10-1-2009

Energy-Efficient Multi-Key Security Scheme for Wireless Sensor Network

Sandeep Kolli

Maciej Jan Zawodniok
Missouri University of Science and Technology, mjzx9c@mst.edu

Recommended Citation

S. Kolli and M. J. Zawodniok, "Energy-Efficient Multi-Key Security Scheme for Wireless Sensor Network," *Proceedings of the 34th IEEE Conference on Local Computer Networks, 2009*, Institute of Electrical and Electronics Engineers (IEEE), Oct 2009.
The definitive version is available at https://doi.org/10.1109/LCN.2009.5355043

The 5th LCN Workshop on Security in Communications Networks (SICK 2009)
Zürich, Switzerland; 20-23 October 2009

Energy-Efficient Multi-key Security Scheme for Wireless Sensor Network

Sandeep Kolli, IEEE Student Member
Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, USA
skkw8@mst.edu

Maciej Zawodniok, IEEE Member
Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, USA
maciej_zawodniok@ieee.org

Abstract— **This paper proposes multi-key encryption scheme and engine architecture (MKE) that increases security and optimizes energy efficiency of sensor networks, while minimizing modifications to existing implementations. The scheme improves security of AES against correlation power analysis (CPA) attack by employing MKE engine, breaking the correlation between power consumption and the used key. Other schemes utilize complex hardware designs, for example by using the inhomogeneous s-boxes that reduce energy efficiency of the engine. In contrast, the proposed hardware engine uses a randomly sequence of few keys to encode subsequent blocks of a messages. Additionally, the scheme improves security of AES against brute-force attacks for a given key size by utilizing multiple keys to encrypt subsequent blocks of a message. In contrast, a typical security upgrade would require a larger key size and encryption engine, which would increase cost and energy consumption of the devices. Both analytical and simulation results are presented in this paper.**

Keywords— Wireless sensor networks (WSN), Advanced Encryption Standard (AES), Correlation power Analysis (CPA), key pre-distribution.

I. INTRODUCTION

The wireless sensor networks are increasingly employed in various applications, for example border security with sensors to detect intrusions, drinking water contamination detection around the water collection point. The secure communication in such networks becomes increasingly important. The system should prevent unauthorized access, tampering with the systems, and other malicious activities. However, the WSN poses several specific challenges related to security. First, a typical WSN consists of hundreds or thousands of battery-powered sensing devices. Hence, the sensor mote's cost and energy consumption have to be minimized in order to make the WSNs viable for commercial applications. Second, the limited processing capabilities prevent from utilizing complex and computationally intensive encryption algorithms. However, the required security level increases over time with the computational power available to attackers. Consequently, the ever-improving processor's speed and capabilities force the sensor designer to increase the size of encryption key thus increasing cost and energy consumption of the devices. Hence, a simple increase of key size is not suitable to handle the tradeoff between the security, energy efficiency, and cost requires a different approach than a simple increase of the key size.

In contrast, the proposed scheme employs multiple keys to improve the overall security. The proposed scheme uses these keys in a random sequence to encrypt a particular message. Hence, if attacker discovers a single key then only a fraction of the message is compromised. Consequently, all the keys have to be compromised before the link becomes unsecure. Moreover, the order and the id of the particular key used are hidden from attacker since the keys are selected randomly and switched on a block-by-block basis instead of per packet or per link basis. The proposed scheme can be applied in any communication network with AES as encryption algorithm, to improve security against CPA attacks. However, it is especially suitable to memory constrained, energy-limited wireless sensor networks.

A. Background and Related Work

The proposed scheme aims at thwarting different attacks on AES algorithm while minimizing necessary changes to the encryption engine implementation. The scheme should be used in conjunction with a key distribution scheme that provides the nodes with sufficient number of keys. Many distribution algorithms [1][2] have been developed and can be used with the proposed scheme, for example Eschenauer and Gligor [3] have developed random key pre-distribution protocols in which a random subset of keys from a large pool of symmetric keys is loaded to each node have to find one common key from their subsets which will be used as shared secret key for protecting their communication. The Eschenauer-Gligor scheme has been progressively improved [4][5][6]. For example, Du et al. [5] proposed the DDHV (Du, Deng, Han, and Varshney) scheme. The DDHV scheme is based on a multiple key space scheme generated from Blom's [7] λ-secure symmetric key generation system, which is randomly assigned to each sensor node in the network. Schemes using random distribution of keys are proposed with different approach techniques to distribute keys among nodes based on different metrics like security, memory, and energy-efficiency. Also, similar mechanisms have been proposed for heterogeneous WSNs [8][9].

