A Survey of Public-Key Cryptographic Primitives in Wireless Sensor Networks

Kyung-Ah Shim, *Member, IEEE*

***Abstract*—Cryptographic primitives are fundamental building blocks for designing security protocols to achieve confidentiality, authentication, integrity and non-repudiation. It is not too much to say that the selection and integration of appropriate cryptographic primitives into the security protocols determines the largest part of the efficiency and energy consumption of the wireless sensor network (WSN). There are a number of surveys on security issues on WSNs, which, however, did not focus on public-key cryptographic primitives in WSNs. In this survey, we provide a deeper understanding of public-key cryptographic primitives in WSNs including identity-based cryptography and discuss their main directions and some open research issues that can be further pursued. We investigate state-of-the-art software implementation results of public-key cryptographic primitives in terms of execution time, energy consumption and resource occupation on constrained wireless devices choosing popular IEEE 802.15.4-compliant WSN hardware platforms, used in real-life deployments. This survey provides invaluable insights on public-key cryptographic primitives on WSN platforms, and solutions to find tradeoffs between cost, performance and security for designing security protocols in WSNs.**

***Index Terms*—Identity-based cryptography, public-key cryptography, public-key encryption, public-key infrastructure, public-key signature, side-channel attack, software implementation.**

I. Introduction

A wireless sensor network (WSN) is a self-configuring network of small sensor nodes communicating among themselves using radio signals, and deployed in quantity to sense, monitor and understand the physical world. Sensor motes are equipped with processor in various modes (sleep, idle, active), power source (AA or coin batteries, solar panels), memory used for the program code and for in memory buffering, radio used for transmitting the acquired data to some storage site and sensors for temperature, humidity, light, etc. WSNs provide a bridge between the real physical and virtual worlds and allow the ability to observe the previously unobservable at a fine resolution over large spatiotemporal scales. They have a wide range of potential applications to industry, science, transportation, civil infrastructure, and security. WSNs differ fundamentally from traditional wireless networks in that WSN nodes have limited computation capabilities and very restricted resource capacities. In environmental and military applications, tiny sensors are deployed and left unattended to report parameters such as the pressure, humidity and chemical activity continuously. The dense deployment and unattended nature of WSNs make it difficult to recharge the node batteries.

WSNs pose a number of unique security challenges that demand innovation in several areas including the design of cryptographic primitives and protocols: how to protect privacy, authentication, and integrity in this distributed and connected computing world, and how to satisfy the requirements of different platforms, ranging from resource constrained embedded devices to high-end servers. Selecting appropriate cryptographic primitives is vital in WSNs, as all security services such as authenticity, integrity and confidentiality are ensured by cryptography. The cryptographic methods used in WSNs should meet the constraints of the sensor nodes and should be evaluated according to the code size, data size, processing time, and power consumption. It is not too much to say that the selection and integration of proper cryptographic primitives into the security protocols of WSNs determines the efficiency of the whole system and the lifetime of the WSNs. Cryptographic techniques are typically divided into two generic types: symmetric-key cryptography (SKC) and public-key cryptography (PKC). An approach using SKC offers advantages in terms of low communication and computational overhead. However, it is not easy to establish a shared secret key in advance and it does not support non-repudiation. PKC solves these problems: it provides a more flexible and simple interface without the need for key pre-distribution and pairwise key sharing. It is known that PKC is considered to be too computationally expensive for small wireless devices if not accelerated by cryptographic hardware.

Earlier studies [1]–[3] showed that it is feasible to apply PKC to small wireless devices with very limited resources by using the right selection of algorithms and associated parameters, optimization, and low-power techniques. They investigated public-key cryptographic primitives including Rabin's scheme [4], RSA [5], and Elliptic Curve Cryptography (ECC) [6]. Since Shor [7] presented efficient quantum algorithms to solve the Integer Factorization problem and Discrete Logarithm problem in 1995, there is an increasing demand in investigating possible alternatives. We call such a class of cryptosystems post-quantum cryptosystems. We focus on post-quantum PKCs based on lattices and multivariate quadratic equations which have been studied on software implementations and optimizations on constrained wireless devices. So, we consider three classes of PKCs as classical PKCs including RSA and ECC, PKCs based on

Manuscript received August 4, 2014; revised April 6, 2015 and June 30, 2015; accepted July 5, 2015. Date of publication July 29, 2015; date of current version January 27, 2016. This research was supported by the National Institute for Mathematical Sciences funded by Ministry of Science, ICT and Future Planning of Korea (B21503-1).

The author is with the Division of Mathematical Modeling, National Institute for Mathematical Sciences, Daejeon 305-390, Korea (e-mail: kashim@nims.re.kr).

Digital Object Identifier 10.1109/COMST.2015.2459691

hard problems over lattices and PKCs based on multivariate quadratic equations. In most of survey papers on WSNs, security issues are divided into from five to seven categories including cryptography, secure routing, secure data aggregation, secure data fusion, location security. There are a number of surveys on security issues on WSNs [8]–[13] focusing security protocols which include many authentication protocols and privacy-preserving protocols [14]–[16], energy-efficient security architectures and protocols [17], secure routing protocols [18], [19], secure data gathering protocols [20], secure data aggregations [21], [22], secure location [23], and secure localization and location verification [24]. However, these surveys didn't focus on public-key cryptographic primitives in WSNs.

The overall aim of this paper is to provide a survey of theoretical backgrounds of the security of the PKCs in the three classes, and discuss their main directions and some open research issues. A main point of this survey is the investigation of state-of-the-art software implementation results for the PKCs on low-power constraint devices choosing popular IEEE 802.15.4-compliant WSN hardware platforms in terms of speed, energy consumption and resource occupation. While software implementations running on general purpose microprocessors are flexible and can be easily updated, hardware implementations either on FPGAs or ASICs can achieve higher performance. Software implementation of cryptographic algorithms may be more cost-effective than corresponding hardware-based mechanisms, enabling the designers to select from alternative cryptographic solutions to meet the accepted level of security requirements for given applications and environments. There are many software implementation results of PKCs for different security levels, parameters, and optimizations. This survey facilitates evaluation and comparison between the software implementation results of PKCs with the same metrics. Additionally, we target at side channel attacks (SCAs) which try to extract secret information from the physical implementation of cryptographic algorithms. Once mathematically strong cryptographic primitives are implemented in either software or hardware, they are known to be vulnerable to various physical attacks such as SCAs. All the cryptographic primitives should be designed in a highly reliable and efficient way to prevent SCAs. This survey provides an overview of SCAs of the PKCs and their countermeasures, which can be used as a guideline for selecting countermeasures in future designs.

The paper is organized as follow. Section II overviews SCAs of cryptographic algorithms on constrained small devices. From Sections III to IV, we survey PKCs including public-key encryption and public-key signature, and identity-based cryptography in WSNs. We then discuss security of PKCs against several attacks including SCAs. We also give theoretical backgrounds of PKCs contained in the three classes and present their software implementation results in terms of several measures on wireless constraint devices. We discuss future research issues in Section V. Concluding remarks are given in Section VI.

II. Side-Channel Attacks

A side-channel attack (SCA) tries to extract secret information from physical implementations of cryptographic algorithms on constrained small devices rather than theoretical weaknesses in the algorithms. While a cryptographic device is performing cryptographic computations, an attacker can monitor the side channel information leakage, such as power consumption, timing, and electromagnetic emanations if nodes are captured or on the spot. Goal of SCAs is to extract a secret key of the implemented cryptographic algorithm from the physical behavior of the target device. The attacker may use techniques such as power analysis, execution cycle frequency analysis, timing information analysis (on data movement into and out of the CPU), electromagnetic radiation analysis, acoustic emission analysis, etc. Simple power analysis (SPA) is a technique that involves directly interpreting power consumption measurements collected during cryptographic operations. No statistical analysis is required in such an attack. Similarly, in simple electromagnetic analysis (SEMA), an adversary is able to extract compromising information from a single electromagnetic sample. In differential power analysis (DPA), an adversary monitors the power consumed by cryptographic devices, and then statistically analyzes the collected data to extract a key. In differential electromagnetic analysis (DEMA), instead of monitoring the power consumption, an attacker monitors electromagnetic emanations from cryptographic devices, and then the same statistical analysis as that for DPA is performed on the collected electromagnetic data to extract secret parameters. Several kinds of these implementation attacks on cryptographic devices can be categorized into two types: passive attacks and active attacks. Passive attacks are based on the observation of side-channel information such as the power consumption of the chip or its electromagnetic emanations. Examples of passive attacks include SPA, DPA, SEMA and DEMA attacks. On the other hand, active attacks, including fault injection attacks, deliberately introduce abnormal behavior in the chip in order to recover internal secret data.

Kocher *et al.* [25] first presented power analysis attacks on Data Encryption Standard (DES) [26], in which an attacker determined a secret key of DES by measuring the power consumption of the algorithm running on a smart card. Almost all block ciphers including Advanced Encryption Standard (AES) [27], which is a current symmetric-key encryption standard of electronic data established by the National Institute of Standards and Technology (NIST) in 2002, are vulnerable to SCAs. Okeya and Iwata [28] showed that message authentication codes (MACs), EMAC, OMAC, and PMAC, are vulnerable to SPA or DPA. Okeya [29] showed that HMACs based on hash functions are vulnerable to DPA. These results show that SCAs can be still mounted on MACs based on block ciphers, even if the underlying block ciphers adopt countermeasures against SCAs. Thus, protecting block ciphers against SCAs is not sufficient, and countermeasures are needed for MACs as well. Security protocols for WSNs are also vulnerable to SCAs, e.g., TinySec [30], which is a link layer security architecture for WSNs use several block ciphers and MACs. SCAs are usually carried out in a context, in which the possible attacker can control the target device. WSNs are particularly vulnerable to SCAs [31] due to the specificities of WSN scenarios: passive acquisition, on-site acquisition, device not controlled, and real-world devices. Node compromise is a critical issue in WSNs. A popular approach to prevent the problem relies on the detection events that arise during the attack (loss of connectivity, removal of a node, etc.). The node capture attack has been illustrated in several works in the contexts of WSNs. Hartung *et al.* [32] recovered the cryptographic secret keys on MICA2 by dumping its internal memory through the JTAG interface. Becher *et al.* [33] showed how to access several hardware components of a node such as the external memory, the bootstrap loader or the JTAG interface. They underlined that the node capture requires the absence of the node from the network for a substantial period, which could be useful to detect captured nodes. There have been proposed SCAs on the implementation of cryptographic algorithms without the node captures.

Recently, Meulenaer *et al.* [34] presented DPA and template-based SPA of AES and ECC implementations on two common types of nodes: MICAz and TelosB. For these attacks, they assumed that the nodes periodically exchange messages encrypted with AES and ECC and the on-site acquisition is convenient for an attacker: the nodes are easily accessible and the presence of the attacker at the site is not detected. Through measuring the power consumption traces of a node without interrupting its network operations, they showed that less than 40 and 80 traces were sufficient to recover the full second AES key on MICAz and TelosB, respectively, minimizing the online complexity of the attack. In the template-based SPA on ECC, first, the attacker builds templates corresponding to the computation of an iteration of the point multiplication. Templates are statistical models deduced from traces for a subset of the key space. In the off-line phase, they are collected on a similar device running the same implementation. It is realistic assuming commercially available nodes (or at least, their microcontrollers) and the use of a freely available ECC implementation, such as the TinyECC library [35] used in their work. After that, the templates are compared with the trace acquired in the online phase. This comparison determines which intermediary values were actually computed by the target device, leading to the recovery of the key bits involved in the iteration. When all the iterations are successfully processed, the ECC 160-bit full key is recovered. These works showed the feasibility of implementing power analysis attacks without being detected in the WSNs. Like these, SCA attacks have been successfully used to break the hardware or software implementations of many cryptosystems including block ciphers (such as DES, AES, Camellia, IDEA, Misty1, etc.), stream ciphers (such as RC4, RC6, A5/1, SOBER-t32, etc.), MACs, public-key ciphers (such as RSA encryption, ElGamal, ECC, XTR, etc.), signature schemes (such as RSA signature, ECDSA, etc.), and cryptographic protocols, and even to break the networking systems [36].

Some countermeasures against SCAs make algorithmic changes to the cryptographic primitives so that attacks are provably inefficient on the obtained implementation, e.g., masking data and key with random mask generated at each run. It had been shown [37], [38] that among all these kinds of countermeasures, algorithmic techniques are the most versatile, all-pervasive, and may be the most powerful. Also, in many contexts, they are the cheapest to put in place. Software-based countermeasures include dummy instructions, randomization of the instruction execution sequence, balancing hamming weights of the internal data, and bit splitting. On the hardware level, the countermeasures usually include clock randomization [39], [40], power consumption randomization or compensation [41], randomization of instruction set execution and/or register usage [42]. However, the effect of these countermeasures can be reduced by various signal processing techniques [42]. Software-based countermeasures against SCAs considerably hinder performance of cryptographic algorithms in terms of memory or execution time or both. Thus, it needs to be considered security against the SCAs as well as traditional cryptanalysis in the design phase of public-key cryptographic algorithms. Countermeasures of each public-key cryptographic algorithms against SCAs will be described in Sections III and IV.

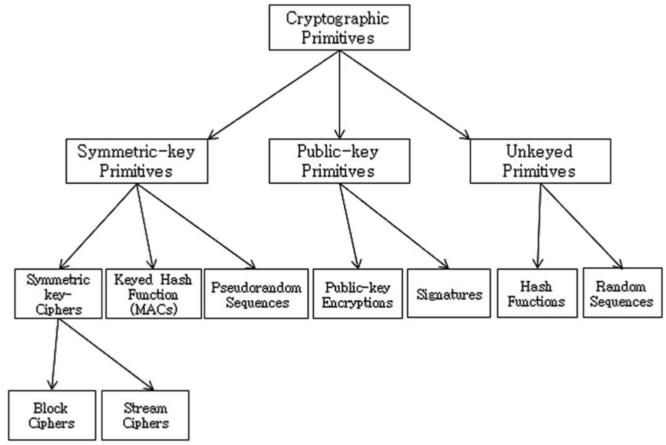

Fig. 1. Taxonomy of cryptographic primitives.

III. Public-Key Cryptography

Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication [43]. Cryptographic goals are confidentiality, authentication, data integrity and non-repudiation. Cryptographic techniques are typically divided into two generic types: symmetric-key and public-key. Fig. 1 provides a schematic listing of the cryptographic primitives considered and how they relate [43]. In this survey, we focus on public-key cryptography (PKC).

