

Chaos Based Key Management Architecture for Wireless Sensor Networks

Rui Miguel Soares Silva

School of Technology and Management of the Polytechnic Institute of Beja
Beja, Portugal
rs@estig.ipbeja.pt

Nuno Sidónio Andrade Pereira

School of Technology and Management of the Polytechnic Institute of Beja
Beja, Portugal
nsap@estig.ipbeja.pt

Mário Serafim Nunes

Technical Superior Institute of the Technical University of Lisbon / INESC
Lisbon, Portugal
mario.nunes@inov.pt

Abstract - In this paper we present a new proposal for key management in Homogeneous Wireless Sensor Networks (HoWSN) that achieves the One Time Pad (OTP) property. The concept of OTP is well known in the security community as the property that allows the encryption with a different and unrelated key for each message that is sent to the communication channel. A new architecture of a cryptographic system is proposed, based on symmetrical encryption and using a chaotic system as the heart of its entropy. The use of symmetrical encryption is due to the well known constraints of the targeted HoWSNs, which however could well support the low resource consumption of the chaotic systems equations. The paper briefly presents the known proposals for key management in HoWSNs, then presents the proposed system, following makes a global evaluation of the proposed system and of the state of the art solutions, and finally presents some conclusions and future work.

Keywords - Key Management; Wireless Sensor Networks; Chaos Cryptography

I. INTRODUCTION

The use of chaotic systems in cryptography is not new, several proposals were done in the past like [1] or [2] and others several cryptanalysis like [4] or [5] were also published. Other previous works could also be referred, and the respective cryptanalysis answers. Nevertheless in [5] a new research effort to the cryptology community on the subject is proposed.

In Wireless Sensor Networks (WSN) there is yet a great field of unsolved issues concerning its security as stated for example in [6]. Due to its low profiles of processing, storage and energy consumption, the small dimension and the common exposure in public places, security issues on WSN are yet a matter of research all around the world. In [7] a classification of the WSN is presented in several aspects, one of them being

Homogeneous (HoWSN) and Heterogeneous (HeWSN). In this paper we will focus on the HoWSN, which are characterized by the fact that all the network nodes have the same properties.

The HoWSN have to be Self-Organized, what means that the nodes need to have some way to establish secure communication links between them from scratch in the bootstrapping process and new nodes must also have the capability of joining securely to the network at any moment. This aspect leads to one of the main security issues in WSN and in particularly in HoWSN, the Key Management.

Several schemes exist and have been published in order to address the best key management scheme that fits to HoWSN concerning its constrained properties. In this paper we propose a new scheme to address the key management process in HoWSN using the properties of the chaotic systems. A main characteristic of this new proposal is the offering of the One Time Pad (OTP) property to HoWSN communications, meaning therefore that each message in the communication channel will be encrypted with a new and different key unrelated with the corresponding encrypted message. Nevertheless the proposed system be independent of the chaotic system used, in this paper we present a targeted solution using the well known Lorenz chaotic system and the Advanced Encryption System (AES) specified in the IEEE 802.15.4 [8] / Zigbee [9] architecture. In this case, through the use of the Lorenz chaotic system we get a system that could generate more than $3.4e38$ different points of space and consequently the same number of keys to encrypt messages in the system. The possibility of use $3.4e38$ different keys leads us to a real OTP scenario of communications as will be proved in section IV. Due to the proposed architecture all the known attacks to chaotic systems, like [3] or [4] for instance, have no application in this system.

The rest of the paper is organized as follows: in section II we briefly present the related work on key management systems to HoWSN; in section III we present the proposed system; in section IV we evaluate the proposed system and the other symmetrical key management systems presented in section II; and finally in section V we present some conclusions and future work on the subject.

II. RELATED WORK ON HOWSN KEY MANAGEMENT

The key management solutions could be based on symmetrical or asymmetrical systems. The asymmetrical ones are more secure, nevertheless have more power requirements than the symmetrical ones due to its greater complexity and could become a burden to the HoWSN low resource equipments. In [10] an evaluation of the energy consumption and memory requirements of several key management protocols over some different sensor node technologies is made. In this paper we focus only on symmetrical key systems, considering that in some real world situations the balance between the improvements of security achieved with the asymmetrical key systems and the low resources of wireless sensor networks demanding for power saving algorithms achieved with the symmetrical key systems, leads to the choice of the symmetrical key systems.