978-1-4244-4487-8/09/$25.00 ©2009 IEEE

In general, sensor nodes will either have to be powered by small nonrenewable batteries or use a modest amount of energy harvested from the environment, for example from solar panel. However, the amount of available energy in both approaches is limited thus thwarting node operational capabilities. Hence, developing energy-efficient cryptographic algorithms and methods is a critical issue in designing protocols for wireless sensor networks, including the security schemes. The sensors' resource constraints, coupled with their limited knowledge of the topology within which they are deployed, render public-key-infrastructure-(PKI) based schemes inappropriate for wireless sensor networks. Some existing studies suggest storing the master key in tamper-resistant memory to reduce the risk [10] [11]. However, there exist attacks that do not require direct access to the memory with the secret key. For example, CPA attack exploits knowledge of hardware implementation of AES to determine a key value using energy signatures.

Power analysis attacks pose a serious threat to implementations of cryptographic algorithms since the attacker does not have to gain access to protected memory. A practical power analysis attack method has been demonstrated [12] [13]. There exist strategies to protect a device against such an attack, for example by masking the intermediate results [14]. However, the existing approaches often require significant modification to the encryption engines [15] and increase energy consumption [16]. For example, a set of different, energy-suboptimal S-Box implementations can be used to hide the key at the cost of higher energy consumption [16]. In contrast, the proposed scheme allows selecting the most energy efficient AES implementation while counteracting CPA attacks by rotating available keys. Each key has a different energy signature, thus breaking correlation between energy consumption and a key value. Hence, the proposed scheme ensures the CPA attacks fails.

This paper presents analysis of the security improvements of the proposed multi-key encryption scheme using the AES cipher. The AES is a symmetric block cipher standard, which was issued by the National Institute of Standards and Technology (NIST) in 2001 [17]. However, it has to be noted that the proposed scheme does not dependent on a specific cipher since the theoretical analysis does not assume usage of AES. Existing multi–key schemes aim at improving the security against the brute force attack by increasing the effective key length. For example the Triple-DES [18] uses the three keys on each block of data subsequently. In contrast, the proposed scheme counters both the CPA and brute force attacks by using a randomly selected key for each block. Consequently, the attacker does not have information which key was used to encrypt the particular block.

The proposed scheme increases the security of AES algorithm with minimal modifications to the hardware design and without any changes to the AES methodology. It is assumed that the keys are available at the sensor nodes. An existing key management schemes [1][2][3][4][5][6][7] can be utilized to satisfy that assumption.

Remark: The dynamic key distribution schemes contribute to communication and power overhead of the whole network. Hence, the future work will include an extended performance analysis of the proposed scheme including the impact of a key management scheme. However, for clarity this work focuses on the analysis of the security aspects of the proposed scheme.

The main contributions of this paper are: (1) novel AES engine architecture that can utilize multiple keys to increase security, (2) theoretical analysis of the performance of the proposed scheme, and (3) a simple and energy-efficient hardware model for implementing the proposed scheme. The paper is organized as follows: in Section II, the main types of attacks on AES algorithm are discussed. Next, the proposed multiple-key encryption technique is presented in Section III. In Section IV, the theoretical analysis of proposed technique is given. Finally, the simulation results for power correlation attack are presented in Section V.

II. Security Attacks

In this paper, the performance analysis is conducted with regards to two types of attacks: (a) the CPA and (b) the brute-force attack on AES scheme. First the details of the attack technique are given, in order to gain understanding how the proposed scheme improves the security.

A. Correlation Power Analysis Attack

The CPA attack exploits the correlation between energy consumption of the circuit and the data processed on it. Often, it is assumed that the circuit's power consumption varies linearly with the bit-wise difference in the processed data [12][13]. For example, if a register bit changes value, the energy consumed is different than if the bit value does not change. For the particular S-Box design, the attacker can analyze correlation between power signatures and the pair of a plain text and a key. When the encryption is repeated for a sufficient number of plaintexts the statistical correlation with simulated keys can be calculated. The key with the highest correlation is considered to be the secret key. Moreover, this method is noninvasive since the attacker does no need to access and read the memory with the key, which often is protected [10][11].

Next, the analysis of CPA attack is performed for the partial key K_s for simplicity. However, the results are valid for the retrieval of the whole key since the method can be repeated to retrieve the other partial keys. For AES, the CPA attacker analyses power signatures for the first and last round of the encryption process. For the CPA attack, a predictable power consumption model is considered:

$$W_S(j) = \mathrm{W}\big(K_S, \mathrm{PT}(j)\big) \tag{1}$$

where $W_S(j)$ is the power consumption for plaintext j, K_S is a partial key, $PT(j)$ is the j^{th} random plaintext, and W is a function of power dissipation. Given N plaintexts, the predicted power $W_S(j)$ can be derived using (1), and the

corresponding power traces calculated. In general, the CPA assumes that power consumption is proportional to number of bits changed in S-Box register. In other words, the power is proportional to Hamming distance [21]. To simplify the analysis while not losing generality, it is assumed that the register, R is initialized to zero (0). Consequently, the power consumed can be expressed as

$$W = a \cdot H(D) + b \cdot D \tag{2}$$

where *a* is a scalar gain between the Hamming distance, *H,* and the power consumed, *W*. The current and previous states of the S-Box register are expressed as *D* and *R* respectively, *b* is power dissipation induced by noise offsets, and time dependent components in a 128-bit random key. If *D* contains *m* independent and uniformly distributed bits, the whole word has an average hamming weight μ=*m*/2 and a variance σ2=*m*/4. Hence, the correlation coefficient between *W* and *H(D)* is equal to:

$$\rho_{wH} = \frac{E(H(D)W) - E(H(D))E(W)}{\sigma(H(D))\sigma(W)} = \frac{a\sqrt{m}}{\sqrt{ma^2 + 4\sigma_b^2}} \tag{3}$$

