34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Biometric Identity Management for Standard Mobile Medical Networks

Alexandru Egner, Alexandru Soceanu, Florica Moldoveanu

***Abstract*—The explosion of healthcare costs over the last decade has prompted the ICT industry to respond with solutions for reducing costs while improving healthcare quality. The ISO/IEEE 11073 family of standards recently released is the first step towards interoperability of mobile medical devices used in patient environments. The standards do not, however, tackle security problems, such as identity management, or the secure exchange of medical data. This paper proposes an enhancement of the ISO/IEEE 11073-20601 protocol with an identity management system based on biometry. The paper describes a novel biometric-based authentication process, together with the biometric key generation algorithm. The proposed extension of the ISO/IEEE 11073-20601 is also presented.**

I. Introduction

A successful introduction and usage of mobile e-health systems on a large scale hinges on two key factors: interoperability and security. ISO/IEEE recently published the final version [1] of the 11073 family of standards which ensure interoperability of data transmission, monitoring and controlling of vital signs between mobile medical devices used in a Personal Area Network (PAN). These specifications do not, however, comprise any security procedures on identity management and data encryption.

As a rising number of patients are moving towards homecare, there is a growing need for creating PANs for mobile medical devices. The usage of this type of network is also contingent on security factors. The clinical data measured, transmitted and archived centrally, must be correctly assigned to the patient using the medical device and not to anyone else. Moreover, another security prerequisite needs to be the privacy of the data transferred.

The paper presents a proposal for enhancing the ISO/IEEE 11073 family of specifications through a secure, easy-to-use and easy-to-implement authentication procedure. The authentication is based on a mutual authentication technique which uses biometric information. The paper demonstrates that the proposed authentication solution is very easily embeddable into the existing ISO/IEEE 11073-20601 Optimized Exchanged Protocol (OEP) standard. The approach also ensures backward compatibility with standard versions already implemented.

A. Egner is with the Department of Information Engineering, University of Padova, Italy (e-mail: egner@dei.unipd.it).

A. Soceanu is with the Department of Computer Science and Mathematics, Munich University of Applied Sciences, Germany (e-mail: soceanu@cs.hm.edu).

F. Moldoveanu is with the Department of Computer Science, University Politehnica of Bucharest, Romania (e-mail: florica.moldoveanu@cs.pub.ro).

II. Overview of ISO/IEEE 11073

The aim of the ISO/IEEE 11073 standard specifications is to highlight and address the aspect of interoperability between a growing number of medical devices that exchange data through wireless networks. This paper refers to the ISO/IEEE 11073-20601(OEP), a standard that defines a point-to-point communication protocol between two entities called Agent and Manager.

The Agent represents the device that collects personal health data directly from patients, i.e. a thermometer, a blood pressure monitoring device, etc. In order to define the behavior of these heterogeneous devices specific medical device specializations were defined, namely the ISO/IEEE 11073-104zz specializations. *The Manager* represents the device that collects personal health data from the Agents. The Manager is a local hosting device which can be a smartphone, a notebook, or a processing engine terminal.

The OEP standard defines the following abstract models for communicating personal health data between Agent and Manager over the Personal Area Network (PAN): the Medical Domain Information Model (MDIB), the Service Model (SM) and the Communication Model (CM). These three models describe the *data model*, the *operations* supported by the entities involved in the communication and the finite state machine describing the *communication process*, respectively.

III. Security of ISO/IEEE 11073 (OEP)

With the protocol becoming increasingly popular, the security issue cannot be ignored. The OEP specification recognizes the importance of security in the context of this protocol and includes the following notice in the final version of the standard [1]: "*This standard is not intended to assure safety, security, health, or environmental protection in all circumstances*".

An important security issue is that wireless networks are not very safe communication channels. Bluetooth, which is the most common choice for existing OEP implementations, has been proven to have important security flaws and vulnerabilities to different types of attacks, such as sniffing, denial of service, or man-in-the-middle. The taxonomy of such Bluetooth threats was defined by Dunning [2]. Another OEP security issue is the impossibility of detecting the source of the data collected from the Agent.

