the starting point of the previously determined subsegment; search for index

$$k = \min\{m|(\exists(i, l < i \leq m)(xs_s^i \geq xs_s^{i-1}) \wedge (\exists(i, l < i \leq m)(xs_s^i \leq xs_s^{i-1})\}.$$

This leads to a sequence of subsegments for each segment. Each subsegment consists of its starting point and the points up to the starting point of the next subsegment.

4.3.5 Analysis and coding of subsegments

In the following, we consider a subsegment that is determined by the steps described in Section 4.3.4. The procedure for analysis is the same for all subsegments. We denote the subsegment's components by $(xu_i, yu_i), i = 0, \ldots, n_u - 1$ whereby n_u denotes the number of pixels of the subsegment. Furthermore, we introduce the identifiers $Useg_s^r$ for subsegment r of segment s and Nu_s for the number of subsegments of segment s.

The monotony property of the x values inside a subsegment guarantees that for each x value there only exists one corresponding y value, i.e. there exists a unique function f with the property $yu_i = f(xu_i) \forall i = 0, \ldots, n_u - 1$. Our goal consists in an approximation of this function with a preferably simple analytic function whose parameters may be used to classify the subsegment. Since the subsegments may be expected to be rather small curves, we use the approximation with quadratic polynomials with minimal average quadratic distance to the functional values delivered by our subsegments (least square method). This means that every subsegment gets assigned a polynomial $P_2(x) = k_1 \cdot x^2 + k_2 \cdot x + k_3$. Depending on the functional parameters, i.e. the type of the quadratic function, we classify the subsegments into four types. It would be self-evident to look for the fundamental curve types "right curve" and "left curve". However, these types only depend on the sign of the quadratic term and completely neglect the two other parameters. Therefore, we use a classification of the subsegments into more abstract curve types which also takes into account the linear term. Furthermore, a subsegment with too few pixels cannot be reasonably represented by a quadratic polynomial. Therefore, we treat subsegments with less than g pixels as an exceptional case denoted by the type "point". This leads to the following coding of the subsegments:

$$Code(Useg_s^r) = \begin{cases} 1 & \text{if } |k_1| < h \text{ (line – very small quadratic part)} \\ 2 & \text{if } k_2 < 0 \text{ and line condition not satisfied (negative linear part)} \\ 3 & \text{if } k_2 \geq 0 \text{ and line condition not satisfied (positive linear part)} \\ 4 & \text{if } n_u < g \text{ (point – exceptional case)} \end{cases}$$

This way, we have 2 bit information per subsegment available. Additional information is provided by the numbers of subsegments Nu_s for each segment $s = 0, \ldots, N_{seg} - 1$ since a given sequence of subsegment codes may be derived from different divisions of segments. The values of g and h are adjustable thresholds. For the prototype values of $g = 5$ and $h = 0.002$ have been chosen.

4.3.6 Additional dynamics features

The derivation method for biometric keys developed within the previous sections provides features of a human signature as input for biometric key generation. In particular, the segment and subsegment features are robust against certain variabilities like signature size, horizontal or vertical alignment or the exact number of sample points. However, the method only considers static signature features, and the robustness against change of pixel numbers in particular allows the reconstruction of the biometric key from a printed signature image.

To enhance the security of the method, it is beneficial to also include some dynamic features of the subsegments. For this purpose, we enrich the subsegment information by velocity information available from the captured x, y, and time values. We now return to the segment information considered in Section 4.3.2, i.e. the information from the normalized signature/sign image (now together with the corresponding time values that remained unchanged during the normalization procedure).

As a first step, we assign a speed value v_s^k to each point except the first and last points of a segment. If the software directly delivers speed values (as an optional part of [ISO07]), we can take these values and just scale the speed components by the scaling value used for normalization in Section 4.3.1. Otherwise, we calculate the speed values with the aid of the time relations to the right and left neighbor points, with duplication of previous speed value if time difference zero occurs. Therefore, it is reasonable to omit the speed values of the first and last segment point.

