Lightweight, Dynamic, and Flexible Cipher Scheme for Wireless and Mobile Networks

Hassan Noura and Damien Couroussé

Univ. Grenoble Alpes, F-38000 Grenoble, France
CEA, LIST, MINATEC Campus,
F-38054 Grenoble, France
hassan.noura@cea.fr, damien.courousse@cea.fr

Abstract. The security of Wireless and Mobile Networks (WN, and MN, respectively) is crucial for effective deployment in various areas and applications such as military and business. The existing security solutions are based on static block /stream cipher to ensure Data Confidentiality (DC). These solutions require multi-round function, and consequently a high computing complexity and energy consumption. However, WN or MN has limited resources that prevent their efficient deployment for a long period. To overcome the previous challenge, a new kind of cipher scheme based on a dynamic permutation packets cipher is presented in this paper to ensure the DC requirements with low computation complexity. Theoretical results show that the proposed algorithm has a reduced computational complexity, which can lead to reduce the energy consumption. It is equally important to note that our proposed solution could be adapted for other kinds of networks that employ packet transmission such as vehicular network.

Keywords: Data Confidentiality, dynamic and lightweight cipher, flexible permutation layer, security analysis.

1 Introduction

Wireless Sensor Networks (WSNs) are used for many purposes, such as monitoring and collecting data as well as accessing and evaluating such information. Indeed, WSNs are appearing in enormous disciplines such as: smart houses, building, environment monitoring, traffic monitoring, military surveillance, health monitoring, or even in bodies as patient monitoring.

Typically, WSN are consisted of small devices that have the capability to gather information about their physical environments. WSNs realize the communication among the nodes by a multi-hop routing protocol. Usually, users of WSNs are divided into two different types (see Fig. 1): Base Station (BS) or **carrier** and sensor node. However, the major problem that threatens WSN is the security, since they are susceptible to several kinds of attacks such as passive and active attack [1], [2]. The former can seriously impair the confidentiality of the network, by trying to extract the content of transmitting packets, while the

N. Mitton et al. (Eds.): AdHocNets 2015, LNICST 155, pp. 225–236, 2015.
DOI: 10.1007/978-3-319-25067-0_18

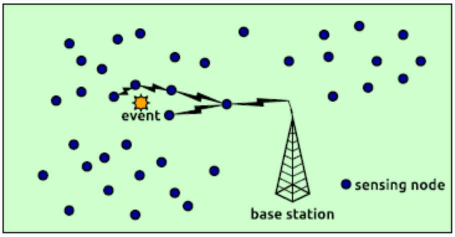


Fig. 1. An example of WSNs scheme

latter can damage the network authentication, by inserting, deleting or modifying the packet contents. One solution to overcome the passive attack, is to encrypt the transmitted packets among sensor nodes.

Hence, it is necessary to ensure that the transmitted data is secure enough from any unauthorized access (DC), as well as that data exchange is occurring only between legitimated parties. With state-of-the-art, low resources, limited computing power, limited power as well as limited battery lifetime are the main characteristic of any WSN architecture. Indeed, these limitations pose several problems from the cryptographic view point and attract many researchers since existing solutions suffer from the limited battery lifetime, where the battery of a sensor is depleted rapidly, and consequently terminates the network lifetime.

A block cipher such as AES [3] is used in real implementationwith secure operation modes such as (OFB) and Counter (CTR) [4], which are more suitable schemes for WSN, where the ciphering process is independent of plain-text, and can be considered as a stream cipher. However, despite its high level of security, AES has a high computational complexity since it uses multi-round structure. The presented implementation in [5] shows that AES decreases rapidly the lifetime of nodes and networks such as in ZigBee [6], WirlessHART [7]. As a conclusion, AES is not suitable for WSN platform, since its average performance has been higher on a range of sensor standard platforms. First, a security Protocol for Sensor Networks (SPINS) based on a block cipher (AES) in Counter mode (CTR) is presented in [8]. SPINS offers several security services such as DC, in addition of low communication overhead (8 bytes per packets). Then, in [9], a new protocol is presented to replace SPINS, while providing similar security services and denoted by TinySec [10] and it is the TinyOS security platform. Furthermore, TinySec suggests to replace AES by other block cipher based on its performance on WSN nodes such as Skipjack [11], or RC5 [12]. Moreover, Skipjack is used in the majority of well-known security platforms for WSN in addition, such as in SenSec [13] and TinyKey-Man [14]. The traditional approach uses the multi-round r function, which can be categorized into two classes: Feistel Networks (FN) and Substitution-Permutation Networks (SPN). Indeed, for each round, several simple iterated functions are applied, which require an important overhead. However, the security level depends on the number of rounds r, which leads to a trade-off between high security level and required computational complexity and consequently overhead energy consumption.

