Secure Processing of Stream Cipher Encrypted Data Issued from IOT: Application to a Connected Knee Prosthesis

Maxime Pistono, Reda Bellafqira, Gouenou Coatrieux

Secure Processing of Stream Cipher Encrypted Data Issued from IOT: Application to a Connected Knee Prosthesis

Maxime Pistono[1], Reda Bellafqira[1] and Gouenou Coatrieux[1]

***Abstract*— In this paper, we propose a secure protocol that allows processing encrypted data emitted by a medical IOT device. Its originality stands on a new fast algorithm which makes possible the conversion of Combined Linear Congruential Generator (CLCG) encrypted data into data homomorphically encrypted with the Damgard-Jurik (D-J) cryptosystem. By doing so, an honest-but-curious third party, like a smartphone, can process data issued from the IOT devices (e.g. raising a health alert) without endangering data privacy while CLCG can be integrated in an IOT of low computation capabilities. Moreover, in order to reduce communication and computation complexities compared to existing solutions and to achieve a real time solution, we further propose a secure packed version of CLCG in the D-J domain. With it a medical IOT can encrypt several pieces of data at once while allowing a third party to independently convert and process them in their D-J homomorphic encrypted form. We theoretically and experimentally demonstrate the performance of our solution in the case of a connected knee prosthesis, the data of which are processed for patient monitoring.**

I. Introduction

Nowadays IOT (Internet Of Things) regroup billions of connected devices [1]. In healthcare, such devices offer the capability to better monitor patient vital signs, This one being at the hospital or at home [2]. Among them, Implantable Medical Devices (IMDs) constitute a specific IOT class. Without being exhaustive, they include pacemaker, insulin pump, neural implant and in our case connected Knee prosthesis. If most of these IMDs have a direct and automatic impact on the patient health, they also communicate different kind of measurements. Extracted data can be made available and analyzed by health professionals or automatically with the help of a smartphone, tablet or computer. Moreover and as illustrated in Fig. 1, these data are more and more collected in the context of big health data [3] where data are reused so as to develop new services such as diagnosis aid support based on machine learning techniques.

Beyond their innovative character, such IMD applications must take into account data security issues as imposed by international regulations (e.g. European General Data Protection Regulation 2016/679, US CFR 164.312 [4], [5]). In particular, data confidentiality and patient privacy are of major concerns. If IMD communications can be secured by means of encryption, an open question is how to make possible the secure processing of transmitted data without endangering their confidentiality and privacy. To the best of our knowledge, this is still an open issue strengthened by the fact IMDs are small resource-constrained devices in terms of: memory, battery and computing capability. In this work, our objective is not only to secure medical data emitted by an IMD but also to allow their secure processing by a HMI (Human Machine Interface) integrated to a mobile device (e.g. a smartphone).

Different solutions have been proposed to ensure the confidentiality of IMD data. Most recent ones, like Cloaker [6] or IMDguard [7], take advantage of a HMI external device that opens a secure communication channel with the IMD and jams communications in case of attacks. But, in this case, HMI has full access to the data sent by IMD. If an attacker takes the control of the HMI device, data privacy is obviously endangered. Basically, such a device should be considered at least as honest but curious (it will not interfere with HMI functionalities but will try to infer data).

Notice also that these approaches and others use classical encryption algorithms like DES (Data Encryption Standard) or AES (Advanced Encryption Standard). With these cryptosystems data cannot be processed or manipulated unless they are decrypted. In our framework, the idea is to let HMI process IMD data so as for example to raise an alarm when necessary to the patient but without accessing to the real IMD data values. HMI has to be able to securely process two signal operations: data filtering and tresholding (i.e. comparing a value to a threshold).