A. PKC Versus Symmetric-Key Cryptography

Symmetric-key cryptography (SKC) is also known as shared-key, single-key, and secret-key cryptography. In this type of message encryption, sender and receiver only have to share the same secret key in the beginning and then they can begin to encrypt and decrypt messages between them using that key. This key predistribution process is very difficult. SKC cannot achieve non-repudiation, as both sender and receiver use the same key, messages cannot be verified to have come from a particular user. PKC, also known as asymmetric cryptography uses two keys: a secret key, which has to be kept private, and a public key, which is publicly known. Any operation done with the secret key can only be reversed with the public key, and vice versa. This nice property makes all PKC-based algorithms useful for secure broadcasting and authentication purposes. It is also an invaluable tool for allowing the secure exchange of secret keys between previously unknown partners. The public key in PKCs must be authenticated. Public-Key Infrastructure (PKI) solves the public key authentication problem using a public-key certificate issued by a Certification Authority (CA). Details of PKI in WSNs will be described in Section III-D. Computational cost of PKC had hindered its application in highly-constrained devices, such as sensor nodes, while an approach using SKC offers advantages in terms of low communication and computational overhead. One may believe that SKC is more suitable for WSN applications which requires only confidentiality or data integrity. To apply SKC to these WSNs, the shared-key distribution is needed. The key predistribution methods have the following three types:

- A single network-wise secret key: this causes a single-point failure, i.e., if the secret key of a node is revealed then the entire network is broken.
- A pairwise key between a node and the BS or between two nodes: the pairwise keying is very difficult and inefficient, i.e., each node must share ${}_nC_2 = \frac{n(n-1)}{2}$ secret keys, where n is the number of the sensor nodes. This creates a problem with managing and ensuring the security of all these keys. If the secret key of a node is revealed, then the other node with the same key is also compromised.
- A group key among a set of nodes: group keying is more inefficient than the pairwise keying as it is required heavy computational overhead and interactions with more than two rounds among nodes. If the group key of a node in a group is revealed, then all the group of nodes is compromised.

To minimize the effects of secret key exposure is an important factor. In fact, the security schemes should guarantee that no matter how many nodes are captured, the secret information extracted from the compromised nodes cannot affect the security among non-compromised nodes, i.e., communications among non-compromised nodes remain secure. However, the three types above cannot satisfy this requirement, while PKC satisfies this requirement.

- In the case of using a public-key encryption scheme, it doesn't matter, as sensor nodes encrypt any message under the BS's public key without requiring the nodes' secret key. However, if the BS's secret key is compromised to an attacker, then the attacker can decrypt all past ciphertexts encrypted by the public key corresponding to the secret key. Thus, the secret key of the BS must be securely stored to prevent such an exposure.
- In the case of using a public-key signature scheme, even though a user's secret key is compromised to an attacker, the security of communications among non-compromised nodes cannot be affected.

A number of applications of WSNs require various security attributes and functionalities including authentication with non-repudiation, homomorphic property, aggregation, batch verification, signature with message recovery, etc. PKC makes it possible to achieve these functionalities. PKC is considered to be too computationally expensive for small devices if not accelerated by cryptographic hardware. Recent studies [1], [2], [44], [45] showed that it is feasible to apply PKCs to small wireless devices with very limited resources by choosing public-key cryptographic algorithms. We will discuss security goals and models for public-key encryption and public-key signature, public key infrastructure, and identity-based cryptography in the following subsections.

B. Public-Key Encryption

In public-key encryption (PKE) schemes, each entity A has a public key e and a corresponding secret key d. In secure systems, the task of computing d given e is computationally infeasible. The public key defines an encryption transformation E_e, while the secret key defines the associated decryption transformation D_d. Any entity B wishing to send a message m to A obtains an authentic copy of A's public key e, uses the encryption transformation to obtain the ciphertext $C = E_e(m)$, and transmits C to A. To decrypt C, A applies the decryption transformation to obtain the original message $m = D_d(C)$. The public key need not be kept secret, and, in fact, may be widely available-only its authenticity is required to guarantee that A is indeed the only party who knows the corresponding secret key. The formal definition of a PKE is ad follows:

Definition 1: A (probabilistic) PKE scheme $\pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is defined by following three algorithms:

- Key Generation $\mathcal{G}$. On input 1^k, where k is a security parameter, the algorithm $\mathcal{G}$ produces a pair (pk, sk) of matching public and secret keys.
- Encryption $\mathcal{E}$. Given a message m (in the space of plaintexts $\mathcal{M}$) and a public key pk, $\mathcal{E}_{pk}(m, r)$ produces a ciphertext C (in the space of ciphertexts $\mathcal{C}$) of m, for a random value r.
- Decryption $\mathcal{D}$. Given a ciphertext $C \in \mathcal{C}$ and the secret key sk, $\mathcal{D}_{sk}(C)$ gives back the plaintext $m \in \mathcal{M}$.

The security of PKE schemes is usually classified from the point of view of their goals and attack models. The standard goals of PKE schemes are as follows:

- **Semantic Security** (SS). Any adversary (probabilistic polynomial-time Turing Machine) cannot obtain any partial information about the plaintext of a given ciphertext [47]. This notion corresponds to a computational version of perfect secrecy introduced by Shannon [46].
- **Indistinguishability** (IND). Given a ciphertext of one of two plaintexts, any adversary cannot distinguish which one is encrypted [47].
- **Non-malleability** (NM). Given a ciphertext of a plaintext, any adversary cannot construct another ciphertext whose plaintext is meaningfully related to the initial one [48].

In a PKE scheme, an adversary has access, as anybody, to the encryption key. It can thus encrypt any plaintext of its choice. Hence, the basic attack is called "Chosen-Plaintext Attack", or in short CPA. However, the adversary may also have access to more information, and namely some decryptions. This is modeled by an access to the decryption oracle. The standard attack models of PKE schemes are as follows.

- **(Non-adaptive) Chosen-Ciphertext Attacks** (Lunch Time Attack, CCA1). An adversary can access to a decryption oracle before it obtains a challenge ciphertext [49], i.e., it can choose a set of ciphertexts and obtain the corresponding plaintexts during this period.
- **Adaptive Chosen-Ciphertext Attacks** (CCA2). In addition to the ability of the CCA1, an adversary can access to a decryption oracle even after it obtains a challenge ciphertext [50]. However, it is prohibited from asking the oracle to decrypt the challenge ciphertext itself.

We describe formal definitions of IND and NM [51], because it is known that SS is equivalent to IND in any attack model [52]. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ ba a two-stage probabilistic adversary whose running time is bounded by t.

Definition 2 (Indistinguishability): Let $\pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a PKE scheme. We define the advantage of $\mathcal{A}$ against the indistinguishability of π as follows:

$$\mathsf{Adv}_{\pi}^{\mathsf{ind}}(\mathcal{A})$$
$$\overset{\mathsf{def}}{=} \left| 2 \times \Pr_{b,r}\left[\begin{array}{l}(pk, sk) \leftarrow G(1^k), (m_0, m_1, s) \leftarrow \mathcal{A}_1(pk), \\ c = \mathcal{E}_{pk}(m_b, r),\ b' = \mathcal{A}_2(m_0, m_1, s, c) : b' = b\end{array}\right] - 1 \right|.$$

We insist above on that $\mathcal{A}_1$ outputs two messages m_0 and m_1 such that $|m_0| = |m_1|$. As usual, we define by $\mathsf{Adv}_{\pi}^{\mathsf{ind}}(t)$ the maximum advantage over all the adversaries $\mathcal{A}$ whose running time is bounded by t. Then we say that π is (t, ε)-IND secure if $\mathsf{Adv}_{\pi}^{\mathsf{ind}}(t)$ is less than ε.

Definition 3 (Non-malleability): Let $\pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be an encryption scheme. We define the advantage of $\mathcal{A}$ against the non-malleability of π by:

$$\mathsf{Adv}_{\pi}^{\mathsf{nm}}(\mathcal{A}) \overset{\mathsf{def}}{=} \mathsf{Succ}_{\pi}^{\mathsf{nm}}(\mathcal{A}) - \mathsf{Succ}_{\pi}^{\mathsf{nm,\$}}(\mathcal{A}),$$

where the two successes use the same probability distribution, for a distribution of plaintexts M and a binary relation R, generated by

$$\mathsf{(pk,sk)} \leftarrow \mathcal{G}(1^k), (M, s) \leftarrow \mathcal{A}_1(\mathsf{pk});$$
$$m, \tilde{m} \leftarrow M; c \leftarrow \mathcal{E}_{\mathsf{pk}}(m, r);$$
$$(R, y) \leftarrow \mathcal{A}_2(M, s, c); x \leftarrow \mathcal{D}_{\mathsf{sk}}(y), \quad \text{and}$$
$$\mathsf{Succ}_{\pi}^{\mathsf{nm}}(\mathcal{A}) \overset{\mathsf{def}}{=} \Pr\left[y \neq c \wedge x \neq \bot \wedge R(x, m)\right]$$
$$\mathsf{Succ}_{\pi}^{\mathsf{nm,\$}}(\mathcal{A}) \overset{\mathsf{def}}{=} \Pr\left[y \neq c \wedge x \neq \bot \wedge R(x, \tilde{m})\right].$$

We define by $\mathsf{Adv}_{\pi}^{\mathsf{nm}}(t)$ the maximum advantage over all the adversaries $\mathcal{A}$ whose running time is bounded by t. Then we say that π is (t, ε)-NM secure if $\mathsf{Adv}_{\pi}^{\mathsf{nm}}(t)$ is bounded by ε.

One can mix-and-match the goals, IND, NM, and attacks, CPA, CCA1, CCA2 in any combination, given rise to six notions of security: IND-CPA, IND-CCA1, IND-CCA2 and NM-CPA, NM-CCA1, NM-CCA2.

Semantic security, defined by Goldwasser and Micali [47], captures the intuition that an adversary should not be able to obtain any partial information about a message given its encryption. A different notion of security, called non-malleability, was proposed by Dolev, Dwork, and Naor [48]. The adversary also has access to a decryption oracle, but its goal is not to obtain partial information about the plaintext of the target ciphertext but rather to create another encryption of a different message that is related in some interesting way to the original, encrypted message. Rackoff and Simon [50] defined the notion of security against CCAs by simply allowing an adversary to obtain decryptions of its choice, i.e., the adversary has access to a "decryption oracle" with a certain restriction. The restriction is that the adversary is not allowed to submit the target ciphertext itself to the oracle, but, it may submit any other ciphertext, including ciphertexts that are related to the target ciphertext.

A PKE scheme secure against CCA2 is a very powerful cryptographic primitive. It is by now generally recognized in the cryptographic research community that security against CCA2 is the "right" notion of security for a general-purpose PKE scheme. This is exemplified by the adoption of Bellare and Rogaway's Optimal Asymmetric Encryption Padding (OAEP) scheme [53] as the Internet encryption standard RSA in PKCS#1 Version 2 and for use in the SET protocol for electronic commerce. Another motivation for security against CCA2 is Bleichenbacher's attack [54] on the widely used SSL key establishment protocol based on RSA PKCS#1 Version 1.5 [55] (SSL still uses RSA-OAEP in PKCS #1 Version 1.5, but the protocol has been patched so as to avoid Bleichenbacher's attack). At Crypto'98, Bleichenbacher [54] presented CCA2 on RSA in PKCS #1 given access to an oracle that, for any ciphertext, indicates whether the corresponding plaintext is PKCS conforming. The attack needed 2 million chosen ciphertexts to find the message for a 1024-bit RSA key. It concluded that RSA encryption should include an integrity check and that the phase between decryption and integrity verification is crucial, as any information leaking from this phase can present a security risk. PKCS #1 Version 2.0 [56] introduced a new algorithm RSAES-OAEP to counteract this attack. However, Manger [57] presented CCA2 on RSAES-OAEP in PKCS #1 Version 2.0 by recovering the plaintext in roughly one thousand oracle queries. OAEP adds an integrity check and masks the structure of the message being encrypted to achieve plaintext-awareness and consequent protection against CCAs. However, translating the octet-aligned OAEP process into integers modulo n in RSAES-OAEP reintroduced sufficient structure to make CCA2 possible, with a high likelihood, in many implementations, where n is an RSA modulus. To avoid implementation weaknesses related to these errors handled within the decoding operation [54], [57], the encoding and decoding operations for RSAES-OAEP and RSAES-PKCS1-v1.5 are embedded in the specifications of the respective encryption schemes in PKCS #1 Version 2.1 and Version 2.2 [58], [59].

A cryptosystem may be semantically secure against CPA or even CCA1, while still being malleable. However, semantic security against CCA2 is equivalent to non-malleability against CCA2 (i.e., IND-CCA2 is equivalent to NM-CCA2) [52]. The relationship of all the security notions of PKE schemes are referred to [52]. In fact, well-known PKEs such as the plain RSA [5] and ElGamal encryption scheme [60] are malleable. In these schemes, it is possible to obtain a ciphertext of $m \cdot \alpha$ from a ciphertext of a message m without knowing the corresponding plaintext. For example, in ElGamal, an adversary increases by α of the encryption of m as $\langle C_1, C_2 \cdot \alpha \rangle$ from $\langle C_1 = g^r, C_2 = (pk)^r \cdot m \rangle$ which is an encryption of m under a public key pk. Implementations of ElGamal often use an element $g \in \mathbb{Z}_p^*$ of a prime order q, where q is much smaller than p. When the set of plaintexts is equal to the subgroup generated by g, the Decision Diffie-Hellman (DDH) assumption implies that ElGamal is semantically secure.

- **Computational Diffie-Hellman Problem (CDHP)**. Given a group $\mathbb{G}$, a generator of g in $\mathbb{G}$, and $)g^x, g^y)$, to compute g^{xy}.
- **Decision Diffie-Hellman Problem (DDHP)**. Given a triple (g^x, g^y, g^z), output `true` if $z = xy$, `false`, otherwise.

Unfortunately, implementations of the scheme often encrypt an m-bit message by viewing it as an m-bit integer and directly encrypting it. The resulting system is not semantically secure, i.e., the ciphertext leaks the Legendre symbol of the plaintext. Thus, in that case, ElGaml is not IND-CPA in $\mathbb{Z}_p^*$, but it is IND-CPA in the prime order subgroup of $\mathbb{Z}_p^*$. In other words, it is IND-CPA if the DDH problem is hard in the underlying group. However, ElGaml is insecure against CCA. The representative example of IND-CCA secure PKE scheme is Cramer-Shoup encryption scheme [61] which is provably secure assuming that the DDH problem is hard and that a hash function used in the scheme is collision-resistant. There is a generic conversion method which converts an IND-CPA PKE scheme to an IND-CCA one. However, the resulting IND-CCA schemes are inefficient compared to their original ones.

C. Public-Key Signatures

A public-key signature (PKS) or digital signature is a contrary concept of a PKE scheme: it signs on a message with a secret key and then its resulting signature is publicly verified with a public key corresponding the secret key. PKSs have many applications in information security, including authentication, data integrity, and non-repudiation. The PKS schemes can be the following two classes [43]:

- **Signature Schemes with Appendix**. It requires an original message as input to the verification algorithm.
- **Signature Schemes with Message Recovery**. It does not require an original message as input to the verification algorithm. In this case, the original message is recovered from the signature itself.

For security of PKS schemes, we consider types of forgeries and attacks on the schemes. Types of forgeries can be divided into the following three classes:

- **Universal Forgery (Total Break)**. An adversary is able either to extract the signing key, or to find an efficient signing algorithm that is functionally equivalent to the signing algorithm equipped with the genuine signing key. So, anyone can forge signatures of any messages.
- **Selective Forgery**. An adversary is able to create a valid signature for a particular message or a class of messages chosen a priori.
- **Existential Forgery**. An adversary is able to forge a valid signed message that signer has not created, but the adversary has little or no control over which message will be the target.