We can classify the symmetrical key systems in the following three classes:

- a) *Single Key Systems* – Use a single key loaded in each node at deployment phase, for all the communications in the network;
- b) *Pair-Wise Key Systems* – Use a different key between every two nodes of the network. In the deployment phase each node is loaded with a different key for each other node with whom it will be able to communicate to;
- c) *Random Key Systems* – These systems share the common property of being probabilistic. In the deployment phase each node is loaded with a subset of keys of a large set, that is itself a subset of a yet bigger super set of keys. The number of keys that each node will own (*its “Key Ring”*) is calculated taking as base the total number of nodes in the network and a predefined desired probability of network connectivity. The first proposal of this class of systems was done in [11] where two nodes could establish a communication link if they share at least one key. In [12] is proposed an improvement on security to the first system where two nodes could establish a communication link only if they share at least q keys.

In section IV we present the evaluation of our system and also of these three classes of systems.

III. PROPOSED SYSTEM

The proposed system for key management in HoWSN is based in the properties of chaotic systems to generate an input that acts as the key stream to be used by the encryption block of the symmetric system used by the nodes. As referred before, the main property of our proposal is the possibility to implement an OTP system in HoWSN, improving this way its security.

A. Architecture

In figure 1 we show the generic architecture of the proposed system, which comprises eight main blocks that will be described next. We do not present the memory block as well the control lines of the architecture since they are not relevant for the purpose of this paper.

- Transmission/Reception (Tx/Rx) responsible for the transmission and reception ciphers over the channel;
- Secret Shared that has a set of information stored in all the nodes at deployment phase and that is the same for all nodes;
- Private Shared that has a set of information stored in all the nodes at the deployment phase and that is unique for each node;
- PRNG that is a Pseudo-Random Number Generator;
- Chaotic System that will be used as key input for the encryption module;
- Encryption Module that will encrypt or decrypt plaintexts or ciphers that are sent over the communication channel;
- Reception Control used to process the ciphers that came from the channel through the Transmission/Reception module;
- One Time Pad Generator stores the data required for all the nodes which whom there is a partnership established.

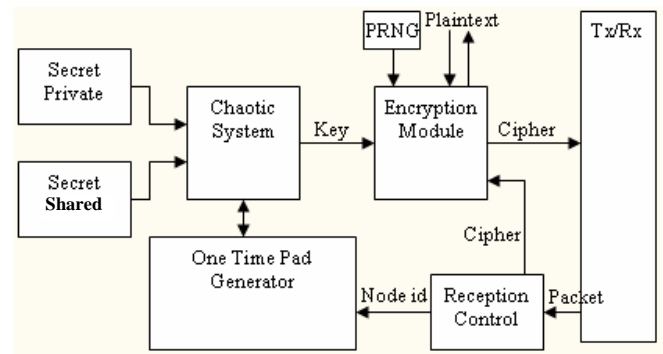


Figure 1 – Generic architecture of the proposed system

The proposed system could be implemented with any Chaotic System and any Encryption Module. Nevertheless we will explain its functionality using as chaotic system the well known Lorenz equations and as Encryption Module the AES with 128 bits as stated in the IEEE 802.15.4 specification [8] and used in ZigBee [9]. Figure 2 shows the resulting architecture.

The Secret Shared (SS) contains five vectors that could be split in two sets. The first set containing SS1, SS2 and SS3 is used to generate the Certification key, and the SS4 and SS5 that are used to the OTP generator of keys. SS1 contains the constants of the Lorenz system (r_c, b_c, σ_c); SS2 contains the coordinates of the starting point of the Lorenz system named as (x_{0c}, y_{0c}, z_{0c}); SS3 is the number of iterations of the Lorenz system needed to get the first key named “ it_c ”. In all these three vectors the index “c” states for “Certification”. SS4 are the constants of the Lorenz system named as (r_k, b_k, σ_k), where the

index “k” states for “Key” and SS5 is the step of progression of the Lorenz System, named Δt_k .

