ECG Dj: Heart Beat Synchronization

Lara Ortiz-Martin[1], Pedro Peris-Lopez[1], Pablo Picazo-Sanchez[2], Gerardo Schneider[2], Juan Tapiador[1]

[1]Dept. of Computer Science. Carlos III University. Leganes. Spain.
[2]Dept. of Computer Science and Engineering. Chalmers | Gothenburg University. Gothenburg. Sweden.

Introduction

Signal synchronizing of two (or more) biometric signals acquired from different sensors has been proven to be challenging. For instance, two similar sensors placed in different parts of the human body measuring the same biological signal, might have delays, data misdetection, noise in the signals or sensors issues among others (Figure 1).

However, once the synchronization is set, authentication protocols, key agreement algorithms and/or encryption schemes will benefit considerably: the same random number can be derived from different signals and thus no key distribution will be needed any more.

In particular, the 4 Least Significant Bit (LSB) of the Inter-Pulse-Interval (IPI) (Figure 2) have been proven to have some entropy degree and many works have used heart signals as key generation for cryptographic protocols [2, 3, 4, 5].

Figure 1: Issues of two heart signals measured in different body locations

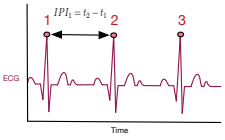

Figure 2: IPI

The current proposal aims to synchronize (at least) two different heart signals measured in distinct parts of the human body. By doing so, the same random number to be used in cryptographic protocols can be generated by different sensors. We propose an architecture where a runtime monitoring technique and a fuzzy extractor [1] scheme work together to reduce possible discrepancies between the random keys generated by two signals simultaneously sampled. As a result, the best possible automatic cryptographic key will be generated.

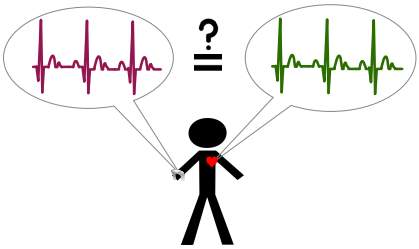

Figure 3: Signal acquisition

In [3], authors propose a statistical-based algorithm to detect where a peak should be and they insert it manually. However, authors model each missed peak using a uniform distribution and do not take into account the security properties of the derived keys [4].

References

[1] H. Kang, Y. Hori, T. Katashita, M. Hagiwara, and K. Iwamura.
Performance Analysis for PUF Data Using Fuzzy Extractor, pages 277–284. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[2] M. Rostami, A. Juels, and F. Koushanfar.
Heart-to-heart (h2h): authentication for implanted medical devices.
In *CCS '13*, CCS '13, pages 1099–1112, New York, NY, USA, 2013. ACM.

[3] R. M. Seepers, C. Strydis, P. Peris-Lopez, I. Sourdis, and C. I. D. Zeeuw.
Peak misdetection in heart-beat-based security: Characterization and tolerance.
In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5401–5405, Aug 2014.

[4] R. M. Seepers, C. Strydis, I. Sourdis, and C. I. D. Zeeuw.
Enhancing heart-beat-based security for mhealth applications.
IEEE Journal of Biomedical and Health Informatics, 21(1):254–262, Jan 2017.

[5] I. Vasyltsov and C. Bak.
Method for seamless unlock function for mobile applications.
In *2016 38th International Conference of the IEEE EMBC*, pages 2614–2617, Aug 2016.

Our proposal: system architecture

In order to detect and correct the imperfection of the random key generation, we propose a hybrid system named ECG Dj. The system architecture can be seen in Figure 4.

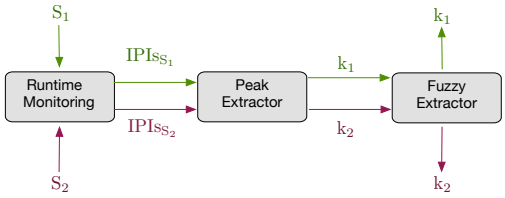

Figure 4: System architecture: ECG Dj

This system has three modules:

Runtime Monitoring. The purpose is to generate in real time two random numbers according to some desirable security (e.g., bit length and randomness quality) properties from two heart signals. This module returns two arrays $IPIs_{S_1}$ and $IPIs_{S_2}$.