In our opinion, the OEP standard should be extended to include two security mechanisms: authentication and data encryption. These mechanisms should ideally be implemented as add-ons to the existing OEP. These add-ons would ensure interoperability, eliminating the need for applications built on top of the protocol that encourage the

development of proprietary implementations. The enhancement of the OEP protocol by security mechanisms does not affect the performance of the communication and does not modify the current version of the protocol, thus enabling backward compatibility with the existing implementations that are currently in use [4].

IV. Authentication Using Biometric Keys

In this context, authentication is seen as a process in which the Manager recognizes and authorizes the tuple consisting of the Agent and the patient. Authentication is carried out in two phases: 1) authentication of the Agent to the Manager; and 2) creation of a link between the patient and the Agent.

The first phase can be implemented in line with the existing specifications of the OEP protocol. Agents authenticate to Managers through *association* mechanisms defined in OEP. Communication between the two devices can only take place if an association of the Agent to the Manager is performed beforehand.

In the second phase, a link between the patient and the Agent is established with the help of the Manager. The paper refers to the link between the Agent and the patient as a *patient authentication*. A patient is considered authenticated when the Manager acknowledges his use of a specific Agent and grants him a set of usage permissions.

The proposed authentication mechanism is a *mutual challenge-response authentication*: entities prove the knowledge of a secret, which can be a password, or a pre-shared key, without sending it to the other party (Fig. 1).

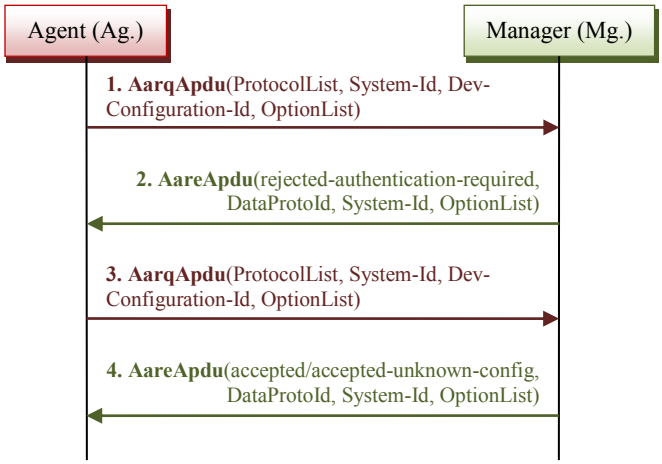

Fig. 1: The proposed extended association procedure that allows authentication

1. *AarqApdu* – request for association. This message is similar to the one in the current version of the protocol.

2. *AareApdu* – Manager's response to the association request. The parameter that holds the actual response contains a new type of response called rejected-authentication-required which denotes that the Manager requests authentication in the association phase. The challenge for the authentication procedure is stored as an attribute in the OptionList parameter.

3. *AarqApdu* – Agent's response to the Manager's challenge. The message contains the response to the Manager's challenge, as well as its own challenge for the Manager. Both pieces of information are stored in the OptionList parameter.

4. *AareApdu* – the authentication result. If the authentication fails the first parameter will be set to rejected-authentication-required and a new challenge is sent to the Agent. If the authentication is successful, the response is implicit. The message is similar to the response in the current version of the protocol except that it also contains the response to the Agent's challenge, stored in the OptionList parameter.

The proposed extension does not change the specification of the current protocol. The authentication is implemented using the types of messages already defined in the protocol, by adding new definitions to the standard nomenclature [1], such as *rejected-authentication-required*, and using the already defined optional fields that can store new information, such as challenges and responses.

Definitions:

Let Σ be an alphabet, defined as $\boldsymbol{\Sigma = \{0, 1\}}$.

Let Σ^l be the set of all strings of length l over Σ.

Let ch_{gen} be a challenge generator function, defined as $\boldsymbol{ch_{gen}: \mathbb{N} \rightarrow \Sigma^l}$, with $\boldsymbol{ch_{gen}(n) = RN_n}$, where RN_n is a random value from Σ^n.