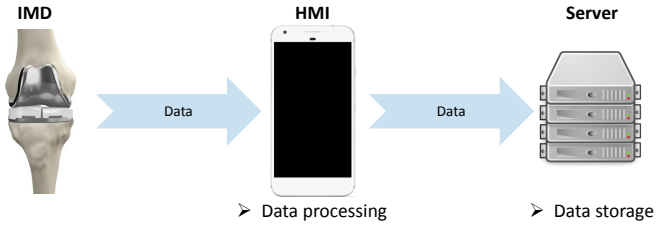

Fig. 1. Our IMD framework : a connected knee prosthesis (IMD) provides various measurements to a Human Machine Interface (HMI) integrated into a mobile device. HMI is able to process data and serves as proxy for data storage in a server.

To secure filtering operations, we propose to exploit homomorphic encryption. Such cryptosystems allow performing operations (e.g. $+$, $\times$) over encrypted data with the guarantee that the decrypted result equals the one carried out onto

This work has received a French government support granted by the National Research Agency in the Investing for the Future program under reference ANR-17-RHUS-0010 - Followknee project, as well as the support of the Joint Laboratory SePEMeD.

[1]M. Pistono, R. Bellafqira, G. Coatrieux, are with the Institut Mines-Telecom Atlantique Bretagne Pays de la Loire Atlantique; Telecom Bretagne; Unite INSERM 1101 Latim, Technopole Brest-Iroise, CS 83818, 29238 Brest Cedex 3 France. {e-mail:{maxime.pistono, reda.bellafqira, gouenou.coatrieux}@imt-atlantique.fr}.

unencrypted data. Regarding thresholding operations, several solutions exist. All correspond to different protocols that combine homomorphic encryption and secure multiparty computation [8], [9]. Whatever these solutions, they are all of high computation or communication complexities and are consequently not suited to our framework. If HMI can process homomorphically encrypted data, that is not the case of IMDs due to their material limitations. In this paper, we propose a new protocol that overcomes these issues. This one takes advantage of lightweight stream cipher CLCG (Combined Linear Congruential Generator) to encrypt data at the IMD side and of a new cryptosystem conversion protocol. With this later, CLCG encrypted data can be turned into data homomorphically encrypted with the Damgard-Jurik (D-J) cryptosystem. HMI can then securely process data while preserving patient privacy. We also introduce a CLCG data packing strategy which allows processing block of data at once. This one significantly reduces communication complexity and makes our solution run in real-time.

The rest of this paper is organized as follows. Section II regroups some preliminaries on different functions and tools on which our protocol relies. We then depict our approach along with its general framework in Section III. Some theoretical and experimental results are given in section IV.

II. Preliminaries

A. Damgard-Jurik (D-J) cryptosystem

Let $((g, K_p), K_s)$ be the public/private key pair

$$K_p = pq \quad \text{and} \quad K_s = LCM((p-1),(q-1)) \tag{1}$$

where LCM is the least common multiple function, p and q are two large prime integers. $\mathbb{Z}_{K_p^n} = \{0, 1, ..., K_p^n - 1\}$. The encryption of a plain-text $m \in \mathbb{Z}_{K_p^n}$ into the cipher-text $c \in \mathbb{Z}^*_{K_p^{n+1}}$ using the public key K_p is given by

$$c = E[m, r] = g^m r^{K_p^n} \mod K_p^{n+1} \tag{2}$$

where $r \in \mathbb{Z}^*_{K_p}$ is a random integer associated to m making the D-J cryptosystem semantically secure.
To get access to the message m from an encrypted message c, we use the function $F(\cdot)$ [10] which computes mK_s from c^{K_s} as follows

$$m = F(c^{K_s})K_s^{-1} \mod K_p^n \tag{3}$$

The D-J cryptosystem has an additive homomorphic property. Considering two plain-texts m_1 and m_2, then

$$\begin{aligned} E[m_1, r_1]E[m_2, r_2] &= E[m_1 + m_2, r_1 r_2] \\ E[m_1, r_1]^{m_2} &= E[m_1 m_2, r_1^{m_2}] \end{aligned} \tag{4}$$

In order to compare D-J encrypted data, we use a difference function $D(\cdot)$ given in [10] as

$$D(E[a, r], E[b, r]) = (a - b) \mod K_p^n \tag{5}$$

As exposed, $D(\cdot)$ gives access to the difference in between a and b modulo the D-J public key. Notice that knowing the modular difference between a and b gives no clues about the value of a and b, respectively.