Types of Attacks are divided into the following three classes:

- **Key-Only Attack**. An adversary knows publicly available information on the scheme.
- **Known-Message Attack**. An adversary can get valid signatures for a set of messages which are known to the adversary but not chosen by it.
- **Chosen-Message Attack**. An adversary can obtain valid signatures from a chosen list of messages before attempting to forge another signed message.
- **Adaptive Chosen-Message Attack**. An adversary is allowed to use a signer as an oracle: the adversary may request signatures of messages which may depend on the signer's signing key and previously obtained signed messages. That is, at any time, the adversary can query the signer with messages chosen at its will, except for the target message.

Definition 4: A PKS scheme $\mathcal{PKS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vfy})$ is specified by three polynomial time algorithms with the following functionality:

- $\mathsf{KeyGen}(1^k)$. A key generation algorithm KeyGen takes a security parameter $k \in \mathbb{Z}^+$ and returns a public/secret key pair (pk, sk).
- $\mathsf{Sign}(sk, m)$. A signing algorithm Sign takes a message m and a secret key sk, and outputs a signature σ.
- $\mathsf{Vfy}(pk, \sigma)$. A verification algorithm Vfy takes a public key pk, a messages m and a signature σ, and outputs `Valid` if $\mathsf{Vfy}(pk, m, \sigma) = 1$, or $\perp$ otherwise.

The most general security notion of a PKS scheme is existential unforgeability under an adaptive chosen-message attack. An adversary is also given access to the signing oracle for any messages except the target message.

Definition 5 (Unforgeability of PKS Schemes): An adversary $\mathcal{A}$'s advantage $Adv_{\mathcal{PKS},\mathcal{A}}$ is defined as its probability of success in the following game between a challenger $\mathcal{C}$ and $\mathcal{A}$:

- Setup. The challenger runs the KeyGen algorithm and pk is given to $\mathcal{A}$.

- Sign Queries. Proceeding adaptively, $\mathcal{A}$ requests a signature on messages m_i for a public key pk, $\mathcal{C}$ returns a signature σ_i.
- Output. Eventually, $\mathcal{A}$ outputs σ^* on m^* for pk and wins the game if i) m^* is not requested to the signing oracle under pk, and ii) $\mathsf{Vfy}(pk, m^*, \sigma^*) = 1$.

A forger $\mathcal{A}(t, q_S, \epsilon)$-breaks a PKS scheme in the model above if $\mathcal{A}$ runs in time at most t and $\mathcal{A}$ makes at most q_S queries to the signing oracle and $Adv_{\mathcal{PKS},\mathcal{A}}$ is at least ϵ. A signature scheme is (t, q_S, ϵ)-existentially unforgeable under an adaptive chosen-message attack if no forger (t, q_S, ϵ)-breaks it.

The plain RSA signature scheme is insecure against all the attacks: Let (e, d) is a public/secret key pair with a modulus $N = pq$, choose $\sigma \in \mathbb{Z}_N^*$ and set $m = \sigma^e \pmod{N}$. Then, by construction, σ is a valid signature on m. To overcome this weakness, the Full-Domain Hash (FDH) RSA is proposed as: generate a signature $\sigma = H(m)^d \pmod{N}$, where H is a hash function onto $\mathbb{Z}_N^*$. The random oracle model, introduced by Bellare and Rogaway [53], is a theoretical framework allowing to prove the security of hash-and-sign signature schemes. In this model, the hash function is seen as an oracle that outputs a random value for each new query. Bellare and Rogaway [62] defined the Full Domain Hash (FDH) signature scheme, which is existential unforgeable against an adaptive chosen message attack in the random oracle model assuming that inverting RSA is hard. They also introduced the Probabilistic Signature Scheme (PSS), which offers better security guarantees than FDH. RSA-PSS added in version 2.1 of PKCS #1 is existential unforgeable against an adaptive chosen-message attack in the random oracle model assuming the RSA problem is hard.

D. PKI

Although PKCs which have some advantages than SKCs are computationally feasible on sensor nodes, one of factors which make it difficult to apply the PKCs to real WSN applications is the public-key authentication problem. PKC has two kinds of keys: a public key and a secret key. Public keys must be authenticated, as one can be absolutely sure that a public key belongs to the person. Public-key infrastructure (PKI) is an arrangement that binds public keys with respective users' identities by means of public-key certificates issued by a Certificate Authority (CA). This PKI causes several problems of certificate management including storage, distribution and the computational cost of certificate verification. According to PKIX which pursued the goal of developing Internet standards to support X.509-based PKIs developed by the ITU-T [63], the major components of a PKI are the following:

- Clients, which are the users of public-key certificates.
- CA, which establishes identities and creates digital certificates.
- Registration Authority (RA), which is responsible for the registration and initial authentication of the clients.
- Repository, which stores the certificates and the Certification Revocation Lists (CRLs).

In order to provide the services of PKI, these components and their functionalities must be mapped to the entities of WSNs. In order to deploy PKI into WSNs, it is also obligatory to select an appropriate hierarchy model. Fortunately, in most cases, the architecture of sensor networks is extremely simple: one BS that serves as the interface to hundreds or thousands of sensor nodes can communicate with the nodes belonging to the same network. Therefore, it is enough to consider that most sensor networks will use a simple hierarchical PKI architecture, with only one root CA. The basic functionalities of PKI, that is, registration, initialization, key generation, certification, and certification retrieval, are done in WSNs as follows:

- The BS creates the public/secret key pair of a sensor node, assigns an unique identification to it, and creates a certificate that links that unique identification with its public key. Later, it initializes the contents of the sensor node (such as configuration data and internal programming), including its certificate and the certificate of the root CA (i.e., the BS itself).
- When a sensor node retrieves the certificate of one of its neighbors, it will be able to check its validity using the root CA's certificate.

In some applications of WSNs with a fixed BS, a PKE scheme is suitable for sensor nodes to achieve end-to-end confidentiality in node-to-BS communications. Because, the BS, a bootstrapper of the WSN, can preload its public key into each sensor node at the predeployment phase. Each sensor node encrypts sensed data via the PKE scheme under the BS's public key and then sends them to the BS or neighboring nodes for relaying. However, in a certain WSN, if the hop-by-hop authentication in node-to-node communications is required, PKCs are not suitable. Because, to authenticate each other, sensor nodes should exchange their public-key certificates and then should verify CA's signatures in the certificates. After these procedures are performed, actual authentication process will be carried out. Thus, communication overhead for this certificate transmission and computational overhead for the verification of CA's signature are very heavy in each node. In this case, there exists a better alternative, an identity-based cryptography.

E. Identity-Based Cryptography

Identity-based cryptography (IBC) introduced by Shamir [64] allows a user's public key to be easily derived from its known identity information such as an email address or a cellular phone number by eliminating the need for public-key certificates. Such cryptosystems alleviate the certificate overhead and solve the problems of PKI technology. A Private Key Generator (PKG) having a master public/secret key pair is responsible for generating private keys for users. IBC is more suitable for WSNs, as the BS can naturally play the role of the PKG. The BS generates sensor nodes' identities and the corresponding private keys and then embeds the private keys in the nodes prior to its use in the field, and no secret channel is needed for key setup. Thus, only the identities of the sensors are exchanged without sending public keys and their certificates. This results in energy saving for the communication

TABLE I
WIRELESS SENSOR NODE PLATFORMS

	MICAz	Tmote Sky	Imote2
Microcontroller	ATmega128L	MSP430	PXA271
Word size	8-bit	16-bit	32-bit
Memory	128KB Program FLASH	10KB RAM	256KB SRAM
	516KB Measurement FLASH	48KB FLASH	32MB SDRAM
	4KB EEPROM	1024kB External FLASH	32MB FLASH
OS	TinyOS	TinyOS	TinyOS, Linux, SOS
Clock	8 MHz	8 MHz	13 MHz, 104 MHz
Power	3.0 V	3.0 V	3.85 V
Current Draw	8 mA	1.8 mA	66 mA
Radio	CC2420	CC2420	CC2420
	2.4 GHz IEEE 802.15.4	2.4 GHz IEEE 802.15.4	2.4 GHz IEEE 802.15.4
Current Draw (receive mode)	19.7 mA	19.7 mA	44 mA (13 MHz), 66 mA (104 MHz)
Current Draw (transmit mode)	17.4 mA	17.4 mA	44 mA (13 MHz), 66 mA (104 MHz)
Data Rate	250 kbps	250 kbps	250 kbps
Battery	2x AA	2x AA	3x AAA

between sensors. In PKI, each sensor node stores its own public key/secret key pair together with the corresponding public-key certificate issued by CA. Then, any external party that wishes to interact with nodes also requires the nodes' public-key certificates. Although the real-time access to the CA is difficult in WSNs, this pre-installation method of the certificates makes it possible to use the PKI.

As mentioned in the previous subsection, this PKI is suitable for node-to-BS communications, but it is not suitable for node-to-node communications, as they require exchange of the nodes' public-key certificates. Thus, ID-based schemes are more suitable for these WSN scenarios: each sensor node which has its unique identification information such as serial numbers gets the corresponding private keys from the BS which serves as the PKG. To authenticate each other, only the identity information should be exchanged without extra public key data. The length of an identity is much shorter than that of a public key and its certificate. Then the validity of the identity information is determined when its signature related to the identity is verified, i.e., if the signature verification ends successfully then the legitimacy of the identity information is also guaranteed. In particular, IBC makes it possible to establish a session key without any interaction. Two parties, each knowing only the identity of the other and without communicating, are then able to derive a secret unknown to any other party, and use that secret to compute the same cryptographic key for secure communications. In node-to-BS communications, the BS stores only nodes' IDs instead of their relatively large-size public keys. Therefore, ID-based schemes are more suitable for these WSN scenarios, as it does not require the transmission of the public-key certifications and verifications of CA's signatures on the public keys for node-to-node communications as well as node-to-BS communications.

IV. IMPLEMENTATION RESULTS OF PKCS ON THE WSN HARDWARE PLATFORMS

Except RSA and ECC, other cryptographic approaches that we focus on are lattice-based schemes and multivariate quadratic equations-based schemes. Since Shor [7] presented efficient quantum algorithms to solve the integer factorization problem and discrete logarithm problem in 1995, there is an increasing demand in investigating possible alternatives. Cryptosystems belong to such a class of cryptography, so-called post-quantum cryptography, are based on lattices or multivariate quadratic equations. We can divide known PKCs into three classes: classical PKCs including RSA and ECC, PKCs based on hard problems over lattices and PKCs based on hard problems related to multivariate quadratic equations. Now, we give theoretical backgrounds of PKCs in the three classes and their software implementation results in terms of execution time, energy consumption and resource occupation for the execution in RAM and ROM on several wireless constraint devices. There are three popular hardware platforms used in real-life deployments: MICAz [65] and Imote2 [65] platforms developed by Crossbow Technology, and Tmote Sky [66] developed by the Moteiv Corporation. Table I shows the comparative specifications of these three wireless sensor platforms.

A. RSA Versus ECC

ECC and RSA are mature public-key cryptographic algorithms that have been researched by the academic community for many years: RSA was conceived by Rivest, Shamir and Adleman in 1977 [5] and Koblitz and Miller independently proposed ECC in 1985 [6], [67].

Base Problems and Algorithms for Solving the Problems: The fundamental operation of RSA is a modular exponentiation in integer rings and its security stems from the difficulty of factoring large integers. ECC operates on the groups of points over elliptic curves and derives its security from the hardness of the elliptic curve discrete logarithm problem (ECDLP). While sub-exponential algorithms can solve the integer factorization problem (IFP) and discrete logarithm problem (DLP), only exponential algorithms are known for the ECDLP except those over pairing-friendly curves.

- **Integer Factorization Problem (IFP)**. Given a composite number $n = pq$, to find prime factors p or q.

- **Discrete Logarithm Problem (DLP)**. Given a group $\mathbb{G}$, a generator g of $\mathbb{G}$, and $h = g^x$, to compute x, where we denote $x = log_g h$.

ECC achieves the same level of security with smaller key sizes and higher computational efficiency than RSA: ECC-160 (resp., ECC 224) provides comparable security to RSA-1024 (resp., RSA-2048). Small key sizes offer potential reduction in processing power, memory, bandwidth, and energy. Some factoring algorithms are tailored to perform better when the integer $n = pq$ being factored is of a special form: these are called special-purpose factoring algorithms. The running time of such algorithms depends on certain properties of the factors of n. The special-purpose factoring algorithms include trial division, Pollard's rho algorithm, Pollard's $p - 1$ algorithm, the elliptic curve algorithm, and the special number field sieve [43]. In contrast, the running times of so-called general-purpose factoring algorithms depend solely on the size of n. The general-purpose factoring algorithms include the quadratic sieve and the general number field sieve [43]. The running time of Pollard's $p - 1$ algorithm for finding the factor p is $\mathcal{O}(Blnn/lnB)$ modular multiplications, where n is an integer having a prime factor p such that $p - 1$ is B-smooth. The elliptic curve method has an expected running time of $L_p[1/2, \sqrt{2}]$ to find a factor p of n. To optimize the running time of the quadratic sieve, the size of the factor base should be judiciously chosen. The optimal selection of $t = L_n[1/2, 1/2]$ is derived from knowledge concerning the distribution of smooth integers close to $\sqrt{n}$. With this choice, the quadratic sieve algorithm has an expected running time of $L_n[1/2, 1]$, independent of the size of the factors of n. A special version of the algorithm (the special number field sieve) applies to integers of the form $n = r^e - s$ for small r and $|s|$, and has an expected running time of $L_n[1/3, c]$, where $c = (32/9)^{1/3} = 1.526$. The general version of the algorithm, called a general number field sieve, applies to all integers and has an expected running time of $L_n[1/3, c]$, where $c = (64/9)^{1/3} = 1.923$. This is, asymptotically, the fastest sub-exponential algorithm known for integer factorization.

The security of ECC is based on the intractability of the ECDLP, which is an elliptic curve version of the DLP. There are several known algorithms for solving discrete logarithms: generic algorithms and group-specific algorithms. The generic algorithms can be generally applied to any type of cyclic group. The group-specific algorithms are specialized algorithms that make use of the structure in the group elements and apply only within certain families of groups. The generic algorithms include Shank's algorithm, which is also called the Baby-Step Giant-Step algorithm, Pollard's Rho and Pollard's Kangaroo algorithms [43] which are applied to any cyclic group including elliptic curve groups and subgroups of $\mathbb{Z}_p$. These are standard "square-root" methods to compute discrete logarithms in a group of prime order l: if we write the group operation multiplicatively, write g for the standard generator of the group, and write h for the DLP input: the objective is to compute $log_g h$, i.e., the unique integer x modulo l such that $h = g^x$. "Square-root" means that the algorithms take $\mathcal{O}(\sqrt{l})$ multiplications on average over all group elements h. The group-specific algorithms are index calculus algorithms. The best algorithm known for computing logarithms in F_{2^m} is a variation of the index-calculus algorithm called Coppersmith's algorithm [68], with an expected running time of $L_{2^m}[1/3, c]$ for some constant $c < 1.587$. The best algorithm known for computing logarithms in $\mathbb{Z}_p^*$ is a variation of the index-calculus algorithm called a number field sieve, with an expected running time of $L_p[1/3, 1.923]$. Considering these algorithms for solving the IFP and ECDLP together with other factors, the lower bounds for computationally equivalent key sizes which give one of useful guidelines for the determining of key sizes for symmetric cryptosystems and conventional PKCs are given Table XVIII in Section IV-E.