When a partial key guess, *Ks*, is considered then

$$P = PT_S \oplus K_S \tag{4}$$

where PT_s denotes the corresponding partial plaintexts. If PT_s contains *n* independent and uniformly distributed bits, it has an average μ=*n*/2 and a variance σ2=*n*/4. Consequently, the correlation between *W* and *P* is expressed as:

$$\rho_{wH} = \frac{E(H(P)W) - E(H(P))E(W)}{\sigma(H(P))\sigma(W)} = \frac{a\sqrt{n}}{\sqrt{na^2 + 4\sigma_b^2}} \tag{5}$$

According to above discussion, the partial key guess is correct, if the intermediate results and the power consumption are correlated, that is the highest correlation coefficient is achieved for that key.

B. Brute Force Attack

The brute force attack is a simple type of attack though requires relatively large effort. The attacker decrypts the cipher text trying every possible key. Assuming the attacker can identify if the decrypted value is the correct ones, the technique requires on average to decrypt half of the total number of possible keys. Hence, for a sufficiently long key the brute force attack becomes impractical. However, the ever-improving performance of modern processing systems quickly ages and weakens the security of AES for a given key size, as shown in Table 1. Simple remedy is to increase key size in par with technical capabilities of a potential attacker. However, in case of resource limited sensor nodes the added memory, processing, and power requirements become prohibitive.

Current practice is to use a 128-bit AES key. Hence, the total number of different keys is equal to 3.4 x 10^{38}. Table 1 illustrates how long it takes to break the AES key using brute force attack. The current results are contrasted with the results from 1995. Additionally, Table 2 shows the decrease in hardware costs per 1000 (millions instructions per second) MIPS [19][20]. Hence, it can be inferred that the security of the AES decreases with time for the same key size.

In conclusion, a simple remedy of increasing key size might increase security against brute-force attacks but for WSNs due to resource limitations in terms of memory, battery power a different approach is needed. The proposed scheme modified architecture of the encryption engine to utilize multiple shorter keys. The security of the proposed scheme increases without need for larger encryption engines. The analytical results are presented in Subsection IV.B.

TABLE 1. COST AND TIME ESTIMATION FOR A BRUTE FORCE ATTACK

Machine cost	Key Search Time in 2009	Key Search Time in 1995
$300M	9.37 x 10^{15} years	4.52 x 10^{23} years
$300K	6.52 x 10^{24} years	5.6 x 10^{33} years
$10K	7.42 x 10^{36} years	Infeasible

TABLE 2. HARDWARE COST OF PROCESSING POWER

Corresponding year	Hardware cost/1000MIPS
1961	$1.1 trillion
1997	$30,000
2007	$0.42

III. MULTI-KEY ENCRYPTION TECHNIQUE

The proposed encryption scheme utilizes multiple keys to encrypt plaintext using AES algorithm. The subsequent blocks of plain text are encrypted using randomly selected keys. In contrast, the standard AES uses a single key to encrypt the whole plaintext. The proposed scheme increases security when compared with the single key approach since complete decryption can be done only with all utilized keys.

Remark: Though the proposed scheme provides uniform security to all data, it is possible to vary the security level (e.g. number of used keys) based on requested protection. In such cases, MKE scheme can be adjusted to use more number of keys for high priority data and fewer keys for low priority data. Also, the routing may vary with requested protection level in order to avoid links with too few keys.

The improved security is achieved with minimal modifications to the AES engine and small increase of complexity and energy. Moreover, the message content is protected even if the intruder compromises few of the shared keys since part of the message will be still protected with the rest of keys. The detailed analysis is presented in Section IV.

The packet is encrypted such that only the destination node can decrypt the data with the corresponding shared keys. The original message is pre-pended with two fields: the first key

ID and a seed, as shown in Fig. 1.

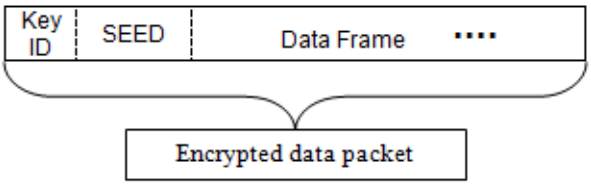

Figure 1. Data payload of a packet

A. Synchronization of Radom Key Sequences

The Key ID and seed fields are utilized for the synchronization purposes. When the destination receives the message it needs to first *synchronize the pseudo random generator* with the source in order to discover the appropriate sequence of keys. Hence, the first block of the message is decrypted using each shared key until the correct key is found, that is when the message's key ID matches the one of the applied key.