B. CLCG Stream cipher encryption in clear and D-J domains

A Combined Linear Congruential Generator (CLCG) [11] is the combination of two Linear Congruential Generators (LCG) which are pseudo number generators based on congruence and linear function. The output X_{i+1} of a LCG is given by

$$X_{i+1} = a_X X_i + c_X \mod m \tag{6}$$

where X_0 is the $seed$; a_X is a multiplier; c_X is an increment; m is the modulo. The output Z_{i+1} of a CLCG is the combination of two LCG sequences [11]

$$\begin{aligned} Z_{i+1} &= X_{i+1} + Y_{i+1} \mod m \\ &= a_X X_i + c_X + a_Y Y_i + c_Y \mod m \end{aligned} \tag{7}$$

Thus the CLCG encryption c of the data d is given by

$$c = d + Z_i \tag{8}$$

The implementation of CLCG in the D-J domain, or more clearly the homomorphic encrypted version of CLCG (i.e. SCLCG for Secure CLCG) is possible if the CLCG modulo (m in eq.7) equals to the user D-J public key K_p. It is such as

$$E[Z_{i+1}, r_{Z,i+1}] = E[X_{i+1}, r_{X,i+1}]E[Y_{i+1}, r_{Y,i+1}] \tag{9}$$

where

$$\begin{aligned} E[\beta_{i+1}, r_{\beta,i+1}] &= E[\beta_i, r_{\beta,i}]^{a_\beta} E[c_\beta, r_{c_\beta}] \\ &= E[a_\beta \beta_i + c_\beta, r_{\beta,i}^{a_\beta} r_{c_\beta}] \end{aligned} \tag{10}$$

with $\beta = X$ or Y. The seed of SCLCG is such as

$$E[seed] = E[X_0, r_{X,0}], E[Y_0, r_{Y,0}], E[c_X, r_{c_X}], E[c_Y, r_{c_Y}] \tag{11}$$

Notice that the knowledge of the parameters $(a_X, a_Y), (c_X, c_Y)$ and m does not endanger the SCLCG security [12].

III. Secure Processing of IMD data

A. Connected Implantable medical device framework

Our framework is part of the Followknee project which aims at developing a connected knee prosthesis (see Fig. 1). This prosthesis collects measures from different sensor like the PH of tissues and sends them to a human machine interface (HMI). HMIs role is twofold. On one hand, it serves as proxy, sending data to a datawarehouse for future big data analysis, and in second, it processes data so as to raise an alarm in case filtered measures are above a specific threshold, asking the patient to see his or her physician.
In a first time, let us consider only one sensor acquiring M samples $\{d_i\}_{i=1,...,M}$ let us also denote $\{w_i\}_{i=1,...,M}$ the weights of the HMI filter and S the alarm threshold, send to HMI by the server. To sum up, HMIs objective is to compute

$$A = \sum_i w_i d_i - S \tag{12}$$

If A is negative (i.e $\sum_i w_i d_i > S$) the HMI emits an alert.

In our framework, prosthesis data are confidential. We also consider the prosthesis and the remote server as honest. That is not the case of HMI. Indeed, the mobile device where it is located, e.g. a smartphone, can be hacked. We thus assume HMI as honest but curious. It can try to infer information about the patient. In our framework, due to the fact the implant is of limited computation capacities, its data $\{d_i\}_{i=1,...,M}$ are sent to HMI CLCG encrypted. In order to allow HMI to process data in a homomorphically encrypted form, there is a need to convert an encrypted data from a cryptosystem to another one. This is one originality of our work. Another one stands in a packing strategy we propose in order to allow processing several pieces of data at once reducing communication complexity drastically.