Implementations of RSA and ECC: Gura *et al.* [1] showed the viability of RSA and ECC on small devices without hardware acceleration. They implemented elliptic curve scalar multiplications (SMs) for 160-bit, 192-bit, and 224-bit NIST/SECG curves over $\mathbb{F}_p$, secp160r1, secp192r1, and secp224r1, respectively, and modular exponentiations for RSA-1024 and RSA-2048 on two 8-bit platforms in assembly code, where p is a prime number. Here, a SM is a multiplication operation of a point, P, on the elliptic curve by a scalar, k, to obtain $Q = kP$: this represents P added to itself k times where the addition is as defined in the elliptic curve group. An exponentiation is a modular exponentiation: a computation of the form a^b mod c. They chose a Chipcon CC1010 8-bit microcontroller which implements the Intel 8051 instruction set. CC1010 contains 32 KB of FLASH program memory, 2 KB of external data memory and 128 bytes of internal data memory. They also chose ATmega128, a popular processor used for sensor network research, on the Crossbow motes platform [65]. ATmega128 is an 8-bit microcontroller based on the AVR architecture and contains 128 KB of FLASH program memory and 4 KB of data memory. On ATmega128 at 8 MHz, they measured 0.81 s for a 160-bit ECC SM and 0.43 s for a RSA-1024 operation with an exponent $e = 2^{16} + 1$. The relative performance advantage of ECC SM over RSA modular exponentiation increases with the decrease in processor word size and the increase in key size. Elliptic curves over fields using pseudo-Mersenne primes as standardized by NIST and SECG allow for high performance implementations and show no performance disadvantage over optimal extension fields or prime fields selected specifically for a particular processor architecture. For both the CC1010 and the ATmega128, ECC-160 SM outperforms the RSA-1024 private-key operation by an order of magnitude and is within a factor of 2 of the RSA-1024 public-key operation. Due to the performance characteristics of Montgomery reduction and pseudo-Mersenne prime reduction, this ratio favors ECC-224 even more when compared to RSA-2048. Table II summarizes performance, memory usage, and code size of the ECC and RSA implementations in [1].

In TinyECC [35], it is shown that the signature generation of ECDSA on MSP430 at an 80-bit security level takes 1.6 s, where ECDSA is the Elliptic Curve Digital Signature Algorithm standardized in ANSI X9.62 [69] and FIPS 186-2 [70]. Recently, Oliveira *et al.* [44] investigated the resource overheads in communication and computation of ECDSA on AVR ATmega128L clocked at 7.3728 MHz and MSP430 clocked at 16 MHz with 802.15.4 radios. They performed software

TABLE II
Average ECC and RSA Execution Times on ATmega128 and CC1010

Algorithm	ATmega128 @ 8MHz			CC1010 @ 14.7456MHz		
	time	data mem	code	time	data mem	code
	s	bytes	bytes	s	ext+int, bytes	bytes
ECC secp160r1	0.81s	282	3682	4.58s	180+86	2166
ECC secp192r1	1.24s	336	3979	7.56s	216+102	2152
ECC secp224r1	2.19s	422	4812	11.98s	259+114	2214
Mod. exp. 512	5.37s	328	1071	53.33s	321+71	764
RSA-1024 public-key e = 2^{16} + 1	0.43s	542	1073	$>$ 4.48s		
RSA-1024 private-key w. CRT	10.99s	930	6292	$\sim$ 106.66s		
RSA-2048 public-key e = 2^{16} + 1	1.94s	1332	2854			
RSA-2048 private-key w. CRT	83.26s	1853	7736			

TABLE III
Computation Costs and Memory Overheads of ECDSA with the Koblitz Curve on MICAz and Tmote Sky

Sensor Motes	Field	Sign (ms)	Energy (mJ)	RAM (Global) (KB)	RAM (Stack) (KB)	ROM (KB)
MICAz	Binary	370	8.9	1.492	1.326	34.5
	Prime	680	17.5	1.361	1.783	36.9
Tmote Sky	Binary	128	3.8	1.260	1.834	27.3
	Prime	134	4.0	1.390	1.923	27.4

implementations on the binary Koblitz curve, sect163k1, and on the prime curve, secp160k1, defined by [71] at an 80-bit security level. Finite fields commonly used in ECC schemes are the prime field $\mathbb{F}_p$ and the binary field $\mathbb{F}_{2^m}$. Both curves over prime or binary fields offer some optimizations for SMs. The special family of binary curves, called Koblitz curves [72], have the property that a SM can be accelerated by exploiting the Frobenius endomorphism $\pi : (x, y) \rightarrow (x^2, y^2)$. For example, the wTNAF [73] method for SMs, replaces the computation of point doubling $2P$ by $\pi(P)$, a much faster operation. On the other hand, for certain ordinary elliptic curves defined over prime fields ($p > 3$), that have an efficiently computable endomorphism, the technique of Gallant, Lambert and Vanstone (GLV) [74] can be used to speed SMs on these curves. In [44], they used the wTNAF method with $w = 5$ on the binary Koblitz curves resulting in 7 precomputed points stored in ROM: in general, when using the wTNAP method, for example, $2^{w-2} - 1$ points are precomputed off-line, where w is the number of bits processed at once. Each precomputed value requires 44 bytes for storage resulting a storage overhead of 308 bytes. This amount of overhead is acceptable in current sensor platforms. Table III shows the computation costs and memory overhead for SMs on the binary Koblitz curve and the prime curve, respectively, which is the best known results [44]. In Table III, execution time for Sign means that for a SM, as the computational overhead of signature generation of ECDSA involves a SM and relatively negligible modular operations. Recently, Chu *et al.* [75] implemented ECC with two different families of elliptic curves, namely Weierstraß-form and Twisted Edwards curves on Optimal Prime Fields (OPFs) as underlying algebraic structure and supports. Due to the combination of efficient field arithmetic and fast group operations, they achieved an execution time of $5.8 \cdot 10^6$ clock cycles for a full 158-bit SM on ATmega128 which is 2.78 times faster than the widely-used TinyECC library. They also showed that the energy cost of SM on MICAz amounts to just 19 mJ when using the Twisted Edwards curve over a 160-bit OPF. Oliveira *et al.*'s implementations for SMs on the binary Koblitz curve on MICAz and Tmote Sky are the fastest among the existing ones.

The microcontrollers from the MSP430 family have many characteristics in common, such as being 16-bit, having the same instruction set and 12 general-purpose registers. The clock frequency and ROM/RAM sizes vary for each member. The MSP430 family instruction set has addition, subtraction and 1-bit only shifts. A swap byte instruction is available, which can be used for cheaper 8-bit shifts. Integer multiplication is carried out with a hardware multiplier. A memory mapped peripheral that is present in all mentioned models. The cost of using this hardware is simply the cost of writing the operands and reading the result to/from a certain memory address. There is no division instruction. Operands can be referenced by four addressing modes: register direct, indexed, register indirect and indirect with auto-increment. Instructions can use immediate constants that are codified in offset words adjacent to the instruction. The number of cycles that an instruction takes to execute can be computed easily, with a few exceptions. It takes one cycle to fetch the instruction and one cycle to read each offset word, if any. Add one cycle for each in-memory source (read) and two cycles for a in memory destination (write). The MSP430 architecture was later expanded into the backward-compatible MSP430X architecture. These microcontrollers (also referred to as MSPX) are able to address up to 1 MB of memory with 20-bit pointers. New instructions are available, such as pushing and popping multiple registers with only one instruction and up to 4-bit shifts with one instruction (which still take the same number of cycles than using separate instructions). Its instructions timings are also different: the most important distinction is that moving data to memory takes one less cycle to execute. Some new MSP430X models feature a new 32-bit hardware multiplier (referred to as MPY32) whose usage is similar to the old 16-bit multiplier and which can be used to greatly improve

TABLE IV
FEATURE OF RELEVANT MSP DEVICES

Microcontroller	MSPX	Clock (MHz)	ROM (KB)	RAM (KB)	MPY32
MSP430F1611	No	8	48	10	No
MSP430F2417	Yes	16	92	8	No
MSP430F1611	Yes	20	32	4	Yes
MSP430F1611	Yes	25	128	8	Yes

TABLE V
TIMINGS IN SECONDS FOR SIGNATURE SCHEMES

Protocol	Curve	Sign			**Verify**		
		MSP	MSPX	MPY32	MSP	MSPX	MPY32
80 bits							
ECDSA	secp160rl [78]	0.315	0.266	0.218	0.625	0.549	0.449
	secp160kl [78]	0.254	0.214	0.179	0.450	0.387	0.327
	K-163 [79]	0.256	0.231		0.481	0.443	
ZSS	BN-158 [45]	0.356	0.284	0.213	5.908	4.077	3.032
128 bits							
ECDSA	p-256 [79]	0.924	0.807	0.557	1.882	1.671	1.156
	secp256kl [78]	0.719	0.618	0.449	1.275	1.099	0.847
	K-283 [79]	0.772	0.718		1.474	1.397	
ZSS	BN-158 [45]	1.056	0.890	0.632	16.611	14.027	9.631

TABLE VI
TIMINGS IN SECONDS FOR PAIRING COMPUTATION

Algorithm	MSP	MSPX	MPY32
80 bits			
η_T	2.856	2.395	
Optimal Ate	3.791	3.202	2.470
128 bits			
Optimal Ate	9.930	8.461	5.967

the performance of cryptographic operations. Table IV [78] presents the features of some relevant microcontrollers, including MSP430F1611 used in Tmote Sky and TelosB, and MSP430F2417 used in TinyNode 184. Gouvêa *et al.* [78] measured the time for sign and verify in ECDSA and Zhang-Safavi-Naini-Susilo short signature scheme (ZSS) [79] on these microcontrollers, which are summarized in Table V.

For the pairing implementation of ZSS, two BN curves [80] over prime fields with 158 bits (BN-158) and 254 bits (BN-254) were chosen, using the Optimal Ate pairing [81]. The fastest pairings are called optimal pairings [81]: the Optimal Ate pairing was chosen for the prime case and the η_T pairing [82] for the binary case. Their implementation results of the η_T pairing and the Optimal Ate pairing at 80-bit and 128-bit security levels on the MSP430 family [78] are given in Table VI. In fact, a pairing-based public-key signature (PBS) scheme is more suitable for node-to-BS communications, as the PBS requires a pairing computation for signature verification on only the BS side and its signature length is shorter than a pairing-free PKS's length (e.g., the signature length of ZSS is half that of ECDSA). However, the PBS schemes are not suitable for node-to-node communications, as certificates of nodes' public keys are exchanged and the pairing computation is very heavy on the node side, so communication overhead and signature verification overhead on the node side are twice than the case of the node-to-BS communication.

Speed-Up Technologies for Scalar Multiplications on Elliptic Curves: In 2001, Gallant *et al.* [74] proposed a new method (GLV method) for accelerating scalar multiplications (SMs) on certain classes of elliptic curves with efficiently computable endomorphisms. Let E be an elliptic curve over a finite field $\mathbb{F}_q$ and let $P \in E(\mathbb{F}_q)$ have prime order r. Given an efficiently computable endomorphism ψ for E s.t. $\psi(P) = \lambda P$, the GLV method consists in replacing the computation kP by a multi-SM with the form $k_1 P + k_2 \psi(P)$, where the decomposition coefficients $|k_1|, |k_2| \approx r^{1/2}$. Since the number of doublings is halved, this method potentially injects a significant speedup in the SM computation on these elliptic curves. This approach might be generalized to m-dimensional case, which can achieve further speedups, if one could get higher degree decompositions with the form $k_1 P + k_2 \psi(P) + \cdots + k_m \psi(P)^{m-1}$, where $|k_i| \approx r^{1/m}$. Constructing efficiently computable endomorphisms is one of the key problems in the GLV method. In 2002, Iijima *et al.* [83] constructed an efficient computable homomorphism on elliptic curves $E(\mathbb{F}_{p^2})$ with $j(E) \in \mathbb{F}_p$ arising from the Frobenius map on a twist of E. Galbraith *et al.* [84] generalized their construction for a large class of elliptic curves over $\mathbb{F}_{p^2}$ (referred to as GLS curves) and applied the GLV method. They gave detailed implementations on these curves, showing that their method ran in between 0.70 and 0.84 the time of the best methods for SMs on general curves at that time. Longa and Sica [85] showed how to merge the two approaches for twists of any GLV curve over $\mathbb{F}_p^2$ to get a four-dimensional decomposition together with fast endomorphisms ϕ, φ over $\mathbb{F}_p^2$ acting on the group generated by a point P of a prime order n, resulting in a proven decomposition for any scalar $k \in [1, n]$ given by

$$kP = k_1 P + k_2 \phi(P) + k_3 \varphi(P) + k_4 \psi\phi(P),$$

with $\max_i(|k_i|) < C_2 n^{1/4}$, for some explicit $C_2 > 0$. They show that the use of the merged GLV-GLS approach supports a SM that runs up to 50% faster than the original GLV method. Morozov *et al.* [86] investigated the design space of ECC on TI's OMAP 3530 platform, with a focus on using OMAP's DSP core to accelerate ECC computations for ARM Cortex A8 core. Their implementation results of SMs over curves, sect163r1, sect283r1 standardized by SEC [87] and NIST on ARM Cortex A8 with DSP are 2.1 ms and 7.965 ms, respectively. Processors based on the ARM architecture are widely used in modern smartphones and tablets due to their low power consumption. They proposed a technique that interleaves ARM-based and NEON-based multiprecision operations, such as multiplication, squaring and modular reduction, in extension field operations in order to maximize the inherent parallelism and hide the execution latency. Recently, Câmara *et al.* [88] implemented SMs of standard NIST curves at the 128-and 256-bit levels of security on ARM Cortex-A8, A9 and A15 processors using NEON. Their records for implementing a fixed-based SM on the standard binary Koblitz curve, K-283, with basic SCA resistance are 404×10^3 cycles and 263×10^3 cycles on Cortex-A9 and A15 processors, respectively.

Faz-Hernádez *et al.* [89] implemented the GLV-GLS Twisted Edwards Curve (Ted127-glv4) defined over $\mathbb{F}_{p^2}$, which supports a four dimensional decomposition of the scalar and is fully protected against timing attacks on ARM processors. They reported that costs of SMs on the curve Ted127-glv4 with full protection against timing-type SCAs at approximately 128-bit security level are 204×10^3 cycles and 116×10^3 cycles on ARM Cortex-A9 and A15 processors using NEON, respectively. These results are about 2 times faster than that of the NIST curve, K-283, at the 128-bit security level on the ARM processors. Recently, Chu *et al.* [88] implemented ECC with Twisted Edwards-form curves on Optimal Prime Fields (OPFs) without GLV-GLS methods: timing for a SM of the Twisted Edwards-form curve is 5.8×10^6 clock cycles for a full 158-bit SM on an 8-bit ATmega128, which is about 2 times slower than that of the binary Koblitz curve [44] at the 80-bit security level. It is because they didn't use any recent optimization techniques. Although there exists no implementation results of the Twisted Edwards Curve with state-of-the-art speeding up methods for SMs such as the GLV-GLS method and protection against several SCAs on the WSN hardware platforms, we expect that the results will be faster than those of the standard elliptic curves on these platforms. Thus, it is required to investigate the feasibility and superiority of the performance of SMs over Twisted Edwards curves on these platforms.

