# General Data Format Security Extensions for Biomedical Signals

S. Daukantas[1], V. Marozas[1], G. Drosatos[2], E. Kaldoudi[2] and A. Lukosevicius[1]

[1] Kaunas University of Technology/Biomedical engineering institute, Kaunas, Lithuania
[2] School of Medicine, Democritus University of Thrace, Alexandroupoli, Greece

*Abstract*— **Biosignals recorded using personal health devices and stored in General Data Format (GDF) are vulnerable when the data is transferred, processed and stored to the external servers. The aforementioned vulnerabilities influence data security and user's privacy. In this paper, we propose modifications of GDF format that enables the encryption both - personal data and biosignals. These modifications do not corrupt the intrinsic structure of the GDF format and allow to encrypt independently the header with personal data and the section of biosignals. The proposed modifications were implemented, embedded and tested in a personal health device – multiparametric scale. The header data and biosignals are encrypted directly in the scale, and saved in the micro-SD card using our modified GDF format. Finally, we present the required resources needed for encryption process.**

*Keywords*— **Biomedical signals, General Data Format, data security and privacy.**

## I. Introduction

We are witnessing the era of personal and wearable health devices, such as smart wristwatches, physical activity meters/pedometers, weight and body composition scales, biopatch monitors, etc. Most of these devices are able to gather and send biosignals together with personal data to cloud services for storage and processing. The data can be potentially intercepted during the transmission and storage. Therefore, security and privacy of such devices is a sensitive issue, restricting the wide use of smart technologies for personal and professional use. Recent vulnerability analysis of wearable devices have shown that 70% of wristwatch firmware data was transmitted without encryption, only 50% of tested devices offered the ability to implement a screen lock [1]. Personal health devices, including wireless body area networks and personal e-Health systems provide benefit to people but there are many security and privacy issues that must be solved [2, 3].

Manufacturers usually provide apps for mobile phones to connect the devices and transfer the collected data to cloud services. Most users' privacy is considered "safe" by agreeing to terms of usage between service provider and the user. On the change of the agreement, e.g. sharing data to third parties, user must accept or stop using services.

When a new sensor is developed, there is a possibility to create a new proprietary file format for storing the data, or to choose from well-documented existing file formats. Since usually there is a need to store multimodal signals with multiple sampling rates, the list of available data formats considerably shrinks. An overview of data formats for biomedical signals [4] shows that the best candidate is General Data Format (GDF), described as a superset of best features from other file formats used to store biomedical signal data. The BioSig project provides toolkits for MATLAB or Octave and open source libraries (in C/C++) [5].

Almost none of biomedical signal file formats support encryption of personal user data. Biosignals (e.g., ECG, EEG) are not considered as a sensitive information when recorded at the hospital, however, when a user monitors health status at home or the data have to be shared, this can become a privacy issue. The OpenXDF [6] file format can use only 128 bit Two-Fish [7] encryption in the CFB [8] block operation mode. However, a limitation of the OpenXDF's proposed encryption method is that the data encryption corrupts the OpenXDF file specification, and thus, the reading tools that support this format cannot acquire information and recognize the file as unreadable.

This paper introduces the GDF format modification, further referred to as the *GDFsec*, which enables an encryption of both the personal data and the multimodal biosignals, recorded using personal health devices. The proposed encryption method does not corrupt the structure of the GDF specification structure.  We show the practical application of this modification to encrypt the data, recorded using mutiparametric scales.

## II. Method

*Implementation of the modified GDF.* The GDF file format consists of five segments: three headers - fixed, variable and optional, the data section for signals, and the event table (see Fig. 1). *Hdr1* defines main information about the file and the user, e.g., user identification, gender, impairments, ICD code of disease, abuses, height, weight, location and date of recording and other. Accordingly, this segment is the most sensitive with respect to privacy issues. *Hdr2* defines signals (signal names, types, durations, etc.) and its attributes – physical and digital units. *Hdr3* is a free header, which can be used for writing additional data in Tag-Length-Value (TLV) structure format starting from the v2.1 GDF specification.

H. Eskola et al. (eds.), *EMBEC & NBC 2017*,
IFMBE Proceedings 65,
DOI: 10.1007/978-981-10-5122-7_183

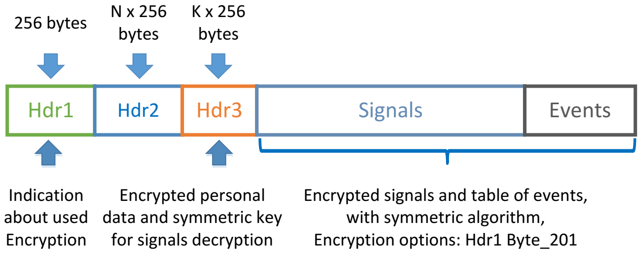
256 bytes
N x 256 bytes
K x 256 bytes
Hdr1
Hdr2
Hdr3
Signals
Events
Indication about used Encryption
Encrypted personal data and symmetric key for signals decryption
Encrypted signals and table of events, with symmetric algorithm, Encryption options: Hdr1 Byte_201

Fig. 1 The structure of the GDF file and changes for added security. Where N is the number of signals and K∈{0, 1, …} is integer value defining *Hdr3* size, enough to fill any required information.