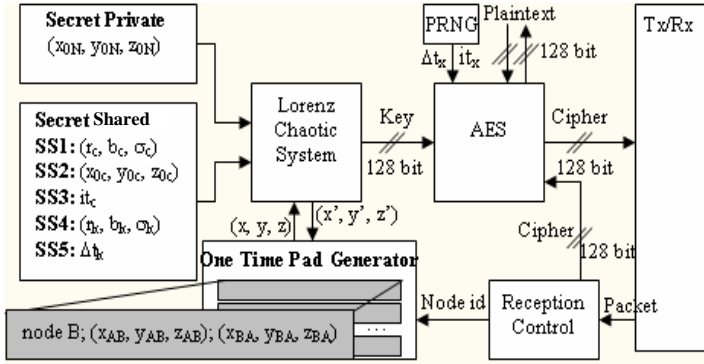


Figure 2 – Architecture of the proposed system targeted to the chaotic Lorenz system and the ZigBee specification

The Private Shared contains the coordinates of the starting point of the Lorenz system named as (x_{0N}, y_{0N}, z_{0N}) , where the index “N” identifies the node.

The One Time Pad Generator contains one row for each node with which there is a relationship established. It stores the following values: nodeid, that is the identification of the communication partner node; $(x_{0AB}, y_{0AB}, z_{0AB})$, that stores the last key issued form A to B; and $(x_{0BA}, y_{0BA}, z_{0BA})$, that stores the last key issued form B to A.

The coordinates (x, y, z) of the Lorenz system will be the keys to be used in the AES module and so should be 128 bit long. As 128 is not divisible by 3 we take advantage of the fact that the z coordinate in the Lorenz system is always positive to represent only the signal of x and y coordinates. So we could store each one of the coordinates with 42 bits and the remaining 2 bits will be used for signal the x and y. Figure 3 illustrates this process.

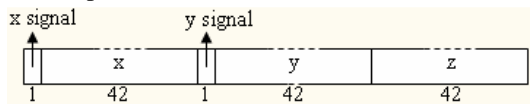


Figure 3 – Translation from the Lorenz coordinates to encryption key

B. One Time Pad bootstrap scheme

The start of a OTP relationship between two nodes takes place in three steps that are described bellow, considering that node B wants to begin the relationship with node A.

1) First Step – Node B:

1. Generates at random the step from A to B, named Δt_B ;
2. Generates at random the number of iterations, named it_{AB} , that A should give on the Lorenz system to get the certified key from A to B, named K_{cAB} ;
3. Calculates the certified key based on the SS1, SS2, SS3 and Δt_B , named $K_{c,\Delta tB}$ to encrypt the it_{AB} ;

4. Encrypts it_{AB} using $K_{c,\Delta tB}$, and gets $E_{K_{c,\Delta tB}} [it_{AB}]$

5. Send Δt_B , and $E_{K_{c,\Delta tB}} [it_{AB}]$ to node A

2) Second Step – Node A:

The first 4 tasks that node A will perform in the second step are symmetrical to that ones performed by node B in the first step.

1. Generates at random the step from B to A, named Δt_A ;

2. Generates at random the number of iterations, named it_{BA} , that B should give on the Lorenz system to get the certified key from B to A, named K_{cBA} ;

3. Calculates the certified key based on the SS1, SS2, SS3 and Δt_A , named $K_{c,\Delta tA}$ to encrypt the it_{BA} ;

4. Encrypts it_{BA} using $K_{c,\Delta tA}$, and gets $E_{K_{c,\Delta tA}} [it_{BA}]$

5. Calculates the certified key $K_{c,\Delta tB}$ based on SS1, SS2, SS3 and the value Δt_B received from B;

6. Decrypts it_{AB} using $K_{c,\Delta tB}$;

7. Calculates the starting point $(x_{0BA}, y_{0BA}, z_{0BA})$ that B should use on the Lorenz system to rise up the OTP generator, based on its own Secret Private SP, on the Secret Shared value SS1, and using the step that B generates at random Δt_B , by iterating the Lorenz system with these parameters the number of times that B has also generated at random, the it_{AB} ;

8. Calculates the certified key K_{cAB} , using SS1, SS2 and Δt_B on the Lorenz system by iterating it again it_{AB} times;

9. Encrypts the starting point for B calculated in 7 using the Certified key calculated in 8, and gets $E_{K_{cAB}} [(x_{0BA}, y_{0BA}, z_{0BA})]$;

10. Sends Δt_A , $E_{K_{c,\Delta tA}} [it_{BA}]$ and $E_{K_{cAB}} [(x_{0BA}, y_{0BA}, z_{0BA})]$ to B;

3) Third Step – Node B:

The tasks that node B must do in the third step are quite similar to those performed by node A on the second step from task 7 to 9.