These WSN protocols ensure secure data transmission over the network, but with a **low network performance**. Hence, the limitations existed in WSN prevent the traditional security tools to achieve the security aspect efficiently. Recently, a new dynamic cipher kinds using dynamic diffusion layers in the integer Galois field were defined in [15]. After that, an enhanced scheme is defined in [16] by using binary diffusion matrix that is based on binary bmixing operation compared to the multiplication operations in integer field that are complex and large in size, slow in speed, and consume much power.

Furthermore, to achieve a secure WSN, a trade-off between security and performance is presented in existing protection techniques. The security in WSN suffers from various limitations and vulnerabilities, which encourages to implement new kinds of packet encryption to achieve a secure data transmission among sensor nodes with low computation complexity.

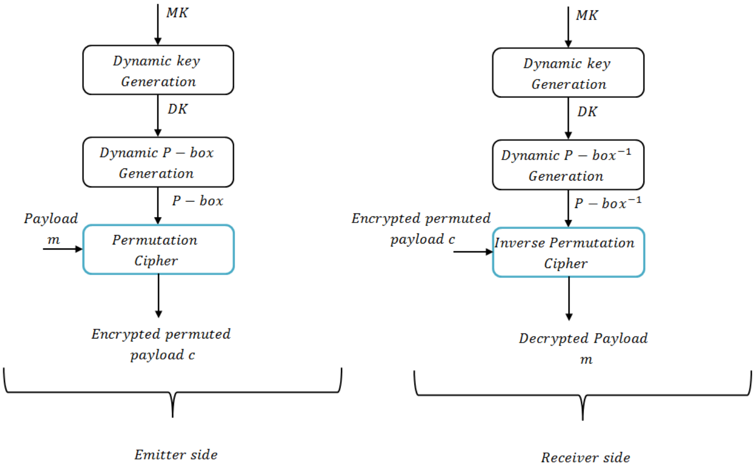
To overcome this problem, especially in constrained resources WSN, a new efficient cipher must involve. From here, comes the idea of our secure cipher scheme that achieves DC in an efficient manner and with the respect of WSN characteristics, in particular the throughput and energy consumption. In this paper, the proposed cipher consists of a dynamic permutation layer for the packet payload. The permutation process is realized using our modified version of GRP, which is our second contribution. The rest of this paper is organized as follows. Section 2 starts by describing our design goals & rationale then presents the proposed secure scheme, and then defines a new construction technique of key dependent and a flexible permutation layer based on our modified scheme of GRP algorithm [17]. Performance and security of the proposed scheme are analyzed in Section 3. Finally, Section 4 presents our conclusion.

2 The Proposed Secure Scheme

The proposed scheme has been designed with the following goals in mind:

1. **Simplicity:** As our approach is designed to be applied on sensor nodes, then it has to cope with the limitations as well as the requirements of the various kinds of WSN which are often fastest and lowest memory using. Our approach was also designed with as simple as possible computational operations in mind, suitable for sensor devices. One round of dynamic permutation operations makes our proposal efficient on a larger number of software platforms. The absence of S-box, diffusion, and key expansion makes our proposal small and efficient in hardware as well.
2. **Security against Attacks:** In fact, the cipher should provide strong resistance against exhaustive search attacks. A relatively large key size (128 bits) was therefore chosen for our approach.