To insure security and privacy protection, we propose the following modification of the GDF format: to encrypt a copy of *Hdr1* containing sensitive user data and to write it into a free header *Hdr3* with specific TAG = 254. Then, a sensitive information from *Hdr1* is deleted before writing to a storage medium. This approach allows restoring original *Hdr1* information if the decryption key is known.

Table 1 Byte encoding structure for encryption method, mode of operation and key size.

| Bits | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| *Encryption key bit length* | | | | | | | | |
| - | | | | | | 0 | 0 | 0 |
| 128 | | | | | | 0 | 0 | 1 |
| 256 | | | | | | 0 | 1 | 0 |
| 2048 | | | | | | 0 | 1 | 1 |
| 4096 | | | | | | 1 | 0 | 0 |
| *Encryption Algorithm used* | | | | | | | | |
| - | | | 0 | 0 | 0 | | | |
| AES | | | 0 | 0 | 1 | | | |
| Blow-Fish | | | 0 | 1 | 0 | | | |
| Two-Fish | | | 0 | 1 | 1 | | | |
| RSA | | | 1 | 0 | 0 | | | |
| *Block cipher mode of operation* | | | | | | | | |
| - | 0 | 0 | | | | | | |
| CFB | 0 | 1 | | | | | | |
| OFB | 1 | 0 | | | | | | |
| CTR | 1 | 1 | | | | | | |

In case when not only user's personal data but also the recorded biosignals are considered as a sensitive information, one can use a signal encryption. To detect whether the GDF file has applied an encryption, two reserved bytes are used. In *Hdr1*, the byte 200 defines encryption used for a copy of *Hdr1* containing personal data, encrypted using RSA 2048 or 4096 bits. In *Hdr1*, the byte 201, if non zero, defines encryption used for signals. Bit encoding structure is the same for both bytes (Table 1).

When RSA 2048 bits encryption is used, *Hdr1* must be broken into a two equal parts of 128 bytes and only then encrypted, because effective message length is 245 bytes with the "v1.5 padding" technique used. There is no such restriction with the RSA 4096 bits encryption (input message length up to 501 bytes). *Hdr1* encryption time in an embedded system is not an issue, as this is done only once, before recording process starts.

Biosignal segment of the GDF file can be encrypted on demand, by using symmetric block cipher, e.g. AES, Blow-Fish or Two-fish. Block cipher mode of operation is selected from CFB, OFB and CTR, because, these modes do not require any special measures to handle messages whose length are not in multiples of block size [8]. Thus, the last partial block does not need padding and ciphertext can be truncated to the input data size, resulting in the same size of the final file. Initialization Vector (IV), passphrase (Key) and hash string of original signals (SHA256) are generated in recording device. IV is generated from biosignal, e.g. from ECG as a biometric key [9]. Deterministic random bit generator (DRBG) is used to generate initial seed and result is XOR'ed with sample set of biosignal data [9, 10], thus resulting in a unique safe key used for encrypting data. These initial stream cipher values are combined in structure (size 192 bytes) as presented in Table 2. This structure is generated for every record, variable sizes can hold up to 512 bit values.

Table 2 Stream cipher data initialization structure.

| Name | Start Byte | Size in Bytes |
|---|---|---|
| Initialization vector | 0 | 64 |
| Encryption key | 64 | 64 |
| SHA hash of the original Data | 128 | 64 |

Entire stream cipher initialization structure is encrypted using the method described in *Hdr1* byte 200, by using a second different RSA public key for the signals. If byte 200 is empty, then RSA 2048 bit encryption is selected with associated public key for signals. Encrypted message is saved in the *Hdr3* under TAG = 253 (with a length according to used cipher).

The recording device needs to know two public keys: one for *Hdr1* encryption, second for the biosignals encryption. The same applies for GDF file decryption: one can access biosignals if knows the private key for biosignals, other private key is required to decrypt *Hdr1* information and both of private keys are required to access full information. Encryption of *Hdr1* only enables de-personalization of biosignals for research or other related purposes.

The proposed modification of GDF format was implemented and validated in the embedded system - CARRE multiparametric scale.