The mutual challenge-response authentication process:

1. Mg. generates its challenge $KM = ch_{gen}(128)$;
2. Mg. sends its challenge, *KM*, to the Agent;
3. Ag. generates its challenge $KA = ch_{gen}(128)$;
4. Ag. calculates its response $RA = hash(KA \oplus KM \oplus K)$, where *K* is a biometric key;
5. Ag. sends to Mg. its *RA*, and its *KA*;
6. Mg. verifies *RA*; if correct, then it authenticates the Ag.;
7. Mg. calculates response $RM = hash(KM \oplus KA \oplus K)$;
8. Mg. sends the Ag. its response *RM*;
9. Ag. verifies *RM*; if correct, it connects to the Mg.

The authentication deploys a cryptographic nonce as a challenge. This challenge is used to ensure that each authentication sequence is unique. Even if an eavesdropper intercepts the messages exchanged between the Agent and the Manager, he cannot replicate the messages and authenticate himself using the same responses. Moreover, since the entities involved in authentication do not reveal the keys they own, it is impossible for the eavesdropper to derive the responses from the previously exchanged messages. These facts protect the protocol against reply attacks.

The mutual challenge-response protocol may be vulnerable to dictionary and brute-force attacks. However, if the response is long enough, 128 or 256 bits, the probability of guessing the response is very low. The paper suggests that the communication should be session based, meaning that the authentication allows message exchange only for defined intervals, such as 5 or 10 minutes. The length of a session

should be set by the implementer of the protocol and should differ from one device to another.

The key K used when calculating the responses in steps 4 and 7 is a biometric key. This key is not pre-shared between Agents and Managers. The key is generated firstly at the Manager, during a registration phase. At the Agent, the key is generated every time the Manager requests authentication. The Agent captures the fingerprint images from the patient and generates the biometric key that is used in the authentication mechanism. This solution eliminates the risk that an attacker intercepts the key through an unsafe communication channel.

The hash function used in the algorithm is a non-invertible function that maps the input string containing both the challenges and the biometric key to a fixed-length output. The implementer is responsible for choosing the adequate hash function. We have tested a solution based on the widely known cryptographic hash function MD5.

V. Biometric Key Generation Procedure

Generating a robust biometric key based on fingerprints is difficult because of the inherent features of the biometric data. Different scans of the same finger usually provide different images. For this reason, ensuring the repeatability of the biometric key is a significant challenge. Methods for generating unique and repeatable keys have been described in the literature [6]. However, these methods are not well suited to systems that are based on healthcare protocols such as OEP.

This section presents a novel method for generating biometric keys based on fingerprint scans. The main challenge was to find a solution for mapping fingerprints to unique biometric keys which does not depend on a classifier that has to be initially trained with a large set of fingerprint templates, such as the one which is part of the solution presented by Bhargav et al. [6, 9, 10]. A second challenge was to find a solution suitable even for devices with low processing power, such as the Agents and Managers involved in OEP.

The biometric key generation process proposed follows the steps below:

1. Obtain the byte streams in ISO-IEC-19794-2 minutiae data format for a set of different scans of the same finger;
2. Parse the byte streams and extract the minutiae points and the cores;
3. Translate the features space to matrices space;
4. Apply rigid transformations to the matrices, so that every core point is translated to a predefined coordinate;
5. Overlap the matrices;
6. Apply the DBSCAN algorithm to form clusters of minutiae points of the same type, in the same region;
7. Compute the clusters' central points;
8. Split the matrix space into areas based on distances to the core point and angles (polar coordinates);
9. Map the clusters' central points to predefined values according to the region of the matrix where they are located;
10. Apply the SVD algorithm to the matrix of mapped values in order to get the hashed key.

The proposed algorithm is based on the fact that although the minutiae points don't have exactly the same coordinates in every scan, their relative distances differ only by a small offset. Fig. 2 shows the resemblance of different image scans. The main idea is to discover the common points which appear in most of the images and use them to create the biometric key. There is a high probability that these points will be obtained from future scans which means the process is highly reliable.