B. Cryptosystem conversion (CrC)

To convert a CLCG encrypted data into a D-J encrypted data, we take advantage of the fact CLCG can be implemented in the D-J domain (see Section II-B). Indeed, it is possible to D-J cipher data already CLCG encrypted and then conduct CLCG decryption in the D-J domain in order access to D-J encrypted data only.

Let us consider the i^{th} CLCG encrypted data, i.e. $d_i + Z_i$ (see eq.8). Its conversion by HMI into its D-J encrypted form $E[d_i, r_{Z,i}^{-1}]$ is given accordingly the following steps

1) HMI computes the D-J encryption of $d_i + Z_i$ with the random value 1: $E[d_i + Z_i, 1]$.
2) HMI computes $E[Z_i, r_{Z,i}]$ using the SCLCG parameterized with $E[seed]$ and a_X, a_Y (see eq.9).
3) Finally HMI computes

$$E[d_i, r_{Z,i}^{-1}] = E[d_i + Z_i, 1] \times E[Z_i, r_{Z,i}]^{-1} \quad (13)$$

Basically, this procedures works in the D-J domain and subtracts the CLCG random value Z_i to $d_i + Z_i$.

Notice that, since HMI has no idea about the random value $r_{Z,i}$, it has no clues about d_i.

C. Processing of Stream Cipher Encrypted Data

Once HMI has accessed to $\{E[d_i, r_{Z,i}^{-1}]\}_{i=1,...,M}$, our objective is to make it securely filter these samples and to threshold the filter output so as to raise or not an alarm (see eq. 12). To reach this goal, we propose a 3-step procedure where we assume: i) HMI and Server *a priori* agree on the filtering weights $\{w_i\}_{i=1,...,M}$ and on SCLCG seed ($E[seed]$); ii) IMD and Server *a priori* agree on the CLCG parameters (see Section 2.2). We recall that the SCLCG modulo m in eq. 7 equals to the D-J public key K_p^n.

Our procedure is thus as follows

1) *IMD data encryption step:* The prosthesis CLCG progressively encrypts and sends M data samples, i.e. $\{d_i + Z_i\}_{i=1,...,M}$.
2) *Server threshold encryption step:* Server SCLCG encrypts the alert threshold $E[S, \prod_{i=1}^{M} r_{Z,i}^{-w_i}]$ and sends it to HMI.
3) *HMI processing step:*
   - Using the same previous procedure as in Section III-B, HMI derives the D-J encryption version of d_i from $d_i + Z_i$, i.e. $E[d_i, r_{Z,i}^{-1}]$.
   - Considering the filtering weights $\{w_i\}_{i=1,...,M}$, HMI calculates $E[d_i, r_{Z,i}^{-1}]^{w_i} = E[w_i d_i, r_{Z,i}^{-w_i}]$.
   - HMI evaluates $W = \prod_{i=1}^{M} E[w_i d_i, r_{Z,i}^{-w_i}] = E[\sum_{i=1}^{M} w_i d_i, \prod_{i=1}^{M} r_{Z,i}^{-w_i}]$
   - Having received a D-J encrypted version of the threshold from the server, HMI computes

$$\begin{aligned} A &= D(E[\sum_i w_i d_i, \prod_i r_{Z,i}^{-w_i}], E[S, \prod_i r_{Z,i}^{-w_i}]) \\ &= \sum_i w_i d_i - S \mod K_p^n \end{aligned} \quad (14)$$

   Finally, HMI checks if A is positive or negative and emits or not a medical alert. As stated in Section II-B knowing the modular difference between S and $w_i d_i$ give no clues on the value of S and d_i, respectively.