When the right key is found for the first block the subsequent blocks are decrypted only once using the correct sequence of keys selected by the pseudo random generator. Also, the same random generator is applied for the subsequent messages in order to reduce the overhead of trying all the shared keys for the first block of every message.

Moreover, the loss of synchronization between transmitter and receiver, for example due to lost packets and retransmission, can easily be detected. The receiver node compares 'key id' and 'seed' fields from each message with the random generator sequence. The match confirms a successful synchronization and the rest of message can be decrypted. Otherwise, the synchronization procedure is repeated using the first block, as described above.

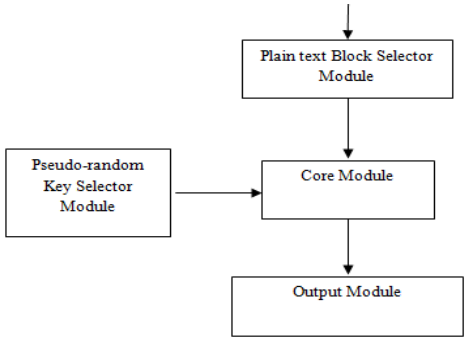

Figure 2. Encryption Module

The encryption process of the proposed MKE scheme is illustrated in Fig. 2. The plaintext data blocks (e.g. 128, 156 or 192 bits long) are encrypted with the key chosen by the pseudo-random key selector module. The first encrypted block includes the *key id and random generator's seed* that are necessary to recreate the same key sequence at both transmitter and receiver. Moreover, these values are *selected randomly and never transmitted in plain*. Hence, the attacker is unable to acquire them.

Decryption process is shown in Fig. 3. Note that an additional processing overhead is introduced when decrypting the first block since the correct key has to be found. Once the first block is decrypted the subsequent blocks are decrypted using keys dictated by the pseudo-random generator. Moreover, in order to minimize the initial overhead, the encoding of the subsequent packets can continue using the same pseudo-random sequence. Then the decoder when using the same sequence will start the key search using the correct one. Hence, no processing overhead will be incurred.

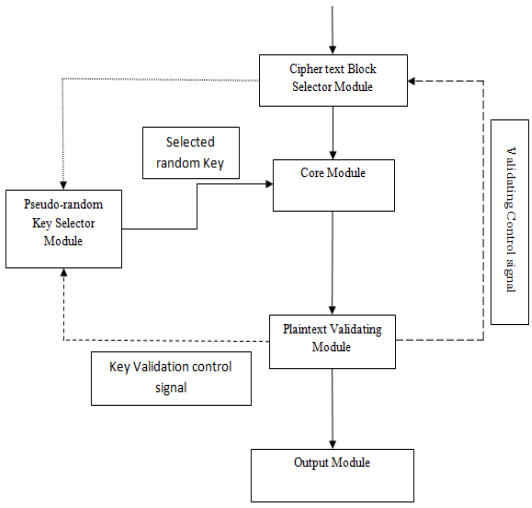

Figure 3. Decryption Module

The existing approaches require significant modification to the encryption engines [16], or use less energy efficient designs [15]. In contrast, the proposed scheme can utilize the S-Boxes with the lowest power consumption that increases energy-efficiency when compared to other schemes [15][16]. Moreover, the security is increased for the same key and cipher size. Next, the analytical and simulation results are presented.

IV. Theoretical Security Analysis of the Proposed MKE Scheme

The security of the proposed multi-key encryption (MKE) scheme is analyzed against the CPA attack and also the brute-force attack. The quantitative results and improvements are presented in next subsections.

A. CPA Attack

Assume that CPA attack on 8 MSBs (Most Significant Bit) of the registers shown in Fig. 4. The key used for this operation is the original key for encryption, N random plaintexts and one fixed but random key have been chosen for the experiment, the total number of bit-changes between the previous and the current values of these M MSBs of the register for the initial key addition are calculated.

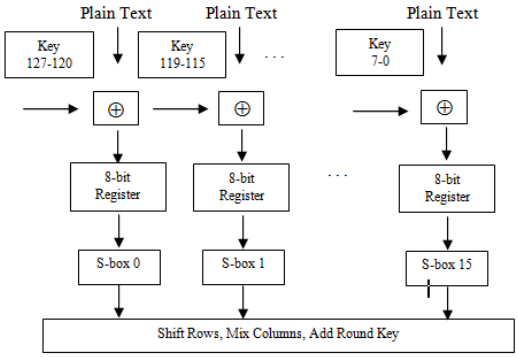

Figure 4: Simplified diagram of AES algorithm

By employing the proposed MKE scheme the correlation between power values with respect to cipher key is broken and correlation coefficients values are reduced. The proposed engine architecture is shown in Figs. 5 and 6. If an attacker employs the CPA method to compromise the secret key, the exact key cannot be decoded since the correlation between the power and the secret key is broken.

