

Towards Secure and Practical MACs for Body Sensor Networks

Zheng Gong¹, Pieter Hartel¹, Svetla Nikova^{1,2}, and Bo Zhu³

¹ Faculty of EWI, University of Twente, The Netherlands

{z.gong, pieter.hartel, s.nikova}@utwente.nl

² Dept. ESAT/SCD-COSIC, Katholieke Universiteit Leuven, Belgium

³ Dept. Computer Science and Engineering, Shanghai Jiaotong University, China
zhuho03@gmail.com

Abstract. In this paper, some practical problems with the Message Authentication Codes (MACs), which are suggested in the current security architectures for wireless sensor network (WSN), are reconsidered. The analysis exploits the fact that the recommended MACs for WSN, e.g., TinySec (CBC-MAC), MiniSec (OCB-MAC), and SenSec (XCBC-MAC), are not exactly suitable for body sensor network (BSN). Particularly a dedicated attack is elaborated on the XCBC-MAC. Considering the hardware limitations of BSN, we propose a tunable lightweight MAC based on the PRESENT block cipher, which is named TuLP. A 128-bit variant TuLP-128 is proposed for a higher resistance against internal collisions. Compared to the existing schemes, our lightweight MACs are time and resource efficient on hardware-constrained devices.

1 Introduction

Traditional wireless sensor networks (WSNs) are used to collect public information in the environment, such as temperature, humidity, fire alarm, etc. Body sensor network (BSN, also called wireless medical sensor network) [33], which can be developed from WSN, is a key technology for long term monitoring of biological events or any abnormal condition of patients for realizing the Ambient Assisted Living (AAL) vision [1]. Since monitored health data from a person with BSN will be a part of personal Electronic Health Record (EHR), a higher level of assessment and protection is required for BSN communications. The existing EHR standards (ISO 27001, 27799, openEHR/ISO 18308, etc.) oblige BSN to be secured with strong cryptography. However, strong cryptography entails more resources. Since BSN nodes are either worn or implanted by a patient, the power consumption should be low to minimize radiation and maximize durability. Moreover, BSN sensors also have limited computational ability and memory, typically with a low-end CPU and RAM in KBytes level. These factors are important not only in the implantable but also in the external sensor settings because they determine how “hidden” and “pervasive” the sensors are.

Considering the highly constrained resources that a BSN node can have, a better trade-off has to be found such that the security is maximized, while minimizing the resource requirement. Unfortunately, because of the heterogeneity of BSN, the secure protocols for static networks might not applicable for BSN. Also the methods proposed

for *ad hoc* networks such as asymmetric cryptography techniques would be costly for BSN applications. Due to the constraints in power consumption and computational ability, it remains a great challenge to design secure and practical cryptographic primitives which are both time and resource efficient for BSN applications.

To ensure the authenticity and integrity of WSN communication, security protocols via different Message Authentication Codes (MACs, different from the term “Medium Access Control”) are proposed. MAC is a symmetric-key primitive that inputs a key-message pair to produce a unique tag. The integrity and the authenticity of the message are protected by the tag and the key respectively. One widely used method is the Security Protocol for Sensor Networks (SPINS) [29], which consists of μ TESLA (micro version of the Timed, Efficient, Streaming, Loss-tolerant Authentication) and SNEP (Secure Network Encryption Protocol) for broadcasting messages. Following SPINS, many lightweight security architectures have been proposed for WSN, e.g., TinySec [21], SenSec [24] and MiniSec [25]. All these architectures considered which MAC will be suitable in the WSN packet/message authentication. For instance, TinySec and MiniSec recommend the well-known CBC-MAC [19] and OCB-MAC [30] respectively, whilst SenSec uses a novel scheme called XCBC-MAC [24]. All the recommended MACs are based on the operation modes of block cipher, and suggest 32-bit length tag for WSN. In contrast, since dedicated hash functions (such as MD5 and SHA-1) are primarily designed to be collision resistant for preventing forgery of digitally signed documents, it was exploited that MACs based on hash functions (e.g., HMAC [16]) might be less competitive than block-cipher-based ones for highly constrained devices [10]. Nevertheless, it is recognized by the BSN research community that authentication in BSN protocols is usually for short messages in network processing [33]. This property implies that the candidates of MACs, which focus more on the one-wayness than on the collision-resistance, will be more practical for BSN applications.

Since typical BSN nodes have limited resources, an appropriate security level should be imposed to realize authenticity and confidentiality in applications. Intuitively, 32-bit security level for WSN is not suitable even for the one-wayness of the transmitted data in BSN. As a comparable case for sensitive data authenticity, the authentication of Electronic Funds Transfer in the US Federal Reserve System uses a 64-bit CBC-MAC, and additionally a secret value for IV is daily changed and synchronized by the member banks. In other applications, certain authorities even recommended to implement a MAC with a longer length of 128-bit. Although an appropriate security level for BSN applications will be ensured case by case, a 64-bit security bound is widely-accepted for resisting sensible threats in such hardware-limited devices. As power and RAM are normally the most constrained resources on a BSN node, the design of a MAC should consider applicable trade-offs towards time and resource efficient in practice.

The contributions of this work are three-fold. Firstly, we describe some practical problems of the MACs recommended in popular security architectures for WSN, such as TinySec (CBC-MAC), MiniSec (OCB-MAC) and SenSec (XCBC-MAC). In particular, we demonstrate an existential forgery attack on XCBC-MAC, which implies that the authenticity of SenSec is broken. Secondly, a performance comparison is presented on efficient MACs from different design principles, e.g., CBC-MAC, OCB-MAC, ALPHA-MAC [12]. Thirdly, taking into account the requirements for authenticity in BSN, we

propose a tunable lightweight MAC based on the PRESENT block cipher [9], which is named TuLP. The structure of TuLP is inspired by the generic construction ALRED [12]. A 128-bit variant TuLP-128 is proposed for the higher resistance against internal collisions. Compared to the existing schemes, our lightweight MACs show a better performance on MICAZ node with less memory costs, and also energy-efficient in the level of gate equivalents.

The remainder of this paper is organized as follows. Section 2 describes some definitions and notions which will be used throughout the paper. The problems with the MACs recommended in the proposed security architectures for WSN are described in Section 3. Section 4 gives a performance comparison of some efficient MACs for BSN authenticity. The designs of TuLP and TuLP-128 follow in Section 5 along with a detailed analysis of the security and the performance. Section 6 concludes the paper.

