# Low-cost Standard Signatures for Energy-Harvesting Wireless Sensor Networks

GIUSEPPE ATENIESE, University of Rome "La Sapienza"
GIUSEPPE BIANCHI, University of Rome Tor Vergata
ANGELO T. CAPOSSELE, University of Rome "La Sapienza"
CHIARA PETRIOLI, University of Rome "La Sapienza"
DORA SPENZA, University of Rome "La Sapienza"

This work is motivated by a general question: Can micro-scale energy harvesting techniques be exploited to support low-cost *standard* security solutions on resource-constrained devices? We focus on guaranteeing integrity and authentication in Internet of Things (IoT) and Wireless Sensor Network (WSN) applications. In this paper we propose techniques to make ECDSA signatures low cost and implementable on resource-constrained devices. By combining precomputation techniques and energy harvesting capabilities of modern sensor nodes, we achieve significant improvement over prior works. In addition, we show the cost of ECDSA signatures can be reduced of up to a factor 10 by using harvesting-aware optimizations.

CCS Concepts: •**Security and privacy** → **Digital signatures;** •**Networks** → **Sensor networks;** •**Hardware** → *Renewable energy;*

Additional Key Words and Phrases: Energy harvesting, ultra low energy systems, low-energy authentication, low-energy digital signatures, IoT and WSNs security, wireless sensor networks

**ACM Reference Format:**
Giuseppe Ateniese, Giuseppe Bianchi, Angelo T. Capossele, Chiara Petrioli and Dora Spenza, 2015. Low-cost Standard Signatures for Energy-Harvesting Wireless Sensor Networks. *ACM Trans. Embedd. Comput. Syst.* V, N, Article A (January 2015), 23 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

## 1. INTRODUCTION

Over the past decade, progress in computing and communication capabilities of embedded devices has played a major role in the advent of the Internet of Things (IoT), a technology that enables smart devices (e.g., mobile phones, wireless sensor nodes, home appliances and industrial machines) to communicate and share data over the Internet [Atzori et al. 2010]. One of the key enablers to realize the vision of the Internet of Things is Wireless Sensor Networks (WSNs) [Alcaraz et al. 2010; Christin et al. 2009]. Due to their low cost and pervasive capability, WSNs have gained increasing popularity in the last decade, as they allow for accurate real-time information in a multitude of application scenarios that conventional cabled or wireless networks are

A portion of this paper is an extension of [Ateniese et al. 2013]. This work has been partially supported by the MIUR project SMARTOUR: INTELLIGENT PLATFORM FOR TOURISM, by the National Technological Cluster CTN001 0034 23154 *Social Museum and Smart Tourism*, funded by the Italian MIUR and by the European Union (European Social Fund) and by the PON project PON03PE00214. G. Ateniese gratefully acknowledges support from a Google Faculty Research Award and an IBM Faculty Award. D. Spenza gratefully acknowledges support from two Sapienza Research Grants for Young Investigators.

Author's addresses: G. Ateniese and A.T. Capossele and C. Petrioli and D. Spenza, Computer Science Department, University of Rome "La Sapienza"; G. Bianchi, Department of Electronic Engineering, University of Rome Tor Vergata.

DOI: http://dx.doi.org/10.1145/0000000.0000000

unable to handle. Acting as a bridge to the physical world, wireless sensor networks will eventually make possible the automatic monitoring of vital signs and health conditions in assisted living and e-health applications, of environmental parameters such as air quality and pollution, and of energy and water use in civil and industrial buildings, among others. In many of such scenarios, security support is a critical requirement [Lopez et al. 2009; Sharma et al. 2012]. However, security solutions must cope with the limited energy resources of WSN platforms, which are typically powered by short-lived batteries. In fact, many WSN applications require the network to operate unattended for extremely long periods of time, tackling contexts where even the physical access to a sensor, once released in the field, may be impossible. Battery replacement and recharging is thus highly impractical and very expensive at best, if not altogether impossible. For this reason, considerable effort has been devoted by the research community to develop carefully-crafted communication and sensing protocols that, along with low-power sensor node architectures, permit an extremely sparing usage of the limited energy resources available to wireless sensor nodes.

Security protocols in WSNs makes no exception to such a design strategy: They should retain effectiveness while using as little energy as possible. One way to accomplish this design goal is to devise novel, *energy-friendly*, lightweight security primitives. However, a *novel* construction is not always advisable in security. In fact, despite its possible technical merits, acceptance of a novel approach requires time for a thorough scrutiny, and may involve multiple revisions along this path (for instance, the NTRU signature was broken multiple times [Gentry and Szydlo 2002] when initially proposed). Moreover, in critical settings it makes sense to leverage standardized security constructions, rather than novel approaches not challenged by a long-lasting real-world practice.

Motivated by this need, in this paper we address the problem of how to *practically* achieve low-cost security in real-world wireless sensor networks, without requiring substantial changes in the security protocols set forth. In particular, we focus on ECDSA (Elliptic Curve Digital Signature Algorithm), the de-facto standard signature scheme employed in WSN and IoT applications to ensure authenticity and integrity of communications [Hummen et al. 2013; Misic 2009; Wei-hong et al. 2009]. Support for ECDSA is also included in CoAP, which is being standardized as an application layer protocol for the IoT [Shelby et al. 2013]. In addition, we remark that NSA-approved products must employ ECDSA[1]. Despite its popularity, however, practical implementations of ECDSA on wireless sensor networks are still challenged by high energy consumption and long delays, which can significantly affect the lifetime of the network [Bicakci et al. 2012], and which prevent the extensive usage of ECDSA-based primitives in many application scenarios.

In this paper, we present a pragmatical approach to reduce the cost of generating ECDSA signatures by jointly exploiting precomputation and energy-harvesting capabilities embedded in modern sensor nodes. Our solution builds on the scheme proposed by Boyko, Peinado and Ventakesan in [Boyko et al. 1998], which we term BPV from the name of the authors. The main idea of BPV is to precompute and store a set of $n$ Discrete Log pairs, a subset of which is randomly chosen and suitably combined to perform costly modular operations with a significant computational gain (see Section 2.2 for details). Despite its simplicity and appeal, however, to the best of our knowledge neither BPV nor similar variants were so far considered in practical sensor networks implementations. This is mainly due to the fact that the number of precomputed pairs must be sufficiently large to thwart Lattice reduction attacks [Nguyen and Stern 1999]. Such a requirement results in a large memory footprint that has long prevented practical

[1] http://www.nsa.gov/ia/programs/suiteb%5Fcryptography/

implementations of BPV on first-generation resource-constrained wireless sensor platforms. To address this limitation, we proposed an improved version of the BPV scheme, which we name I-BPV. I-BPV requires only relatively small fraction of the memory available to last-generation motes [Bischoff et al. 2009] (see Section 5.3 for details). The second key component of our proposed approach is support for energy harvesting. The recent emergence of cost-effective low-scale power scavenging technologies are making possible to supplement the limited battery energy of wireless motes with energy gathered from the environment (e.g., solar, wind, etc.) [Basagni et al. 2013]. In some cases the energy available to environmentally-powered motes can be even *excessive*, i.e., greater than the amount that can fit into the energy storage devices of a node (energy overflow), and thus it would be wasted if not immediately used. As pointed out in our prior work [Ateniese et al. 2013], the occurrence of such *energy peaks* very well fits precomputation-based schemes, as it permits to push part of the computation to excess energy periods. We thus propose a set of harvesting-aware optimizations to exploit periods of high energy availability.

Our specific contributions are the following.

— We present I-BPV, a precomputation scheme for ECDSA signatures that reduces the memory overhead of existing precomputation schemes by a factor of 5, making them feasible on resource-constrained wireless sensor platforms.
— We propose a set of specific harvesting-aware optimizations that exploit energy-harvesting capabilities of modern sensor nodes to enhance the performance of I-BPV.
— We implement I-BPV on three off-the-shelf sensor node platforms, the MagoNode++, TelosB and MICA2 motes, characterized by widely different design aspects, and provide an in-depth experimental assessment of the performance, energy cost, and emerging trade-offs.
— Through both simulations and real-life experimentations, we perform a thorough assessment of the performance of I-BPV in energy-harvesting WSNs. Our results show that leveraging periods of high energy availability allows to significantly reduce the energy consumption of performing ECDSA signatures to up to a factor 10 w.r.t. prior implementations.

The rest of the paper is organized as follows. In Section 2 we provide the background on the known results we have exploited and extended in this work. I-BPV is presented and discussed in Section 3. In Section 4, harvesting-aware optimizations are presented. Performance of I-BPV are evaluated in Section 5. We review related works in Section 6. Section 7 concludes the paper.

## 2. BACKGROUND

### 2.1. Elliptic Curve Digital Signature Algorithm (ECDSA)

We recall the construction of an ECDSA signature [Johnson and Menezes 1998]. In what follows, unless otherwise specified, we resort to multiplicative notation. Select an elliptic curve $E$ defined over $\mathbb{Z}_p$ such that the number of points in $E(\mathbb{Z}_p)$ is divisible by a large prime $q$. Let $g \in E(\mathbb{Z}_p)$ be a point of order $q$. Let the integer $x \in [1, q-1]$ be a randomly chosen private key, and let the elliptic curve point $g^x \in E(\mathbb{Z}_p)$ be the corresponding public key (along with the public setup information $q, E, g$). Let $H(.)$ be a secure hash function. Then the ECDSA signature for a message $m$ is constructed as shown in Algorithm 1.

Security of ECDSA relies on the choice of the integer $r$, which must be unique and unpredictable for each signature. Indeed, if $r$ can be predicted, then it would be trivial to derive the secret key $x$ from the linear modular equation:

$$s = r^{-1}(H(m) + xw) \mod q \rightarrow x = w^{-1}(sr - H(m)) \mod q$$

**ALGORITHM 1:** Algorithm of ECDSA Signature for a message $m$.

```
Input: The node's private key $x$. A message $m$ to be signed.
Select a random value $r \in [1, q-1]$.
Compute the elliptic curve point $g^r = (x_1, y_1)$.
Compute $w = x_1 \mod q$ (if $w = 0$, restart).
Compute $r^{-1} \mod q$.
Compute $s = r^{-1}(H(m) + xw) \mod q$ (if $s = 0$, restart).
Output: The pair of integers $(w, s)$ is the signature for message $m$.
```

Similarly, if a same $r$ is used for signing two different messages $m$ and $m'$, then the secret key $x$ would be readily derived from the known signatures $(w, s)$ and $(w, s')$.