## III. Results

### A. *CARRE multiparametric scale*

CARRE multiparametric scale is a device developed during the European FP7-ICT project CARRE (Grant no. 611140). A commercially available body composition scales with footpad and handlebar electrodes (HBF-508/510, Omron, Kyoto, Japan) served as an enclosure; inner electronics was developed in-house. A study of using built-in electrodes to acquire biomedical signals, e.g. electrocardiogram (ECG) and impedance plethysmogram (IPG) was done using external hardware [11]. In a later work, it was shown that signal quality is sufficient to extract short-term heart rate variability and pulse arrival time (PAT) [12]. The development of the multiparametric scales is upgraded with each iteration and new types of signals are registered simultaneously (see Fig. 2).

The main processor is Cortex-M3, LPC1765 (NXP Semiconductors). It was chosen due to the large number of general purpose inputs/outputs and required speed. Its task is to acquire data from all on board sensors and write them to the micro-SD card as a GDF file. Additional tasks are to display information in the graphical color LCD screen with resolution of 320 x 240 pixels and keyboard input control. There is a sound codec connected by I2S and I2C communication channels for the audible guidance and information when the measurement starts and stops. Real time clock is implemented as a separate module in order to have time stamps on demand. WiFi module ESP8266 serves for data transfer tasks.

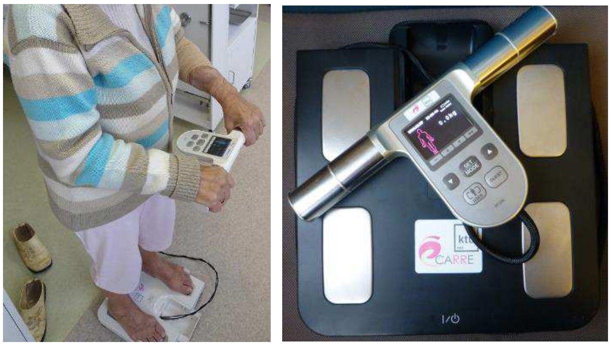

Fig. 2 CARRE multiparametric scale.

Three lead ECG signals are acquired by the biosignal frontend ADS1294R (Texas Instruments Inc.). Signals are sampled at 500Hz with 24 bits resolution. Weight and bioimpedance signals are measured by the front-end AFE4300 (Texas Instruments Inc.). External circuit, controllable by main MCU, was developed for the AFE4300 in order to increase injected current magnitude, to get bigger pulsatile impedance changes arising from blood pulsations. There is a third microcontroller - nRF52832 (Nordic Semiconductor). It combines ARM Cortex-M4f processor with multipurpose radio transceiver and can be used for communication by proprietary protocols or by Bluetooth Low Energy (BLE V4.0+) in central or device modes. Central mode allows to receive data from other BLE devices and transfer information via WiFi channel to the server (HUB function). This processor monitors battery level, sends alarm if battery level is low and the secondary function is reading optional photoplethysmogram (PPG) signals from AFE4490 sensors mounted in handlebar.

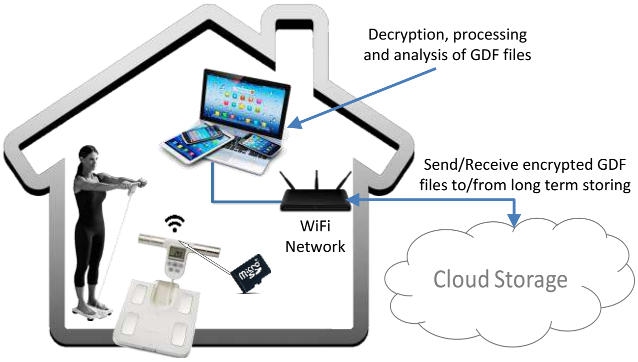
Decryption, processing and analysis of GDF files
Send/Receive encrypted GDF files to/from long term storing
WiFi Network
Cloud Storage

Fig. 3 Communication of CARRE multiparametric scale.

Signals recorded using the scale are encrypted into the GDF format by LPC1765 MCU running at 100 MHz, then stored in the micro-SD card and uploaded to a storage service in a cloud provider (see Fig. 3). Public keys and scales configuration files are stored in a micro-SD card.

### B. *Encryption analysis of GDF file*

A subset of mbed TLS crypto library [13] functions was used to implement the proposed encryption. Multiple functions required dynamic memory allocation and were converted to static declarations in order to avoid possible memory leaks and to gain stability of the embedded system. Analysis of timing and memory resources used for encryption functions are presented in Table 3.

Public key reading from the micro-SD card and decoding functions are executed in 206 ms and takes 33 KB of CODE memory. Later used functions requires fewer FLASH resources, because most of sub-functions in mbed TLS library are reused. RSA 4096 bits block encryption time is close to three seconds, resulting in 512 bytes of ciphertext.

Real biosignal values are acquired in order to generate biometric based key along with CTR-DRBG generated output.

Combined key generation should provide higher level of uniqueness than CTR-DRBG key generator alone, using dueling clocks as an entropy source [14].