D. Data packing and communication complexity reduction

With the previous solution, one sample issued from the prosthesis is encoded onto 1024 bits in the case of a D-J cryptosystem public key K_p of 1024 bits (see Section II-A). To reduce the communication complexity, we propose a SCLCG packing strategy that takes into account the prosthesis provides measures from N different sensors. Our idea is to CLCG encrypts a block of sensors samples at once on the prosthesis side and to process them simultaneously in the D-J domain on the HMI side. A block D_i is then constituted of N measures $D_i = \{d_{i,j}\}_{j=1..N}$. For one given sensor p, the objective is that HMI securely computes:

$$A = \sum_{i=1}^{M} w_i d_{i,p} - S_p \quad (15)$$

The procedure we propose organizes sensors samples in a block in a specific way, which depends on their binary encoding. Let us consider that for the p^{th} sensor sample, weight and threshold values are encoded on b_d^p, b_w^p and b_t^p bits, respectively. Let us also assume that the D-J public key is of 1024 bits. Our packing method consists in allocating consecutive sequence of bits of these 1024 bits to encode the measure of a sensor. Based on this organization, all operations in the encrypted domain, i.e. additions and multiplications, for a given sensor will be conducted on its respective sequence of bits. Operation results being limited to a number of bits, one must avoid overflows. More clearly, the measure of the p^{th} sensor will be encoded on $b^p = max(b_d^p + b_w^p, b_t^p) + log_2(M)$ bits. It can be seen that this value depends on the number of additions of the filtering operations. Considering that $b^p = b, \forall p \in [\![0, N-1]\!]$, a block of data is encoded as

$$D_i = \sum_{j=0}^{N-1} 2^{j \cdot b} d_{i,j} \quad (16)$$

Notice that on its side, Server has also to pack filters thresholds, that is to say $S = \sum_{j=0}^{N-1} 2^{j \cdot b} S_j$.

At the end of our protocol according to eq.14 we obtain

$$A = \sum_i w_i D_i - S \mod K_p^n \tag{17}$$

A result we can unpack in order to obtain set of differences on which decisions can be taken

$$\{\sum_{i=1}^{M} w_i d_{i,p} - S_p\}_p \tag{18}$$

IV. Theoretical and experimental results about communication and computation complexity

To the best of our knowledge, the solution we propose based on data packing is the first that combines in a single process a filtering operation with a difference computation without having to use fully homomorphic encryption, unexploitable in real time applications.

In table I, we compare the theoretical computation complexity of our approach and the ones of the KBH method. This latter is only able to compare homomorphically encrypted data, but actually ensures the best trade-off in terms of complexity. Results from KBH cryptosystem and from our solution are given in terms of modular multiplications. As it can be seen, thanks to packing, our solution outperforms KBH, and additionally allows data filtering.

A. Experimental results

Our protocol was implemented simulating the IMD and the HMI device as virtual machines equipped of 1.3 GHz CPU with 1GB memory (equivalent to an IPhone 5). The prosthesis provides the information of 34 sensors and the HMI filter is of length 10. In experimental simulation, 125 secure HMI filtering and comparisons operations require less than 1 second in real conditions. As consequence, our protocol can be used in real applications.

B. Security analysis under the semi-honest model

In our protocol, all data are encrypted by the SCLCG or D-J cryptosystems the security analyzes of which have been shown in [10], [11], [12] and [14], respectively. The crypto-system conversion also relies on these cryptosystems, SCLCG being the implementation of CLCG into D-J cryptosystem domain. It is consequently secure. With our protocol HMI accesses to the difference value between the filtered data and the alert threshold in a clear form (see eq. 17). It has been shown in [14] that an attacker cannot infer information about the filtered data or the threshold from this difference. To sum up, our solutions prevents a semi-honest HMI from learning: the private keys of the D-J cryptosystem; the sequence generated by the SCLCG and the data emitted by the IMD as well as the thresholds.