### 2.2. Precomputation of Discrete Log pairs: Simple BPV generator and full BPV generator

Let $g \in \mathbb{G}_q$ be a generator of a cyclic group of order $q$. Boyko, Peinado and Venkatesan first introduced in [Boyko et al. 1998] a surprisingly simple technique for speeding up the generation of pairs of the form $(r, g^r)$, which is generally the most expensive operation in Discrete log based schemes. The technique they proposed, hereafter referred to as *simple BPV generator*, speeds up the computation by preliminary precomputing (and storing in a table) a number $n$ of randomly-chosen pairs. Whenever a random pair $(r, g^r)$ is needed, the generator randomly selects $k$ out of the $n$ precomputed pairs, sets the "random" value $r$ as the sum of the chosen terms $\kappa_i$, and computes the corresponding term $g^r$, by simply multiplying the corresponding precomputed values $g^{\kappa_i}$. This algorithm is extremely efficient, as it requires only $k-1$ multiplications. Of course, the generated value $r$ is not uniformly distributed. However, with an appropriate choice of the parameters $n$ and $k$, the distribution of the generated values is statistically close to the uniform random distribution [Nguyen et al. 1999].

The simple BPV generator is further extended in [Boyko et al. 1998] by combining it with a random walk on a Cayley graph expander. Hereafter, we refer to this extension with the name *full BPV generator*, or *BPV* for brevity. The two phases of the full BPV generator are shown in Algorithm 2. We recall that, intuitively, a graph is an expander if it is easy to reach any vertex from any other in very few steps. In other words, a graph is an expander when, starting from any initial probability distribution on its vertices, a random walk on the graph will rapidly converge to the uniform distribution on all vertices. Obviously, expanders are of practical interest whenever their degree is *low* but their expansion "speed" is large. The expansion performance of a graph can be quantified via a (vertex) expansion parameter $\gamma$. Clearly, we wish to have $\gamma > 1$ as large as possible. Most of the results concerning expanders (including all results presented in the next section 3) are expressed in terms of an alternative (spectral) parameter $\epsilon < 1$, an $\epsilon$-spectral expander being a $\gamma$-vertex expander with $\gamma = 2/(1+\epsilon^2)$.

The full BPV generator builds on a theorem proved by Alon and Roichman in [Alon and Roichman 1994].

THEOREM 2.1. *Let $\mathbb{G}_q$ be a group of order $q$, and let $\mathbb{S}$ be a random set of group elements. Let $X(\mathbb{G}_q, \mathbb{S})$ be a Cayley graph of the group $\mathbb{G}_q$ with respect to a set $\mathbb{S}$ of elements. For any $1 > \epsilon > 0$ there exists a constant $c(\epsilon) > 0$ such that, for any random set $\mathbb{S}$ of $c(\epsilon)\log_2 q$ elements of $\mathbb{G}_q$, the Cayley graph is an $\epsilon$-spectral expander almost surely.*

Based on this result, the full BPV generator includes an *additional* table comprising $n_e$ randomly chosen pairs (Table $T2$ in Algorithm 2). The generator has an extra cost in terms of storage due to the additional table $(d_j, g^{d_j})$ and the pair $(t, R)$, and requires

**ALGORITHM 2:** Algorithm of the full BPV generator.

```
Preprocessing:
    Generate $n$ integers $\kappa_1, \ldots, \kappa_n \in \mathbb{Z}_q$.
    Create an empty table $T1$ of size $n$.
    for i = 1 to n:
        Compute $g^{\kappa_i}$ and set $T1_i = (\kappa_i, g^{\kappa_i})$.
    Generate $n_e = c(\epsilon) \log_2 q$ integers: $d_1, \ldots, d_{n_e} \in \mathbb{Z}_q$.
    Create an empty table $T2$ of size $n_e$.
    for j = 1 to $n_e$:
        Compute $g^{d_j}$ and set $T2_j = (d_j, g^{d_j})$.
    Initialize a value $t$ to a random element in $\mathbb{Z}_q$.
    Randomly select $d_j \in \{d_1, \ldots, d_{n_e}\}$ and initialize a value $R = g^{d_j}$.
Online Pair generation:
    Randomly generate $S \subset [1, n]$ of size $k$.
    Select a random $d_u, u \in [1, n_e]$.
    Set $r = t + d_u \mod q$ and $g^r = R \cdot g^{d_u}$, using pair $(d_u, g^{d_u})$ stored in table $T2$.
    for i = 1 to k:
        Set $r = r + S_i \mod q$.
    for i = 1 to k:
        Set $g^r = g^r \cdot g^{S_i}$, using pair $(S_i, g^{S_i})$ stored in table $T1$.
    Return the pair $(r, g^r)$.
```

two extra multiplications in addition to the $k - 1$ ones. However, for an appropriate choice of $n_e \approx \log_2 q$, i.e., $c(\epsilon) = 1$, it permits to reduce the value $k$ by a factor of two, i.e., the full BPV generator with parameters $n, k$ behaves as the simple generator with parameters $n, 2k$.

### 2.3. Application of BPV to ECDSA

As the BPV scheme does not depend on the specifically chosen group, it can be directly applied to the Elliptic Curve setting [Coron et al. 2001], and to the relevant Elliptic Curve Digital Signature Algorithm (ECDSA) construction. The security of the BPV generator relies on the hardness of the *Hidden Subset Sum problem*.

DEFINITION 1 (HIDDEN SUBSET SUM PROBLEM). *Given integers $M, b_1, \cdots, b_m \in \mathbb{Z}_M$, find $\alpha_1, \cdots, \alpha_n \in \mathbb{Z}_M$ such that each $b_i$ is some subset sum of $\alpha_1, \cdots, \alpha_n$ modulo $M$.*

This problem is conjectured to be hard if the ratio $n / \log_2 M$ is sufficiently large, more precisely greater than a given threshold approximately equal to 0.94. As noted in [Nguyen and Stern 1999], the reliance upon the Hidden Subset Sum problem holds also when the generator is used such that the integers $b_i$ are not directly disclosed, but indirectly provided to a passive attacker via Discrete log terms such as $g^{b_i}$. This is indeed a case of significant practical interest when BPV is used for ECDSA. In fact, when the truly random terms $r$ used by ECDSA (Section 2.1) are replaced with those produced by the generator (Section 2.2), security of signature schemes depends on a slightly modified variant of the Hidden Subset Sum problem, called the *Affine Hidden Subset Sum problem*, which does *not* appear to be more complex than the original problem [Nguyen and Stern 1999].

DEFINITION 2 (AFFINE HIDDEN SUBSET SUM PROBLEM). *Given a positive integer $q$, and $b_1, \cdots, b_m$, $c_1, \cdots, c_m \in \mathbb{Z}_M$, find integers $x, \alpha_1, \cdots, \alpha_n \in \mathbb{Z}_M$, such that each $b_i + xc_i$ is some subset sum modulo $M$ of $\alpha_1, \cdots, \alpha_n$.*

Indeed, this obviously holds for ECDSA. It is sufficient to note that $r$, which, owing to the generator, is a hidden subset sum, can be expressed as:

$$r = s^{-1}(H(m) + xw) = s^{-1}H(m) + xws^{-1} = b + xc \mod q$$

where, for each signed message, $b = s^{-1}H(m) \mod q$ and $c = ws^{-1} \mod q$ are known to a passive attacker.

## 3. I-BPV: IMPROVED BPV GENERATOR

The BPV full generator combines the simple generator with a random walk on expanders based on Cayley graphs on abelian group. The distribution of the outputs of the simple generator is shown to be at most $2^{-(e+1)}$ statistically distinguishable from the uniform distribution, where $e = \frac{1}{2}\left(\log \binom{n}{k-m}\right)$ and $m = |p|$ for a prime $p$. Thus, for large values of $\binom{n}{k}$, the outputs of the simple generator follow essentially the uniform distribution. In BPV, the simple generator is improved by using expanders which will preserve randomness even when decreasing $k$. This is due to the Alon–Roichman theorem [Alon and Roichman 1994] which asserts that random Cayley graphs are expanders:

THEOREM 3.1 (RANDOM CAYLEY GRAPHS ARE EXPANDERS). *For every $\epsilon > 0$ there exists a constant $c(\epsilon)$ such that the Cayley graph, obtained by selecting $n_e$ elements independently and uniformly at random from a finite group G, has expected second largest eigenvalue less than $\epsilon$ (i.e., it is an expander with high probability), whenever $n_e \geq (c(\epsilon) + o(1)) \log |G|$.*

Because of this theorem, the value $n_e$ is set to $c(\epsilon) \log |G|$ in BPV. Here the leading constant $c(\epsilon)$ is $4e/\epsilon^2$ which is about $10.87/\epsilon^2$.

The full BPV generator can be improved by showing that $n_e$ can be smaller, thus saving in space. Our improved BPV generator, which we call I-BPV, relies on the result from Christofides and Markstrom [Christofides and Markstrom 2008], who showed that the constant $c(\epsilon)$ can be reduced from $10.87/\epsilon^2$ to $2/\epsilon^2$. More specifically, by stressing the relationship between graph expansion and the second eigenvalue, the bounds on the expected expansion of the Cayley graph is $n_e = (2 \ln 2/\epsilon + o(1))^2 \log |G|$. Recalling the Theorem 3.1, the expected second largest eigenvalue has to be $\mathbf{E}]\lambda_2(X(G, S))] \leq \epsilon$ where the Cayley graph $X(G, S)$ is an undirected graph formed by taking the elements of $G$ as vertices with $G$ as a finite group and $S$ a set of generators for $G$. Thus, fixed an $\epsilon$, the value $n_e$ in I-BPV will be about $1/5$ of the $n_e$ used in BPV. In practice, this means that the extra table stored in I-BPV, and thus the memory overhead of the precomputation scheme for ECDSA, will be *five times smaller* than the table in BPV.

I-BPV is still safe against birthday attacks, even though $n_e$ is significantly smaller than previously intended. In particular, when the ratio $n/\log_2 q$ is in the order of 1 or more, based on [Nguyen and Stern 1999], the security of the I-BPV generator depends on its resistance to birthday attacks, which directly derives from the relevant theorem in BPV.

THEOREM 3.2 (FROM [BOYKO ET AL. 1998]). *If $G$ is a cyclic group of order $q$, then the expected number of repetitions in a run of I-BPV of length $l$ is at most:*

$$\frac{\binom{l}{2}}{q} + \frac{l}{\binom{n}{k}}\left(\frac{1}{1-2^{-c}} + \frac{1}{c} k \log n\right)$$

*for some constant $c$.*

The first term of Theorem 3.2 is the expected number of repetitions in an ideal sequence whose elements are independent random elements of $\mathbb{Z}_q$. The second term represents the number of additional collisions due to the generator. This term contains the parameter $l$ only as a linear function, which is important to minimize the effect of the birthday attack (which aims at increasing the expected number of collisions of a factor proportional to $l^2$).