SCAs on RSA and ECC: At last, we discuss SCAs on RSA and ECC. A modular exponentiation of RSA is insecure against several SCAs including timing attacks, SPA, DPA and fault injection attacks [90]–[93]. However, the implementations of RSAES-OAEP can be made resistant to timing attacks and power analysis by ensuring that all the steps in the computation of a secret key operation take the same amount of time or consume the same amount of power [94]. A more elegant approach to providing resistance to timing attacks [95] is to use "blinding" suggested by Ron Rivest. Similar approach could be incorporated for countermeasures for Diffie-Hellman and DSA. Fault analysis is potentially applicable to an implementation of RSAEP/RSADP which uses the Chinese Remainder Theorem [96] during decryption. However, the use of OAEP padding means that this attack does not pose a threat to RSAES-OAEP [94]. In practice, execution of an elliptic curve SM, kP, can leak information of k in many ways. The goal of an attacker is to retrieve the entire bit stream of k using physical attacks. Note that for some scenarios, the attacker only needs to recover a few bits of k to break the scheme. Nguyen and Shparlinski [97] have shown that a few bits of k from a couple of signatures are enough to break ECDSA. Many protection methods have been proposed to counteract the SCAs. However, countermeasures are normally proposed to prevent an implementation from a specific attack. Recently, Fan and Verbauwhede [98], and Gio [99] provided a summary paper on known physical attacks and countermeasures on ECC and three principles for the selection of countermeasures. For implementers of ECC, their papers can be used as a road map for countermeasure selections in design stages. Error message-based SCAs on public-key cryptosystems are possible [36].

The most prominent and convincing example of SCAs exploiting error messages is Bleichenbacher's attack [54] on the RSA encryption in PKCS #1 Version 1.5 [55]. This RSA encryption, which specifies a method for formatting the plaintext message prior to application of the RSA function, is widely deployed in practice including in the SSL protocol for secure web communications. It was assumed that an attacker has access to an oracle that returns a bit telling whether the ciphertext corresponds to data encrypted according to RSA in PKCS #1 Version 1.5. For 1024-bit RSA moduli, Bleichenbacher's attack enables an adversary to obtain the decryption of a target ciphertext C by submitting about one million carefully chosen ciphertexts related to C to the victim and learning whether the ciphertexts were rejected or not. The attack necessitated a patch to numerous SSL implementations. The RSA-OAEP encryption that uses Optimal Asymmetric Encryption Padding (OAEP) was proposed by Bellare and Rogaway [53] and proved secure in the random oracle model by Shoup [100] and Fujisaki *et al.* [101]. PKCS #1 Version 2.0 introduced a new algorithm RSA-OAEP to counteract the attack. Manger [56] presented CCA2 on RSAES-OAEP in PKCS #1 Version 2.0 [55]. After the publication of the results of Bleichenbacher and Manger [54], [57], it is widely believed to be important to include a strong integrity check into the RSA encryption. Klima *et al.* [102] introduced a new SCA on EME-OAEP in PKCS #1 Version 2.1 [58]. What they attacked is the part of the plaintext which is shielded by the OAEP method. They also showed that Bleichenbacher's and Manger's attack on the RSA encryption in PKCS #1 Version 1.5 and RSAES-OAEP in PKCS #1 Version 2.0 can be converted to attacks on the RSA signature scheme with any message encoding. Furthermore, Klima *et al.* [103] pointed out that incorporating a version number check over PKCS #1 plaintext used in the SSL/TLS also creates a side channel that allows an attacker to invert the RSA encryption. Using this attack, one can either recover the premaster-secret or sign a message on behalf of the server in an SSL/TLS session. Even so, one can also propose adding a cryptographic checkable redundancy code (crypto-CRC) of the whole padded message (like a hashed value) in the plaintext and encrypt $(message||padding||H(message||padding))$, where H is a secure hash function. In this way, any forged ciphertext will have a negligible probability to be accepted as a valid ciphertext. Basically, attackers are no longer able to forge valid ciphertexts, so the scheme is virtually resistant against chosen ciphertext attacks. Obviously, it is important to pad before hashing: padding after hashing would lead to the a similar attack. The right enciphering sequence is thus (*pad, hash, encrypt*). Conversely, the right deciphering sequence consists of decrypting, checking the hashed value, then checking the padding value. Invalid hashed value must abort the decipherment [36].

B. Pairing-Based Cryptography

Shamir proposed an ID-based signature (IBS) scheme in [64], he left as an open problem to find an ID-based encryption (IBE) scheme. Over the years a number of researchers tried to propose secure and efficient IBE schemes, but with little success. This state of affairs changed in 2001 when an IBE scheme based on the Weil pairing was proposed by Boneh and Franklin [104]. Since Boneh and Franklin's ID-based encryption (IBE) scheme based on the Weil pairing [104], bilinear pairings of algebraic curves have made it possible to realize cryptographic primitives that were previously unknown or impractical. These pairings have greatly contributed to the construction of a short signature scheme [105], a tripartite key agreement protocol [106], a non-interactive ID-based key agreement protocol [107], an efficient broadcast encryption scheme [108], a keyword-searchable encryption scheme [109] and an IBE scheme [104]. Despite the several advantages of pairing-based cryptosystems (PBC), the implementation of the pairing is very expensive.

Base Problems: Underlying hard problems of PBCs are as follows:

- **Gap Diffie-Hellman Problem (CGHP)**. Given a triple (P, xP, yP), find xyP with the help of a DDH Oracle.
- **Bilinear Diffie-Hellman Problem (BDHP)**. Given a triple (xP, yP, zP), compute $e(P, P)^{xyz}$.

Security of PBCs depends on the difficulty of solving the DHP or the discrete logarithm problem (DLP) in the pairing groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, where the pairing is $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Since $\mathbb{G}_1$ and $\mathbb{G}_2$ are prime subgroups of elliptic curve groups, they must have at least 160 and 256 bits at the 80-bit and 128-bit levels of security, respectively. The finite field from which $\mathbb{G}_T$ is a subgroup, however, must have a larger size than $\mathbb{G}_1$ and $\mathbb{G}_2$ due to the existence of sub-exponential attacks to its DLP. For prime fields, the standard [110] recommends orders with 1,024 and 3,072 bits at the 80-bit and 128-bit levels of security, respectively. For binary fields, due to Coppersmith's attack [68], a more conservative approach is required: at least approximately 1,412 and 4,036 bits at the 80-bit and 128-bit levels, respectively.

Implementations of Pairings: The standard algorithm for computing pairings is the Miller algorithm [111]. The first pairings were the Weil and Tate defined on supersingular curves. First, TinyTate [112] took around 31 s to compute the Tate pairing with a security level of 512-bit RSA using TinyECC [35] on ATmega128L. With NanoECC [113], the η_T pairing and Tate pairing with an 80-bit security level (1024-bit RSA) can be computed in 10.96 s and 17.93 s on ATmega128L, and in 5.25 s and 11.82 s on MSP430, respectively. Ishiguro *et al.* [114] implemented the η_T pairing over ternary fields in 5.79 s on ATmega128L. Szczechowiak *et al.* [115] implemented the η_T pairing in only 2.66 s, 1.71 s and 0.46 s on ATmega128L, MSP430 and PXA27x, respectively. More recently, Oliveira *et al.* [116] experimented an authenticated ID-based non-interactive protocol via TinyPBC. They showed that the η_T pairing could be computed in 1.9 s, 1.27 s and 0.46 s on ATmega128L, MSP430 and PXA27x, respectively. The implementation of the η_T pairing defined over binary fields which is proposed by Barreto *et al.* [82] is the fastest at this 80-bit security level. Like most pairings, the η_T pairing uses a variant of Miller's algorithm to evaluate pairings. However, the η_T pairing requires only half the number of iterations of the Miller's loop compared with other pairings. Recently, Gouvêa *et al.* [78] implemented the Optimal Ate pairing defined over Barreto-Naehrig curves, BN-158, BN-254, on prime fields at 80-bit and 128-bit security levels, respectively, on MSP430. According to [78], their execution timings are 3.791 s and 9.930 s at 80-bit and 128-bit security levels, respectively. Considering the state-of-the-art implementation results, the computation time for a pairing computation is at least two (at most seven) times slower than that of a scalar multiplication (SM) on the elliptic curve group depending on the selection of the parameters and on the WSN hardware platforms.

TABLE VII
PERFORMANCE EVALUATION OF TWO OPERATIONS ON THREE WSN HARDWARE PLATFORMS

	MICAz	Tmote Sky	Imote (13MHz)
Pairing	2.66s / 62.73mJ	1.71s / 17.70mJ	0.46s / 12.12mJ
`MapToPoint`	1.55s / 36.55mJ	1.07s / 11.07mJ	0.28s / 7.38mJ

Among several ID-based cryptographic primitives, IBE, IBS schemes and special-purposed IBS schemes can be constructed without using the pairings. However, there is no ID-based non-interactive key agreement (IBNKA) protocol without using the pairings. The IBNKA protocol require the pairing computations in the session key computation. In the scheme, the `MapToPoint` function which is used to map identity information into a point on an elliptic curve is indispensably accompanied. Although there have been much discussion regarding the construction of such an algorithm, there have been no efficient deterministic polynomial time algorithm proposed for it thus far. Their experimental results of the pairing and `MapToPint` on the three WSN platforms in [117] are summarized in Table VII. To implement the pairing, they used MIRACL [118] library which provides all of the necessary tools to perform operations on elliptic curves. A `MapToPoint` operation also takes a considerable amount of time. One of the most important issues for sensor devices is efficient memory utilization. The pairing calculation requires a significant amount of ROM on all platforms. This is mainly due to the large size of the ECC library (more than 40 KB on MICAz, around 20 KB on Tmote Sky and 25 KB on Imote2). The code needed to perform

`MapToPoint` is also included in this library. Open research issues in this area are to construct a pairing-free IBE scheme and a pairing-free IBNKA protocol, and to accelerate speed of the pairing computation which is almost the same as that of a SM.

Three Types of Pairings: There have also been several proposals of security analysis for solving the DLP on $\mathbb{G}_T$ in the case of small characteristic [119], [120]. Hayashi *et al.* [120] showed that the DLP over $\mathbb{F}^{397\cdot 6}$ can be solved. Subsequently, Adj *et al.* [119] reported that the actual security level of the curves with characteristic 3 is lower than was previously estimated. In the case of characteristic 2, Joux [121] reported that the DLP in $\mathbb{F}^{2254\cdot 24}$ can be solved in practical time. $\mathbb{G}_T$ is included in the extension field of degree 4 or 12, thus $\mathbb{G}_T$ is also included in $\mathbb{F}^{2254\cdot 24}$. These woks of record-breaking discrete logarithm computations defined over fields with characteristic 2 or 3 mean that "Type 1" pairings are insecure [122]. The pairings defined by $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ are classified into the following three types [123]:

- **Type 1**. $\mathbb{G}_1 = \mathbb{G}_2$.
- **Type 2**. $\mathbb{G}_1 \neq \mathbb{G}_2$, but there is an efficiently computable homomorphism $\phi : \mathbb{G}_2 \to \mathbb{G}_1$.
- **Type 3**. $\mathbb{G}_1 \neq \mathbb{G}_2$ and there are no efficiently computable homomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$.

Although Type 3 pairings are usually efficient, some researchers have preferred Type 1 pairings as they are convenient in certain applications. Hence, there are a lot of cryptographic protocols that have been designed only for Type 1 pairings. To implement Type 1 pairings, there are traditionally two choices:

- using fields of characteristic 2 or 3
- using supersingular elliptic curves (embedding degree 2) over $\mathbb{G}_p$, where p is at least a 500-bit prime (and, nowadays, probably at least a 1500-bit prime).

The first option was by far the most efficient, but the recent discrete logarithm algorithms and computational records mean it must now be considered insecure. Hence, we have to choose the relatively slow choice of elliptic curves over $\mathbb{F}_p$. However, Type 1 pairings over $\mathbb{F}_p$ are be much slower than Type 3 pairings. Therefore, it should use Type 3 pairings for secure an efficient implementations in future protocol designs. As seen in the paragraph of "Implementations of Pairings", Oliviera *et al.*'s software implementation results of the η_T pairing defined over supersingular curves on binary fields with an embedding degree of four on the three microcontrollers are the fastest among all the existing results. However, Type 1 pairings defined over characteristic 2 finite fields are insecure, and so Type 3 pairings are the desirable choice for pairing implementations. The typical example of type 3 pairing is the optimal ate pairing defined on BN curves. The implementation results of the Optimal Ate pairing at 80-bit and 128-bit security levels on the MSP430 family are given in Table VI.

SCAs on PBC: Bilinear pairings are realized on groups of elliptic curves. Although pairing-based cryptography (PBC) uses methods from elliptic curve cryptography (ECC), the vulnerability of PBCs against SCAs is not well understood. In ECC, the secret is a scalar multiplier of a point on the curve. In PBC, the secret is usually a point on the curve. Although the pairing itself is an interesting target for SCAs, but the effort that has been spent on the analysis of SCAs on PBC is much smaller than in the case of ECC. There are some results that analyze the vulnerability of pairings against passive attacks as well as against active attacks [124]–[127]. Target implementation is mainly the η_T pairing [82] defined over binary fields which is known as the fastest at an 80-bit security level. Several countermeasures have been proposed to protect SCAs on PBC. They can be divided into three types: i) The first type uses the bilinearity of pairings [124]. ii) The second type, the most widely researched type, multiplies intermediate variables by random values during pairing computations [125]–[127]. iii) The third type, proposed by Shirase [128], adds some random values, i.e., random value addition (RVA), to intermediate variables suspicious of being targets of DPA for secure implementation of the η_T pairing over $\mathbb{F}_{3^n}$. Recently, Seo *et al.* [129] proposed an efficient RVA-based DPA countermeasure for securing the η_T pairing algorithms over $\mathbb{F}_{2^n}$. Due to insecurity of the pairings defined over fields with characteristic 2 or 3, countermeasures of the Type 3 pairings against SCAs such as the optimal ate pairing should be proposed.

C. PKCs Based on Hard Problems Over Lattices

Lattices have been studied by cryptographers for quite some time, both in the field of cryptanalysis [130]–[132] and as a source of hard problems on which to build encryption schemes [133]–[135]. Lattice-based cryptography (L-PKC) is also a promising area due to the simple additive, parallelizable structure of a lattice. A lattice $\mathcal{L}$ is a discrete additive subgroup of $\mathbb{R}^n$. By discrete, we mean that there exists an $\varepsilon > 0$ such that for any $\mathbf{v} \in \mathcal{L}$, and all $\mathbf{w} \in \mathbb{R}^m$, if $||\mathbf{v} - \mathbf{w}|| < \varepsilon$ then $\mathbf{w}$ does not belong to the lattice $\mathcal{L}$. Let $\mathbf{v}_1, c \ldots, \mathbf{v}_k$ be a set of vectors in $\mathbb{R}^n$. An n-dimension lattice $\mathcal{L}$ is a set of vectors $\{\sum_{i=1}^{n} a_i \mathbf{v}_i | a_i \in \mathbb{Z}$, where $\mathbf{v}_1, \cdots, \mathbf{v}_n\} \in \mathbb{Z}^n$ is a set of linear independent vectors, called the basis of the lattice.