Peak Extractor. This module is responsible of transforming an array of IPIs into binary numbers. Note that this method is agnostic to the way of measuring the heart signal. In this module, the cryptographic key is built by taking the 4 LSBs and applying a Grey Code to correct errors in the signals. The final outputs are two similar keys (k_1 and k_2).

Fuzzy Extractor. This last module is responsible of the extraction of a stable signal from the noisy data (k_1, k_2). After the fuzzy extractor is run both values will have the same bits.

Once the system is executed, both keys can be used in different devices without any key agreement protocol.

Runtime Monitoring

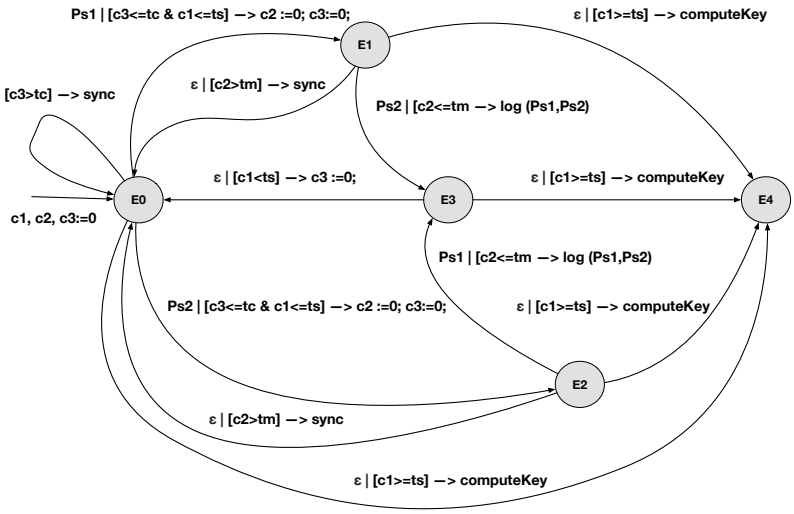

Figure 5: Formal proof of the monitor

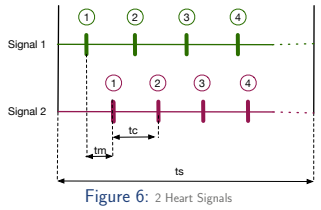

Figure 6: 2 Heart Signals

Constants	
Tc	Acceptable time between two consecutive peaks.
Tm	Acceptable diff between two matching peaks in the Signal 1 and Signal 2
Ts	Max time to sampling.
Clocks	
c1	This clock simulates the time of sampling. If c1 == ts automaton must finish.
c2	This clock measures the time between consecutive peaks of two signals.
c3	This clock measures the time between consecutive peaks of one signal.
Actions	
sync	This action controls what IPIs are saved in the output array.
computeKey	This action goes to next module and has the Array of IPIs well detected like output.
States	
E0	Initial state.
E1	When a peak of the first signal is detected.
E2	When a peak of the second signal is detected.
E3	When one peak of each signal is detected.
E4	When the max time is detected and the key is ready.

Table 1: Properties of the automaton

Acknowledgements

This work was supported by the MINECO grant TIN2013-46469-R (SPINY: Security and Privacy in the Internet of You); by the CAM grant S2013/ICE-3095 (CIBERDINE: Cybersecurity, Data, and Risks), and by the MINECO grant TIN2016-79095-C2-2-R (SMOG-DEV - Security mechanisms for fog computing: advanced security for devices).

This research has been supported by: the Swedish funding agency SSF under the grant Data Driven Secure Business Intelligence, the Swedish Research Council (Vetenskapsrådet) under grant Nr. 2015-04154 (PolUser: Rich User-Controlled Privacy Policies), and the European ICT COST Action IC1402 (Runtime Verification beyond Monitoring (ARVI)).

laortiz@inf.uc3m.es pperis@inf.uc3m.es jestevez@inf.uc3m.es

July 4, 2017

pablop@chalmers.se gersch@chalmers.se