More specifically, based on the work presented in [Boyko et al. 1998], we can observe that the probability that any particular number output by the full generator repeats after exactly $m$ steps is at most:

$$\min\left\{\frac{1}{\begin{pmatrix}n\\k\end{pmatrix}}, \frac{1}{q} + 2^{-cm}\right\} \tag{1}$$

If there exists an integer $m < l$ such that $1/q + 2^{-cm} \le 1/\binom{n}{k}$, then let $\sigma$ be the smallest such integer. Otherwise, let $\sigma = l$. Let the random variable $C$ denote the number of collisions. Then:

$$\begin{aligned} EC &= \sum_{ij} Pr(x_i = x_j) \\ &\le \sum_{i<j;j-i<\sigma} \frac{1}{\binom{n}{k}} + \sum_{i<j;j-i\ge\sigma} \left(\frac{1}{q} + 2^{-c(j-i)}\right) \\ &< \frac{\binom{l}{2}}{q} + l\sigma\left(\frac{1}{\binom{n}{k}} - \frac{1}{q}\right) + \sum_{i<j;j-i\ge\sigma} 2^{-c(j-i)}, \end{aligned} \tag{2}$$

where $x_i$ is the $i$-th element in the output sequence and the sums go over all ordered pairs $(i, j)$ such that $1 \le i < j \le l$ and either $j - i < \sigma$ or $j - i \ge \sigma$. By definition of $\sigma$, we obtain $\sigma \ge \lceil -\log D/c \rceil$, where $D = \frac{1}{\binom{n}{k}} - \frac{1}{q}$. For sufficiently large $\binom{n}{k}$, the second term of 2 is at most:

$$l\sigma D \le lD\lceil \log(1/D/c) \rceil < \frac{l}{c}\frac{1}{\binom{n}{k}} \log\begin{pmatrix}n\\k\end{pmatrix},$$

Concerning the third term of 2 we can observe that:

$$\sum_{i<j;j-i\ge\sigma} 2^{-c(j-i)} < l\frac{2^{-c\sigma}}{1-2^{-c}} < \frac{l}{\binom{n}{k}}\frac{1}{1-2^{-c}},$$

as $2^{-c\sigma} < \frac{1}{\binom{n}{k}}$. Finally, by combining these bounds with 2 we obtain the proof of Theorem 3.2.

Our choice of parameters $n$ and $k$ is justified by the study of Nguyen and Stern in [Nguyen and Stern 1999], where they used the discrete Fourier transform to prove that the distribution of the BPV output is indistinguishable from the uniform distribution, and this holds without the addition of the expander.

Given a fixed value of expected number of repetitions in a run of I-BPV of length $l$, in order to resist to Birthday attacks, the $n$ precomputed pairs should be periodically *refreshed*. In particular, a refresh operation should be performed after $l$ runs of I-BPV. Table I shows the expected number of repetitions for increasing values of $l$. Reported values are computed with parameters $k = 8$, $n = 160$, $q = 160$.

Table I. Expected number of repetitions in a run of I-BPV for increasing values of $l$.

| l | Expected repetitions |
|---|---|
| $3 * 10^2 - 3 * 10^3$ | $10^{-9}$ |
| $3 * 10^3 - 3 * 10^4$ | $10^{-8}$ |
| $3 * 10^4 - 3 * 10^5$ | $10^{-7}$ |
| $3 * 10^5 - 3 * 10^6$ | $10^{-6}$ |
| $3 * 10^6 - 3 * 10^7$ | $10^{-5}$ |

## 4. I-BPV OPTIMIZATIONS FOR ENERGY-HARVESTING WIRELESS SENSOR NETWORKS

Environmentally-powered nodes experience significantly changes in the power they harvest over time, due to varying weather conditions, monthly trends and seasonal patterns [Jeong and Culler 2012]. This results in an alternation between periods in which energy must be sparely used, and situations in which there may even be an excess of energy available, which would be wasted unless used in the short term. An *energy peak* occurs whenever a node is harvesting power at a rate that exceeds its current power consumption, while having its energy storage at capacity or, more generally, exceeding a given charging level threshold. More formally, an energy peak occurs at time $t$ if:

$$P_t^h > P_t^c \wedge E_t^s > E_{th}, \tag{3}$$

where $P_t^h$ is the amount of power being harvested at time $t$, $P_t^c$ is the power consumption of the node at time $t$, $E_t^s$ is charging level of the energy storage of the node at time $t$ and $E_{th}$ is the charging threshold (e.g., $E_{th}$ is typically the maximum amount of energy that can be stored in the supercapacitor or in the rechargeable battery of the node). Harvested energy would be lost whenever a node experiences an energy peak. In addition, supercapacitors, which are commonly used for energy storage, suffer from leakage, i.e., energy that is harvested and not used progressively leaks and is wasted. To reduce energy waste that occurs in these situations, we propose harvesting-enabled optimizations that leverage energy harvesting for precomputations to enhance performance.

### 4.1. Signature precomputations

This optimization aims at reducing the cost of performing an ECDSA signature by partially precomputing the combination of the terms produced by the generator (Section 2.3). It is applied whenever a node detects it is experiencing an energy peak, based on its storage level and harvesting rate. The optimization works as follows. Whenever there is an energy peak and there is free space in the RAM, $(\kappa_j, g^{\kappa_j})$ pairs are read from the flash, point multiplication is performed, and the precomputed result is stored in the node's RAM. The results of such precomputation can then be directly used to sign future messages, deallocating the corresponding RAM whenever a stored value is used. By precomputing point multiplications during periods of high energy availability, the energy cost and the time needed to perform an ECDSA signature can be significantly reduced (see Table III, Section 5.1.4). Precomputations are carried for the whole duration of the energy peak. If there is no space left to store precomputed results in the nodes' RAM, they can be stored in the node's flash (see Section 5.3.4).

### 4.2. Pairs refresh optimization

This optimization is meant to pro-actively exploit periods of high energy availability and energy peaks to reduce the energy cost of performing demanding computations. The most energy-expensive operation required by I-BPV is by far pairs refresh, which requires the nodes to perform $n$ modular exponentiations (see Table III, Section 5.1.4). However, pairs refresh is a critical operation required to maintain the security level of

Table II. Main characteristics of the MagoNode++, TelosB and MICA2 motes.

| | Telos B | MICA2 | MagoNode++ |
|---|---|---|---|
| Program memory (KB) | 48 | 128 | 128 |
| RAM (KB) | 10 | 4 | 16 |
| Nonvolatile storage (KB) | 1024 | 512 | 16384 |
| MCU Active power (mW) | 3 | 33 | 12.3 |
| Minimum Operation (V) | 1.8 | 2.7 | 1.8 |

I-BPV over time. By pushing modular exponentiations to energy harvesting periods, nodes can reduce energy waste and mitigate the cost of performing such demanding computations. The frequency with which pairs refresh should be performed depends on the security parameter $l$. In particular, according to Table I, in order to resist to Birthday attacks a refresh operation should be performed after a given number of signatures, between $S_{min}$ and $S_{max}$, have been performed. When the number of signatures generated by a node reaches the $S_{min}$ threshold, the node determines whether it can wait until the next predicted recharging opportunity to perform pairs refresh. In fact, the energy availability over time of environmental sources such as solar light and wind, even if non-controllable, can typically be predicted with a good level of accuracy. In order to do so, nodes run an energy prediction algorithm to forecast when the next high energy-availability phase will occur [Cammarano et al. 2016; Recas Piorno et al. 2009; Kansal et al. 2007]. The decision on waiting until the next recharge is based on both energy prediction and the expected rate of signature generation, which is estimated based on past history. If the number of signatures generated by the node reaches the $S_{max}$ threshold (e.g., because of a large and unexpected prediction error), pairs refresh is performed immediately.

## 5. PERFORMANCE EVALUATION

In this section, we systematically evaluate the performance of I-BPV in different settings by means of both simulation-based evaluation and experimental validation in a testbed of energy-harvesting wireless motes. In Section 5.1, we describe the implementation of I-BPV on a recent wireless sensor platform, the MagoNode++ [Paoli et al. 2016], and on two widely-deployed off-the-shelf wireless sensor node families, TelosB [Crossbow Technology 2004] and MICA2 [Crossbow Technology 2003] motes, and we thoroughly evaluate its performance in terms of cryptographic primitives performance and of energy consumption. Table II summarizes the characteristics of the considered platforms. The MagoNode++ is built upon the MagoNode [Paoli et al. 2014], a 802.15.4 compliant WSN mote operating in the ISM 2.4 GHz band. The MagoNode rev. B features the ATmega256RFR2 microcontroller and transceiver bundle and the Texas Instruments CC2530 radio front-end, which provides superior radio performance with low-power consumption. In addition, the MagoNode++ features an energy-harvesting subsystem composed by a light or thermoelectric harvester, a battery manager and a power manager module. It further integrates a state-of-the-art RF Wake-Up Receiver [Spenza et al. 2015] that enables low-latency asynchronous communication, virtually eliminating idle listening at the main transceiver. In section 5.3, we validated our proposed harvesting-enabled optimizations, confirming their effectiveness in further improving the performance of I-BPV.

### 5.1. Performance evaluation of I-BPV

Standard Elliptic Curve Cryptography (ECC) is typically implemented by using elliptic curves either over a large prime field (e.g., $\mathbb{F}_p$) or over a field of characteristic two (e.g., $\mathbb{F}_{2^m}$). Several standard based security protocols, such as TLS, support ECDSA implemented over both primary and binary fields. Among these two fields, we have selected two curves, based both on the security level they provide and on their computational efficiency. In the following, we describe the implementation of I-BPV on both ordinary elliptic curve of prime order, (e.g., Miyaji, Nakabayashi and Takano (MNT) non-supersingluar curves [Miyaji et al. 2001]) and on anomalous binary elliptic curves (e.g., Koblitz curves [Koblitz 1987]). In both cases, their group order is at least 160 bits to resist Pohlig-Hellman attacks [Pohlig and Hellman 1978].

*5.1.1. Implementation on MNT curve.* We have implemented I-BPV in nesC for the operating system TinyOS 2.x. In our implementation, based on the TinyECC library [Liu and Ning 2008], the elliptic curve is an MNT curve, which can be written in the simplified Weierstrass form as:

$$E(\mathbb{F}_p) : y^2 = x^3 + ax + b. \tag{4}$$

The elliptic curve E is defined over a prime field $\mathbb{F}_p$ where $p = 2^{160} - 2^{31} - 1$ as recommended by enisa[2], the European Union Agency for Network and Information Security. According to NIST, this guarantees a security level of 80 bits.