Base Problems and Algorithms for Solving the Problems: Let $\mathcal{L}$ be an n-dimension lattice.

- **Shortest Vector Problem (SVP)**. Find the shortest nonzero vector in $\mathcal{L}$, i.e., find $0 \neq v \in \mathcal{L}$ such that $||\mathbf{v}||$ is minimized.
- **Closest Vector Problem (CVP)**. Given a vector $\mathbf{w}$ which is not in $\mathcal{L}$, find the vector $\mathbf{v} \in \mathcal{L}$ closest to $\mathbf{w}$, i.e., find $v \in \mathcal{L}$ such that $||\mathbf{v} - \mathbf{w}||$ is minimized.
- **Approximate Shortest Vector Problem (ASVP)**. Find a nonzero vector $\mathbf{v} \in \mathcal{L}$ satisfying $||\mathbf{v}|| \leq \psi(n)||\mathbf{u}||$ for any (other) vector $u \in \mathcal{L}$, where ψ is some slowly growing function of n.
- **Approximate Closest Vector Problem (ACVP)**. Find a nonzero vector $\mathbf{v} \in \mathcal{L}$ satisfying $||\mathbf{v} - \mathbf{w}|| \leq \psi(n)||\mathbf{u} - \mathbf{w}||$ for any (other) vector $u \in \mathcal{L}$.

The two basic hard problems SVP and CVP are known to be NP-hard to solve exactly [136], [137] and also NP-hard to approximate [138], [139] within at least constant factors. Time

TABLE VIII
COMPLEXITY OF KNOWN SVP/CVP ALGORITHMS

Algorithm	Time	Memory	CVP	SVP	Proven/Heuristic
Kannan-Enumeration	$n^{n/(2e)+o(n)}$	poly(n)	✓	✓	proven
Voronoi-cell	2^{3n}	2^{n}	✓	✓	proven
List-Sieve	$2^{2.465n+o(n)}$	$2^{1.233n+o(n)}$	×	✓	proven
GaussSieve	-	$2^{0.41n+o(n)}$	?	✓	heuristic
Nguyen-Vidick sieve	$2^{0.415n+o(n)}$	$2^{0.2075n+o(n)}$	×	✓	heuristic
WLTB sieve	$2^{0.3836n+o(n)}$	$2^{0.2557n+o(n)}$	×	✓	heuristic
Becker-Gama-Joux	from $2^{0.415n}$ to $2^{0.377n}$	$2^{0.2075n}$ $2^{0.292n}$	✓	✓	heuristic

TABLE IX
COMPARISON OF NTRU AND RSA

System	Security (MIPS yrs)	Public Key (Ciphertext) Size (bits)	Private Key Size (bits)	Create Key (millisecs)	Encrypt (blks/sec)	Decrypt (blks/sec)
RSA 512	$4.00 \cdot 10^{5}$	512	512	260	2441	122
NTRU 167	$2.08 \cdot 10^{6}$	1169	2238	4.0	5941	2818
RSA 1024	$3.00 \cdot 10^{12}$	1024	1024	1280	932	22
NTRU 263	$4.61 \cdot 10^{14}$	1841	3682	7.5	3676	1619
RSA 2048	$3.00 \cdot 10^{21}$	2048	2048	4195	310	3
RSA 4096	$2.00 \cdot 10^{33}$	4096	4096	—	—	—
NTRU 503	$3.38 \cdot 10^{35}$	4024	8048	17.3	1471	608

complexity of known algorithms that find the exact solution are at least exponential in the dimension of the lattice. These algorithms also serve as subroutines for strong polynomial time approximation algorithms. Algorithms for the exact problem hence enable us to choose appropriate parameters. A shortest vector can be found by enumeration [140], [141], sieving [142]–[145] or the Voronoi-cell algorithm [146]. Enumeration uses a negligible amount of memory and its running time is between $n^{\mathcal{O}(n)}$ and $2^{\mathcal{O}(n^2)}$ depending on the amount and quality of preprocessing. Probabilistic sieving algorithms, as well as the deterministic Voronoi-cell algorithm are simple exponential in time and memory. A closest vector can be found by enumeration and by the Voronoi-cell algorithm, however, state-of-the-art sieving techniques cannot be directly applied to solve CVP instances. Table VIII presents a summary of the complexities of currently known SVP and CVP algorithms.

Constructions and Implementations of L-PKCs: Goldreich *et al.* [134] proposed a PKE scheme and a PKS scheme relying on the computational difficulty of lattice reduction problems, in particular, the ACVP. Hoffstein *et al.* [135] proposed a PKE scheme NTRUEncrypt based on a particularly efficient class of convolution modular lattices, called NTRU lattices. Hoffstein *et al.* [147] proposed a PKS scheme NTRUSIGN based on solving ACVP in the special class of NTRU lattices. These schemes require arithmetic in quotient polynomial rings, $R = \mathbb{Z}(x)/(x^N - 1, q)$, called convolution polynomial rings. Due to its simple primitive operations for encryption and signing, just mere polynomial multiplications, NTRUEncrypt and NTRUSIGN claim to be faster than other asymmetric schemes. A secret key for NTRUEncrypt consists of a good basis for an N-dimensional sublattice of a $2N$-dimensional NTRU lattice, but in order to solve the ACVP efficiently for arbitrary message digest points, one must know a full good basis for the lattice. According to the results of encryption, decryption, and key creation performed using Tao Group's Tumbler implementation of NTRUEncrypt, programmed in C and running on a 300 MHz Pentium II operating under Linux, the biggest advantage of NTRU is that key creation time is much too fast [148]. Comparison of RSA and NTRUEncrypt in terms of keys sizes and execution time is given in Table IX, where RSA key creation done on a 255 MHz Digital AlphaStation and RSA encryption/decryption programmed in Microsoft Visual C++ 5.0 (optimized for speed, Pentium Pro code generation) on a Pentium II 266 MHz under Windows NT 4.0 [148]. However, in WSNs, it may not be a big advantage, as the key creation is performed by powerful administrators related the BS before deployment to the WSN. Thus, fast encryption decryption or fast signature generation/verification are preferred than the fast key creation on sensor nodes. The measurements of [135], [147], [149] imply that for the parameters $(N, p, d) = (251, 128, 72)$, a lattice attack on the secret key requires about 6.6×10^{12} MIPS years at an 80-bit security level. ECIES is an Elliptic Curve Integrated Encryption Scheme which is a hybrid scheme integrating a symmetric-key encryption algorithm and message authentication code (MAC) into the elliptic curve ElGamal encryption scheme. Due to the integration of different functions, ECIES is considered to be secure against CCA without having to increase the number of operations or the key length in the elliptic curve ElGamal encryption scheme [150], which is included in the SEC 1 document (version 2.0) [71]. In ECIES, compared to a SM on the elliptic curve, other cryptographic operations including hash, MAC, symmetric-key encryption operations are relatively negligible. Time and energy consumption required to execute NTRUEncrypt-251 with a parameter set $(N, q, p) = (251, 128, 3)$ and ECIES-160 are given in Table X and XI [151].

At Eurocrypt'06, Nguyen and Regev [152] presented the first successful key-recovery experiments on NTRUSIGN-251 without perturbation, as proposed in half of the parameter choices in the NTRU standards [153] being considered by IEEE

TABLE X
EXECUTION TIME OF NTRUENCRYPT AND ECIES ON MICAZ

	Initialization (ms)	Encrypt (ms)	Decrypt (ms)	ROM (B)	RAM (B)	Public Key (Ciphertext) (B)
NTRUEncrypt-251	23546	3895	2202	2214	0	251
ECIES-160	1839	3907	2632	18136	2156	20

TABLE XI
ENERGY CONSUMPTION OF NTRUENCRYPT AND ECIES ON MICAZ

	Initialization (mJ)	Encrypt (mJ)	Decrypt (mJ)	Transmit (μJ)	Receive (μJ)
NTRUEncrypt-251	576.1	93.5	53.5	367.21	397.08
ECIES-160	44.1	93.8	63.2	66.88	72.32

TABLE XII
NTRUSVES KEY SIZES

Parameter set	k	$s_{privKey}$	s_{pubKey}
ees251ep6	80	218 bytes	296 bytes
ees347ep2	112	529 bytes	740 bytes
ees397ep1	128	595 bytes	840 bytes
ees491ep1	160	723 bytes	1028 bytes
ees587ep1	192	853 bytes	1220 bytes
ees787ep1	256	1118 bytes	1620 bytes
ees251ep7	80	194 bytes	548 bytes
ees347ep3	112	462 bytes	740 bytes
ees397ep2	128	518 bytes	840 bytes
ees491ep2	160	630 bytes	1028 bytes
ees587ep2	192	738 bytes	1220 bytes
ees787ep2	256	969 bytes	1620 bytes

P1363.1 [154]: experimentally, 400 signatures are enough to disclose the NTRUSIGN-251 secret key due to symmetries in NTRU lattices. After Nguyen and Regev's attack, countermeasures have been proposed to repair the scheme, such as the perturbation used in NTRUSIGN standardization proposals, and the deformation proposed by Hu *et al.* [155] in 2008. These two countermeasures were claimed to prevent the attack. Recently, Ducas and Nguyen [156] revisited Nguyen-Regev attack to show that it is much more powerful than previously expected. In particular, an optimized Nguyen-Regev attack can surprisingly break in practice both NTRU's perturbation technique [147] as recommended in standardization proposals [154], [157], and Hu *et al.*'s countermeasure [155]. They can recover the secret key in a few hours, using 8,000 signatures for NTRUSIGN-251 with one perturbation submitted to IEEE P1363 standardization [158], or only 5,000 signatures for the latest 80-bit-security parameter set proposed in [157]. These are the first successful experiments fully breaking NTRUSIGN with countermeasures. NTRUEncrypt has decryption failures: with standard parameters, validly generated ciphertexts may fail to decrypt. Howgrave-Graham *et al.* [159] showed that decryption failures cannot be ignored, as they happen much more frequently than one would have expected. If one strictly follows the recommendations of the EESS standard [160], decryption failures happen as often as every 2^{12} messages with $N = 139$, and every 2^{25} messages with $N = 251$. It turns out that the probability is somewhat lower (around 2^{-40}) with NTRU products, as the key generation implemented in NTRU products surprisingly differs from the one recommended in [160]. In any case, decryption failures happen sufficiently often that one cannot dismiss them, even in NTRU products. Howgrave-Graham *et al.* [161] considered the impact of the possibility of decryption failures in proofs of security for padding schemes, where these failures depend on both message and key. They claimed that an average case failure analysis is not necessarily sufficient to achieve provable security with existing CCA2-secure schemes.

A CCA2-secure padding scheme for NTRUEncrypt was given in [162], but the proof neglects to take decryption failures into account, and is thus flawed when instantiated with current NTRUEncrypt parameter sets. Such ideas are explored further in [163]. Howgrave-Graham *et al.* [159] proposed NTRUEncrypt padding scheme, called NEAP, based on hash functions in a similar way to PSS-E [164]. They provided the full security proof of NEAP against CCA2 in the random oracle model by showing that it is still secure in the presence of decryption failures. The EESS #1 v2 standard [153] specifies an instantiation of NAEP, known as SVES-3, which is currently undergoing a standardization process and will presumably be included in the upcoming IEEE standard 1363.1 [165]. Buchmann and Döring [166] refer to SVES-3 proposed in the draft standard as NTRUSVES. They provided time measurements as well as key sizes for all parameter sets proposed by IEEE P1363.1-D9. The key sizes of NTRUSVES are given in Table XII from IEEE P1363.1-D9: column "k" denotes the bit security level of NTRUSVES with the given parameter set, and columns "$s_{privKey}$" and "s_{pubKey}" denote the size of the DER-encoded secret key and public key ASN.1 structures, respectively. We omit time measurements of NTRUSVES in [166], as they are not results implemented on the WSN platforms. Stehlé and Steinfeld [167] showed that a slight variant of NTRUEncrypt (called pNE) can be shown to achieve IND-CPA security based on worst-case lattice problems over ideal lattices. Like the original NTRUEncrypt scheme, the pNE scheme is trivially insecure against CCAs, due to its homomorphic properties. Steinfeld *et al.* [168] proposed a variant of NTRUEncrypt, called NTRUCCA, which is IND-CCA2 secure in the standard model assuming the worst-case quantum hardness of problems in ideal lattices, and only incurs a constant factor overhead in ciphertext and key length over the pNE variant shown to be IND-CPA in [167]. The scheme still preserves a key and ciphertext length and encryption/decryption computation costs quasi-linear in the security parameter, given the best known attacks. They claimed that it is the first efficient variant of NTRUEncrypt achieving IND-CCA2 security based on standard cryptographic assumptions.

SCAs on L-PKCs: Silverman and Whyte [169] presented timing attacks on NTRUEncrypt based on variation in the number of hash calls made on decryption. The attacks were applied to the parameter sets of [160], [161]. To mount the attack, an attacker performs a variable amount of precomputation, then submits a relatively small number of specially constructed ciphertexts for decryption and measures the decryption times. Comparison of the decryption times with the precomputed data allows the attacker to recover the key in greatly reduced time compared to standard attacks on NTRUEncrypt. For parameter sets in [170] that claim k-bit security but are vulnerable to this attack, they found that an attacker can typically recover a single key with about $k/2$ bits of effort. Finally, they suggested a simple means to prevent these attacks by ensuring that all operations take a constant number of SHA calls. The recommended countermeasure does not break interoperability with the parameter sets of [160], [161] and has only a slight effect on performance.

D. PKCs Based on Multivariate Quadratic Equations

MQ-PKCs are cryptosystems whose security depends on the hardness of solving multivariate quadratic equations over a finite field (MQ-Problem). Thus, estimating the hardness of the MQ-Problem is important for the security of MQ-PKCs. It is known that the MQ-Problem over a finite field is NP-hard when the coefficients are randomly chosen, and no quantum algorithm efficiently solving the MQ-Problem has been presented. Therefore, MQ-PKCs are one of candidates for post-quantum cryptography. In MQ-PKCs, a system $\mathcal{P} = (p^{(1)}, \cdots, p^{(m)})$ of multivariate quadratic polynomials with m equations and n variables is defined by

$$p^{(k)}(x_1, \cdots, x_n) = \sum_{i=1}^{n}\sum_{j=i}^{n} p_{ij}^{(k)} x_i x_j + \sum_{i=1}^{n} p_i^{(k)} x_i + p_0^{(k)},$$

for $k = 1, \cdots, m$. Inverting a system of generic multivariate quadratic (MQ) polynomials $\mathcal{P}$ is known to be hard, as the corresponding MQ-problem is proven to be NP-complete. However, the trapdoor is given by a structured quadratic central map $\mathcal{F} = (f^{(1)}, \cdots, f^{(m)})$. To hide the trapdoor, we choose two invertible affine or linear transformations S, T and define $\mathcal{P} = T \circ \mathcal{F} \circ S$. Then the public/secrte key pair is $(\mathcal{P}/(S, \mathcal{F}, T))$. Thus, how to design secure multivariate trapdoors is a core of the design for an invertible quadratic map $\mathcal{F}$.