2 Preliminaries

Here we review some definitions and primitives which will be used in the following sections. Exclusive-or (xor) will be denoted by \oplus . A message $M = a||b$ denotes the concatenation of two strings a and b . Let \mathcal{M} and \mathcal{K} be the message and key spaces respectively.

ALRED. The ALRED construction is a generic MAC design introduced by Daemen and Rijmen [12]. The ALRED construction consists of the following steps:

1. **Initialization:** Fill the state with an all-zero block and encrypt it with a full encryption E with an authentication key k .
2. **Chaining:** For each message, iteratively perform an *injection layout* to map the bits of the message to the same dimensions as a sequence of r round keys of E . Then apply a sequence of r times round function of E to the state by using the output of the injection layout as the round keys.
3. **Finalization:** Apply a full encryption E with the authentication key k to the final state. The tag is the first ℓ_m bits of the output.

By using AES as the underlying block cipher, Daemen and Rijmen also presented two paradigms called ALPHA-MAC [12] and Pelican [13] based on ALRED. Recently, many papers exploited that ALPHA-MAC and Pelican might be threatened under the internal collisions [18], the side-channel attack [5] and the impossible differential analysis [32]. We note that all those cryptanalyses are based on the internal structures of ALPHA-MAC and Pelican, which do not endanger the security of ALRED.

PRESENT. At CHES 2007, Bogdanov *et al.* proposed an ultra-lightweight block cipher which is named PRESENT [9]. PRESENT is an example of an SP-network and consists of 31 rounds. The block length is 64 bits and two key lengths of 80 and 128 bits are supported. The hardware requirements for PRESENT are competitive. Using the *Virtual Silicon* (VST) standard cell library based on *UMC L180 0.18 μ m 1P6M Logic Process* (UMCL18G212T3), PRESENT-80 and PRESENT-128 are estimated to require 1570 and 1886 gate equivalents, respectively [9]. Since Bogdanov *et al.* do not expect the

128-bit key version to be used until a rigorous analysis is given, the term PRESENT means 80-bit key version in hereafter.

Further details about the specification of PRESENT can be found in Bogdanov *et al.* [9], including basic results of the differential and linear cryptanalyses, which can be summarized as follows.

Theorem 1. *Any five-round differential characteristic of PRESENT has a minimum of 10 active S-boxes.*

Theorem 2. *Let ϵ_{4R} be the maximal bias of a linear approximation of four rounds of PRESENT. Then $\epsilon_{4R} \leq 2^{-7}$.*

Based on PRESENT, Bogdanov *et al.* [10] propose some low-energy block-cipher-based hash functions (e.g., single and double block length construction DM-PRESENT and H-PRESENT respectively) which are more practical than dedicated or AES-based hash functions on highly constrained devices, such as RFID tags.

Recently, many cryptanalysis results have been given on the PRESENT block cipher. Wang [31] presents a differential attack on 16-round PRESENT with the complexities of about 2^{64} chosen plaintexts, 2^{32} 6-bit counters, and 2^{64} memory accesses. Collard and Standaert [11] show a statistical saturation attack against 24-round PRESENT. The saturation attack [11] depends on a simplified key schedule algorithm such that the same subkey should be used in each round. Özen *et al.* [27] provide a related-key rectangle attack on 17-round PRESENT-128. Albrecht and Cid [2] present an algebraic differential attack on 19-round PRESENT-128. However the known attacks on PRESENT with 80-bit keys, without any simplification, so far are bounded with 16 rounds [31].

3 Problems with the MACs Recommended for WSN

For ensuring the security of the communication in WSN, many schemes have been proposed for the different layers of WSN. Basically, data link layer security is fundamental for other security properties in the higher layers, e.g., secure routing in network layer and non-repudiation in application layer. In practice, there exist three widely-cited schemes for the security of data link layer, which are TinySec [21], SenSec [24], and MiniSec [25]. For confidentiality, all the three schemes suggest using a lightweight block cipher for data encryption. But for authenticity, three totally different MAC functions are recommended, which are claimed to be suitable for WSN. In this section, we will give a comparative analysis of the three recommended MAC functions in the three schemes [21,24,25].

CBC-MAC. In TinySec [21], Karlof *et al.* suggest to use CBC-MAC [19] as the underlying MAC function. CBC-MAC uses a cipher block chaining construction for computing and verifying MACs. The first advantage of CBC-MAC is simplicity, as it relies on a block cipher which minimizes the number of cryptographic primitives that must be implemented on BSN nodes with a limited memory. For BSN applications, the disadvantage of CBC-MAC is that independent keys should be used for encryption and authentication. Furthermore, the one-key CBC-MAC construction [4] is not secure for

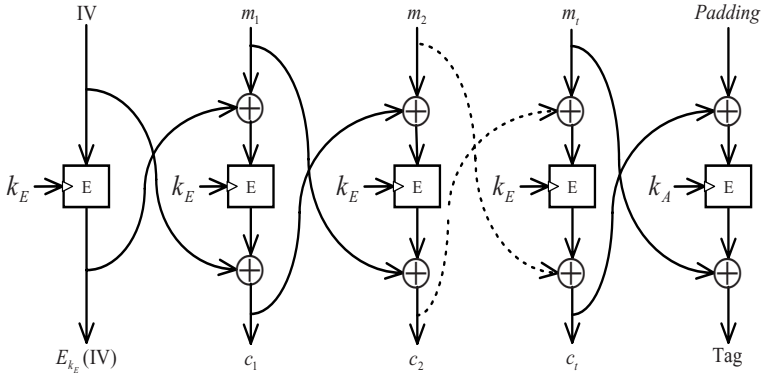


Fig. 1. The XCBC algorithm proposed in SenSec [24]

arbitrary length messages, which allows adversaries can forge a tag for certain messages. To preserve the provable security for arbitrary length messages, a variant of CBC-MAC uses three different keys for the authentication [7]. Although the three-key construction solves the arbitrary length message problem and avoids unnecessary message padding, it raises another typical risk with respect to the key management in BSN. Compared to the one-key construction, the extra keys will impose a heavy burden on key generation, distribution and storage. The risk of the key management indicates that a *provably secure* CBC-MAC might be less practical for BSN applications.