Let us consider scenario where two keys are used. Their partial keys are ks1 and ks2 respectively. Then corresponding power values are expressed as

$$P_1 = PT_S \oplus K_{S1} \tag{6}$$

$$P_2 = PT_S \oplus K_{S2}$$

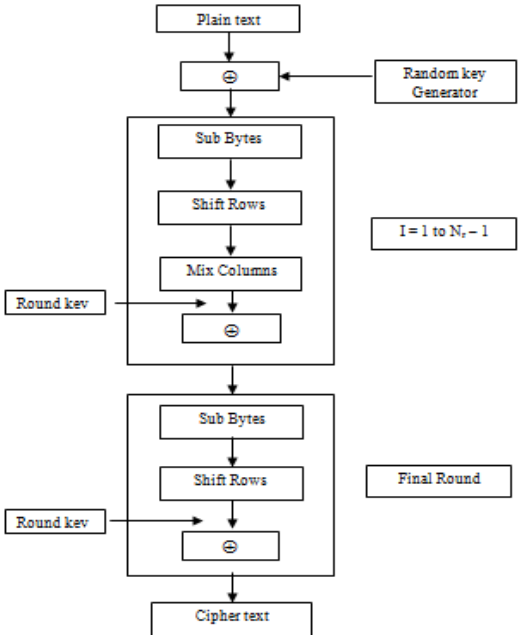

Figure 5. Modified Architecture for AES Encryption Algorithm

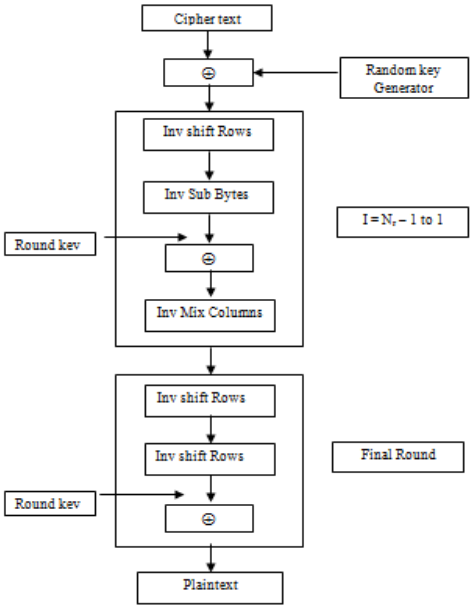

Figure 6. Modified Architecture for AES Encryption

The correlation can be calculated from (5). When the attacker encrypts N plaintexts, it can calculate correlation between the corresponding power usage for each plaintext and the simulated power consumption of the same plaintexts. The attacker needs to calculate the correlation for every key combination. For a particular key the correlation values are summed up for all plaintexts:

$$\rho_k = \rho_{wp\,1} + \rho_{wp\,2} + ... + \rho_{wpN} \cong N \cdot \rho_{wp} \tag{7}$$

where p1,p2,…,pN denote the keys used with each plaintexts.

Theorem 1: The correlation factor value decreases with number of employed keys.

Proof: The claim of the Theorem 1 is analyzed for two cases: with two and *M* keys. In short, the combined correlation factor obtained for multiple keys is smaller compared to the correlation factor obtained when employing one key over *N* plaintexts.

1) Case 1: Encryption using two keys

First, a case with two keys used for encryption is considered. The keys are randomly used to encrypt *N* subsequent message blocks. Let n_{k1} and n_{k2} denotes the number of blocks encrypted with the key *k*1 and *k*2 respectively. Now the correlation factor is equal to

$$\rho_k = n_{K1} \cdot \rho_{wp1} + n_{K2} \cdot \rho_{wp2} = n_{K1} \cdot \rho_{wp1} + (N - n_{K1}) \cdot \rho_{wp2} \tag{8}$$

where ρ_{wp1} and ρ_{wp2} are the correlation factors corresponding to keys k1 and k2. The difference in correlation factor value due to implementation of multiple keys can be expressed as:

$$\Delta\rho_k = N \cdot \rho_{wp1} - \left[n_{k1} \cdot \rho_{wp1} + (N - n_{k1}) \cdot \rho_{wp2} \right] \quad (9)$$

where $N > n_{k1}$, and $\Delta\rho_k$ is the difference between correlation factors when single key and multiple keys are used. If the difference factor is positive, then correlation factors get altered such that exact key used cannot be found.

For example, in Fig. 7, the key at I=156 has the highest correlation coefficient ρ_{wp1} averaged over *N* rounds. If a different key is used for encryption and averaged over *N* rounds the correlation for I=156, denoted as ρ_{wp2}, is always lesser than the ρ_{wp1} from the first encryption instance since second key will have its peak at other time instance. So, using both keys subsequently decreases the overall correlation coefficient value for *N* rounds.