Base Problems and Algorithms for Solving the Problems: We first introduce the MQ-problem. Let q be a power of prime and $K = GF(q)$.

- **Solving Polynomial System (SPS) Problem**: Given a system $\mathcal{P} = (p^{(1)}, \cdots, p^{(m)})$ of m nonlinear polynomial equations defined over a finite field K with degree of d in variables $x_1, \cdots, x_n$ and $y \in K^m$, find values $(x'_1, \cdots, x'_n) \in K^n$ such that $p^{(1)}(x'_1, \cdots, x'_n) = \cdots = p^{(m)}(x'_1, \cdots, x'_n) = y$.

The SPS problem is proven to be NP-complete even for the simplest case of quadratic polynomials over $\mathbb{F}_2$. For efficiency, MQ-PKCs restrict to quadratic polynomials. For the special case of all polynomials $p^{(1)}, \cdots, p^{(m)}$ having degree 2, the SPS problem is called MQ-Problem for multivariate quadratic. The MQ-Problem is efficiently solved under special n and m conditions. In particular, the algorithm by Kipnis *et al.* [171] can solve the MQ-Problem over a finite field of even characteristic in a polynomial-time of n when $n \geq m(m+1)$. The complexity of this Kipnis-Patarin-Goubin's algorithm is

$$\begin{cases} \mathcal{O}\left(n^w m (logq)^2\right), & \text{char}(K) \text{ is } 2; \\ \mathcal{O}\left(2^m n^w m (logq)^2\right), & \text{char}(K) \text{ is odd}, \end{cases}$$

where $2 \leq w \leq 3$ is the exponent of the Gaussian elimination. It is also known that Gröbner basis algorithms [172]–[174] solve the MQ-Problem, and these algorithms are more effective in the overdetermined ($n \ll m$) MQ-Problem [175], [176]. Also, this problem can be solved efficiently when the number of unknowns n is sufficiently greater than that of equations m in the underdetermined. Courtois *et al.* [177] proposed an algorithm which extend Kipnis-Patarin-Goubin's algorithm when char(K) is odd, and this algorithm can be applied when n and m satisfy $n \geq 2^{m/7}(m+1)$. This algorithm can solve the MQ-Problem of m equations and n unknowns over K in time about $2^{40}(40 + 40/logq)^{m/40}$. Thomae *et al.* [178] proposed an algorithm which can be applied when n and m satisfy $n > m$. This algorithm uses Gröbner basis algorithm, so the complexity of this algorithm is exponential-time. They claimed that the MQ-Problem of 28 equations and 84 unknowns over $GF(2^8)$ has the 80-bit security. Recently, Miura *et al.* [179] proposed an algorithm which can solve the MQ-Problem in a polynomial time of n when $n \geq m(m+3)/2$. The complexity of their algorithm is the same as Kipnis-Patarin-Goubin's. The problem is suitable for use in cryptographic primitives, while the problem alone is not sufficient. To be able to make the system solvable for the owner of some secrets and still not solvable for the rest, it needs a way to embed some hidden trapdoors. The way is built on the Isomorphism of Polynomials problem.

The security of MQ-schemes is not solely based on the MQ-Problem, but also on some variants of the Isomorphism of Polynomials (IP) problem. There exist three versions of this problem.

- **Problem IP1S (Isomorphism of Polynomials with 1 Secret)**: Given nonlinear multivariate systems $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{B} = \mathcal{A} \circ T$ for a linear or affine map T, find a map T' such that $\mathcal{B} = \mathcal{A} \circ T'$.
- **IP2S (Isomorphism of Polynomials with 2 Secrets) Problem**: Given nonlinear multivariate systems A and B such that $B = T \circ A \circ S$ for linear or affine maps S and T, find two maps S' and T' such that $B = T' \circ A \circ S'$.

MQ-PKCs relied on a somewhat heuristic construction to build trapdoor one-way functions, whose security was based

TABLE XIII
MINIMAL SECURITY PARAMETERS ACHIEVING CERTAIN LEVELS OF SECURITY

Security Level	0/1-UOV (o, v)	Rainbow (v_1, o_1, o_2)	enTTS (l, m, n)
2^{80}	(28, 37)	(18, 13, 14)	(9, 36, 52)
2^{128}	(44, 59)	(36, 21, 22)	(15, 60, 88)

on the hardness of the IP problems. Starting with an easy-to-invert quadratic map $\mathcal{F}$, one builds an apparently random-looking one by setting $\mathcal{P} = T \circ \mathcal{F} \circ S$. The idea is that the changes of coordinate would hide the structure of $\mathcal{F}$ that makes it easy to invert, so that $\mathcal{G}$ would look random. Inverting random quadratic maps is extremely hard, and the best options in general are exhaustive search, or the computation of a Gröbner basis, both techniques being exponential in n. In this setting, $\mathcal{P}$ is the public key, while S and T are the secret key. When $\mathcal{F}$ is public, then recovering the secret-key precisely means solving an instance of the IP problem. Several cryptosystems have been built on this idea [180]–[182], but they have all been broken [183]–[189]. The main reason behind this fiasco is that the specific instances of the IP problem exposed by these schemes were weak because $\mathcal{F}$ was too special, so that polynomial-time and/or efficient algorithms to crack them have eventually been designed [190], [191].

Constructions of MQ-PKCs: There are several ways to build the central map $\mathcal{F}$. Due to the uncertainty of the problem, almost all MQ schemes have been broken, including the balanced Oil and Vinegar scheme [192], SFLASH [193] and much more [194]–[199]. More precisely, Matsumoto-Imai [200] proposed an MQ-scheme which supports both encryption and digital signatures. Patarin broke Matsumoto-Imai system [195] and proposed a generalized and improved Matsumoto-Imai scheme system [201], called "Hiddden Field Equations" (HFE). The HFEs have been used to construct digital signature schemes, e.g., Quartz [202] and SFLASH [203]. SFLASH which is a modified and secured version of Matsumoto-Imai system, has been developed to suit smart-card environments. To avoid several attacks presented in [197], [201], [204], four important variations were proposed, HFE$^-$, HFE$^+$, HFEv and HFEf by removing public equations, adding public equations, adding vinegar variables and fixing variables of the public key, respectively. Patarin [195] proposed a new simple signature scheme, called "Oil and Vinegar". The idea consists in hiding quadratic equations in n unknowns called "Oil" and $v = n$ unknowns called "Vinegar" over a finite field K, with linear secret functions. This scheme is broken by Kipnis and Shamir [192]. Kipnis *et al.* [171] showed that the attack in [192] also works, exactly in the same way, when $v > n$. However, they have seen that the cases $v > n$ are much more difficult. When $v \geq n$, they called the scheme "Unbalanced Oil and Vinegar (UOV)".

At CHES 2011, Petzold *et al.* [205] showed that large parts of the public key are redundant in order to prevent key recovery attacks: $\mathcal{S}$ can be chosen of a special structure due to equivalent keys and thus large parts of the public and secret map are equal. They proposed 0/1-UOV by choosing this parts of $\mathcal{P}$ of a special structure, such that direct attacks on the public key do not become easier, they were able to reduce the key size and running time of the verification algorithm. Ding and Schmidt [206] proposed a Rainbow signature scheme, which is a generalization of the Oil-Vinegar construction to improve the efficiency of UOV. Rainbow uses the same idea as UOV but in different layers. The security parameters (q, v_1, o_1, o_2) or (v_1, o_1, o_2) of Rainbow with two layers is chosen to prevent MinRank attacks [207] and preserve short signatures size at the same time, where q, v_1 and o_i are the size of the underlying small finite field, the number of vinegar variables, and oil variables, respectively. Yang and Chen [208] proposed the enhanced TTS (enTTS) scheme which is an enhanced version of the Tame Transformation Signatures. The idea of the scheme was to use several layers of UOV trapdoors and to make them as sparse as possible. In contrast to UOV, this would prevent Kipnis-Shamir attack [192] without increasing the number of vinegar variables. All attacks against Rainbow can also be applied to enTTS. Minimal security parameters achieving certain levels of security of these three schemes are given in Table XIII [205]. In summary, nearly all MQ-encryption schemes and most of the MQ-signature schemes have been broken. There are only very few exceptions like the signature schemes UOV and its layer based variants Rainbow.

Implementations of MQ-PKCs: At CHES 2012, Czypek *et al.* [45] implemented the MQ-signature schemes, UOV, Rainbow and enTTS, on an 8-bit microcontroller with 32 MHz. The parameter sets of UOV, 0/1-UOV, Rainbow and enTTS for several security levels and their implementation results are summarized in Table XIV. Yang *et al.* [209] implemented enTTS(20,28) on MSP430 running at 8 MHz: the average signing time is 71 ms, while the average verification time is 726 ms, measured with the time provided by TinyOS at the granularity of 1/32,768 seconds, averaging over 1,000 runs. Table XV compares the implementations of enTTS, ECC, RSA, and NTRU on comparable 8-bit platforms: for a fair comparison with Czypek *et al.*'s implementation running at 32 MHz, timings at lower frequencies were scaled accordingly [45]. Compared to ECDSA with the binary Koblitz curve on Tmote Sky [44], enTTS(20, 28) is about 1.8 times faster than ECDSA in signing, while ECDSA is about 2.84 times faster than enTTS(20, 28) in verifying. These results are summarized in Table XVI. At the 80-bit security level, the sizes of public keys, secret keys and signatures of RSA, ECDSA, NTRUSIGN and enTTS are encapsulated in Table XVII. According to Table XIV, the secret key size of enTTS is 4.5 KB and the public key size is 49.6 KB at the 80-bit security level. A typical sensor node like Tmote Sky has 10 KB RAM and 48 KB flash. The secret keys can be fit into the working RAM and the public keys are too big to store on the memory. For signature verification, motes need to store the sender's public key in flash memory, as they are too big to fit into RAM: it is needed occasionally by the motes when they need to verify the authenticity of a message coming from a BS or another mote. As the security level increases, the MQ-schemes cause a significative storage cost due to the size of

TABLE XIV
IMPLEMENTATION RESULTS OF UOV, 0/1-UOV, RAINBOW AND ENTTS

	Scheme	Key Size	[Byte]	System	Parameter	Clockcyles x 1000		Time[ms]@32MHz		Code Size [Byte]	
		private	public	private	public	sign	verify	sign	verify	sign	verify
	enTTS(5,20,28)	1351	8120	*	*	153	1,126	4.79	35.22	12890	827
	enTTS(5,20,28) [211]	1417	8680	*	*	568[1]	5,808[1]	17.75[2]	181.5[2]	-	-
2^{80}	uov(28,37)	49728	60060	*	*	3,637	3,911	113.66	122.23	2188	466
	0/1 uov(28,37)	30044	11368	19684	48692	3,526	3,211	110.20	100.37	2258	578
	rainbow(18,13,14)	19682	27945	*	*	1,740	2,214	54.38	69.19	4162	466
	enTTS(9,36,52)	4591	49608	*	*	609	6,658	19.03	208.07	41232	827
2^{128}	uov(44,59)	194700	235664	*	*	13,314	14,134	416.07	441.70	2188	466
	0/1 uov(44,59)	116820	43560	77880	192104	12,782	13,569	399.43	424.04	2258	578
	rainbow(36,21,22)	97675	135880	*	*	8,227	9,216	257.11	288.01	4162	466
	enTTS(15,60,80)	13051	234960	*	*	2,142	30,789	66.94	962.17	116698	827

[*] Not applicable, [1] Derived from values in original work, [2] Scaled to the same clock frequency

TABLE XV
OVERVIEW OF IMPLEMENTATIONS OF FOUR PKCS ON 8-BIT PLATFORMS WITH 32 MHZ

Method	Time[ms]@32MHz	
	Sign	Verify
enTTS(5,20,28) [211]	17.75	181.5
ECC-P160 (SECG) [1]	203	203
ECC-P192 (SECG) [1]	310	310
ECC-P224 (SECG) [1]	548	548
RSA-1024 [1]	2,748	108
RSA-2048 [1]	20,815	485
NTRU-251-127-31 Sign [212]	143	-

TABLE XVI
COMPARISON BETWEEN ENTTS(20, 28) AND ECDSA ON TMOTE SKY

	Sign (ms)	Verify (ms)
enTTS(20, 28)	71	726
ECDSA	128	256

the keys in RAM. As seen in Table XV, MQ-signature schemes are much faster than other signature schemes. However, in reality, to use the MQ-schemes in actual WSN applications, PKI or ID-based infrastructure must be adopted. Public-key certificates required in PKI for authentication of public keys in the MQ schemes have very large size (compared to simplified ECC-160 certificate of 86 bytes [2]). In wireless communications, generally, data transmission is very expensive in terms of energy consumption. In this point of view, their high performance can be offset by these heavy transmissions.

To reduce overhead related to public-key certificate, we consider ID-based MQ-schemes. As in a generic construction [211] that converts a PKS scheme into an IBS scheme, an ID-based MQ-signature scheme is possible by sending a public key and a signature on the public key signed by a PKG's master secret together with a signature on a message. Since there is no function to map identity information into a public key $\mathcal{P} = T \circ \mathcal{F} \circ S$ in the MQ-scheme, the size of the ID-based signature is the public key size plus two MQ-signatures without reducing the sizes of public and secret keys. Thus, it needs to construct an efficient ID-based MQ-signature scheme. To adopt cryptographic primitives to WSN environments, the size of system parameters and several types of keys stored in sensor nodes must be small, at the same time, the time and energy consumption of public key encryption and signature generation/verification must be minimized. At present, MQ-PKCs are not suitable for WSNs due to their much heavy key sizes and communication cost, and the lack of proper infrastructure despite their fast speeds. Nevertheless, there are several possibilities open to many researchers.

SCAs on MQ-PKCs: There are two different secret keys for SFLASH, namely the proper secret key (S, T) and the random seed $\triangle$ used for the hash function SHA-1. Whereas many papers discussed the security of (S, T), little is known about that of $\triangle$. Steinwandt *et al.* [212] proposed a theoretical DPA on finding $\triangle$ by observing the XOR operations. Okeya *et al.* [213] proposed another DPA on $\triangle$ using the addition operation modulo 232, and presented an experimental result of the DPA. They demonstrated that if the implementation of SHA-1 is naive or careless, the attacker can recover $\triangle$: about 1408 message-signature pairs can generate a dummy signature function in a few hours, so Patarin's attack [195] can be totally performed off-line, without asking the signing oracle. In order to resist the attack, the developers of SFLASH should carefully implement SHA-1 related to $\triangle$. Recently, Hashimoto *et al.* [214] presented general fault attacks on MQ-PKCs including Big Field type (e.g., Matsumoto-Imai, HFE and SFLASH) and Stepwise Triangular System (STS) type (e.g., UOV, Rainbow and TTM/TTS).