1. Calculates the starting point $(x_{0AB}, y_{0AB}, z_{0AB})$ that A should use on the Lorenz system to rise up the OTP generator, based on its own Secret Private SP, on the Secret Shared value SS1, and using the step that A generates at random Δt_A , by iterating the Lorenz system with these parameters the number of times that A has also generated at random, the it_{BA} ;

2. Calculates the Certified key K_{cBA} , using SS1, SS2 and Δt_A on the Lorenz system by iterating it again it_{BA} times;

3. Encrypts the starting point for A calculated in 1 using the Certified key calculated in 2 and gets $E_{K_{cBA}} [(x_{0AB}, y_{0AB}, z_{0AB})]$;

4. Sends the encrypted starting point $E_{K_{cBA}} [(x_{0AB}, y_{0AB}, z_{0AB})]$ to A.

Figure 4 shows a message sequence chart of this process.

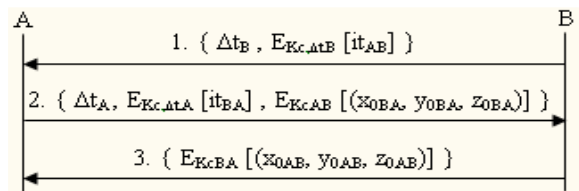


Figure 4 – Message Sequence Chart for the One Time Pad bootstrap

C. One Time Pad scheme evolution

After the bootstrap procedure, the first message that node A will send to node B will be encrypted using the key that results from the translation of the coordinates of the Lorenz system with the inputs SS4, SS5, and the starting point $(x_{0AB}, y_{0AB}, z_{0AB})$ that B has determined, after “ it_{AB} ” iterations. The translation from these coordinates of the Lorenz system to the key results directly from figure 3. These coordinates will be stored in the OTP Generator table as stated in the architecture. The second message will be encrypted using the key that results from the translation of the coordinates of the Lorenz system with SS4, SS5 as for the first key, but now using the previous coordinates that were generated to be translated to the first key and through the iteration of the Lorenz system just one time (or a single step of iteration). All the next keys will be obtained through the same process stated for the second key, using always the SS4, SS5, and the last coordinates generated by the system that are stored on the OTP Generator table, and iterating the system always just one time. The first key generated by the node B will be generated as well by the Lorenz system with SS4, SS5, the starting point $(x_{0BA}, y_{0BA}, z_{0BA})$ that A has determined, iterated it_{BA} times also determined by A. The following keys will be calculated using the same procedure stated to node A.

IV. EVALUATION

In this section we evaluate the proposed system concerning Storage Requirements, Scalability, and Key Attacks. Then we evaluate in the same aspects, the key managements systems based on symmetrical keys briefly described in section II.

A. Evaluation of the proposed system

Concerning the Storage requirements the proposed system demands in each node the capacity to store the Secret Private material, the Secret Shared material and two keys for each communication link. For simplicity of analysis we will not consider of relevance the storage space needed for the identification of the corresponding node to each key, and in the case of the Chaotic System of Lorenz presented in this paper we will not even consider the storage space needed for the SS3 and SS5. We will focus just on the space needed to store the keys. Considering the use of the Chaotic System of Lorenz and the AES encryption module, as can be observed in figure 2, each node of the proposed system will need space to store 4 keys plus 2 more keys for each communication link. The growth of the Storage requirements (i.e. the Ring Size) for each node of the proposed system could be related with the number of existing communication links by the following linear equation: $RingSize = 4 + 2 \times NumberOfCommunicationLinks$.