For the next steps, we need the following two auxiliary functions to determine indices of points inside segments and subsegments where the numbering of points and subsegments always starts from 0 for each segment:

$$\begin{aligned} Lowindex(s,r) &= \text{index of first pixel of subsegment } r \text{ inside segment } s \\ Highindex(s,r) &= \text{index of last pixel of subsegment } r \text{ inside segment } s \end{aligned}$$

Since no speed values exist for the first pixel of each first subsegment as well as for the last pixel of the last subsegment, we define the following additional auxiliary values:

$$\begin{aligned} Low(s,r) &= \begin{cases} 1 & \text{if } r = 0 \\ Lowindex(s,r) & \text{else} \end{cases} \\ High(s,r) &= \begin{cases} Highindex(s,r) - 1 & \text{if } r = Nu_s - 1 \\ Highindex(s,r) & \text{else} \end{cases} \end{aligned}$$

Next, we calculate the average speed value for each subsegment, followed by the maximum and minimum values of average speeds over the whole signature:

$$\begin{aligned} &For(s = 0, \ldots, N_{seg} - 1) \\ &\quad For(r = 0, \ldots, Nu_s - 1)\{ \end{aligned}$$

$$\overline{v_s^r} = \frac{1}{High(s,r) - Low(s,r) + 1} \cdot \sum_{k=Low(s,r)}^{High(s,r)} v_s^k$$

$$vmin = \min\{\overline{v_s^r} | s = 0, \ldots, N_{seg} - 1, r = 0, \ldots, Nu_s - 1\}$$

$$vmax = \max\{\overline{v_s^r} | s = 0, \ldots, N_{seg} - 1, r = 0, \ldots, Nu_s - 1\}$$

$$diff = vmax - vmin$$

}

According to these minimum and maximum values, the average speeds are now quantified in the following way:

$$Quant(\overline{v_s^r}) = \begin{cases} 1 & \text{if } \overline{v_s^r} < vmin + \frac{diff}{4} \\ 2 & \text{if } vmin + \frac{diff}{4} \leq \overline{v_s^r} < vmin + \frac{diff}{2} \\ 3 & \text{if } vmin + \frac{diff}{2} \leq \overline{v_s^r} < vmin + \frac{3 \cdot diff}{4} \\ 4 & \text{if } \overline{v_s^r} \geq vmin + \frac{3 \cdot diff}{4} \end{cases}$$

These quantified speed values offer an additional 2 bit code per subsegment. The subsegment codes are not equally distributed, i.e. the perfect entropy of 2 bits per subsegment is slightly disturbed. However, there is no strong mutual strong dependency between the speed codes except that a minimum and a maximum value (resulting in speed codes 1 and 4) must occur at least once. Furthermore, none of the curve types tends to be typically drawn slower or faster than the other ones, i.e. there is also no direct dependence of the speed codes on the corresponding curve codes. Hence, we now practically have 4 bit information per subsegment.

The speed codes do not depend on particular speed values of the writer but represent the writer's behavior in writing particular parts faster or slower than others. Hence, they provide robust features against variability in individual speeds.

5 Prototype and experimental results

5.1 Static features

For a first prototype, a Wacom Intuous 3 tablet with a resolution of 5080 lpi was used. The testing program was designed with the GUI framework QT (V4.7.1) that allows the operation of the Intous tablet in mouse mode. This way, the separation of pen-up and pen-down may be recognized with the aid of mouse events.

With this technology combination, the capturing of data points works asynchronously. Therefore, time values are captured (with the aid of QT standard functions) with a measuring unit of 1 ms. As mentioned in Section 4.3.6, the speed values are calculated according to the time difference between the left and right neighbor points also under consideration of exceptional cases with time differences of zero (which really occurred some times in case of fast movement).

It turns out that the algorithm works better for people with the habit to present a very exact appearance of their handwritten signature and whose signature is more similar to printed characters than to cursive writing.

For our tests, several series of live data were captured from different persons. The first data generated more than once was considered as "reference data" to determine the acceptance rate, i.e. the rate of complete matching of two password data strings. The test persons could all decide about the type of their signature (e.g. only last name, first name and last name, without full stop in between, etc.). However, they were instructed to always keep their writing habits and to present their signature along the horizontal direction of the tablet. First tests were performed with static features only (before the implementation and testing of the enlargement with dynamic features). The best results were reached by a very experienced test person frequently operating signature tablets and also involved in the initial implementation phase. This person reached an acceptance rate of 18.4% with a first test series of 30 samples which could be increased to over 50% in the third test series. For unexperienced persons, it also became obvious that the acceptance rate increases with further test series. However, the rates were much lower. The best achieved results consisted in an acceptance rate of 13% within the first test series (30 samples) and 33% within the second test series (33 samples).

5.2 Additional dynamics features

After these first results, the testing program was extended with the dynamic velocity feature whereby the static curve feature are still visible, i.e. the success for static and dynamic features can be determined separately. During the tests of this extension with further test persons, it was confirmed that the method really works with only few experienced persons since no one could reach again the same rates stated before.