Overall, the following conditions hold: $N > n_{k1}$ and $\rho_{wp1} > \rho_{wp2}$. Consequently, the first difference of correlation factor $\Delta\rho_k$ is always positive which shows that the final correlation values decrease if two keys are used.

2) Case 2: General formulation for M keys

Following the analysis from the case 1, the correlation factor difference for *M* keys can be expressed as:

$$\Delta\rho_k = N \cdot \rho_{wp1} - \left[n_{k1} \cdot \rho_{wp1} + n_{k2} \cdot \rho_{wp2} + \ldots + n_{kM} \cdot \rho_{wpM} \right] \quad (10)$$

where $N = \sum_{i=1}^{M} n_{ki}$. The ρ_{wp1} is the highest correlation when compared to the other M-1 keys. Hence, the $\Delta\rho_k$ difference is always positive.

B. Brute Force Attack

The proposed scheme uses random sequence of *N* keys to encrypt block plaintext using AES algorithm. Hence, if *k* keys out of *N* are compromised, the attacker does have to have sufficient information to determine the sequence used to encrypt the message blocks. In this technique the keys are selected randomly and attacker uses every key out of the known *k* keys to decrypt the cipher text for every cipher text block. The number of trials increases with number of blocks in the message and number of known keys, *k.* Moreover, only a fraction of a message is decrypted, as he knows only few out of total number of keys.

In order to find the correct 128-bit key the attacker needs to consider all 2^{128} possible keys. In terms of probability the probability of selecting a correct key is equal to $P = 2^{-128}$. The probability of finding the particular key by brute force, P(T), can be expressed as

$$P(T) = (1-P)^T P \quad (11)$$

In the MKE scheme, each block is encrypted using randomly selected key out of the N shared ones. Hence, even if intruder compromises few keys, it is not possible to decode whole message until every key used for encryption is compromised.

Consider an attacker who compromised *k* out of *N* keys. Now, the attacker has to select a key each time from the group of known *k* keys for every cipher text block, then the probability of selection a key is

$$P(k) = \binom{k}{1} \Big/ \binom{N}{1} = (k/N) \quad (12)$$

Probability of finding exact key from known *k* keys out of *N* keys until correct key is found is given as:

$$P_{correct} = (1 - P(k))^L \quad (13)$$

where *L* denotes the number of trials to be performed for each message block to decrypt it until attacker finds the correct key from the known group of keys and it has a maximum value of *k* (i.e., total number of keys known by the intruder).

The final probability becomes lower than *P*(T) in (11), where only one key is used. The final probability P_{final}, is the product of *P*(k) and $P_{correct}$ and is given by

$$P_{final} = P(k) \cdot P_{correct} \quad (14)$$

where P_{final} denotes final probability of finding the correct key for each plaintext block and it denotes increase in security, even if some of the keys out of *k* keys are compromised attacker has to check each time for the correctness of the key, which results in time consumption as well as only partial detection of the whole message.

Remark: It has to be noted that the proper design of the random generator (RNG) is required to ensure that attacker cannot finding the order of the keys and thus predict which one is used for each block. A weak RNGs are known to undermine otherwise strong security mechanisms [22] [23].

The percentage of total message decrypted is proportional to number of *k* keys known to intruder out of *N* keys

$$P(percent) \propto k/N \quad (15)$$

The proposed scheme increases the security by using multiple-keys. The security increases proportionally to the number of used keys.

V. Simulation Results

Simulation has been conducted in Matlab to analyze the performance of the proposed MKE schemes in thwarting the CPA attack. Attack on a partial 8-bit key is studied. However, the results can be easily expanded to the general case of *L*-bit key. It is assumed that the attacker simulates encryption engine for the *g*=256 keys (for 8-bit partial key/register). The attacker collects the power signatures for *N*=1000 plaintexts,

each plaintext representing a block of message data. Then, the correlation factors for (N x g) cases are calculated and averaged for the N=1000 plaintexts. The results are presented based on reference to the respective keys. The key with highest correlation is expected to be the secret key used for the actual encryption. The proposed technique uses multiple keys subsequently for N plaintexts, which results in the reduced correlation.

Remark: In CPA attack, the complexity of attack increases with number of plaintexts, N. Also, the confidence of finding the correct key increases with number of plaintexts, N, since the individual correlation coefficients of N plaintexts are being summed up. However, the confidence saturates at some level due to noise in measurements and quality of the power correlation for the given circuitry. Furthermore, in case of the MKE scheme, the signal correlation reduces when compared to a single key scheme since the random selection of keys breaks the correlation between power consumption and the key. This tradeoff between increasing complexity and saturating confidence leads to a practically justifiable size of the plaintext set, N. Henceforth, the N=1000 value is typically considered for this experiment[15][16].