XCBC-MAC. The XCBC-MAC algorithm proposed by Li *et al.* [24] is part of the authenticated encryption mode for SenSec. Let k_A and k_E be the authentication key and the encryption key, respectively. Let message $M = m_1||m_2||\dots||m_t$. Figure 1 depicts the construction of XCBC-MAC. In general, the XCBC-MAC algorithm can be viewed as a variant of the two-key CBC mode. Unfortunately, we have found a practical existential forgery on XCBC-MAC by implementing a chosen-message attack. One can easily build two different messages with the same tag under the XCBC mode. The forgery can be described in the following steps:

1. First, adversary \mathcal{A} obtains IV, $E_{k_E}(IV)$ from the first block of any former ciphertext under k_E .
2. Next, \mathcal{A} requests the encryptions on the two different blocks $E_{k_E}(IV) \oplus m_1$ and $E_{k_E}(IV) \oplus m'_1$ in the XCBC mode. The ciphers will be $E_{k_E}(m_1) \oplus IV$ and $E_{k_E}(m'_1) \oplus IV$. \mathcal{A} obtains $E_{k_E}(m_1)$ and $E_{k_E}(m'_1)$ by xoring the ciphers with IV.
3. Finally, \mathcal{A} arbitrarily selects a message M' , and then outputs two different messages M_1, M_2 , where $M_1 = E_{k_E}(IV) \oplus m_1||E_{k_E}(m_1)||0||M'$ and $M_2 = E_{k_E}(IV) \oplus m'_1||E_{k_E}(m'_1)||0||M'$.

It is easy to see that two different prefixes $E_{k_E}(IV) \oplus m_1||E_{k_E}(m_1)||0$ and $E_{k_E}(IV) \oplus m'_1||E_{k_E}(m'_1)||0$ will produce the same zero output to the next step. Thus the two different messages M_1 and M_2 will have the same tag. The attack is difficult to detect since IV is a public-known value and the prefixes are computationally indistinguishable from a randomized query. Although the above attack can be avoided by using a

one-time randomized IV, this assumption is impractical in WSN and BSN. If IV can frequently be updated, all nodes should immediately synchronize the value. Otherwise the receiver cannot correctly decrypt any packet from the sender. Since synchronization is costly in sensor networks, it is impractical for an IV to be distributed just for one-time usage. Due to the above analysis, the XCBC-MAC algorithm proposed in SenSec [24] is insecure under the chosen message attack and should be abandoned in any circumstance of WSN/BSN authentication.

OCB-MAC. In MiniSec [25], Luk *et al.* suggest using the OCB mode [30], which is an efficient authenticated encryption scheme, as the MAC function for message authenticity and integrity. Since its publication OCB has received some attention, but little cryptanalysis. We believe this has two reasons. First, the security proof of OCB [30] seems to imply that cryptanalysis is useless. The proof is quite complicated and analysis of the proof details is restricted to those people who are well-versed in formal proof techniques. Second, the OCB mode has been patented. There is a significant cost, both directly and indirectly, associated with using a patented algorithm. The last reason is the main reason for the lack of rigorous cryptanalysis. Spending time on OCB will only help the patent-holders to sell their licenses without any further compensation to the cryptanalyst. Moreover, Ferguson also presents a collision attack on OCB with arbitrary length messages [15]. To keep adequate authentication security of OCB, one has to limit the amount of data that the MAC algorithm processes. Since the offset values used in OCB require extra time/memory costs with respect to the message length, the area and the power consumption will be increased for the computation and storage. The above reasons are relevant to real-life applications on BSN, and cast doubts on the wisdom of using OCB.

4 A Comparison of Some Practical MACs for BSN

We have shown that the MAC functions proposed for WSN in the literature are not exactly suitable for BSN. Many different MAC Functions have been proposed in the past decades. Driven by the highly constrained resources of BSN node, the performance and security of those candidates should be rigorously examined before they are implemented. Basically, there are three approaches towards designing MAC functions. The first is to design a new primitive from scratch, such as UMAC [6]. The second is to define a new mode of operation for existing primitives. Such as variants of encryption modes of block ciphers: CBC-MAC [19] and OCB-MAC [30]; Or variants mode of hash functions: HMAC/NMAC [3,16]. The third approach, which can be viewed as a hybrid of the first and the second approach, is to design new MAC functions using components of existing primitives, such as ALPHA-MAC [12].

Based on the security and performance requirements of BSN, we will give a detailed comparison of some popular MAC candidates, which are claimed to be efficient from the three different approaches. To be fair, all MACs based on block cipher use AES-128 as the underlying block cipher, as well as input messages can be of arbitrary length. The timing of the keysetup and the message processing are estimated from the performance data given by the NESSIE consortium [26] (Pentium III/Linux Platform), such that the

Table 1. The comparison of some practical MAC functions

	CBC-MAC [19]	OCB-MAC [30]	ALPHA-MAC [12]	HMAC (SHA-1) [16]
Based on	cipher mode	cipher mode	AES components	hash mode
Keysetup	616	644	1032	1346
Finalization	1440	1444	416	3351
Message processing	26	30	10.6	15
Area in GE (estimate)	4764	6812	4424	8120 [14]

message processing time is measured in cycles/byte, while the keysetup and keysetup + finalization are measured in cycles. The area in *gate equivalents* (GE) can be calculated from two parts: the area of the underlying component or primitive, and the area for internal operations and storages. In order to compare the area requirements independently it is common to state the area in GE, where one GE is equal to the area which is required by two-input NAND gate with the lowest driving strength of the appropriate technology [28]. By following the same method [10,14], we also use the *Virtual Silicon* (VST) standard cell library based on *UMC L180 0.18 μ m 1P6M Logic Process* (UMCL18G212T3) to estimate each area in GE of the candidates. According to the related experiments [14], the area for AES-128 encryption is estimated to be 3400 GE, as well as 64-bit storing and exclusive-or require 512 GE and 170 GE, respectively.

For chips built with CMOS technology, the power consumption is the sum of two parts: the static and the dynamic costs. The static part is roughly proportional to the area, namely the larger size of the chip the larger energy costs, whilst the dynamic part is proportional to the operating frequency. For the devices with a lower operating frequency, the static power consumption is the most significant. For this reason, the area of gate equivalents is often used as a simplified benchmark for energy efficiency. The comparison in Table 1 shows that ALPHA-MAC has merits on both of the message processing speed and the area of GE, which indicates that one could also build a time and energy efficient MAC from the ALRED construction by using a lightweight block cipher.