In this section, a new efficient and secure permutation scheme is discussed. Usually, the term efficiency means achieving the security conditions of WN with



(a) (b)

Fig. 2. Proposed cipher scheme at the emitter (a) and receiver side (b)

a little amount of time. The proposed scheme overcomes the disadvantages presented in the previously discussed techniques, and defines new kinds of cipher. In addition, it ensures a simple implementation when operating with constrained devices.

Table 1. Dynamic keys generation

```
procedure KEY_UPDATE($Mk$, $SK_{c1}$, $adin$, $i, c1$ , $c2$)
    if ($Ctr_2 \% w == 0$) then
                                                    ▷ Update the session key

        $Ctr_1 \leftarrow Ctr_1 + 1$
        $SK_{Ctr_1} \leftarrow SHA - 512(MK||c1||adin)$
    end if
                                                    ▷ Produce the dynamic key

    $O_{Ctr_2} \leftarrow SHA - 512(SK_{Ctr_1}||Ctr_1||Ctr_2)$
    $DK_{Ctr_2} \leftarrow LSB(O_i,\ 4 \times l)$
    return $DK_{Ctr_2}, SK_{Ctr_1},\ Ctr_1,\ Ctr_2$
end procedure
```

First, an extension for the packet header is introduced to express the sequence number of packets as NP with length equal to 1 byte (optimal value in order to reduce the trade-off between security and communication overhead). This header, similar to the sequence number used in IPSec, is involved in the proposed key derivation function to provide robustness against replay attacks. Indeed, a key exchange is supposed to be realized.

2.1 Secure Scheme at Emitter Side

In practical WSNs scenarios, a source node needs to transmit some data denoted by M. First, at the source side, the required data to be transmitted is divided into different packets M_1, M_2, $\ldots$, M_n. The different steps of the proposed scheme at the emitter side are described below in details:

Dynamic Key Generation. The dynamic key is produced using HASH-CTR DRBG, where its theoretical security analysis was analyzed in [18] and its robustness and performance have been proved. **Dynamic** key approach is used in our scheme instead of a static key approach in order to overcome the fixed key problem.

This process is presented in the following pseudo code as in Table 1 and seen in Fig. 3. In this section, the process of dynamic key generation is deeply explained starting with the generation of the session keys and ending with the reconstruction of dynamic keys from each session key.

First, a secret key is agreed between the sensor nodes and the sink. This single private key called 'Master Key' is designed by MK. Besides, Ctr_1 is a counter that is incremented for each w packets ($w = 251$). The Master key MK_{c1}, $c1$ value and some additional information related to the source/sink node $adin$ are concatenated together, then hashed using SHA-512 in order to perform at the end the session key required for the Ctr_1^{th} interval, denoted by SK_{Ctr_1}. Then, for each d (with $d \leq w$, $d = 251$) packets, the session key value SK_{Ctr_1} is combined with Ctr_1 and Ctr_2 values to perform O_{Ctr_2} value. Finally, a dynamic key, denoted by DK_{Ctr_2} is obtained directly by truncating $4 \times l$ -bits of Least Significant Bit (LSB) of O_{Ctr_2}. Noting that the size of the Master MK_{c1} and session SK_{c1} keys are 512 bits, while the dynamic key DK has a variable size.

Encryption Cipher Based on Dynamic, Flexible *P_box*. After the dynamic key generation, a key dependent permutation layer is fulfilled. The GRP permutation algorithm is defined in [17], can be considered as simple, flexible, and efficient in software and hardware implementation that is the reason why it chooses. In the following, the GRP permutation algorithm, used as a basic element of our permutation scheme, is described: As in Fig. 4, $R1$ is the source array or the original vector, CR is the configuration vector (control register) and $R3$ is the destination vector. The basic idea of the GRP instruction is to divide the index into two groups according to the pseudo-random bit sequence (CR). If the bit in CR is 0, this index is moved into the first group. Otherwise, this element is put into the second group.