Fig. 7 illustrates the raw correlation factors for a single key scenario [16]. It exhibits a high correlation with the secret key that was used during the actual encryption. In contrast, for MKE scheme the correlation between secret key and correlation coefficients decrease with number of used keys, as shown in Figs. 8 and 9. The main reason is that the subsequent blocks are encrypted with different keys, which power correlation with the simulated results for a single key decreases.

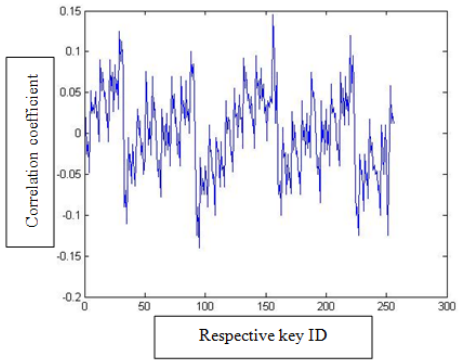

Figure 7. The correlation coefficients for one key [17]

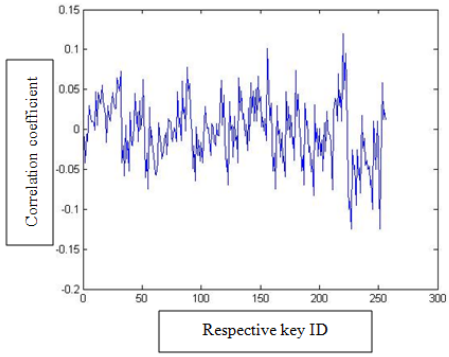

Figure 8. The correlation coefficients when two keys are used

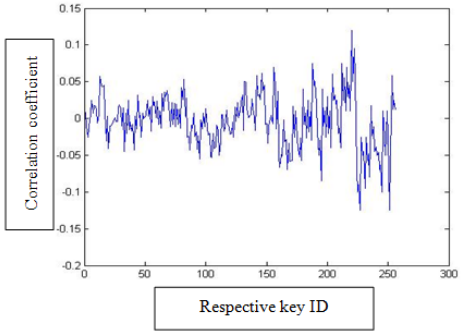

Figure 9. The correlation coefficients when five keys are used

For the two key case, as shown in Fig. 8, the highest correlation occurs for I=256. This is different from single-key case presented in Fig. 7 since the MKE technique randomly select keys that reduces overall correlation factor averaged over N blocks of data. In this case, two keys are randomly chosen to encrypt N=1000 plaintexts or blocks of data. Assuming that attacker have no idea of MKE scheme, correlates N*g cases (g=256) for each of the g keys with power consumption values of N=1000 plaintexts (uses two keys for encryption). It results in lower correlation values with only one of the key among the two having the highest correlation (one key match with original key in 'g' keys and other key negating). Moreover, the equation (10) can be interpreted as an average of the correlations of the used keys. Hence, the overall correlation reduces with the number of used keys since the average over the whole domain space is equal to zero. Next, the proposed scheme is analyzed from energy-efficiency point of view.

A. Energy Efficiency Analysis

The energy efficiency is measured as the energy consumed by encryption engine to transmit a given message. The analysis assumes the AES-based engine that operates on 128-bit blocks (128-bit key size). In general, the AES algorithm can be implemented using various S-Box designs, for example LUT, SOP, etc. These S-boxes have different energy efficiency and power correlation. In the scheme proposed in [16] the authors utilize a combination of several different designs in order to reduce power correlation. However, the power consumption increases since the scheme uses S-boxes

with a high-energy consumption.

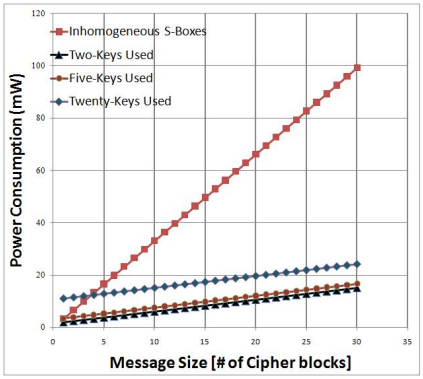

Figure 10. Comparison of power consumption for multi-key encryption (MKE) scheme

In contrast, the proposed MKE scheme improves the energy-efficiency compared to the technique in [16] since it can employ the S-box design with lowest power consumption, as shown in Fig. 10. The comparison between power consumption of the technique in [16] and the proposed scheme illustrates that even with the initial synchronization overhead (i.e. finding the first key in sequence) the proposed scheme outperforms the other scheme. Moreover, the proposed scheme scales better with size of the message since it allows using the S-boxes with the lowest power footprint.

VI. Conclusion

The proposed MKE technique has been shown to improve the security of AES algorithm against CPA attack while minimizing power consumption. Additionally, it improves security of AES against brute-force attacks.