5 Two New Lightweight MACs from ALRED

In this section, we will propose a tunable lightweight MAC based on PRESENT, which is named TuLP. To raise the security bound of resisting internal collisions, we will also give a wide-pipe version of TuLP, which is called TuLP-128. Both of our schemes use the experiences of ALPHA-MAC [12] and Pelican [13]. Next, the security of our schemes will be analyzed. Finally, the performance of our lightweight schemes will be given. Compared to the results in Table 2, our new MAC functions are time-efficient with less memory usage, and also energy-efficient in the number of gate equivalents.

5.1 TuLP and TuLP-128

By using the round function of PRESENT [9], first a new MAC function TuLP is built from a modification of the ALRED construction. TuLP is a lightweight MAC function with an 80-bit key length at maximum and 64-bit block length, which consists of the following steps:

1. **Padding.** Let k be an authentication key such that $|k| \leq 80$ bits. If $|k|$ is less than 80 bits, it should be iteratively padded with 1 and 0 as 10101 \dots . First pad M with $\lambda(M, k)$ where $\lambda(M, k)$ returns the concatenation of bitwise lengths of M and k . Then pad the concatenated string to a multiple of 64 bits, e.g., appending a single bit 1 followed by necessary d bits 0. Finally Split the result $pad(M)$ into 64-bit blocks $m_1, m_2, \dots, m_t, t = \frac{|pad(M)|}{64}$, such that

$$pad(M) = M || \lambda(M, k) || 10^d.$$

2. **Initialization.** Apply one full-round PRESENT encryption E to the initial value IV with the (padded) authentication key k , then obtain $s_0 = E_k(IV)$ as the initial state.
3. **Compression.** For each message block m_i where $i \in \{1, 2, \dots, t\}$, xor m_i with the current state s_i as the 64 most significant bits of the key k_i for current r times PRESENT round function ρ . The rest 16 bits of the key k_i is derived from the 16 most significant bits of the authentication key k (denote by $MSB^{16}(k)$). By executing the same key schedule algorithm of PRESENT, apply r times ρ on the state s_{i-1} , such that

$$s_i = \rho_{m_i \oplus s_{i-1} || MSB^{16}(k)}^r(s_{i-1}).$$

4. **Finalization.** Apply one full-round PRESENT encryption to the state s_t under the key k , and then truncate the first ℓ_m bits of the final state s_{t+1} as the tag of the message M .

$$s_{t+1} = E_k(s_t), tag_M = Trunc^{\ell_m}(s_{t+1}).$$

Since the length of internal state is only 64 bits, TuLP is not strong enough to resist the birthday attack on internal states for an existential forgery. Although this “weakness” is not fatal in some BSN applications, we still provide a wide-pipe version, which is called TuLP-128, to increase the state and the maximum tag lengths to be 128 bits. The key length of TuLP-128 is up to 160 bits. We note that the design principle is inspired by MDC-2 [20] and the padding rule is identical to TuLP.

1. **Padding.** Let k be an authentication key such that $|k| \leq 160$ bits. By using the same padding rule of TuLP, split the result $pad(M) = M || \lambda(M, k) || 10^d$ into 64-bit blocks $m_1, m_2, \dots, m_t, t = \frac{|pad(M)|}{64}$.
2. **State Initialization.** Divide the (padded) authentication key k into two 80-bit key $k_l || k_r$. Then apply one full-round PRESENT encryption to two different 64-bit initial values IV_1 and IV_2 under k_l and k_r , respectively. Obtain the outputs as the *left* and *right* initial states $s_{l,0}$ and $s_{r,0}$, such that

$$s_{l,0} = E_{k_l}(IV_1), s_{r,0} = E_{k_r}(IV_2).$$

3. **Compression.** For each message block m_i where $i \in \{1, 2, \dots, t\}$, first split the last left and right states $s_{l,i-1}$ and $s_{r,i-1}$ into four 32-bit blocks. Then exchange the least significant 32 bits of the left state (denoted by $LSB^{32}(\cdot)$) with the most significant 32 bits of the right state. The exchanged input states are denoted by

$\hat{s}_{l,i-1}$ and $\hat{s}_{r,i-1}$. By following the same algorithm of the compression in TuLP, apply r PRESENT round functions on the exchanged input states $\hat{s}_{l,i-1}$ and $\hat{s}_{r,i-1}$ respectively.

$$\begin{aligned}\hat{s}_{l,i-1} &= \text{MSB}^{32}(s_{l,i-1}) \parallel \text{MSB}^{32}(s_{r,i-1}), \\ \hat{s}_{r,i-1} &= \text{LSB}^{32}(s_{l,i-1}) \parallel \text{LSB}^{32}(s_{r,i-1}); \\ s_{l,i} &= \rho_{m_i \oplus s_{l,i-1} \parallel \text{MSB}^{16}(k_l)}^r(\hat{s}_{l,i-1}), \\ s_{r,i} &= \rho_{m_i \oplus s_{r,i-1} \parallel \text{MSB}^{16}(k_r)}^r(\hat{s}_{r,i-1}).\end{aligned}$$

4. **Finalization.** Apply one full-round PRESENT encryption to the left and the right states under the divided keys k_l and k_r respectively. Then truncate the first ℓ_m bits of the concatenation of the final states as the tag of the message M .

$$\begin{aligned}\hat{s}_{l,t} &= \text{MSB}^{32}(s_{l,t}) \parallel \text{MSB}^{32}(s_{r,t}), \\ \hat{s}_{r,t} &= \text{LSB}^{32}(s_{l,t}) \parallel \text{LSB}^{32}(s_{r,t}); \\ s_{l,t+1} &= E_{k_l}(\hat{s}_{l,t}), \quad s_{r,t+1} = E_{k_r}(\hat{s}_{r,t}); \\ \text{tag}_M &= \text{Trunc}^{\ell_m}(s_{l,t+1} \parallel s_{r,t+1}).\end{aligned}$$

Figure 2 and 3 depict the high-level algorithms of TuLP and TuLP-128, respectively. Referring to the security issues of ALPHA-MAC and Pelican [5,10,32], the advantages of our schemes are as follows.