A modification scheme of GRP is proposed here and described in Table 2. This modification was done in order to enhance the level of random of permutation but with acceptable computation. It consists of iterating two times the GRP's

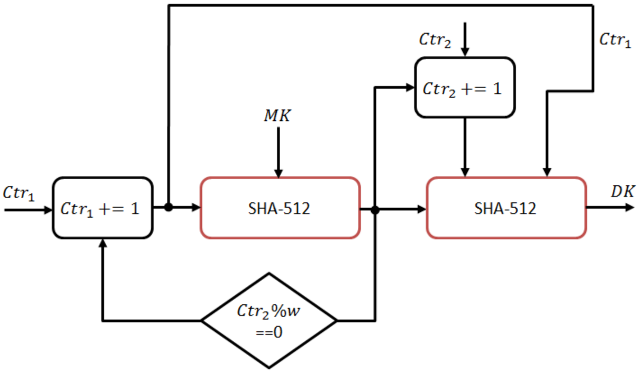


Fig. 3. Proposed key derivation function

algorithm for each round of permutation rp. CR and the bitwise complement of CR, are used as a control register permutation for the first and second time, respectively. The final permuted vector P_box is obtained after rp (round of permutation). Another contribution is to use different CR for each iteration of $Perm$ round function.

Table 2. Proposed permutation algorithm

```
procedure PERM($DK$, $l$, $rp$)
                                                ▷ $L$ is the length of input vector

    $P\_box \leftarrow 1$ to $l$

    for $i \leftarrow 1$ to $rp$ do
        $CR_i \leftarrow DK[(i-1) \times l \ : \ (i-1) \times l - 1]$
        $P\_box = GRP(P\_box,\ CR_i)$
        $P\_box = GRP(P\_box,\ \overline{CR_i}\ )$
    end for
                                                ▷ $P\_box$ is a dynamic $Pbox$

    Return $P\_box$
end procedure
```

Where i and ($P\_box[i]$) are the original and permuted positions of the input packet, where l represents the size of a packet. This transformation is iterated 4 times to ensure a good cryptographic performance (see Fig. 6). Hence, as result 4 control registers are needed to be used to calculate $Pbox$. In this context, the dynamic key DK of size $4 \times l$ bits is divided into 4 components $CR = CR_1,\ CR_2,\ CR_3,\ CR_4$ each of size of l bits, and used for its corresponding iterations. Indeed, this transformation requires for each iteration a control parameter CR_i, $i = 1,\ 2,\ \ldots,\ rp$ and each one can be obtained directly from

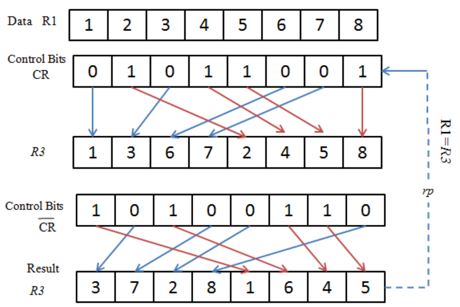


Fig. 4. Example of the proposed permutation $PERM$ algorithm with $l = 8$

DK. After producing the $Pbox.$, each element of the packet contents is permuted by using its correspondent permuted index and the permutation process is applied byte-by-byte as seen below:

$$c[i] = m[P\_box[i]] \tag{1}$$

where $m[i]$ and $c[i]$ are the $i - th$ original and encrypted (permuted) byte of packets respectively. $Pbox[i]$ is a permutation coefficient for the i^{th} elements. After applying the permutation process, the output encrypted payload C is the permuted encrypted packets.

2.2 The Proposed Secure Scheme at the Receiver Side

The different steps of the proposed scheme at the receiver side are described below:

1. The receiver buffering model sorts the packet stream into packets according to their NP.
2. then, DK is generated using the same approach, that was investigated at the emitter side.
3. After that, the destination produces the inverse secret permutation process P_box^{-1} by using the same permutation scheme but applied in reverse order of control parameters.

After producing the $Pbox^-1$, each element of the encrypted packet contents is permuted by using its correspondent inverse permuted index as seen below:

$$d[i] = m[P\_box^{-1}[i]] \tag{2}$$

where $c[i]$ and $d[i]$ are the $i - th$ encrypted (permuted) and decrypted byte of packets respectively. $Pbox^{-1}[i]$ is an inverse permutation coefficient for the i^{th} elements. After applying the process of inverse permutation, the output decrypted payload D is the original content of packet.