In real world applications, the number of communication links in HoWSNs should be reduced to only the adjacent nodes, in order to reduce the transmission power and to minimize inter-node interference. Besides that, the real world topologies for HoWSN applications could be approximated by two classes: linear topology and grid topology. For the linear topology it could be thought for instance a HoWSN along a road or highway in which each sensor will communicate with just the two adjacent nodes. For the grid topology it could be

thought as a HoWSN that covers a vineyard with sensor nodes along the grape-vine rows. In the grid topology, the typical number of nodes with which each node establishes a communication link is eight. The proposed system needs space to store simultaneously just 8 keys for the linear topology and 20 keys for the grid.

Relating to the Scalability the proposed system is limited to the possible number of different Secret Privates that could be generated. In the case under analysis that use the Lorenz chaotic system, the total number is 2 powered by 128 bits, giving us the number of $3.4e38$ nodes. This means “unlimited” in a real world point of view.

In what concerns to Key Attacks our proposal never repeats a key in its lifetime. This is guaranteed by the entropy inherent to the Secret Private that is different for all the nodes and that will be used to determine the initial conditions of the chaotic system together with the generation of some random numbers as exposed in section III. Complementing this entropy is the chaotic system itself. It guarantees an evolution that strongly depends on the initial conditions, and even by little changes in the initial conditions, the evolution of the chaotic system is unpredictable different. Our system is also resistant to the known chaotic systems cryptanalysis; good references on this subject are [3] or [4]. The protection of the chaotic system in our architecture is achieved through the use of the output of the chaotic system as input to the symmetrical cryptographic module, as the key to encryption/decryption. In the case exposed in this paper, the cryptographic module is the AES with 128 bits of key that have well known security properties. This way the security of the proposed system is the same as the cryptographic module used (in the case exposed, the AES) but greatly increased with the fact that it never use the same encryption key, offering this way the One Time Pad property.

B. Evaluation of other symmetrical key management systems

The Single Key based systems are the better ones concerning Storage requirements, because they just require one key in each sensor node. Concerning Scalability they are also the better ones because they are completely scalable, through the storage of the same single key in each new node. In what concerns to Key Attacks this systems are the worst ones because the same key is used to exchange every message in the network.

The Pair-Wise Key based systems are bad relating to the Storage requirements, because each node has to store one key for each other node in the network with whom it could be able to establish a communication channel. Considering that we don't know who will be the neighbors of each node, it is necessary to store in every node one key to establish a communication channel with any other node in the network. Relating to Scalability this class of systems are the worst ones because it is not possible to add more nodes to the network due to the need of storage in every other nodes a new key to communicate with the new node. In what concerns to Key Attacks this class of systems are better then the previous class because each key is used just in one communication channel of the network, minimizing the number of messages exchanged

that use the same key, and besides that the capture of one of those keys just compromise one communication channel.

The Random Key based systems like [11] and [12] present some practical drawbacks of implementation due to the trinomial “Storage requirements”, “Scalability” and “Network connectivity”. As briefly exposed in section II this class of systems are based on the establishment of communication channels between neighbor nodes that share common keys in their key rings. But it could happen that two neighbor nodes do not share any common key. So the network connectivity is completely dependent on this. From [11] it is possible to derive the *probability of network connectivity* (P_c) as a function of the total number of keys that exist to be distributed by the nodes (S), the number of keys in the key ring of each node (m), the total number of nodes in the network (n), and the number of neighbors of each node (n'), as this:

$$P_c = e^{-\alpha(S,m,n,n')} \quad \text{with} \quad \alpha(S,m,n,n') = ne^{\frac{n(n-1)}{n-1}p'(S,m)}$$

The $p'(S, m)$ represents the probability of two nodes share at least one key, given by:

$$p' = 1 - \frac{((S - m)!)^2}{(S - 2m)!S!}$$

In figure 5 we draw an example for $S = 10^5$ and $n = 10^4$, where is possible to observe the relation between the number of keys that is needed in the key ring of each node (m) to achieve a certain probability of network connectivity (P_c) that is dependent on the number of neighbors of each node (n').