- In ALPHA-MAC [12], all message blocks directly become the round keys after the message injections, so the attacker can execute side-channel attacks in the *known message scenario*. Biryukov *et al.* [5] present a side-channel attack on ALPHA-MAC, which relies on the fact that the round keys of ALPHA-MAC are public-known by the attacker. In TuLP, round keys are not computed from a deterministic function of input message blocks. Thus, a side-channel attack is unlikely to make a hypothesis on any intermediate states of the algorithm. The xor operation between the state and the input message block can resist the attacker to implement similar side-channel attacks [5] on TuLP and TuLP-128.
- Like in Pelican [13], the message injection layer is also removed in TuLP and TuLP-128 for simplicity. Because it can hardly improve the resistance against linear and differential attacks. In Pelican, the message block is xored with the last output state as the input for current round. But in our schemes, the message block is xored with the state as a part of the subkey for next round. We note that the iteration of $E_{k \oplus m}(k)$ is proven to be collision and preimage resistant in the black-box analysis of the PGV constructions [8].
- The bitwise lengths of message and key are appended to the end of the message. This message padding rule can avoid some trivial attacks on the internal collision and the extension. ALPHA-MAC and Pelican only pad message with a single 1 followed by the minimum number of 0 bits to suffice a block.
- Benefit from the ALRED construction, the security of our schemes can be reduced to the security of PRESENT if internal collisions are not involved. The proofs are provided in the security analysis of Section 5.2. Since the compressions in TuLP and TuLP-128 are different from the PRESENT encryption, encryption and authentication can use the same secret key.

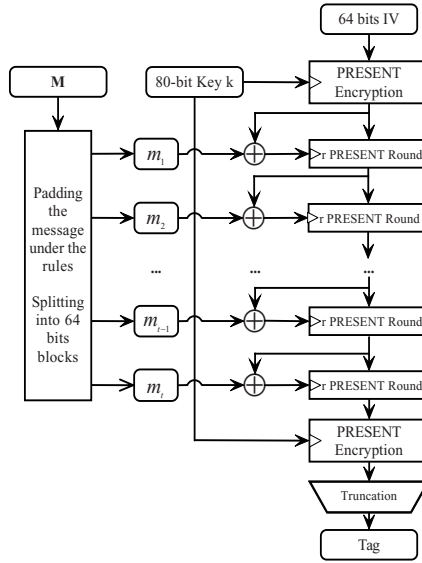


Fig. 2. The illustration of TuLP

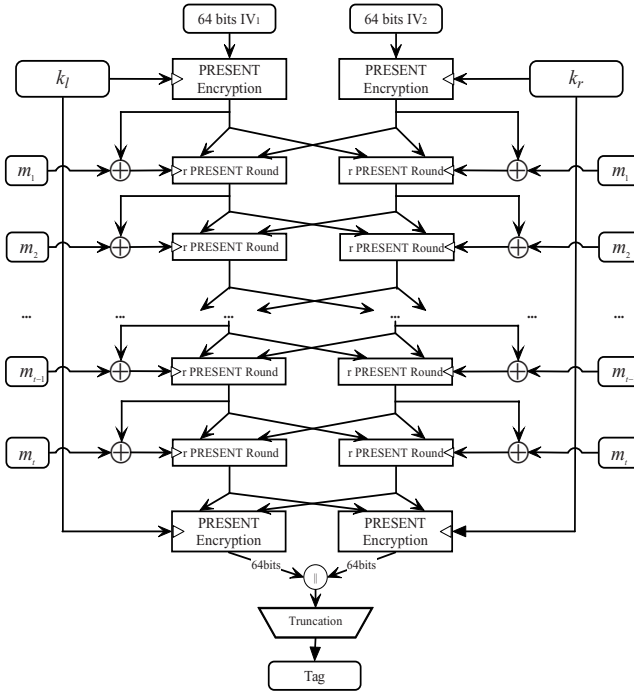


Fig. 3. The illustration of TuLP-128

- TuLP is designed for rapid message processing. The computational costs of the message processing are equivalent to $\frac{r}{31}$ of one PRESENT encryption. Whilst TuLP-128 provides a wider intermediate state and maximum 128-bit tag length for collision resistance, such that the costs of message processing only require $\frac{2-r}{31}$ of one PRESENT encryption.
- The choice of r rounds PRESENT in the compression is *tunable* by consideration of the practical balance of security and performance. Since key management in sensor network is expensive on computation and energy, the length of authentication key is *tunable* since the padding rules considered dynamic key length. To give practical instances for the analysis in the following section, we will consider $r=16$ in the compression of TuLP and TuLP-128, whilst $IV = IV_1 = 0123456789ABCDEF$ and $IV_2 = FEDCBA9876543210$.

5.2 Security Analysis

In this section, we first prove that TuLP is as strong as the PRESENT block cipher with respect to key recovery and existential forgery attacks without internal collisions. Then we give a synthetic analysis of TuLP when internal collisions are considered. Finally, a similar security analysis is given on TuLP-128.

Since the ALRED construction has a similar internal structure with the CBC mode, which typically implies the security between the construction and the underlying cryptographic primitives. Derived from the provability results of the ALRED construction in [12], it is easy to derive a similar result on TuLP as follows.

Theorem 3. *Any key recovery attack on TuLP requiring t (adaptively) chosen messages, can be converted to a key recovery attack on the PRESENT block cipher requiring $t + 1$ adaptively chosen plaintexts.*

Proof. Let \mathcal{A} be a successful attacker requiring t tag values corresponding to t (adaptively) chosen messages m_i yielding the key k , where $i \in \{1, 2, \dots, t\}$. Then we derive a key recovery attack on the PRESENT block cipher as follows.

1. Request the first state $s_0 = E_k(IV)$.
2. For $i = 1$ to t , compute the intermediate state $s_i = \chi(s_0, m_i)$, where χ denotes the compression function of TuLP.
3. For $i = 1$ to t , request $tag_i = \text{Trunc}(E_k(s_i))$.
4. Submit t tag values to \mathcal{A} to recover the key k .

The above attack requires t chosen messages and one chosen message on $E_k(IV)$. So the theorem follows. \square

Similar to Theorem 3, the provability of TuLP can be extended to the existential forgery attack and the fixed point attack as follows. The proofs are omitted here due to the page limit.