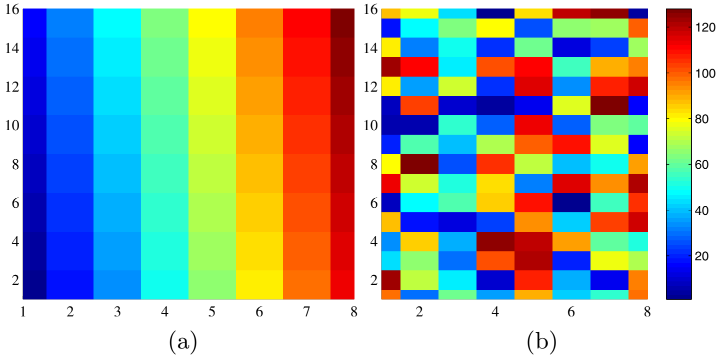


Fig. 5. Original and Permuted indexes for a random produce dynamic P-box in a matrix form with $l = 128$

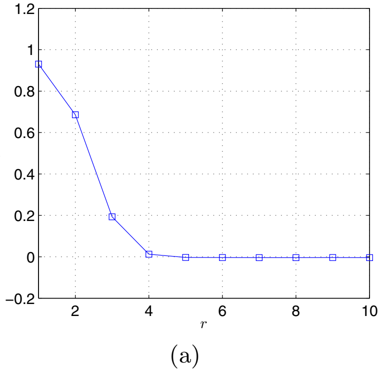


Fig. 6. Variation of the average of ρ of the recurrence of producing P-boxes versus rp for 1000 random dynamic keys

3 Cryptographic Strength and Performance

3.1 Cryptographic Performance of the Proposed Dynamic Permutation Layer

The performances of the proposed dynamic permutation scheme should be quantified in order to demonstrate its safe implementation. Indeed, a new criterion, which is the coefficient correlation (described in [19]) between the recurrence of permuted vector $((P\_box(t),\ P\_box(t+1),\ t = 1,\ 2,\ \ldots,\ l-1)$ is used in order to quantify the round number of permutations rp. These tests were applied for $nk = 2^{15}$ random dynamic keys. Fig. 6 shows the average of the coefficient correlation between the recurrence of permuted index versus rp for $l = 116$, which is the maximum length of the payload in WSN. It is clear that for $rp \geq 4$, the coefficient correlation becomes close to zero (ideal value). Consequently, rp should be 4, and the choice of this value is justified.

3.2 Key Sensitivity

Sensitivity refers to a huge change in the cipher-text, responding to a slight change in the keys K. The sensitivity of K is analyzed for 1000 random keys, using the percent Hamming distance PH that is calculated between two vectors X and Y with the same length l as following:

$$PH = \frac{\sum_{j=1}^{l} Byte2Bin(X_j \oplus Y_j)}{l \times 8} \times 100\% \tag{3}$$

In this case, the sensitivity of K becomes as follows:

$$\begin{aligned} KS_w =& \frac{E_{K_w,\, IV}(M) \oplus E_{K'_w,\, IV}(M)}{l \times 8} \times 100\% \\ =& \frac{\sum_{j=1}^{l} Byte2bin(C_j^w \oplus C_j^{w'})}{l \times 8} \times 100\% \end{aligned} \tag{4}$$

where C_w, C'_w are the corresponding cipher packets using K_w and K'_w respectively. All the elements of K'_w are equal to those of K_w, except one element, which is the random Least Significant Bit (LSB), which was flipped to show the sensitivity of the scheme with a little change in key. Fig. 8, show the sensitivity of the secret key and its distribution, where only a LSB is changed on the secret key versus 1000 random keys for initial and enhanced scheme.