Due to the limited computational capabilities and the internal (RAM/ROM) memory constraints of sensor platforms, optimizations are necessary for a practical implementation. We used curve-specific optimizations to speed up modular multiplication and modular square, applicable to our case of group size $p$ being a pseudo Mersenne prime. To decrease the high computational cost of performing a modular inversion, elliptic curve operations are implemented in projective coordinates using Jacobian representation. The affine coordinates can be transformed into projective coordinates which use three elements to represent a point $(X, Y, Z)$, allowing the numerator and the denominator to be calculated separately. The elliptic curve defined in (4) is converted to Jacobian coordinates as follows:

$$E(\mathbb{F}_p) : Y^2 = X^3 + aXZ^4 + bZ^6, \tag{5}$$

where $X = xZ^2$, $Y = yZ^3$. We used the OS function to generate randomness. We also experimented with PRNGs and both HMAC-SHA1, as a PRF, and SHA-512 truncated at 384 bits to behave like a "random oracle" [Dodis and Puniya 2008].

*5.1.2. Implementation on Koblitz curves.* Following NIST recommendations, we have also implemented I-BPV on a Koblitz curve (sect163k1[3]) defined over $\mathbb{F}_{2^m}$:

$$E(\mathbb{F}_{2^m}) : y^2 + xy = x^3 + ax^2 + b, \tag{6}$$

where $m = 163$ and the representation of $\mathbb{F}_{2^{163}}$ is defined by:

$$f(x) = x^{163} + x^7 + x^6 + x^3 + 1. \tag{7}$$

The NIST irreducible polynomial for the finite field $\mathbb{F}_{2^{163}}$ allows us to exploit optimizations such as a fast modular reduction algorithm, Solinas' $\tau$-radic nonadjacent form (TNAF) representation [Solinas 2000] and an extensive use of the Frobenius map $\tau$. We based our implementation on Koblitz curves on RELIC[4], a modern cryptographic meta-toolkit with emphasis on efficiency and flexibility.

*5.1.3. Methodology.* To evaluate the performance of I-BPV, we have measured both its computational overhead, expressed in terms of time needed to perform the needed operations, and its energy consumption. We experimentally evaluated the computational overhead by performing selected operations 10000 times, and recording the time

[2] https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/algorithms-key-sizes-and-parameters-report
[3] http://www.secg.org/sec2-v2.pdf
[4] http://code.google.com/p/relic-toolkit

Table III. Computational overhead and energy consumption of ECC operations on different platforms.

| | Telos B | MICA2 | MagoNode++ |
|---|---|---|---|
| Exponentiation | 3701ms/19.98mJ | 2244ms/53.85mJ | 976ms/13.76mJ |
| Multiplication | 193ms/1.04mJ | 130ms/3.12mJ | 57ms/0.8mJ |
| Conversion of coordinates A/P/A | 179ms/0.97mJ | 121ms/2.90mJ | 53ms/0.74mJ |
| Precomputation | 292ms/1.6mJ | 192ms/4.6mJ | 78ms/1.09mJ |

needed to perform the overall cycle. This allows to estimate the average time needed to perform each operation. We derived power consumption of the nodes via in-lab measurements. In particular, we measured the current consumption of the node when its microcontroller (MCU) is in active mode and its radio is off. For all the three platforms, we have observed a negligible difference between actual measurements and the values reported in the datasheets.

*5.1.4. Cost of atomic Elliptic Curve operations.* Table III shows the computational overhead and energy cost of ECC operations over the MagoNode++, TelosB and MICA2 platforms. Consistently with Section 5.1.1, we use the multiplicative group notation. Estimation of energy cost is determined as detailed in Section 5.1.3. The two basic operations are exponentiation (i.e., computation of an EC group point $g^s$ with $s$ a randomly chosen integer in $[1, q-1]$) and multiplication between two randomly chosen group points. Exponentiation is used in ordinary ECDSA, whereas I-BPV only uses multiplications for signature generation.

Exponentiation is, as expected, the most expensive operation. Table III shows that one exponentiation is executed in about 3.7s over a Telos B mote, in about 2.2s over a MICA2 and in about 1s over a MagoNode++. The difference between these values is due to hardware differences between the three motes and to the effect of platform-specific optimizations of the assembly code. The energy consumption associated with exponentiation is $19.98\,\mathrm{mJ}$, $53.85\,\mathrm{mJ}$ and $13.76\,\mathrm{mJ}$ for TelosB, MICA2 and MagoNode++ motes, respectively. This large difference is mostly due to the different current consumption of the CPU of the three platforms ($1.8\,\mathrm{mA}$ for TelosB, $8\,\mathrm{mA}$ for MICA2 and $4.7\,\mathrm{mA}$ for MagoNode++). Our experimental evaluation also shows that a multiplication costs about a factor $17-19$ less than an exponentiation, in terms of both time and energy. A multiplication requires $193\,\mathrm{ms}$ with an energy consumption of $1.04\,\mathrm{mJ}$ on TelosB motes, $130\,\mathrm{ms}$ with an energy consumption of $3.12\,\mathrm{mJ}$ on MICA2 motes, and $57\,\mathrm{ms}$ with an energy consumption of $0.8\,\mathrm{mJ}$ on a MagoNode++. By themselves, these results might (erroneously) suggest that the saving in using precomputations might be limited to the case of up to $17-19$ terms. As shown later on in table IV this is *not* the case, and, for instance, 60 multiplications are performed in almost 1/4 of an exponentiation time. Indeed, our implementation performs (faster) operations in the Jacobian projective coordinates. The cost in converting from affine coordinates to projective coordinates and vice versa (labeled as "Conversion of coordinates A/P/A" in Table III) is thus a fixed overhead that applies once to both exponentiation and multiplication. This cost is non negligible, being, in terms of time, of 179ms on TelosB, of 121ms on MICA2 and 53ms on MagoNode++ platforms. Note that the conversion of coordinates affine → projective → affine accounts for almost all the cost of performing a multiplication, being the latter step (projective → affine) the dominant cost. Nevertheless, backward conversion to affine is recommended as security may be affected by leaving results in projective coordinates [Naccache et al. 2004]. The cost of online pairs generation as described in Algorithm 2 is shown in Table III as the precomputation cost. A precomputation is performed in 292ms on a Telos B, 192ms on a MICA2 and 78ms on a MagoNode++, consuming $1.6\,\mathrm{mJ}$, $4.6\,\mathrm{mJ}$ and $1.09\,\mathrm{mJ}$ for TelosB, MICA2 and MagoNode++ motes, respectively.

Table IV. Anatomy of an ECDSA signature cost using MNT curve: Experimental measurements on TelosB, MICA2 and MagoNode++.

| | Telos B | | | MICA2 | | | MagoNode++ | | |
|---|---|---|---|---|---|---|---|---|---|
| $k$ | $\sum \kappa_j$<br>mod $q$ | $\prod g^{\kappa_j}$ | ECDSA | $\sum \kappa_j$<br>mod $q$ | $\prod g^{\kappa_j}$ | ECDSA | $\sum \kappa_j$<br>mod $q$ | $\prod g^{\kappa_j}$ | ECDSA |
| 60 | 10ms<br>0.05mJ | 1026ms<br>5.54mJ | 1229ms<br>6.63mJ | not supported | | | 3ms<br>0.04mJ | 276ms<br>3.89mJ | 330ms<br>4.65mJ |
| 30 | 5ms<br>0.03mJ | 604ms<br>3.26mJ | 802ms<br>4.33mJ | 3ms<br>0.07mJ | 381ms<br>9.14mJ | 523ms<br>12.55mJ | 2ms<br>0.03mJ | 165ms<br>2.33mJ | 217ms<br>3.06mJ |
| 15 | 2ms<br>0.01mJ | 391ms<br>2.11mJ | 586ms<br>3.16mJ | 2ms<br>0.05mJ | 252ms<br>6.05mJ | 393ms<br>9.43mJ | 1ms<br>0.01mJ | 103ms<br>1.45mJ | 155ms<br>2.19mJ |
| 8 | 1ms $\epsilon$ | 291ms<br>1.57mJ | 485ms<br>2.62mJ | ≅1ms<br>≅0.02mJ | 191ms<br>4.58mJ | 331ms<br>7.94mJ | ≅1ms<br>0.01mJ | 77ms<br>1.09mJ | 128ms<br>1.8mJ |

Table V. Performance comparison with NTRUSign, other optimizations of ECDSA, and XTR-DSA (MICA2 motes). The last two rows report performance results of I-BPV on both MNT and Koblitz curves, in which ECDSA exponentiations are performed by using multiplications of precomputed elliptic curve points.

| Reference | Scheme | ROM | RAM | \|Sig\| | $\|key_{priv}\|$ | $\|key_{pub}\|$ | $t_{sign}$ | $E_{CPU}(t_{sign})$ |
|---|---|---|---|---|---|---|---|---|
| [Gura et al. 2004] | RSA | 7.4KB | 1.1KB | 128B | 128B | 131B | 10.99s | 263.8mJ |
| TinyECC [Liu and Ning 2008] | ECDSA | 19.3KB | 1.5KB | 40B | 21B | 40B | 2.001s | 48.1mJ |
| [Driessen et al. 2008] | NTRUSign | 11.3KB | 542B | 127B | 383B | 127B | 0.619s | 22.3mJ |
| | ECDSA | 43.2KB | 3.2KB | 40B | 21B | 40B | 0.918s | 22.0mJ |
| | XTR-DSA | 24.3KB | 1.6KB | 40B | 20B | 176B | 0.965s | 23.2mJ |
| This work (MNT) | ECDSA | 18.2KB | 1.2KB | 40B | 21B | 40B | 0.346s | 8.1mJ |
| This work (Koblitz) | ECDSA | 64.5KB | 1.8KB | 40B | 21B | 40B | 0.298s | 6.9mJ |

*5.1.5. Cost of I-BPV for precomputation-based ECDSA.* The cost in terms of memory, time, and energy consumption of I-BPV depends on the parameters used in the generator, i.e., the number $n$ of precomputed pairs $(\kappa_j, g^{\kappa_j})$, the number $n_e$ of the elements $(d_j, g^{d_j})$ comprising the set used for the random walk over the Cayley graph expander, and the number $k$ of elements drawn at each signature. The parameters $n$ and $n_e$ are only related to storage, and hence do not impact the cost of an ECDSA signature in terms of time and energy consumption. Parameter $k$, instead, affects performance results as it is related to the number of multiplications to be performed.

Table IV provides an overview of the various time/energy costs involved in an ECDSA signature with I-BPV, along with the cost of the whole signature, for four values of the security parameter $k$, with $n = 160$. Specifically, the table reports, for each sensor node platform, the time and energy consumption needed to perform: i) the modular sum of the coefficients $\kappa_j \in \mathbb{Z}_q$, ii) the product of the $k$ corresponding elliptic curve points $g^{\kappa_j}$, and iii) the total ECDSA signature cost. Results show that the cost, as expected, grows with the size of the parameter $k$, but it remains significantly lower than the cost of an exponentiation even for large $k$. Pairs are assumed to be stored in RAM. Note that the first row for MICA2 is left blank because MICA2 mote can not store $k = 60$ entirely in RAM.