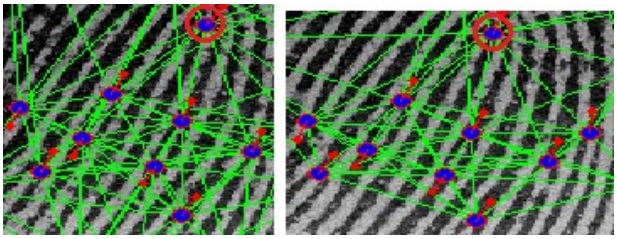

Fig. 2: Similarity between different fingerprint scans

Step 1 is performed on a set of multiple scans of the same finger in order to obtain a higher degree of repeatability of the biometric key. Generating the key based on a single scan may lead to an erroneous result. By using several biometric measurements, we ensure that only the common features are used in the process, which leads to a more robust process.

We have tested our solution with *number_of_scans* = 10 samples of the same fingerprint. The threshold can, however, be set to a different number depending on aspects such as the image quality of the fingerprint reader. Current fingerprint readers are capable of automatically obtaining several fingerprint scans in a short space of time without requiring the patient to place a finger more than once on the surface. We used a Zvetco P5000 fingerprint reader [5] with Griaule's Fingerprint SDK 2009 [7] which uses the ISO-IEC-19794-2 minutiae data format standard [8] to store the features of a fingerprint.

Step 2 is represented by the parsing of the ISO-IEC-19794-2 byte streams and the extraction of the cores and the minutiae's coordinates, angles and types.

Step 3 performs the translation from the features space to the matrices space. The image area of Zvetco P5000 is 256x360 pixels. Each pixel in the image is mapped to a matrix field, creating 256x360 matrices. Fig. 3 shows this transformation and the resulting matrix.

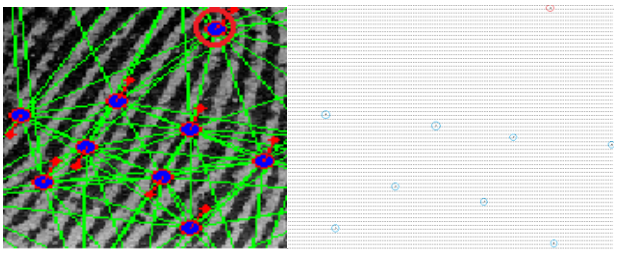

Fig. 3: Transformation between features space and matrices space

Step 4 carries several transformations applied to the matrices, enabling the core points to be translated into a specific predefined position. The coordinates we have chosen for this position are (*(xmax-xmin)*/2, *(ymax-ymin)*/2), where *xmin*, *xmax*, *ymin* and *ymax* represent the coordinates of the Regions of Interest (ROI).

In *Step 5*, the transformed matrices are overlapped. By overlapping the matrices a single matrix that contains all the minutiae points extracted in *Step 2* can be obtained. The minutiae points will tend to accumulate in the same regions of the image. This is helpful in determining which minutiae points are relevant to the key generation algorithm and what their estimated coordinates are.

Step 6 is when the minutiae points are clustered. All the minutiae points have mapped with one of the two values: 1 (*ridge_ending*) and 2 (*ridge_bifurcation*). We use these values to run the DBSCAN algorithm in order to detect clusters that have elements of the same type: 1 or 2. A validation process is performed to obtain the clusters. Only clusters that have a certain number of elements, in other words, a certain *cluster_size*, are validated. The *cluster_size* should meet the following conditions:

$$0.7 * number\_of\_scans \leq cluster\_size \leq number\_of\_scans$$

These clusters represent regions of the image where minutiae points will be detected with very high probability in future scans as well.

Step 7 calculates the points that best represent all the minutiae points of these clusters. These points are not the center of the bounding boxes of the clusters. They are calculated as the average of all the minutiae points within that cluster.

Step 8 uses an algorithm that divides the matrix area into different regions. Each region is mapped to a coded value, which means that every minutia point contained in that region is mapped to that value. We observed that the distance between the minutia point and the core is directly proportional to the deviation. Thus, the solution is designed to divide the area in such way that the minutia points that have a higher probability of large deviation belong to a bigger region. Fig. 4 describes the regions defined around the core point.