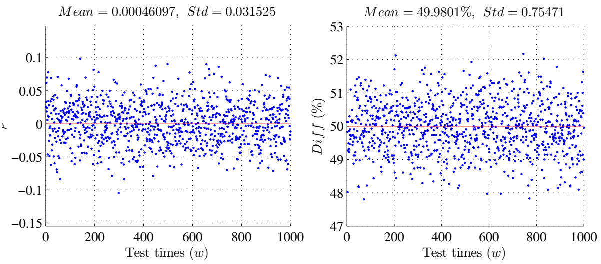
Additionally, Fig. 7-b shows the percent of hamming distance PH between original and permuted packet. From these figures, it can be seen that the majority of samples are close to the optimal value in bit level (50%). Therefore, we can consider that the proposed cipher block is strong enough to make the chosen/known plain-text attacks ineffective, while a dynamic key is used for each input packet.

3.3 Cryptanalysis

A cryptographic scheme is considered secure if it can resist attacks. The cryptographic security of our scheme relies on two properties:

- The use of a dynamic key (using counter mode).
- The unpredictability and high sensitivity of the secret permutation layer P.

However, all the packet contents are permuted via the dynamic secret permutation layer *Pbox*, at the source node before transmitting, and the intermediate nodes have no knowledge about DK. Thus, making the reconstruction of the original packet content very difficult.



(a) (b)

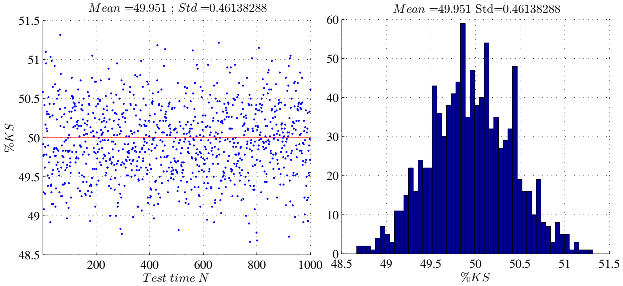
Fig. 7. The variation of the coefficient correlation between the original and encrypted contents packets (a) and PH between plain and cipher-packet (b) versus 1000 random dynamic keys respectively, with $l = 116$

Moreover, in Fig. 7-a, the average coefficient correlation between the original and encrypted packets for 10000 different secret permutation layers is shown. These results indicate that no detectable correlation exists between the original and its corresponding cipher packets which indicates that our proposal ensure security against statistical attacks. This discussion is presented in order to prove that the cryptographic security of our proposal is similarly powerful like traditional cryptographic solutions, and moreover, it ensures computational difficulties for a global attack to recover any meaningful information. Additionally, the proposed scheme works on dynamic manner, which means that the use of special encrypted packets will not lead to obtain any useful information about dynamic key and consequently about the session and master key respectively. Therefore, the key space of the master or the dynamic key of our scheme is sufficiently large to make the brute-force attack unfeasible. Moreover, the key space of the master keys is 2^{128}.

Besides, using the dynamic key method will limit the ability of the attackers to break our proposed scheme. The sensitivity of the master and dynamic keys is proved since our proposed scheme uses the cryptographic keyed hash function $SHA-512$.

3.4 Flexibility and Execution Time

Our proposed scheme ensures the flexibility against the packet length l, while it is able to extend (increase/decrease) the length of the payload. On the other hand, the execution time is a very important factor for any cipher algorithm, since less computational time means low computation complexity, and consequently less energy consumption and minimum resource requirements for ciphering/ deciphering process. This is considered as paramount for practical importance, especially for recent kinds of networks, where huge amounts of data are transmitted. The computation complexity of the proposed permutation cipher is $O\ (l)$ in addition to the process of generation of P-box, which is also linear $(O\ (l))$.



(a) (b)

Fig. 8. The sensibility results for change a random LSB bit of the secret key versus 1000 random keys (a) and its corresponding density function (b)

Let us note that an iteration of unkeyed hash function SHA-512 (with small input block 512 bits) is also required for each input packet (high level of security), or for a set of packets (depend on the configuration). This shows that the proposed method is sufficiently fast for applications and especially for real time constraint applications.