*5.1.6. Comparison with other techniques.* Table V compares the performance attained by our I-BPV-based ECDSA signature in both MNT and Koblitz versions (parameters: $n = 160$, $n_e = 32$, and $k = 8$) with alternative signatures, as well as other ECDSA implementations. The data reported in the table for benchmarking schemes are adapted from [Driessen et al. 2008], which provides a comparative assessment of the reported schemes when implemented over a MICAz sensor node (which uses the same micro-

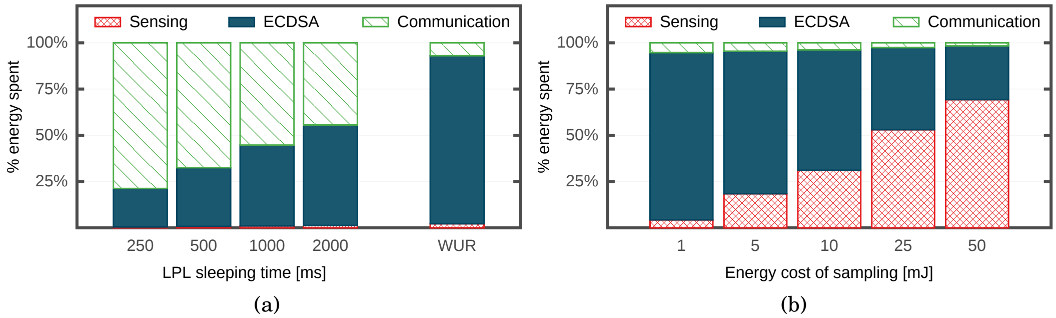


Fig. 1. Percentage of per-day energy consumption associated with sensing, communication and signature generation using standard ECDSA without precomputations in scenarios with: a) different cost of sensing; and b) different cost of communication, i.e., using a wake-up radio (WUR) or with different duty cycle values.

controller employed by the MICA2). The remaining entries in the table are provided for the reader's convenience, and report the size (Sig), in bytes, of the considered signatures, along with the size, in bytes, of the private key ($key_{priv}$) and the public key ($key_{pub}$). In the comparison, we further account for the fact that the $n$ precomputed pairs $(\kappa_j, g^{\kappa_j})$ and the $n_e$ pairs $(d_j, g^{d_j})$ cannot be entirely stored in the RAM. Indeed, MICA2 motes have only a 4KB RAM, whereas each pair requires 63 bytes of memory: 19 bytes for the integers $\kappa_i$, and 22 bytes for each of the two coordinates of the elliptic curve points $g^{\kappa_i}$. Even if 2.8 KB were in principle available (the implementation of our scheme requires 18.2KB of ROM and 1.2KB of RAM for the MNT version and 64.5KB of ROM and 1.8KB of RAM for the Koblitz version), we considered the worst-case approach of storing *all* the pairs in the flash memory. Access to the flash brings about an extra time/energy cost. Specifically, reading one pair takes 1.94 ms and causes an energy consumption of $0.023\,\mathrm{mJ}$. This supplementary flash access overhead explains the slightly worse results with respect to the performance reported in Table IV for the same setting of the parameters.

Table V clearly shows that precomputation permits to significantly outperform other scheme reported in the table: our I-BPV-based signature over Koblitz curves is *three times faster* than the best ECDSA implementation reported in the table, and it is *twice as fast* as than NTRU$_{\mathrm{SIGN}}$. Similar improvements are shown also in terms of energy consumption.

### 5.2. Relative energy cost of ECDSA signatures

To better motivate the need of optimizing ECDSA signatures, we evaluate the relative energy cost of performing ECDSA signatures with respect to the cost of sensing and communication in different scenarios.

We run simulations using GreenCastalia [Benedetti et al. 2013], an open-source extension of the Castalia simulator [Boulis 2007] that we develop for accurate modeling of energy-harvesting WSNs. The energy model we use is that of TelosB, which uses the IEEE 802.15.4-compliant CC2420 transceiver. For accurate modeling of time and energy consumption of security-related operations, we further extend GreenCastalia to include a realistic model of the microcontroller of TelosB motes, according to which time and energy consumption of security-related operations are modeled based on experimental measurements (Section 5.1). In addition, we also provide simple models to account for time and energy spent to read/write from/to the flash and for sensing activity.

We consider a Structural Health Monitoring application in which nodes in the network are deployed for monitoring of a critical structure, such a bridge. As a practical example, we consider Telos B nodes equipped with on-board Sensirion SHT1x sensors that perform temperature and humidity measurements twice per minute. Based on the sensor datasheet specifications, we set the power consumption of sensing to $3$mW, and the time needed for the measurement to complete to $171$ms [Sensirion AG 2011]. A data packet containing sample measurements is then generated, signed to ensure integrity and authentication, and sent to the sink. We use the default settings of GreenCastalia for channel and radio models. The channel data rate is set to 250 Kbps. The average path loss between nodes in the network follows the lognormal shadowing model. Packet reception probability for each link is computed based on SINR, packet size, and modulation type. The additive interference model is used, so that the effect of simultaneous transmissions from multiple nodes is linearly added at the receiver. Simulations are run for ten days.

Figure 1(a) reports the percentage of per-day energy consumption associated with sensing, communication and signature generation (using standard ECDSA without precomputations) in scenarios with different energy cost of communication. In particular, we consider both the cases in which duty cycling is employed and in which a wake-up receiver (WUR) is used for on-demand communication. We implemented duty cycling in GreenCastalia through the Low Power Listening (LPL) MAC-layer technique. We set LPL parameters based on the TinyOS 2.1 implementation of BoX-MAC-2 [Moss and Levis 2008]. Nodes using LPL follow asynchronous wake-up schedules, performing periodic receive checks every $l$ms, $l \in \{250, 500, 1000, 2000\}$. In other words, nodes sleep for $250$ms, $500$ms, $1$s or $2$s between successive checks for channel activity. In the scenario in which nodes use duty cycling, the relative energy cost of performing ECDSA signatures varies between 20% and 55% of the total energy spent by the node, depending on the value of $l$. When using a wake-up receiver with nano ampere current consumption, such as [Spenza et al. 2015], rather than duty-cycle-based communication, the relative cost of performing ECDSA signatures is as high as 90% of the total energy consumed by the node.

Figure 1(b) shows the percentage of per-day energy consumption associated with sensing, communication and signature generation in scenarios with different energy cost of sensing. In these tests, motes use a wake-up receiver. Depending on the energy cost of sensing, the relative energy cost of performing ECDSA signatures varies between 29% and 90% of the total energy spent by the node.

### 5.3. Performance evaluation of I-BPV with harvesting-enabled optimizations

To assess the performance improvement achieved by using our proposed harvesting-enabled optimizations, we carried out both simulations-based experiments using real-life energy traces, and practical experiments in a testbed of solar-powered Telos B motes.

*5.3.1. Harvesting-aware optimizations.* In this set of experiments, we implement the harvesting-aware optimizations described in Section 4, and estimate their impact on reducing the energy toll associated with security operations.

We implemented I-BPV in GreenCastalia. In simulations, all the precomputed pairs, accounting to about $12$ KB, are stored in the flash memory of the nodes. This is a worst case scenario, as part of them could be stored in the RAM. The storage requirement is determined as follows: The number $n$ of pairs $(k, g^{\kappa_i})$, each using $63$ bytes, must be set to a value not lower than 160, the size in bit of the Elliptic Curve group, to prevent lattice reduction attacks [Nguyen and Stern 1999]. Thanks to our optimization

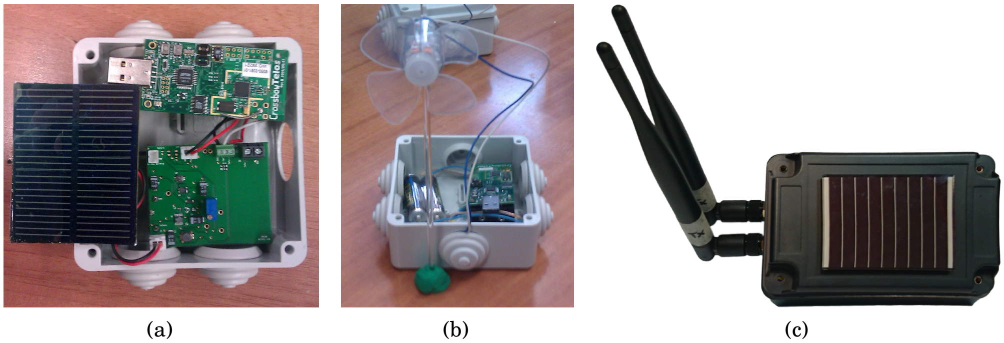


Fig. 2. Telos B motes interfaced with (a) photovoltaic cell and (b) micro wind turbine, and (c) MagoNode++ with photovoltaic cell.

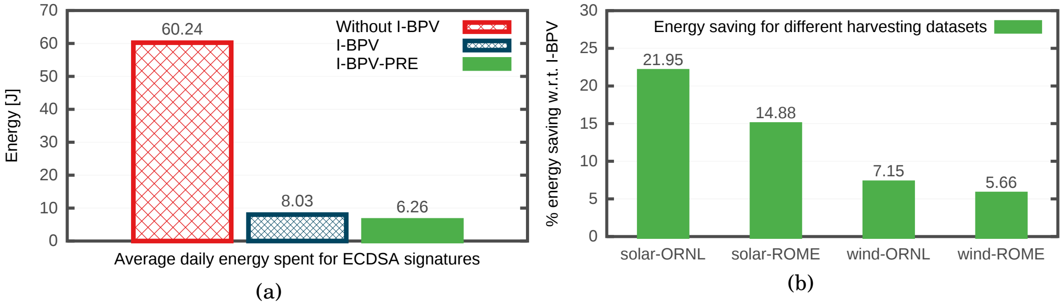
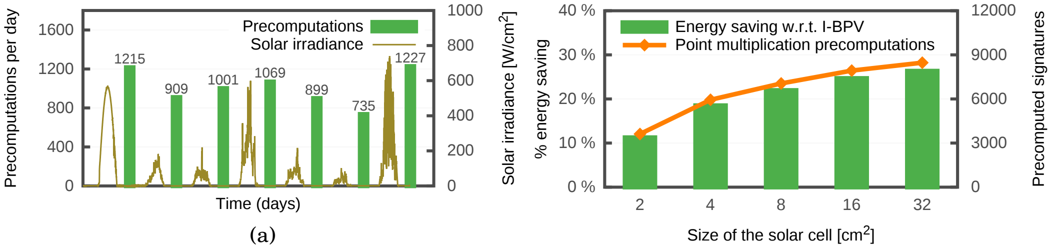


Fig. 3. Effect of harvesting-aware optimizations on energy spent to sign messages: (a) Average energy spent per day by standard ECDSA (no optimization), by I-BPV and by I-BPV with point multiplication precomputations (I-BPV-PRE); (b) Energy saving with point multiplication precomputations in different solar and wind harvesting scenarios.