In the case of a CPS attach, the MKE scheme thwarts a CPA type attack by reducing correlation between power consumption and the key. The proposed scheme, using 5 keys can decrease the correlation by 80% between power and data. Also, the energy consumption of the proposed MKE scheme reduces by over 70% when compared to the inhomogeneous S-boxes scheme while maintaining high security.

Moreover, when a single key is compromised only a small fraction of the message is compromised thus increasing security against brute-force attacks. Consequently, all the keys have to be compromised before the link becomes unsecure.

VII. References

[1] Gupta, A.; Kuri, J," Deterministic schemes for key distribution in wireless sensor networks", Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference,6-10 Jan. 2008

[2] Park, E.C.; Blake, I.F.," Reducing Communication Overhead of Key Distribution Schemes for Wireless Sensor Networks", Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference,13-16 Aug 2007.

[3] L. Eschenauer and V. D. Gligor, "A key management scheme for distributed sensor networks," in Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS '02), pp. 41–47, Washington, DC, USA, November 2002.

[4] H. Chan, A. Perrig, and D. Song, "Randomkey predistribution schemes for sensor networks," in Proceedings of IEEE Symposium on Security and Privacy, pp. 197–213, Oakland, Calif, USA, May 2003.

[5] W. Du, J. Deng, Y. S. Han, P. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," ACM Transactions on Information and System Security, vol. 8, no. 2, pp. 228–258, 2005.

[6] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," ACM Transactions on Information and System Security, vol. 8, no. 1, pp. 41–77, 2005.

[7] R. Blom, "An optimal class of symmetric key generation systems," in Proceedings of the Workshop on the Theory and Application of CryptographicTechniques (EUROCRYPT '84), pp. 335–338, Paris, France, April 1985.

[8] Zhihong Liu, Jianfeng Ma , Qiping Huang and sangjae moon " A pairwise key Establishment scheme for Heterogeneous sensor networks", 2008 ACM(Hetersanet08, may 30,2008, Hong Kong SAR,china).

[9] Sajid hussain, Firdous kausar and Ashraf Masood " An Efficient key distribution scheme for Heterogeneous sensor networks" 2007 ACM(IWCMC'07,August 12-16,2007,Hawaii,USA).

[10] S. Basagni, K. Herrin, D. Bruschi and E. Rosti, Secure pebblenets, in Proceedings of the 2001 ACM International Symposium on Mobile ad hoc networking & computing, Long Beach, CA, USA (2001) pp. 156–163. ACM Press.

[11] R. Di Pietro, L.V. Mancini and S. Jajodia, Providing secrecy in key management protocols for large wireless sensors networks, Journal of AdHoc Networks, 1(4) (2003) 455–468.

[12] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis", Advances in Cryptology -Crypto 1999, LNCS 1666, pp.388-397, Springer-Verlag, 1999.

[13] Siddika Berna Örs, Frank K. Gurkaynak, Elisabeth Oswald, and Bart Preneel. "Power-Analysis Attack on an ASIC AES Implementation", In Proceedings International Conference on Information Technology-ITCC 2004, Las Vegas, USA, Proceedings, 2004.

[14] Norbert Pramstaller, Elisabeth Oswald, Stefan Mangard, et al., "A Masked AES ASIC Implementation", in Proceedings of Austrochip 2004, Villach, Austria, Oct. 8, 2004.

[15] Zheng Zhoxia,Zou Xuecheng, Liu Zhenglin, Chen Yicheng "Secure AES Coprocessor against Power Analysis for Wireless Sensor Networks",Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference ,sept 2007 IEEE.

[16] Zheng Zhoxia,Zou Xuecheng, Liu Zhenglin, Chen Yicheng "Security Analysis and Optimization of AES S-boxes Against CPA attack in Wireless Sensor Network" , Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference ,sept 2007 IEEE.

[17] Advanced Encryption Standard (AES)", Federal Information Processing Standards Publication 197, Nov. 2001.

[18] D.Coppersmith, D.B.Johnson and S.M.Matyas "A Proposed mode for triple-DES encryption", IBM J.RES Develop. Vol 40, No.2,March 1996.

[19] IBM 1961 BRL Report

[20] Halfill, Tom R. (2006-10-10). "204101.qxd Ambric's New Parallel Processor". Microprocessor Report (Reed Electronics Group): 1–9.

[21] S. Guilley, P. Hoogvorst, R. Pacalet. Differential Power Analysis Model and some Results. In proceedings of CARDIS 2004, Kluwer Academic Publishers, pp. 127-142, 2004.

[22] John Kelsey, Bruce Schneier, David Wagner, and Chris Hall S. Vaudenay, "Cryptanalytic attacks on Pseudorandom key generators", FSE'98, LNCS 1372, pp. 168{188},Springer-Verlag Berlin Heidelberg 1998.

[23] M. Dichtl,"How to Predict the Output of a Hardware Random Number Generator", Proceedings of the Workshop on Cryptographic Hardwareand Embedded Systems – CHES 2003, LNCS 2779, pages 181–188, Springer-Verlag Berlin Heidelberg.