	KBH	Paper
Computation: IMD	$2n(\frac{3}{2}\lambda + 5) + 3$	$3M$
Computation: HMI & Server	$5n(\frac{3}{2}\lambda + 5) + 2$	$n(2n + 8\lambda + 7) + 5\lambda + M(6\lambda + 9) - 2$
Data number	1	MN

TABLE I

Comparison between KBH and our optimized packed protocol in terms of computation complexity for one execution considering a given security parameter λ [13]. n is a D-J cryptosystem parameter see Section II-A

V. Conclusion

In this paper, we have proposed a protocol which allows a third party to process stream ciphered data. Its originality stands on a cryptosystem conversion (CrC) and a data packing strategy. Our CrC procedure converts CLCG encrypted data into homomorphic encrypted data, and is suitable to most secure processing based on partially homomorphic encryption. It further allows jointly filtering-thresholding data. Experimental results show that our solution is practical in real application contrarily to the state of the art based on fully homomorphic cryptosystems. Thus any IOT having enough capacity to implement a CLCG cryptosystem can take advantage of our solution.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645 – 1660, 2013.

[2] R. Zgheib, E. Conchon, and R. Bastide, "Engineering iot healthcare applications: towards a semantic data driven sustainable architecture," in *eHealth 360*. Springer, 2017, pp. 407–418.

[3] G. Bouzillé and al., "Sharing health big data for research - A design by use cases: the INSHARE platform approach," in *The 16th World Congress on Medical and Health Informatics (MedInfo2017)* , 2017.

[4] C. for Medicare, M. Services *et al.*, "Security standards: Technical safeguards," *HIPAA Secur. Ser*, vol. 2, pp. 1–17, 2007.

[5] C. Camara, P. Peris-Lopez, and J. E. Tapiador, "Security and privacy issues in implantable medical devices: A comprehensive survey," *Journal of biomedical informatics*, vol. 55, pp. 272–289, 2015.

[6] T. Denning, K. Fu, and T. Kohno, "Absence makes the heart grow fonder: New directions for implantable medical device security," in *Proceedings of the 3rd Conference on Hot Topics in Security*, 2008, pp. 5:1–5:7.

[7] F. Xu, Z. Qin, C. C. Tan, B. Wang, and Q. Li, "Imdguard: Securing implantable medical devices with the external wearable guardian," in *2011 Proceedings IEEE INFOCOM*, April 2011, pp. 1862–1870.

[8] I. Damgård, M. Geisler, and M. Krøigaard, "Efficient and secure comparison for on-line auctions," in *Information Security and Privacy*, J. Pieprzyk, H. Ghodosi, and E. Dawson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 416–430.

[9] F. Kerschbaum, D. Biswas, and S. d. Hoogh, "Performance comparison of secure comparison protocols," in *2009 20th International Workshop on Database and Expert Systems Application*, Aug 2009, pp. 133–136.

[10] R. Bellafqira, G. Coatrieux, D. Bouslimi, G. Quellec, and M. Cozic, "Proxy re-encryption based on homomorphic encryption," in *Proceedings of the 33rd Annual Computer Security Applications Conference*. ACM, 2017, pp. 154–161.

[11] B. A. Wichmann and I. D. Hill, "Algorithm as 183: An efficient and portable pseudo-random number generator," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 31, no. 2, pp. 188–190, 1982.

[12] P. L'Ecuyer, "Tables of linear congruential generators of different sizes and good lattice structure," *Math. Comput.*, vol. 68, no. 225, pp. 249–260, Jan. 1999.

[13] I. Damgård, M. Keller, E. Larraia, V. Pastro, P. Scholl, and N. P. Smart, "Practical covertly secure mpc for dishonest majority – or: Breaking the spdz limits," in *Computer Security – ESORICS 2013*, 2013, pp. 1–18.

[14] I. Damgård and M. Jurik, "A length-flexible threshold cryptosystem with applications," in *Information Security and Privacy*, R. Safavi-Naini and J. Seberry, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 350–364.