(Section 3), the number $n_e$ of supplementary pairs for constructing the Cayley graph are set to $32$, one fifth of the group size in bits. Hence, $192 \times 63$ bytes are used in total.

To simulate energy harvesting, we obtained real-life solar and wind traces by interfacing Telos B motes with photovoltaic cells (Fig. 2(a)) and with wind micro turbines (Fig. 2(b)). Additional solar and wind harvesting data from the National Renewable Energy Laboratory (NREL) at Oak Ridge, Tennessee [NREL: Measurement and Instrumentation Data Center 2011] were also used in simulations. In order to use such traces in our performance evaluation, we converted raw weather data, i.e., irradiance and wind speed values, into energy harvesting estimations. In particular, we calculate the power $P_s$ harvested by a solar cell of size $A$ and efficiency $\eta$ as: $P_s = A \cdot \eta \cdot I$, where $I$ is the radiant energy incident onto surface. For wind energy harvesting, we estimate the output power $P_w$ of the wind micro turbine as: $P_w = 0.5 \cdot v^3 \cdot A \cdot \rho \cdot C_p$, where $v$ is the wind speed in m/s, $A$ is the rotor swept area in m$^2$, $\rho$ is the air density (typically $1.25$ kg/m$^3$), and $C_p < 1$ is the power extraction coefficient.

*5.3.2. Signature precomputations.* In this simulations, we consider the same application scenario detailed in Section 5.2, in which the nodes are also equipped with energy harvesters. Simulations are run for ten days by using different energy-harvesting datasets. In this setting, the average energy spent per day to sign messages is more than 60J for ECDSA without I-BPV and of around 8J with I-BPV, resulting in a 86% reduction in energy consumption (Figure 3(a)). These results are computed based on the energy spent by the MCU of the node to perform security operations and on the en-



(a)

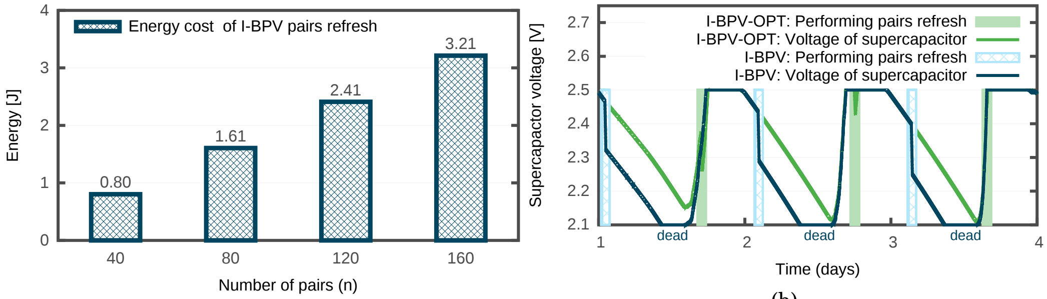
(b)

Fig. 4. Effect of energy availability on point multiplication precomputations: (a) Number of precomputations performed per day by I-BPV-PRE over 7 days with variable harvesting conditions; (b) Impact of the solar cell size on number of precomputations and energy saving obtained by I-BPV-PRE.

ergy spent for reading from and writing to the node's flash. Energy harvested in excess, when available, is used to precompute point multiplications as detailed in Section 4.1. Using I-BPV with point multiplication precomputations (denoted as I-BPV-PRE in the following) allows to further reduce the average energy spent to sign message of up to an additional 22%. This results in an energy saving of approximately a factor 10 with respect to the case in which I-BPV is not used and non-optimized ECDSA signatures are employed. Figure 3(b) shows the additional energy saving (in percentage) w.r.t. I-BPV that is achieved when using different energy harvesting datasets. As expected, the number of precomputations performed by I-BPV-PRE, and thus the energy saving it achieves, is higher in the solar energy harvesting scenario than in the wind energy harvesting scenario. This is due to the higher amount of energy harvested by solar cells with respect to wind micro turbines, which results in a greater number of energy peaks. Fig. 4(a) details the energy harvesting profile of a node, and the number of precomputations performed during each day. Since the number of point multiplication precomputations performed by the nodes depends on the power harvested during the day, in the same scenario we also evaluate the impact of the solar cell size (and thus on the amount of energy harvested by the nodes) on the number of precomputations performed by I-BPV-PRE. Fig. 4(b) shows the energy saving achieved by I-BPV-PRE w.r.t. I-BPV and the average number of point precomputations performed per day by a node powered by a solar cell whose size is varied between 2 and 32 cm$^2$.

In scenario in which the harvesting energy availability is limited, such as in the wind harvesting one, the number of point precomputations performed by I-BPV-PRE also depends on the energy charging threshold $E_{th}$. For example, in the wind-ROME scenario I-BPV-PRE performs an average of 173 point precomputations per day when the energy charging threshold is set to the maximum amount of energy that can be stored in the supercapacitor of the node. In this case, an energy peak only occurs if the energy storage is fully charged while the node is harvesting power at a rate that exceeds its current power consumption. Using less conservative values for $E_{th}$ increases the average number of precomputations. For example, setting $E_{th} = 90\%$ results in around $20\%$ more precomputations performed per day with respect to the case in which precomputations are performed only when the energy storage is fully charged.

*5.3.3. Pairs refresh optimization.* In the last set of experiments, we evaluate the impact of the pairs refresh optimization proposed in Section 4.2 in terms of efficient utilization of the harvested energy. Figure 5(a) shows the energy spent by I-BPV for pairs refresh (without harvesting-aware optimizations). The pool size n has been varied between 40 and 160. The energy spent for pairs refresh varies between $0.80$ J and $3.21$ J, depending on the pool size. Results show similar trends when considering the MagoNode++ and



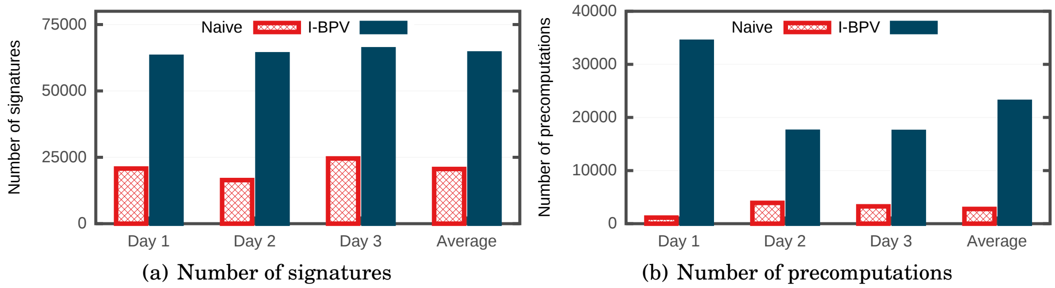
(a) (b)

Fig. 5. Impact of pairs refresh optimization on efficient utilization of harvested energy: (a) Energy spent for pairs refresh by I-BPV for different size of the pairs pool; (b) Three-days snapshot of the supercapacitor voltage of a node with pairs refresh optimization (I-BPV-OPT) and without pairs refresh optimization (I-BPV). With I-BPV-OPT, pairs refresh is performed during periods of high energy availability, using harvested energy in excess without consuming energy stored in the supercapacitor. This significantly reduces the amount of “dead’ time.

the MICA2 platforms. In particular, when n=160, refreshing the pool of pairs requires $2.22\,\mathrm{J}$ of energy on a MagoNode++ and $8.62\,\mathrm{J}$ of energy on a MICA2 mote.

To assess the impact of harvesting-aware optimizations, we use the same setup as detailed in Section 5.3.2, and consider energy-harvesting nodes powered by an harvesting subsystem that includes a solar cell of size 2cm$^2$ and a 10F Panasonic Gold supercapacitor. Nodes estimate when the next high energy-availability phase will occur by using the Pro-Energy energy prediction algorithm [Cammarano et al. 2016]. We measure the amount of time during which each node is considered “dead” due to its capacitor being empty. When using I-BPV, performing pairs refresh requires significant amount of energy from the supercapacitor, which results in nodes being “dead” for more than 36 hours over our 10-days experiment. When using pairs refresh optimization, pairs refresh is performed during periods of high energy availability, which allows to directly use harvested energy in excess without consuming stored energy. This matching between energy consumption and energy harvesting profile significantly reduces the amount of “dead’ time from more than 36 hours to less than half an hour over a 10-days experiment. Figure 5(b) depicts a three-days snapshot of traces extracted from simulations, which show that pairs refresh optimization allows better usage of the harvested energy.

*5.3.4. I-BPV optimizations vs. naive approaches: experimental evaluation.* In this section, we discuss and motivate the use of the harvesting-aware optimization presented in Section 4.1 with respect to a naive approach, in which the harvested energy is used to directly compute a full exponentiation as part of precomputations. The naive solution works as follows: extra energy, which would be wasted if not used, is used to compute pairs $(\kappa, g^\kappa)$, which are then stored in RAM and FLASH until they are both filled up. We show that this naive solution, although seeming quite promising as it requires no extra storage, is much less performing than our approach. To this end, we consider a simple application in which each node is required to continuously sense and send signed messages containing sensors measurements. The node has two main states: either having extra energy from harvesting or not. When it is in the first state, it uses energy in excess to populate the available memory with pairs. Its only task is thus to precompute and store pairs $(\kappa, g^\kappa)$. If no extra energy is available, the node keep generating sensing data and transmitting signed messages. In the event the node uses up all pairs, it will compute new ones on the fly. That is, the node first generates a pair $(\kappa, g^\kappa)$



(a) Number of signatures

(b) Number of precomputations

Fig. 6. Experimental comparison of I-BPV and naive approach over three days: signatures and precomputations performed daily by a solar-powered TelosB mote.

and then uses it to compute a single signature. The on-the-fly generation continues until the node reaches the first state again. We compare the performance of computing pairs $(\kappa, g^{\kappa})$ by using full exponentiation (naive approach) and with our I-BPV-based technique (with $n = 160$ and $k = 8$). To this end, we evaluate how many signatures the node can generate and transmit in a typical day using these two strategies. We run such an experiment in a testbed of solar-powered motes with energy-harvesting capabilities. The sensor node is built around the Telos B platform, as shown in Figure 2(a), and it includes an harvesting subsystem composed of a 0.5W solar panel, a custom charging board and three rechargeable 1.2 V / 2450 mAh batteries. The harvesting subsystem also implements a maximum power point controller (MPPC), which dynamically maximizes the harvesting efficiency, in order to get as much power as possible from the solar panel under any lighting condition. Nodes can determine whether they are experiencing an energy peak by sampling the voltage of their solar cell and energy storage through dedicated test points connected to the ADC input ports of the Telos B mote. A direct voltage look up table is used to determine the current harvesting power based on the harvester's voltage. We implemented in TinyOS both the naive solution and I-BPV with the proposed harvesting-aware optimizations, and deployed outdoors two solar-powered motes. Figure 6 shows the number of signatures and the number of precomputations performed daily by the motes over a snapshot of three days by our I-BPV-based method and by the naive approach. Results show that, on average, I-BPV outperforms the naive approach by a factor of 3 in terms of number of signatures per day, and up to a factor of 30 in terms of precomputations.