Lemma 1. *Any existential forgery attack on TuLP without internal collisions requiring t (adaptively) chosen messages, can be converted to a ciphertext guessing attack on the PRESENT block cipher requiring $t + 1$ adaptively chosen plaintexts.*

Lemma 2. Any existential forgery attack on *TuLP*, requiring t (adaptively) chosen messages for a fixed point $\{(m, s) | E_{m \oplus s}(s) = s, m \in \mathcal{M}, s \in \mathcal{K}\}$, can be converted to a fixed point attack $\{(m', k) | E_{m'}(k) = k, m \in \mathcal{M}, k \in \mathcal{K}\}$ on the *PRESENT* block cipher requiring $t + 1$ adaptively chosen plaintexts.

Now we analyze the security with respect to internal collisions. The reason why we choose $r=16$ in the compression of *TuLP* (and *TuLP-128*) to resist the internal collisions from the linear and differential cryptanalysis are briefly described as follows.

Theorem 4. Consider $r=16$ in the compression of *TuLP*. The minimum extinguishing differential in *TuLP* imposes a differential characteristic of about 2^{-64} . Whilst the maximum bias of the linear analysis with the probability of about 2^{-28} with 2^{56} known plaintext/ciphertext pairs.

Proof. Based on the differential and the linear cryptanalyses that are given by Bogdanov *et al.* [9], any 5 rounds differential characteristic of *PRESENT* has a minimum of 10 active S-boxes. One round *PRESENT* has one S-box, all 31 rounds use the same. For differential cryptanalysis, we have:

1. One S-box provides maximum 2^{-2} possibility for differential characteristic, thus 16 rounds provide a lower bound $(2^{-2})^{r \cdot 10/5} = 2^{-64}$ for the probability of a characteristic. The probability is not greater than the birthday attack on the intermediate states (2^{-32} and 2^{-64} for *TuLP* and *TuLP-128* respectively).
2. This differential cryptanalysis would require the memory complexity of about 2^{64} known plaintext/ciphertext pairs.

For linear cryptanalysis, we have:

1. Any 4 rounds provide the maximal bias of a linear approximation $\epsilon_{4R} \leq 2^{-7}$. Hence 16 rounds provide the maximum bias of a linear approximation $(2^{-7})^{r/4} = 2^{-28}$.
2. This linear cryptanalysis would require the memory complexity of about $1/(2^{-28})^2 = 2^{56}$ known plaintext/ciphertext pairs.

So the theorem follows. □

Consider a typical BSN application consisting of 100 nodes, each node transfers an 8-byte message under the same authentication key per 15 seconds for monitoring. Although the above linear analysis has a non-negligible bias, the time and the memory complexities of obtaining 2^{56} plaintext/ciphertext pairs (about 2^{19} TB) would be impractical.

By using multi-collisions, Knudsen *et al.* [22] provide a collision attack and preimage attacks on the MDC-2 construction with the time complexities of about $(\log_2(n)/n) \cdot 2^n$ and 2^n where the block length is n . The preimage attacks make new trade-offs so that the most efficient attack requires time and memory of about 2^n . Whilst the meet-in-the-middle attack on MDC-2 [23] requires time and memory about $2^{3n/2}$ and 2^n . Based on the security analysis of the MDC-2 construction and *TuLP*, the security of *TuLP-128* with the internal collisions is as follows.

Theorem 5. Consider $r=16$ in the compression of TuLP-128. The internal collision and preimage attacks on TuLP-128 have the complexities of about $2^{61.3}$ and 2^{64} , respectively.

Proof. The proof is based on the security that $r=16$ in the compression of TuLP-128. One S-box provides a maximum 2^{-2} possibility for differential characteristic, 16-round PRESENT provide a lower bound 2^{-64} for the probability of a characteristic. The minimum extinguishing differential in TuLP-128 imposes a differential characteristic of about 2^{-64} in the left state and the same in the right state. 16 rounds provide a maximum bias of a linear approximation 2^{-28} . But both the differential analysis and the linear cryptanalysis require a memory complexity no less than 2^{56} known plaintext/ciphertext pairs, which is impractical in BSN. Since PRESENT is an SP-network block cipher and the iteration of $E_{k \oplus m}(k)$ is proven to be collision and preimage resistant in the black-box analysis by Black *et al.* [8], and TuLP-128 has a MDC-2 like construction. Each round of the compression in TuLP-128 exchanges the right most 32 bits of the left state with the left-most 32 bits of the right state. Due to Knudsen *et al.*'s cryptanalysis of MDC-2 [22], the internal collision attack and the preimage attack on TuLP-128 would require the time complexity of about $(\log_2(64)/64) \cdot 2^{64} \approx 2^{61.3}$ and 2^{64} , respectively. Therefore, the complexity of an internal collision is about $2^{-61.3}$ via the multi-collision attack with a negligible memory requirement. Whilst the preimage attack requires time and memory of about 2^{64} . So the theorem follows. \square

Although TuLP-128 does not achieve the ideal upper bounds of collision and preimage resistances, the MDC-2 like structure in TuLP-128 still yields many practical advantages. For example, symmetric left and right pipes can minimize the area in hardware, or the memory usage in software implementation. And the simple permutation layer between left and right states saves redundant logical gates. Nevertheless, a $2^{61.3}$ level of time complexity on finding an internal collision is still beyond the computational bound in practice. Now we consider the security of TuLP-128 without internal collisions.

Theorem 6. Any key recovery attack on TuLP-128 requiring t (adaptively) chosen messages, can be converted to a key recovery attack on PRESENT requiring $t + 2$ adaptively chosen plaintexts.

Proof. Consider the situation that $k_l = k_r = k$. Let \mathcal{A} be a successful attacker requiring t tag values corresponding to t (adaptively) chosen messages m_i yielding the key k , where $i \in \{1, 2, \dots, t\}$. Let χ be the compression function of TuLP. $\text{MSB}^{32}(\cdot)$ and $\text{LSB}^{32}(\cdot)$ denote the truncation of the most and the least significant 32 bits, respectively. Then we derive a key recovery attack on the PRESENT block cipher as follows.