4 Conclusion and Perspectives

Security in MN and WN becomes more and more crucial, due to the vastness use of this field such as WSN. The existing schemes using cryptographic algorithms cannot achieve a low execution time for high security level such as AES. In fact, a new security scheme has been proposed and realized to ensure safe data exchange, while providing less complexity and consequently less energy consumption. After that, simulation results are discussed and analyzed to validate the robustness of the proposed packet encryption scheme, its degree of randomness, and its key sensitivity as well as its cryptographic strength against different traditional and physical attacks (dynamic keys). These results indicate a significant improvement compared to AES, which leads to achieve the required security level with lower computational complexity. Indeed, our proposed scheme can be well deployed in any kind of networks, and also it appears to be adequate for the use with real time application with constrained devices.

References

1. Huang, Y.: Research of efficient security scheme in wireless network. In: Liu, X., Ye, Y. (eds.) Proceedings of the 9th International Symposium on Linear Drives for Industry Applications, volume 4. LNEE, vol. 273, pp. 717–724. Springer, Heidelberg (2014)
2. Karygiannis, T., Owens, L.: Wireless network security. In: NIST Special Publication, vol. 800, p. 48 (2002)

3. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer, Heidelberg (2002)
4. Dworkin, M., Dworkin, M., Gallagher, P.D., Director Nist Special Publication -f : Recommendation for block cipher modes of operation: Methods and techniques (2001)
5. Lee, H., Lee, K., Shin, Y.: Aes implementation and performance evaluation on 8-bit microcontrollers. CoRR, abs/0911.0482 (2009)
6. Evans-Pughe, C.: Bzzzz zzz [ZigBee wireless standard]. IEE Review 49(3), 28–31 (2003)
7. Raza, S., Slabbert, A., Voigt, T., Landernäs, K.: Security considerations for the wireless hart protocol. In: Proceedings of the 14th IEEE International Conference on Emerging Technologies & Factory Automation, ETFA 2009, pp. 242–249. IEEE Press, Piscataway (2009)
8. Perrig, A., Szewczyk, R., Tygar, J.D., Wen, V., Culler, D.E.: Spins: security protocols for sensor networks. Wirel. Netw. 8(5), 521–534 (2002)
9. Karlof, C., Sastry, N., Wagner, D.: Tinysec: a link layer security architecture for wireless sensor networks. In: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys 2004, pp. 162–175. ACM, New York (2004)
10. Karlof, C., Sastry, N., Wagner, D.: Tinysec: a link layer security architecture for wireless sensor networks. In: ACM, pp. 162–175 (2004)
11. Skipjack, N.: KEA algorithm specifications (1998)
12. Rivest, R.L.: The rc5 encryption algorithm. In: Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008, pp. 86–96. Springer, Heidelberg (1995)
13. Li, T., Wu, H., Wang, X., Bao, F.: SenSec design. Institue for InfoComm Research, Tech. Rep. TR-I2R-v1, vol. 1 (2005)
14. Du, W., Deng, J., Han, Y.S., Varshney, P.K., Katz, J., Khalili, A.: A pairwise key predistribution scheme for wireless sensor networks. ACM Transactions on Information and System Security (TISSEC) 8(2), 228–258 (2005)
15. Noura, H., Martin, S., Agha, K.A.: E3sn - efficient security scheme for sensor networks. In: SECRYPT, pp. 615–621 (2013)
16. Noura, H., Martin, S., AI Agha, K., Grote, W.: Key dependent cipher scheme for sensor networks. In: 2013 12th Annual Mediterranean Hoc Networking Workshop (MED-HOC-NET), pp. 148–154, June 2013
17. Lee, R.B., Shi, Z., Yang, X.: Cryptography efficient permutation instructions for fast software. IEEE Micro 21(6), 56–69 (2001)
18. Campagna, M.J.: Security bounds for the nist codebook-based deterministic random bit generator (2006), matthew.campagna@pb.com 13453 received November 1, 2006. http://eprint.iacr.org/2006/379
19. Thirteen Ways to Look at the Correlation Coefficient. The American Statistician 42(1), 59–66 (1988)