## 6. RELATED WORKS

Effective precomputation techniques have been proposed in the past [Brickell et al. 1993; Rooij 1995] to accelerate modular exponentiations at the basis of several standard signature and key management schemes, such as the (Elliptic Curve) Digital Signature Algorithm and the Diffie-Hellman key exchange protocol. Despite their promises, however, the actual application of such techniques in the IoT and WSN security arena has been apparently overlooked. Rather, driven by the common goal of energy conservation, most of the research effort in this field has specifically targeted the design of alternative energy-friendly lightweight security primitives [Dini and Savino 2011; Zia and Zomaya 2011]. In many IoT and WSN application scenarios, however, using standardized security constructions remains the approach of choice, being supported by both rigorous security analyses and long-lasting real-world practice. The Elliptic Curve Digital Signature Algorithm is a standard security primitive that has received widespread consideration in emerging security protocols for low-power devices (see, e.g., the IETF working groups ROLL and CORE-CoAP), due to

ECDSA signatures being significantly cheaper than RSA signatures at the same security level [Wander et al. 2005]. Previous works, such as [Driessen et al. 2008], have presented ECDSA implementations for Wireless Sensor Networks. However, delay and energy consumption of generating ECDSA signatures are still significant, currently being in the order of seconds and tens of mJ, respectively [Capossele et al. 2015]. For this reason, solutions specifically tailored to WSNs have been proposed to avoid intensive use of ECDSA [Liu et al. 2012]. To the best of our knowledge, the possibility of exploiting energy harvesting to speed up generation of ECDSA signatures, thus reducing its energy toll, has so far be investigated only by our prior work [Ateniese et al. 2013]. Indeed, despite the extremely rich literature on solutions specifically tailored to WSNs [Ren et al. 2011; Zhou et al. 2008], to date only a handful of works have focused on the possibility offered by energy harvesting to support and improve security schemes. One of the earliest work in this field is the optimization mechanism proposed by Taddeo et al. in [Taddeo et al. 2010]. Their proposed scheme enables to dynamically change communication security settings of an energy-harvesting wireless sensor network (EH-WSN) over time based on the energy state of the network. Pabbuleti et al. investigate ECC-based and hash-based signature schemes on autonomous, energy-harvesting sensor nodes, demonstrating the trade-off between computation energy and communication energy in PKC signature schemes [Pabbuleti et al. 2014]. Use of precomputations with partitioned execution modes has been recently investigated by Aysu and Schaumont on constrained energy-harvesting platforms as a potential optimization technique for a post-quantum hash-based signature scheme [Aysu and Schaumont 2015]. Bianchi et al. propose a solution for data access control in EH-WSNs deployed for health care and assisted living applications [Bianchi et al. 2013], which combines smart caching and energy intake prediction to make computationally-heavy asymmetric cryptography schemes feasible in real WSNs with energy harvesting. In [Shakhov et al. 2013], Shakhov et al. investigate survivability of EH-WSNs nodes under flooding-based attacks and discuss counteracting methods against them. Lim and Huie propose a countermeasure to selective forwarding attack in energy harvesting WSNs in [Lim and Huie 2015]. A more comprehensive taxonomy of attacks specifically focused on energy-harvesting scenario is provided in [Kang et al. 2015].

Energy prediction models, such as [Cammarano et al. 2016; Recas Piorno et al. 2009; Cammarano et al. 2013; Kansal et al. 2007], are widely used in energy-harvesting WSNs as a building block of harvesting-aware solutions. For example, algorithms for task scheduling [Zhang et al. 2015] and task allocation [Porta et al. 2014], harvesting-aware communication protocols [Le et al. 2013] and power management strategies based on dynamic load adaptation [Renner et al. 2014; Mohaqeqi et al. 2013] have been proposed that use energy prediction to optimize the usage of the harvested energy. However, prediction-based approaches are still seldom used to support security solutions.

## 7. CONCLUSIONS

In this paper, with focus on a concrete implementation of an ECDSA signature over three mote platforms (MagoNode++, TelosB and MICA2) and its extensive assessment, we have shown that precomputations permit to significantly reduce the energy cost and accelerate the speed of signatures in wireless sensor nodes. By using MNT curves, we achieved an ECDSA-signature generation time below $350\,\mathrm{ms}$ over MICA2 motes, with an energy consumption below $10\,\mathrm{mJ}$. We further improved these results by using Koblitz curves, generating a signature in $300\,\mathrm{ms}$ and consuming about $7\,\mathrm{mJ}$. Our outcomes have shown that, with precomputations, an ECDSA signature attains performance superior to lightweight approaches such as NTRUsign. We believe these results make a significant case for considering precomputation techniques for making standard signatures practical in the wireless sensor domain, rather than choosing alternative signature schemes. As a further argument in favor of precomputation, we pointed out the emergence of energy harvesting technologies that opportunistically draw energy from the environment. We implemented optimizations by leveraging the energy that micro solar cells and wind microturbines can make available to cryptographic processing. Specifically, we applied precomputation techniques moving the computation of the most resource demanding operations to times when the energy is at peak. Through simulations and real-life experimentation, we showed that harvesting-enabled optimizations can significantly improve the performance of the system. Given these promising results, we believe that the exploitation of harvested energy for security protocols is a very compelling playground for future creative constructions.

## REFERENCES

Cristina Alcaraz, Pablo Najera, Javier Lopez, and Rodrigo Roman. 2010. Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration?. In *Proceedings of SecIoT 2010*. Tokyo (Japan).

N. Alon and Y Roichman. 1994. Random Cayley graphs and expanders. *Random Structures Algorithms* 5, 2 (1994), 271–284.

Giuseppe Ateniese, Giuseppe Bianchi, Angelo Capossele, and Chiara Petrioli. 2013. Low-cost Standard Signatures in Wireless Sensor Networks: A Case for Reviving Pre-computation Techniques?. In *Proceedings of NDSS 2013*. San Diego, CA.

Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The Internet of Things: A survey. *Comput. Netw.* 54, 15 (Oct. 2010), 2787–2805.

Aydin Aysu and Patrick Schaumont. 2015. Precomputation Methods for Hash-based Signatures on Energy-Harvesting Platforms. *IEEE Trans. Comput.* (2015).

Stefano Basagni, M. Yousof Naderi, Chiara Petrioli, and Dora Spenza. 2013. Wireless Sensor Networks with Energy Harvesting. In *Mobile Ad Hoc Networking: The Cutting Edge Directions*. John Wiley and Sons, Inc., Hoboken, NJ, Chapter 20, 701–736.

David Benedetti, Chiara Petrioli, and Dora Spenza. 2013. GreenCastalia: An Energy-harvesting-enabled Framework for the Castalia Simulator. In *Proceedings of ACM ENSSys 2013*. Rome, Italy, 7:1–7:6.

Giuseppe Bianchi, Angelo T. Capossele, Chiara Petrioli, and Dora Spenza. 2013. AGREE: exploiting energy harvesting to support data-centric access control in WSNs. *Elsevier Ad Hoc Networks* 11, 8 (2013), 2625 – 2636.

Kemal Bicakci, Ibrahim Ethem Bagci, and Bulent Tavli. 2012. Communication/computation tradeoffs for prolonging network lifetime in wireless sensor networks: The case of digital signatures. *Information Sciences* 188, 0 (2012), 44 – 63.

Reinhard Bischoff, Jonas Meyer, and Glauco Feltrin. 2009. Wireless Sensor Network Platforms. In *Encyclopedia of Structural Health Monitoring*. John Wiley & Sons, Ltd.

Athanassios Boulis. 2007. Castalia: Revealing Pitfalls in Designing Distributed Algorithms in WSN. In *Proceedings of ACM SenSys 2007*. Sydney, Australia, 407–408.

Victor Boyko, Marcus Peinado, and Ramarathnam Venkatesan. 1998. Speeding up Discrete Log and Factoring Based Schemes via Precomputations. In *Proceedings of EUROCRYPT 1998*. Springer, Santa Barbara, California, USA, 221–235.

Ernest Brickell, Daniel Gordon, Kevin McCurley, and David Wilson. 1993. Fast Exponentiation with Precomputation. In *Proceedings of EUROCRYPT 1992*, Vol. 658. Santa Barbara, California, USA, 200–207.

Alessandro Cammarano, Chiara Petrioli, and Dora Spenza. 2013. Improving Energy Predictions in EH-WSNs with Pro-Energy-VLT. In *Proceedings of ACM SenSys 2013, Poster Session*. New York, NY, USA, 41:1–41:2.

Alessandro Cammarano, Chiara Petrioli, and Dora Spenza. 2016. Online Energy Harvesting Prediction in Environmentally-Powered Wireless Sensor Networks. *IEEE Sensors Journal* 16, 17 (Sept 2016), 6793–6804.

A. T. Capossele, V. Cervo, G. De Cicco, and C. Petrioli. 2015. Security as a CoAP resource: an optimized DTLS implementation for the IoT. In *Proceedings of the IEEE International Conference on Communications, ICC 2015*.

Delphine Christin, Andreas Reinhardt, Parag Mogre, and Ralf Steinmetz. 2009. Wireless Sensor Networks and the Internet of Things: Selected Challenges. In *Proceedings of the 8th GI/ITG KuVS Fachge-*

*spräch Ďrahtlose Sensornetze, Hamburg, Germany*, Technische Universität Hamburg-Harburg Institut für Telematik (Ed.). 31–34.

Demetres Christofides and Klas Markstrom. 2008. Expansion properties of random Cayley graphs and vertex transitive graphs via matrix martingales. *Random Structures Algorithms* 32, 1 (2008), 271–284.

Jean Sebastien Coron, David M. Raihi, and Christophe Tymen. 2001. Fast Generation of Pairs (k, [k]P) for Koblitz Elliptic Curves. In *Selected Areas in Cryptography*. Springer-Verlag, London, UK, 151–164.

Crossbow Technology. 2003. MICA2 MOTE PLATFORM Datasheet. (2003). Document Part Number: 6020-0042-04.