1. Request the initial left and right states $s_{l,0} = E_k(\text{IV}_1)$ and $s_{r,0} = E_k(\text{IV}_2)$.
2. For $i = 1$ to t , compute the left state $s_{l,i} = \chi(\text{MSB}^{32}(s_{l,i}) \parallel \text{MSB}^{32}(s_{r,i}), m_i)$ and the right state $s_{r,i} = \chi(\text{LSB}^{32}(s_{l,i}) \parallel \text{LSB}^{32}(s_{r,i}), m_i)$.
3. For $i = 1$ to t , request $\text{tag}_i = \text{Trunc}(E_k(s_{l,i}) \parallel E_k(s_{r,i}))$.
4. Submit t tag values to \mathcal{A} to recover the key k .

The above attack needs t chosen messages except $E_k(IV_1)$ and $E_k(IV_2)$. So the theorem follows. \square

Similar to Theorem 6, it is easy to obtain the following lemmas on TuLP-128. The proofs are omitted here due to the page limit.

Lemma 3. *Any existential forgery attack on TuLP-128 without internal collisions of requiring t (adaptively) chosen messages, can be converted to a ciphertext guessing attack on PRESENT requiring $t + 2$ adaptively chosen plaintexts.*

Lemma 4. *Any existential forgery attack on TuLP-128 with a fixed point of requiring t (adaptively) chosen messages, can be converted to a fixed point attack on PRESENT requiring $t + 2$ adaptively chosen plaintexts.*

5.3 Performance

Before we study the performance of TuLP and TuLP-128, first we program an optimized implementation of PRESENT by using 1K bytes look-up table on MICAz nodes. From our performance tuning, we find that the bit permutation of PRESENT is costly in software implementation. Compared to the best known result of AES-128 software implementation on MICAz nodes [17], our optimized implementation of PRESENT still shows a competitive processing speed per block and promising lower memory costs. Since PRESENT has already been proven to be a better choice than AES in hardware implementation [10], our optimized implementation shows that PRESENT is also practical in software.

Table 2. The comparison of AES and PRESENT implementations

Encryption	Software (MICAz)			Hardware [10]		
	RAM (byte)	ROM (byte)	Processing speed	Logic process	Cycles per block	Area
AES-128 [10,17]	1915	12720	1.46ms / 16Bytes	0.35 μ m	1032	3400 GE
PRESENT-80	1040	1926	1.82ms / 8Bytes	0.18 μ m	32	1570 GE

As a point of comparison, we select DM-PRESENT [10], which is derived from the Davies-Meyer construction and the PRESENT with an 80-bit key, as the underlying hash function for HMAC [16]. We also choose one-key CBC-MAC based on PRESENT as a benchmark for comparability. The area in GE is estimated by using the *Virtual Silicon (VST)* standard cell library based on *UMC L180 0.18 μ m 1P6M Logic Process* (UMCL18G212T3). All experiments are based the MICAz nodes (*TinyOS version 2.10*), which are popular in both of WSN and BSN. The results in the entries of processing speed (in milliseconds) are averaged by iterating 100 times experiments with/without the optimization in the keysetup.

If we choose $r=16$ in the compression of TuLP, TuLP will be about 2 times faster than PRESENT encryption in message processing. Table 3 shows that TuLP approaches 1.6 and 1.8 times faster than HMAC with DM-PRESENT and one-key CBC-MAC based on PRESENT respectively, where message length from 8 bytes to 1024 bytes. The keysetup costs in our schemes, which require one (or two) PRESENT encryption(s) to generate an

Table 3. The comparison amongst some PRESENT-based MAC functions

	TuLP	TuLP-128	CBC-MAC (PRESENT)	HMAC (DM-PRESENT)
Key length (bit)	80	160	80	80
Intermediate state (bit)	64	128	64	64
RAM / ROM (byte)	1048 / 3302	1056 / 3718	1040 / 2970	1056 / 3484
Area in GE (estimate)	2252	2764	2252	2213 [10]
Processing Speed (ms)	TuLP	TuLP-128	CBC-MAC (PRESENT)	HMAC (DM-PRESENT)
8 bytes	4.46 / 6.63	8.91 / 13.24	6.51	10.90
16 bytes	5.59 / 7.75	11.17 / 15.49	8.70	13.08
32 bytes	7.87 / 10.03	15.72 / 20.05	13.05	17.43
64 bytes	12.39 / 14.56	24.76 / 29.09	21.77	23.97
128 bytes	21.43 / 23.59	42.84 / 47.17	39.20	37.04
256 bytes	39.50 / 41.67	79.00 / 83.33	74.06	65.35
512 bytes	75.65 / 77.81	151.53 / 155.66	143.78	122.01
1024 bytes	147.94 / 150.10	295.97 / 300.31	283.21	233.04

encrypted IV, mainly lack TuLP (or TuLP-128) in processing the messages shorter than 32 bytes. We note that the keysetup can be optimized by precomputing the encrypted IV before the authentications with the same keys, and the values can be reused in the latter authentication with the same keys. Same optimization can be implemented in TuLP-128 to boost the processing of short messages. We note that HMAC also can precompute the initialization values for optimization, but the values must be treated and protected (128 bits for a certain key in DM-PRESENT) in the same manner as secret keys [16]. While the optimization for our schemes only increases a smaller storage (one encrypted IV is 64-bit) without need to be insulated. Although the lengths of internal state and tag are doubled, the performance of TuLP-128 is still comparable to one-key CBC-MAC based on PRESENT. Obviously, TuLP-128 will be faster than HMAC with a double block length hash function based on PRESENT. Nevertheless, if a higher security bound is required, one can tweak the rounds in the compressions of TuLP and TuLP-128. For instance, increase 16 rounds to 20 will decrease about $4/16=25\%$ performance in message processing. In return, a 20-round PRESENT will have a lower bound $(2^{-2})^{20 \cdot 10/5} = 2^{-80}$ for a differential characteristic. And the maximal bias of a linear approximation $(2^{-7})^{20/4} = 2^{-35}$, which requires 2^{70} known plaintext/ciphertext.

6 Conclusion

By considering the restrictions of BSN, two lightweight MACs TuLP and TuLP-128 have been proposed. The security of our schemes is analyzed with respect to the cryptanalyses on ALRED and the results on PRESENT. The key length and the number of round functions in the compression are tunable in our lightweight schemes, which support practical trade-offs between security and performance in BSN applications. The statistics strongly support that TuLP and TuLP-128 are promising on devices with constrained resources. Since both PRESENT and ALRED are new proposals, we suggest

that rigorous analysis should be imposed to avoid any potential weakness inside the cryptosystems based on them.