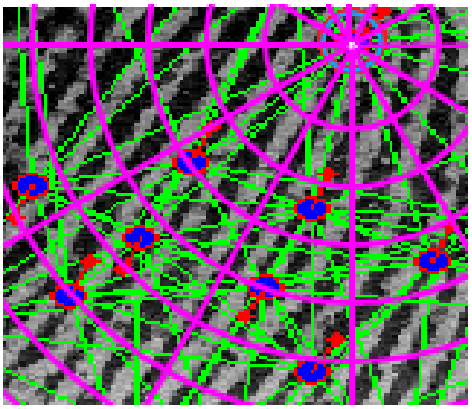

Fig. 4: Dividing the area into different sized regions

In *Step 9*, unique values are assigned to the central points of the clusters according to the region where they are located. Mapping between the regions and the codes is carried out in the preceding step.

Step 10 completes the process of generating the biometric key. Firstly, the coded values of the clusters' center points are stored in a new matrix, A. A unitary reduction to the diagonal form is performed to obtain $A = USV^t$, where S is a diagonal matrix whose values σ_i represent the singular values of A. These values are then concatenated to obtain the biometric key.

VI. Conclusion

The publication of the final versions of the various specifications of the ISO/IEEE 11073 family of standards between 2008 and 2010 elicited a huge response from hardware manufacturers and software developers [3,4]. The next step expected on the path of this rapid technical evolution is the implementation of the above components into commercially available mobile interoperable health systems.

For such health systems to succeed through to the commercial market they have to prove they can offer a very high degree of security for patients and care personnel. The security methods supported by the standard Bluetooth layer (BT 2.1 +EDR), including the Health Data Profile (HDP), are insufficient. Consequently, the application layer needs to compensate for this. Considering that the existing ISO/IEEE 11073 standards do not address the security aspects at all, the paper presented a proposal for enhancing security by introducing an identity management procedure based on biometric technology.

The non-invertible biometric key used for the mutual challenge-response Agent-Manager authentication is derived from a fingerprint measure. An innovative solution for generating the biometric key was also presented.

Selected References

[1] IEEE Engineering in Medicine and Biology Society, *Health informatics-Personal health device communication, Part 20601and Part 10408:,* N.Y, 2008, 2010.

[2] J. P. Dunning, "Taming the Blue Beast - A survey of Bluetooth-Based Threats," *Security and Privacy, IEEE,* pp. 20-27, 2010.

[3] Bluegiga, [Online]. Available: http://www.bluegiga.com/home [Accessed 1/7/2012].

[4] Andago Ingineria S.L., "Open Health Assistant Project," [Online]: http://openhealthassistant.andago.com/ [Accessed 1/9/2012].

[5] Zvetco Biometrics LLC, "P5000 Fingerprint Reader Overview," [Online]. Available: http://www.zvetcobiometrics.com/Business/Products/P5000/overview.jsp [Accessed 1/9/2012].

[6] A. Bhargav-Spantzel, A. Squicciarini, E. Bertino, X. Kong and W. Zhang, "Biometrics-based identifiers for digital identity management", in *IDTRUST '10: Proceedings of the 9th Symposium on Identity and Trust on the Internet*, pages 84-96, Gaithersburg, MD, 2010.

[7] Griaule Biometrics, "Fingerprint SDK 2009" [Online]. Available: http://www.griaulebiometrics.com/page/en-us/fingerprint_sdk. [Accessed 3/15/2012].

[8] ISO/IEC FCD 19794-2, *Biometrics – Biometric Data Interchange Formats – Part 2: Finger Minutiae Data*, 2007.

[9] W. Zhang, Y.-J. Chang, and T. Chen, "Optimal thresholding for key generation based on biometrics", in *ICIP '04: International Conference on Image Processing*, pages 3451–3454, 2004.

[10] A. Juels and M. Wattenberg, "A fuzzy vault scheme", in *Proceedings of IEEE International Symposium on Information Theory*, 2002.