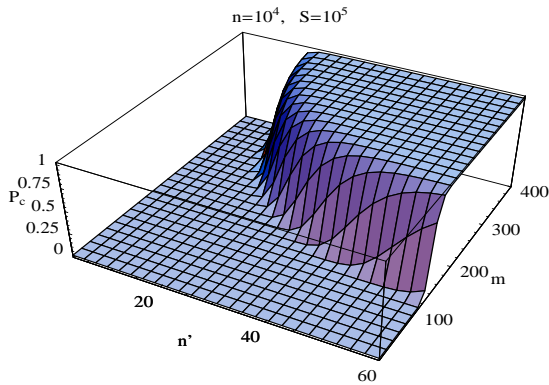


Figure 5. Connectivity probability as a function of the number of neighbours and the size of the key ring

From this draw its possible to observe that under about 20 neighbors there is no chance to achieve a probability of network connectivity near the 100% (in this calculus we use 0.99999 as was used in [11]). Considering the real world application topologies as described in the previous sub-section this is a real drawback of application of this class of systems. Increasing the number of neighbors trying to overcome this drawback will rise on problems of inter-node interference and besides that, the economical cost increase, as well it could also rise on some ecological issues depending on the application of the sensor network (as in wild life monitoring for instance). For example to achieve a probability of network connectivity near the 100% with 30 neighbors it will be needed 354 keys stored in they key ring of each node. To reduce this number we must increase the number of neighbors, with 60 neighbors the number of keys needed is reduced to 208. However the increase of the number of neighbors could not be done

infinitely. Besides all the previous exposed problems that rise on from this procedure, there are also physical limitations related with the number of communications that could be established in a limited geographical area using the wireless medium. In fact we could imagine a graph with millions of nodes in a square meter, having each node multiple connections to other nodes. But if we change those nodes by wireless sensor nodes, the establishment of the communication channels will be strongly limited. Another aspect of this class of systems is the need to know the total number of nodes in order to calculate the number of keys needed in each key ring to achieve a certain probability of network connectivity. This establishes from the beginning a Scalability limit for the network. Increasing the total number of nodes in order to maximize the upper limit will also increase the number of keys needed in each key ring, rising on again the Storage requirement issue.

V. CONCLUSIONS AND FUTURE WORK

The system proposed in this paper presents an architecture that overcomes the drawbacks of the actual symmetrical based systems, however some disadvantages of this type of systems still remains, like attacks to single nodes to extract the common key, that in our case are the Secret Shared and the Secret Private of the chaotic system. We are working on the evolution of this system to a hybrid symmetrical and asymmetrical system to improve the security services of the system, namely to make it resistant to single node attacks, a property that is difficult to achieve with pure symmetrical system.

REFERENCES

- [1] M. Hasler, “Synchronization of chaotic systems and transmission of information”, International Journal for Bifurcation and Chaos, vol.8, no.4, pp.647-659, 1998.
- [2] M. Feki, “An adaptive chaos synchronization scheme applied to secure communications” Chaos Solitons and Fractals., vol.18, pp.141-148, 2003.
- [3] T. Yang, L. B. Yang, and C. M. Yang, “Cryptanalyzing chaotic secure communications using return maps”, Physics Letters A, vol.245, pp. 495-510, 1998.
- [4] G. Álvarez, and S. Li, “Breaking network security based on synchronized chaos”, Computer Communications, vol.27, no.16, pp.1679-1681, 2004.
- [5] F. Dachselt and W. Schwarz, “Chaos and Cryptography”, IEEE Trans. on Circuits and Systems-1, vol.48, no.12, 2001.
- [6] A. Perrig, J. Stankovic, and D. Wagner, “Security in Wireless Sensor Networks”, Communications of the ACM, vol.47, no.6, 2004.
- [7] M. Eltoweissy, M. Moharrum, and R. Mukkamala, “Dynamic Key Management in Sensor Networks”, IEEE Communications Magazine, April 2006.
- [8] IEEE Computer Society Press, “Wireless medium access control and physical layer specifications for low-rate wireless personal area networks. Standard 802.15.4-2003”, May 2003.
- [9] ZigBee Alliance, “ZigBee Specification”, June 2005.
- [10] D. W. Carman, P. S. Kruus, and B. J. Matt, “Constraints and approaches for distributed sensor network security”, NAI Labs Technical Report #00-010, September 2000.
- [11] L. Eschenauer, and V. Gligor, “A key-management scheme for distributed sensor networks”, 9th ACM Conference on Computer and Communication Security, 2002.
- [12] H. Chan, A. Perrig, and D. Song, “Random Key Predistribution for Sensor Networks”, IEEE Symposium on Security and Privacy, 2003.