Acknowledgement. We would like to thank Vicent Rijmen and Xuejia Lai for their helpful advice. And also thank many anonymous reviewers for their valuable comments. The first author acknowledges the financial support of SenterNovem for the ALwEN project, grant PNE07007. The last author is supported by NSFC (No.60573032, 60773092, 60803146), National “863” Program of China (No. 2009AA01Z418) and National “973” Program of China (No.2007CB311201).

References

1. AAL: The Ambient Assisted Living Joint Programme. European Union (January 2008), <http://www.aal-europe.eu/about-aal>
2. Albrecht, M., Cid, C.: Algebraic Techniques in Differential Cryptanalysis. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 193–208. Springer, Heidelberg (2009)
3. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 1–15. Springer, Heidelberg (1996)
4. Bellare, M., Kilian, J., Rogaway, P.: The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences* 61(3), 362–399 (2000)
5. Biryukov, A., Bogdanov, A., Khovratovich, D., Kasper, T.: Collision Attacks on AES-Based MAC: Alpha-MAC. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 166–180. Springer, Heidelberg (2007)
6. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC: Fast and Secure Message Authentication. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 216–233. Springer, Heidelberg (1999)
7. Black, J., Rogaway, P.: CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions. *Journal of Cryptology* 18(2), 111–131 (2005)
8. Black, J., Rogaway, P., Shrimpton, T.: Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 320–335. Springer, Heidelberg (2002)
9. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
10. Bogdanov, A., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y.: Hash Functions and RFID Tags: Mind the Gap. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 283–299. Springer, Heidelberg (2008)
11. Collard, B., Standaert, F.-X.: A Statistical Saturation Attack against the Block Cipher PRESENT. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 195–210. Springer, Heidelberg (2009)
12. Daemen, J., Rijmen, V.: A New MAC Construction ALRED and a Specific Instance ALPHA-MAC. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 1–17. Springer, Heidelberg (2005)
13. Daemen, J., Rijmen, V.: The Pelican MAC Function. Unpublished manuscript, <http://eprint.iacr.org/2005/088>
14. Feldhofer, M., Rechberger, C.: A Case Against Currently Used Hash Functions in RFID Protocols. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4277, pp. 372–381. Springer, Heidelberg (2006)

15. Ferguson, N.: Collision attacks on OCB. Preprint (February 2002)
16. Federal Information Processing Standard 198, The Keyed-Hash Message Authentication Code (HMAC), NIST, U.S. Department of Commerce (March 2002)
17. Healy, M., Newe, T., Lewis, E.: Analysis of Hardware Encryption Versus Software Encryption on Wireless Sensor Network Motes. In: Mukhopadhyay, S.C., Gupta, G.S. (eds.) *Smart Sensors and Sensing Technology*. Springer, Heidelberg (2008)
18. Huang, J., Seberry, J., Susilo, W.: On the internal Structure of ALPHA-MAC. In: Nguyễn, P.Q. (ed.) *VIETCRYPT 2006*. LNCS, vol. 4341, pp. 271–285. Springer, Heidelberg (2006)
19. ISO/IEC 9797-1, Information technology - Security Techniques - Message Authentication Codes (MACs) - Part 1: Mechanisms using a block cipher, ISO (1999)
20. ISO/IEC 10118-2:1994. Information technology - Security techniques - Hash-functions - Part 2: Hash-functions using an n-bit block cipher algorithm, Revised in (2000)
21. Karlof, C., Sastry, N., Wagner, D.: TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. In: *SenSys 2004*, Baltimore, Maryland, USA, November 3-5 (2004)
22. Knudsen, L., Mendel, F., Rechberger, C., Thomsen, S.: Cryptanalysis of MDC-2. In: Ghilardi, S. (ed.) *EUROCRYPT 2009*. LNCS, vol. 5479, pp. 106–120. Springer, Heidelberg (2009)
23. Lai, X., Massey, J.: Hash Functions Based on Block Ciphers. In: Rueppel, R.A. (ed.) *EUROCRYPT 1992*. LNCS, vol. 658, pp. 474–494. Springer, Heidelberg (1993)
24. Li, T., Wu, H., Wang, X., Bao, F.: *SenSec Design. I²R Sensor Network Flagship Project (SNFP: security part): Technical Report-TR v1.0* (February 2005)
25. Luk, M., Mezzour, G., Perrig, A., Gligor, V.: MiniSec: A Secure Sensor Network Communication Architecture. In: *IPSN 2007*, Cambridge, Massachusetts, USA, April 25-27 (2007)
26. Performance of optimized implementations of the NESSIE primitives, v2.0, The NESSIE Consortium (2003),
<https://www.cosic.esat.kuleuven.be/nessie/deliverables/D21-v2.pdf>
27. Özen, O., Varici, K., Tezcan, C., Kocair, Ç.: Lightweight Block Ciphers Revisited: Cryptanalysis of Reduced Round PRESENT and HIGHT. In: Boyd, C., Nieto, J.G. (eds.) *ACISP 2009*. LNCS, vol. 5594, pp. 90–107. Springer, Heidelberg (2009)
28. Paar, C., Poschmann, A., Robshaw, M.: New Designs in Lightweight Symmetric Encryption. In: Kitsos, P., Zhang, Y. (eds.) *RFID Security: Techniques, Protocols and System-on-Chip Design*, pp. 349–371. Springer, Heidelberg (2008)
29. Perrig, A., Szewczyk, R., Wen, V., Culler, D., Tygar, J.D.: SPINS: security protocols for sensor networks. In: *Proceedings of the 7th annual international conference on Mobile computing and networking*, Rome, Italy, pp. 189–199 (July 2001)
30. Rogaway, P., Bellare, M., Black, J.: OCB: A block-cipher mode of operation for efficient authenticated encryption. *ACM Transactions on Information and System Security (TISSEC)* 6(3), 365–403 (2003)
31. Wang, M.: Differential Cryptanalysis of Reduced-Round PRESENT. In: Vaudenay, S. (ed.) *AFRICACRYPT 2008*. LNCS, vol. 5023, pp. 40–49. Springer, Heidelberg (2008)
32. Wang, W., Wang, X., Xu, G.: Impossible Differential Cryptanalysis of PELICAN, MT-MAC-AES and PC-MAC-AES, <http://eprint.iacr.org/2009/005>
33. Yang, G.Z. (ed.): *Body Sensor Network*. Springer, London (2003)