Stream cipher block size is selected to be micro-SD card sector size (512 bytes) as the same amount of data is read/written to/from micro-SD card. Plaintext block encryption takes 7.36 ms only.

Table 3 Required resources for the encryption process.

| Function group | FLASH (bytes) | RAM (bytes) | Instruction count | Execution time (ms) |
|---|---|---|---|---|
| - Read & parse public key from SD card | 33112 | 1592 | 2059883 | 205.988 |
| - Encryption seed generation | 5276 | 8744 | 1790200 | 179.0 |
| - RSA block encryption (4096 bits) | 360 | 5428 | 29153010 | 2915.3 |
| - Stream cipher initialization (AES 256 bits) | 392 | 224 | 14527962 | 1452.7 |
| - Stream cipher block encrypt (512 bytes) | 132 | 38 | 73668 | 7.36 |
| - Signal block digest process (512 bytes) | 12 | 69 | 76063 | 7.6 |

The required encryption functions use almost 40KB of FLASH and 16KB of RAM. In CARRE Multiparametric Scale, main MCU has 64KB of RAM, out of which 16KB buffer is used for GDF packet saving prior write to micro-SD, resulting in 32KB of RAM free to other tasks.

## IV. Discussion & Conclusions

*Main result.* Proposed modification to the GDF specification allows to encrypt biosignals on the device and to maintain file integrity and readability by available reading tools. If a copy of *Hdr1* is encrypted and moved in *Hdr3*, standard reading tools e.g., BioSig toolkit for Matlab or SigViewer application can read GDF file as if no personal data exists in it. When signal encryption is only used, the readers can still load the personal data without any problem, but the signals will represent noise. Two encryption/decryption modalities could be used independently.

*Limitations.* Additional information increases the header size of the GDF file, but this is one time process and can be prepared before actual recording starts. Recording device needs information about the two public keys, in order to fully encrypt the GDF file, and a public key infrastructure is required for the management of them. It may be possible to use a threshold cryptography based method [15] and thus reduce the number of public keys to one.

*Conclusions.* The proposed modification of the GDF file increases the security level of the data obtained from personal health devices thus enabling more wide applications of such devices and secure sharing of data between authorized users, physicians and institutions.

## Acknowledgment

This work was supported by the FP7-ICT project CARRE (Grant No. 611140), funded in part by the European Commission.

## Conflict of Interest

The authors declare that they have no conflict of interest.

## References

1. Ching K.W. & Singh M.M. Wearable Technology Devices Security and Privacy Vulnerability Analysis. Int. Journ. Of Network Security & Its Applications, 8(3), 2016.
2. Ming Li at. al. Data Security and Privacy in Wireless Body Area Networks. IEEE Wireless Communications, 17(1), 2010.
3. Drosatos G. et al. Towards Privacy by design in Personal e-Health Systems. Proc. of 9th Int. Joint Conf. on Biomedical Engineering Systems and Technologies, vol. 5, Rome, 2016.
4. Schlögl A. An overview on data formats for biomedical signals. Springer, Berlin, Heidelberg, pp. 1557-1560, 2009.
5. The BioSig Project. http://biosig.sourceforge.net
6. OpenXDF Consortium. Open eXchange Data Format Specification. 2009.
7. Schneier B. et al. Twofish: A 128-Bit Block Cipher. 1998.
8. Dworkin M. Recommendation for Block Cipher Modes of Operation Methods and Techniques. *Natl. Inst. Stand. Technol. Spec. Publ. 800-38A 2001 ED*, no. December, p. 66, 2001.
9. Yao L. et al. A Biometric Key Establishment Protocol for Body Area Networks. International Journal of Distributed Sensor Networks, 7(1), 2011.
10. Revett K. & de Magalhães S.T. Cognitive Biometrics: Challenges for the Future (pp. 79–86). Springer, Berlin, Heidelberg, 2010.
11. Paliakaite B. et al. Estimation of pulse arrival time using impedance plethysmogram from body composition scales. In *IEEE Sensors Applications Symposium (SAS)*, pp. 1–4, 2015.
12. Paliakaitė B., Daukantas S. & Marozas V. Assessment of pulse arrival time for arterial stiffness monitoring on body composition scales. *Comput. Biol. Med.*, Apr. 2016.
13. SSL Library mbed TLS / PolarSSL. https://tls.mbed.org
14. Dunigan T. Random bits from dueling clocks. https://developer.mbed.org/users/manitou/code/rng/
15. Zhang C. et al. A new construction of threshold cryptosystems based on RSA. Information Sciences, 363, pp. 140-153, 2016.

Corresponding author:

Author: Saulius Daukantas
Institute: Kaunas University of Technology
Street: K. Baršausko str. 59 - A459
City: Kaunas
Country: Lithuania
Email: saulius.daukantas@ktu.lt