Crossbow Technology. 2004. TELOSB MOTE PLATFORM Datasheet. (2004). Document Part Number: 6020-0094-01 Rev B.

Gianluca Dini and Ida M. Savino. 2011. LARK: A Lightweight Authenticated ReKeying Scheme for Clustered Wireless Sensor Networks. *ACM Transactions on Embedded Computing Systems* 10, 4 (2011), 41:1–41:35.

Yevgeniy Dodis and Prashant Puniya. 2008. Getting the Best Out of Existing Hash Functions; or What if We Are Stuck with SHA? In *Applied Cryptography and Network Security*, StevenM. Bellovin, Rosario Gennaro, Angelos Keromytis, and Moti Yung (Eds.). Lecture Notes in Computer Science, Vol. 5037. Springer Berlin Heidelberg, 156–173.

B. Driessen, A. Poschmann, and C. Paar. 2008. Comparison of Innovative Signature Algorithms for WSNs. In *Proceedings of ACM WiSec 2008*. Alexandria, Virginia, USA.

Craig Gentry and Michael Szydlo. 2002. Cryptanalysis of the Revised NTRU Signature Scheme. In *Proceedings of EUROCRYPT 2002*. Amsterdam, The Netherlands, 299–320.

Nils Gura, Arun Patel, Arvinderpal Wander, Hans Eberle, and SheuelingChang Shantz. 2004. Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. In *Proceedings of CHES 2004*, Marc Joye and Jean-Jacques Quisquater (Eds.). Vol. 3156. 119–132.

René Hummen, Jan H. Ziegeldorf, Hossein Shafagh, Shahid Raza, and Klaus Wehrle. 2013. Towards Viable Certificate-based Authentication for the Internet of Things. In *Proceedings of ACM HotWiSec 2013*. New York, NY, USA, 37–42.

Jaein Jeong and David Culler. 2012. Predicting the Long-Term Behavior of a Micro-Solar Power System. *ACM Transactions on Embedded Computing Systems* 11, 2, Article 35 (July 2012), 38 pages.

Don B. Johnson and Alfred J. Menezes. 1998. Elliptic curve DSA (ECSDA): an enhanced DSA. In *Proceedings of USENIX SSYM 1998*. Berkeley, CA, USA, 13–13.

J. Kang, R. Yu, S. Maharjan, Y. Zhang, X. Huang, S. Xie, H. Bogucka, and S. Gjessing. 2015. Toward secure energy harvesting cooperative networks. *IEEE Communications Magazine* 53, 8 (August 2015), 114–121.

Aman Kansal, Jason Hsu, Sadaf Zahedi, and Mani B. Srivastava. 2007. Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems* 6, 4 (2007), 32.

Neal Koblitz. 1987. Elliptic curve cryptosystems. *Mathematics of computation* 48, 177 (1987), 203–209.

Trong Nhan Le, Michele Magno, Alain Pegatoquet, Olivier Berder, Olivier Sentieys, and Emanuel Popovici. 2013. Ultra Low Power Asynchronous MAC Protocol Using Wake-up Radio for Energy Neutral WSN. In *Proceedings of ACM ENSSys 2013*. Rome, Italy, 10:1–10:6.

Sunho Lim and L. Huie. 2015. Hop-by-Hop cooperative detection of selective forwarding attacks in energy harvesting wireless sensor networks. In *Proceedings of IEEE ICNC 2015*. 315–319.

An Liu and Peng Ning. 2008. TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. In *Proceedings of IEEE IPSN 2008*. Washington, DC, USA, 245–256.

Yongsheng Liu, Jie Li, and M. Guizani. 2012. PKC Based Broadcast Authentication using Signature Amortization for WSNs. *IEEE Transactions on Wireless Communications* 11, 6 (June 2012), 2106–2115.

Javier Lopez, Rodrigo Roman, and Cristina Alcaraz. 2009. Analysis of Security Threats, Requirements, Technologies and Standards in Wireless Sensor Networks. In *Foundations of Security Analysis and Design V*, Vol. 5705. 289–338.

J. Misic. 2009. Cost of secure sensing in IEEE 802.15.4 networks. *IEEE Transactions on Wireless Communications* 8, 5 (May 2009), 2494–2504.

Atsuko Miyaji, Masaki Nakabayashi, and Shunzou Takano. 2001. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE transactions on fundamentals of electronics, communications and computer sciences* 84, 5 (2001), 1234–1243.

Morteza Mohaqeqi, Mehdi Kargahi, and Maryam Dehghan. 2013. Adaptive Scheduling of Real-time Systems Cosupplied by Renewable and Nonrenewable Energy Sources. *ACM Transactions on Embedded Computing Systems* 13, 1s, Article 36 (Dec. 2013), 28 pages.

D. Moss and P. Levis. 2008. *BoX-MACs: Exploiting Physical and Link Layer Boundaries in Low-power Networking*. Technical Report SING-08-00. Stanford University, Stanford, CA.

David Naccache, Nigel Smart, and Jacques Stern. 2004. Projective Coordinates Leak. In *Proceedings of EuroCrypt 2004*. Springer Verlag LNCS 3027, Interlaken, Switzerland, 257–267.

Phong Nguyen, Igor Shparlinski, and Jacques Stern. 1999. Distribution of modular sums and the security of server aided exponentiation. In *Proceedings of the Workshop on Comp. Number Theory and Crypt.* Springer, Singapore, 1–16.

Phong Nguyen and Jacques Stern. 1999. The Hardness of the Hidden Subset Sum Problem and Its Cryptographic Implications. In *Proceedings of CRYPTO 1999*, Vol. 1666. Santa Barbara, California, USA, 786–786.

NREL: Measurement and Instrumentation Data Center 2011. NREL: Measurement and Instrumentation Data Center. (2011). http://www.nrel.gov/midc/.

Krishna Pabbuleti, Deepak Mane, and Patrick Schaumont. 2014. Energy Budget Analysis for Signature Protocols on a Self-powered Wireless Sensor Node. In *Radio Frequency Identification: Security and Privacy Issues*, Nitesh Saxena and Ahmad-Reza Sadeghi (Eds.). Vol. 8651. 123–136.

Mario Paoli, Antonio Lo Russo, Ugo Maria Colesanti, and Andrea Vitaletti. 2014. MagoNode: Advantages of RF Front-ends in Wireless Sensor Networks. In *Real-World Wireless Sensor Networks*. Vol. 281. 125–137.

Mario Paoli, Dora Spenza, Chiara Petrioli, Michele Magno, and Luca Benini. 2016. MagoNode++: A Wake-Up-Radio-Enabled Wireless Sensor Mote for Energy-Neutral Applications. In *Proceedings of ACM/IEEE IPSN 2016 (Poster Session)*. Vienna, Austria, 1–2.

Stephen Pohlig and Martin Hellman. 1978. An Improved Algorithm for Computing Logarithms over GF (p) and Its Cryptographic Significance Function. *IEEE Transactions on information Theory* 24, 1 (1978), 106–110.

Thomas La Porta, Chiara Petrioli, Cynthia Phillips, and Dora Spenza. 2014. Sensor Mission Assignment in Rechargeable Wireless Sensor Networks. *ACM Transactions on Sensor Networks* 10, 4 (June 2014), 60:1–60:39.

Joaquin Recas Piorno, Carlo Bergonzini, David Atienza, and Tajana Simunic Rosing. 2009. Prediction and management in energy harvested wireless sensor nodes. In *Proceedings of CTIF Wireless Vitae 2009*. Aalborg, Denmark, 6–10.

Y. Ren, V. Oleshchuk, F.Y. Li, and X. Ge. 2011. Security in Mobile Wireless Sensor Networks – A Survey. *Journal of Communications* 6 (2) (2011), 128–142.

Christian Renner, Stefan Unterschütz, Volker Turau, and Kay Römer. 2014. Perpetual Data Collection with Energy-Harvesting Sensor Networks. *ACM Transactions on Sensor Networks* 11, 1 (Sept. 2014), 12:1–12:45.

Peter Rooij. 1995. Efficient exponentiation using precomputation and vector addition chains. In *Proceedings of EUROCRYPT 1994*, Vol. 950. Santa Barbara, California, USA, 389–399.

Sensirion AG. 2011. SHT1x Datasheet: Humidity and Temperature Sensor IC. (2011).

Vladimir Shakhov, Sangyep Nam, and Hyunseung Choo. 2013. Flooding Attack in Energy Harvesting Wireless Sensor Networks. In *Proceedings of ACM ICUIMC 2013*. New York, NY, USA, 49:1–49:5.

Saurabh Sharma, Amit Sahu, Ashok Verma, and Neeraj Shukla. 2012. Wireless Sensor Network Security. In *Advances in Computer Science and Information Technology*, Vol. 86. Springer, Bangalore, India, 317–326.

Zach Shelby, Klaus Hartke, and Carsten Bormann. 2013. Constrained Application Protocol (CoAP). Working Draft. (December 30 2013).

Jerome A. Solinas. 2000. Efficient Arithmetic on Koblitz Curves. *Designs, Codes and Cryptography* 19, 2-3 (2000), 195–249.

Dora Spenza, Michele Magno, Stefano Basagni, Luca Benini, Maoli Paoli, and Chiara Petrioli. 2015. Beyond duty cycling: Wake-up radio with selective awakenings for long-lived wireless sensing systems. In *Proceedings of IEEE INFOCOM 2015*. 522–530.

Antonio Vincenzo Taddeo, Marcello Mura, and Alberto Ferrante. 2010. QoS and security in energy-harvesting wireless sensor networks. In *Proceedings of ICETE SECRYPT 2010*. Athens, Greece, 1–10.

A.S. Wander, N. Gura, H. Eberle, V. Gupta, and S.C. Shantz. 2005. Energy analysis of public-key cryptography for wireless sensor networks. In *Proceedings of IEEE PerCom 2005*. 324–328.

Wang Wei-hong, Cui Yi-ling, and Chen Tie-ming. 2009. Design and implementation of an ECDSA-based identity authentication protocol on WSN. In *Proceedings of IEEE MAPE 2009*. 1202–1205.

Daming Zhang, Yongpan Liu, Xiao Sheng, Jinyang Li, Tongda Wu, C.J. Xue, and Huazhong Yang. 2015. Deadline-aware task scheduling for solar-powered nonvolatile sensor nodes with global energy migration. In *Proceedings of ACM/EDAC/IEEE DAC 2015*. 1–6.

Yun Zhou, Yuguang Fang, and Yanchao Zhang. 2008. Securing wireless sensor networks: a survey. *IEEE Communications Surveys & Tutorials* 10, 3 (2008), 6 – 28.

T. A. Zia and A. Y. Zomaya. 2011. A lightweight security framework for wireless sensor networks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications* 2, 3 (2011), 53–73.

Received December 2015; revised May 2016; accepted August 2016