At a later stage, another tablet was integrated into the test program, namely a Wacom STU-500 tablet. This device only provides a resolution of 2540 lpi, but it is better to handle for the user in particular due to the direct optical feedback on the tablet itself. The STU-500 tablet cannot be operated in mouse mode with QT. Instead, a special capturing software needed to be integrated. This software directly delivers signature templates compliant with [ISO07] containing information about x, y, time, and pressure. Explicit pen-up and pen-down information (with the aid of the switch-state channel) are not given, but the separation of segments may be discovered by points with the pressure value 0. A time channel is also not present, but a unique sampling rate of 200 samples per second is used. All test users in fact felt more comfortable with this type of tablet, and the signature images already looked more stable than with Wacom Intuous. However, no more stable results could be reached for the final biometric keys.

Furthermore, the algorithm was tested with publicly available sample data of the Signature Verification Competition (SVC) 2004 [YCX+04]. No complete matching of biometric keys – for static as well as for dynamic features – obtained from SVC files were reached. The best achieved results were recognition rates of 10% for the static features only. But

the resulting curve and speed codes contained some larger matching parts. In total, the currently implemented version of the biometric key generation algorithm only provided a few real success cases, otherwise only partially usable results. Therefore, no more reasonable quantitative results (e.g. ROC curves) may be given at the current state.

6 Conclusions

We discussed a method for deriving characteristic codes of signature/sign segments to be used as biometric keys. It becomes obvious that an acceptable reproducibility of the biometric keys may only be achieved for few persons who practised writing their signature/sign on a tablet for a long time. In addition, a clear and exact signature appearance is needed. This means that the method will work better for people writing block letters than for people with joined-up handwriting [Sch11]. This also means that signatures/signs yielding reproducible biometric keys will be easier to forge.

Considered separately, the dynamics features with velocity codes appear with similar stability and variability as the static curve codes. The method is not accurate enough to deliver biometric keys ready to be used for applications. Nevertheless, the test series with captured live data as well as SVC 2004 data always contained stable, frequently re-occurring partial number sequences. The investigations have shown the difficulty of directly extracting passwords from biometric data without the storage of helper data. Such methods may only work under certain ideal conditions and with special experienced people.

Acknowledgements

This work has been part of the Fraunhofer "Challenge" programme for projects with high risk and high attractiveness in case of success. The authors are very grateful to Ankit Singh for assistance in coding and testing the biometric key generation algorithm.

References

[HGS06] J. Hofer, C. Gruber, and B. Sick. Biometric analysis of handwriting dynamics using a script generator model. In *IEEE Montain Workshop on Adaptive and Learning Systems*, pages 36–41, Logan, UT, USA, 2006.

[ISO07] Information technology – Biometric data interchange formats – Part 7: Signature/sign time series data. International Standard ISO/IEC 19794-7, 2007.

[JS02] A. Juels and M. Sudan. A fuzzy vault scheme. In *IEEE International Symposium on Information Theory*, Lausanne, Switzerland, 2002.

[JW99] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Sixth ACM Conference on Computer and Communications Security*, Singapore, 1999.

[MRLW01] F. Monrose, M.K. Reiter, Q. Li, and S. Wetzel. Cryptographic key generation from voice. In *IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 2001.

[MRW02] F. Monrose, M.K. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. *International Journal of Information Security*, 1(2):69–83, 2002.

[Pla09] R. Plaga. Biometric keys: Suitable use cases and achievable information content. *International Journal of Information Security*, 8(6):447–454, 2009.

[Sch11] B. Schneier. Decline in Cursive Writing Leads to Increase in Forgery Risk? http://www.schneier.com/blog/archives/2011/05/decline_in_curs.html, May 2011.

[SIT06] The PasswordSitter. White paper, Fraunhofer Institute SIT, 2006.

[TY09] A.B.J. Teoh and W.K. Yip. Secure Dynamic Signature-Crypto Key Generation. In L. Wang and X. Geng, editors, *Behavioural Biometrics For Human Identification: Intelligent Applications*. IGI Global, 2009.

[VSM02] C. Vielhauer, R. Steinmetz, and A. Mayerhöfer. Biometric hash based on statistical features of online signatures. In *IEEE International Conference on Pattern Recognition*, 2002.

[YCX+04] D.-Y. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto, and G. Rigoll. SVC2004: First International Signature Verification Competition. In D. Zhang and A.K. Jain, editors, *1st International Conference on Biometric Authentication*, Hong Kong, China, 2004.