For the Big Field type scheme (e.g., MI [200], HFE [201], SFLASH [215] and ℓIC [181]). In these schemes, the attacker can simplify the target problem to an easier one (e.g., HFE to MI) and can recover the secret affine transforms S and T by only single fault and sufficiently many pairs of messages and signatures given by the faulty central map. For the STS type scheme (e.g., Tsujii's scheme [216], Shamir's scheme [217], UOV [171], Rainbow [206] and TTM/TTS [208], [218]), the attacker can recover a part of the secret affine transform T on the quadratic forms directly from a pair of message and signature given by the faulty central map. On their attacks in practice, the fault can not always change the coefficients in $\mathcal{G}$, as there were three possible system parameters ($\mathcal{G}$, S, or T). However, the success probability of changing the central map $\mathcal{G}$ by one fault attack is high enough for attackers, and they are able to distinguish whether the location of the fault

TABLE XVII
THE SIZES OF PUBLIC KEY, PRIVATE KEY AND SIGNATURE FOR THE FOUR PKS SCHEMES AT THE 80-BIT SECURITY LEVEL

80-bit Security Level	Public Key Size (B)	Secret Key Size (B)	Signature Size (B)
Rainbow(17, 13, 13)	25,740	19,546	44
enTTS(9,36,28)	49,608	4,591	52
ECDSA-160	20	20	40
RSA-1024	128	128	128
NTRU-167	147	294	147

TABLE XVIII
NIST RECOMMENDATION FOR CRYPTOGRAPHIC KEY LENGTHS

Date	Minimum of	Symmetric	Factoring	Discrete	Discrete	Elliptic
	Strength	Algorithms	Modulus	Logarithm Key	Logarithm Group	Curve
2010	80	2TEA	1024	160	1024	160
2011-2030	112	3TDA	2048	224	2048	224
>2030	128	AES-128	3072	256	3072	256
>>2030	192	AES-192	7680	384	7680	384
>>>2030	256	AES-256	15360	512	15360	512

is in $\mathcal{G}$ or not. Unlike Okeya *et al.*'s attack, they do not use the secret information discovered by the SCA attacks but use pairs of messages and signatures given by the faulty secret information to recover the secret affine maps S and T. Though these fault attacks do not necessarily break the schemes directly, partial information of secret keys recovered by the fault attacks is usually critical in terms of preserving the security against known attacks.

E. Selecting Cryptographic Key Sizes

To protect data beyond the year 2010, RSA Security recommends RSA with 2048-bit keys as the new minimum key size which is equivalent to ECC with 224-bit keys [219]. To determine appropriate cryptographic key sizes corresponding to security levels, Lenstra *et al.* [220] presented the following four points on which the choice of cryptographic key sizes depends primarily:

- **Life Span**: the expected time the information needs to be protected.
- **Security Margin**: an acceptable degree of infeasibility of a successful attack.
- **Computing Environment**: the expected change in computational resources available to attackers.
- **Cryptanalysis**: the expected developments in cryptanalysis.

Although some of their default setting are debatable, it is believed that their lower bounds for computationally equivalent key sizes give one of useful guidelines for the determining of key sizes for symmetric cryptosystems and conventional PKCs. For years ranging from 1982 to 2050, the computationally equivalent key size recommendations resulting are given in [220], [221]. The web site in [222] summarizes reports from well-known organizations to quickly evaluate the minimum security requirements for cryptographic primitives. We can also easily compare all these techniques and find the appropriate key length for our desired level of protection. Table XVIII introduce NIST Recommendation (2012) for key lengths which is one of them in [222]. In Table XVIII, TDEA (Triple Data Encryption Algorithm) and AES (Advanced Encryption Standard) are block ciphers [223], all key sizes provided in bits are the minimal sizes for security.

V. DISCUSSIONS

Here, we discuss open research issues to improve the efficiency and security of public-key cryptographic primitives in WSNs.

First, research on a new family of elliptic curves for small wireless devices, which can be replaced with the current elliptic curve standards, is needed. As seen in Table XV and 21, the speed of L-PKC (NTRU) is 1.4 times faster than that of ECC, but the size of public key (resp., secret key) is 7.4 (resp., 14.7) times heavier than that of ECC at the 80-bit security level. Also, the speed for signing (resp., verifying) of MQ-PKC (enTTS) is 11.4 (resp., 1.12) times faster than that of ECC, but the size of public key (resp., secret key) is 2,480 (resp., 229) times heavier than that of ECC. Therefore, at present, ECC has the best balance in terms of speed, memory requirements, communication cost and energy consumption on wireless devices with very limited resources: ECC defined on the standard binary Koblitz curve is the best PKC suitable for WSNs in terms of computational complexity, communication overhead, key size, memory and storage. Although lots of efforts for speeding up on the standard elliptic curves, it still seems to be heavy on the WSN hardware platforms. So, it needs a new family of curves suitable for these platforms. Twisted Edward (TED) curves are a candidate of this new family. Although there exists no implementation results of the TED Curve with state-of-the-art speeding up methods for SMs such as the GLV-GLS method and protection against several SCAs on the WSN hardware platforms, we expect that the results will be faster than those of the standard elliptic curves on these platforms. Thus, it is required to investigate the feasibility and superiority of the performance of SMs over TED curves on these platforms.

Second, one of the challenges is to achieve secure implementation of public-key cryptographic primitives against SCAs with as little extra cost as possible. SCAs are easy-to-implement whilst powerful attacks against cryptographic implementations, and their targets range from primitives, protocols, modules, and devices to even systems. These attacks pose a serious threat to the security of cryptographic modules. It is necessary to propose principles for selecting a set of countermeasures against SCAs reflecting the constraints of the WSN hardware platforms and applications. The works [98], [99] provided three principles, Complete, Specific and Additive, to choose countermeasures against SCAs on ECC. However, they didn't cover a range of applications and platforms with different requirements and constraints. Also, it has never proposed countermeasures of MQ-signature schemes such as UOV and Rainbow against timing attacks and power analysis. These countermeasures and the principles for selecting a set of countermeasures against SCAs for the WSN platforms is useful in designing secure cryptographic primitives and security protocols for given WSN applications.

Third, it needs to the study of optimizations for Type 3 pairings such as the Optimal Ate pairings on BN curves. Compared to Type 1 pairings which turned out insecure on the finite fields with small characteristics, there are relatively few works of efficient implementations on WSN hardware platforms. In fact, pairing-free cryptographic primitives is better than pairing-based ones in terms of efficiency. However, in the case of ID-based non-interactive key agreement protocol, it is impossible to replace the protocol with a pairing-free one. At present, time required for an Optimal Ate pairing on BN curve at an 80-bit security level on MSP430 is 3.791 s, which is about 30 times (resp. 3 times) slower than that for a scalar multiplication on the Koblitz curve on the binary field (resp. Type 1 pairing, the η_T pairing on the binary field). Thus, it should be studied various optimizations of Type 3 pairings for its practical use.

Finally, it is important to study for the security and efficiency of MQ-PKCs. To guarantee the security of communications in the post-quantum era, we need alternatives to the classical ones. Post-quantum cryptography are L-PKC (e.g., NTRU), code-based cryptography (e.g., McEliece), hash-based cryptography (e.g., Merkle's hash-trees signatures), and MQ-PKCs. All of these systems are believed to resist classical computers and quantum computers. Due to simplicity of operations (matrices and vectors) and small fields avoid multiple-precision arithmetic, MQ-PKCs are very fast, especially for signatures, compared to conventional schemes and require only very moderate resources. This makes MQ-PKCs excellent candidates for use in resource constraint devices. Still there are several issues that pose obstacles on the way of using MQ-PKCs:

- **Large Key Size**. The major obstacle of MQ-PKCs is large-size public and secret keys. From Table XVII, the size of secret key (resp, public key) for Rainbow(18, 13, 14) is 153 times (resp., 218 times) larger than that of RSA-1024 at the 80-bit security level. Many works reducing the sizes of secret key and public key have been published. In [205], [224]–[226], Petzoldt *et al.* showed different possibilities to decrease the public key sizes of UOV and Rainbow. However, their methods damage the randomness of the quadratic part of the public keys by using circulant matrices as some quadratic parts of the public key. Thus, to exploit MQ-PKCs practically, it needs to find a new way to reduce the key sizes dramatically without damaging the randomness of the quadratic part of the public key.
- **Lack of Advanced MQ-PKCs**. Another problem is a lack of "advanced" MQ-PKCs like key exchange protocols and signature schemes with special properties. In particular, to design ID-based MQ-signature is an another approach for eliminating the transmission of public-key certificates. If we use MQ-IBS scheme which enable a user's public key to be easily derived from the user's identity information, then it is obvious to reduce the public key size, as it uses short identities instead of large public keys.

Consequently, it should be continued to systematically study the security and efficiency of MQ-PKCs, so that we can identify the highest-security systems that fit the speed and space requirements imposed by cryptographic applications and users. We have no doubt that if these problems are solved, the MQ-PKCs are the best promising alternative.

VI. Conclusion

In this survey, we have given a deeper understanding of state-of-the-art public-key cryptographic approaches in WSNs, and have discussed their main directions. We have presented theoretical backgrounds and security of PKCs contained in the three classes: classical PKCs including RSA and ECC, PKCs based on hard problems over lattices (L-PKC) and PKCs based on multivariate quadratic equations (MQ-PKC). Also, we have investigated their software implementation results on low-cost microcontrollers choosing popular IEEE 802.15.4-compliant WSN hardware platforms in terms of execution time, energy consumption, communication overhead and resource occupations. Additionally, a digest of SCAs on PKCs and their countermeasures have provided. Finally, we have discussed some open research issues that can be further pursued. This survey provides valuable insights on public-key cryptographic primitives on WSN platforms, and tradeoffs to make the decision-making process more effective and direct for designing the security schemes.

References

[1] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," *Cryptogr. Hardware Embedded Syst.*, vol. 3156, pp. 119–132, 2004.

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

[3] G. Gaubatz, J. P. Kaps, and B. Sunar, "Public key cryptography in sensor networks-revisited," in *Proc. 1st ESAS*, 2004, pp. 1–17.

[4] M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep., 1979.

[5] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 32, no. 2, pp. 130–126, Feb. 1978.

[6] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, pp. 203–209, 1987.

[7] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997.

[8] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 8, no. 2, pp. 2–23, 2nd Quart. 2006.

[9] J. Sen, "A survey on wireless sensor network security," *Int. J. Commun. Netw. Inf. Security*, vol. 1, no. 2, pp. 55–78, 2009.

[10] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: A survey," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 2, pp. 52–73, 2nd Quart. 2009.

[11] A. Perrig, J. Stakovic, and D. Wagner, "Security in wireless sensor networks," *Commun. ACM*, vol. 47, no. 6, pp. 53–57, 2004.

[12] J. P. Walters and Z. Liang, " *Wireless Sensor Network Security: A Survey, Security in Distributed, Grid, and Pervasive Computing*. New York, NY, USA: Auerbach, Ch. 17.

[13] M. K. Jain, "Wireless sensor networks: Security issues and challenges," *Int. J. Comput. Inf. Technol.*, vol. 2, no. 1, pp. 62–67, 2011.

[14] D. Boyle and T. Newe, "Security protocols for use with wireless sensor networks: A survey of security architectures," *Proc. IEEE ICWMC*, 2007, p. 54.

[15] C. Dhivya Devi and B. Santhi, "Study on security protocols in wireless sensor networks," *Int. J. Eng. Technol.*, vol. 5, no. 1, pp. 200–207, 2013.

[16] A. Sangwan, D. Sindhu and K. Singh, "A review of various security protocols in wireless sensor network," *Int. J. Comp. Tech. Appl.*, vol. 2, no. 4, pp. 790–797, 2011.

[17] K. Daniluk and E. Niewiadomska-Szynkiewicz, "A survey of energy efficient security architectures and protocols for wireless sensor networks," *J. Telecommun. Inf. Technol.*, vol. 3, pp. 64–72, Mar. 2012.

[18] R. Gupta and H. Dhadhal, "Secure multipath routing in wireless sensor networks," *Int. J. Electron. Comput. Sci. Eng.*, vol. 1, no. 2, pp. 585–589, 2012.

[19] A. Kellner, O. Alfandi, and D. Hogrefe, "A survey on measures for secure routing in wireless sensor networks," *Int. J. Sens. Netw. Data Commun.*, vol. 1, pp. 1–17, 2012.

[20] P. Mohanty, S. Panifrahi, N. Sarma, and S. S. Satapathy, "Security issues in wireless sensor network data gathering protocols: A survey," *J. Theoretical Appl. Inf. Technol.*, vol. 13, no. 1 pp. 14–27, 2010.

[21] J. Jose, J. Jose, and F. Jose, "A survey on secure data aggregation protocols in wireless sensor networks," *Int. J. Comput. Appl.*, vol. 55, no. 18, pp. 17–21, Oct. 2012.

[22] H. Alzaid, E. Foo, and J. G. Nieto, "Secure data aggregation in wireless sensor network: A survey," in *Proc. AISC*, 2008, pp. 95–105.

[23] R. A. Parate, P. Patil, and G. Agarwal, "Survey on location privacy preserving schemes in wireless sensor network," *Int. J. Eng. Res. Technol.*, vol. 1, no. 9, pp. 1–5, 2012.

[24] Y. Zeng, J. Cao, J. Hong, S. Zhang, and L. Xie, "Secure localization and location verification in wireless sensor networks: A survey," *J. Supercomput.*, vol. 64, no. 3, pp. 685–701, Jun. 2013.

[25] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Crypto*, 1999, pp. 388–397.

[26] "Data encryption standard," Federal Inf. Process. Std. Publication 46, Nat. Bureau Std., Gaithersburg, MD, USA, Jan. 1977.

[27] Advanced Encryption Standard (AES), *Federal Information Processing Standards Publication 197*, United States National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, 2002.

[28] K. Okeya and T. Iwata, "Side channel attacks on message authentication codes," *Security Privacy Ad-hoc Sens. Netw.*, vol. 3813, pp. 478–488, 2006.

[29] K. Okeya, "Side channel attacks against HMACs based on block-cipher based hash functions," *Inf. Security Privacy*, vol. 4058, pp. 432–443, 2006.

[30] C. Karlof, N. Saatry, and D. Wagner, "Tinysec: A link layer security architecture for wireless sensor networks," in *Proc. 2nd ACM Conf. Embedded Netw. Sens. Syst.*, 2004, pp. 162–175.

[31] T. Moabalobelo1, F. Nelwamondo and H. D. Tsague, "A survey on the cryptanalysis of wireless sensor networks using side-channel analysis. [Online]. Available: http://researchspace.csir.co.za/dspace/bitstream/10204/6107/1/Moabal-obelo_2012.pdf

[32] C. Hartung, J. Balasalle, and R. Han, "Node compromise in WSN: The need for secure systems," Colorado Univ., Boulder, CO, USA, Tech. Rep. CU-CS-990-05, 2005.

[33] A. Becher, Z. Benenson, M. Dornseif, "Tampering with motes: Real-world physical attacks on wireless sensor networks," in *Proc. SPC*, 2006, pp. 104–118.

[34] G. Meulenaer and F. Standaer, "Stealthy compromise of wireless sensor nodes with power analysis attacks," in *Proc. MOBILIGHT*, 2010, pp. 229–242.

[35] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks," in *Proc. IPSN*, 2008, pp. 245–256.

[36] Y. B. Zhou and D. G. Feng, "Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing," Cryptology ePrint Archive, Rep. 2005/388, 2005. [Online]. Available: http://eprint.iacr.org

[37] S. Chari, C. Jutla, J. Rao, and P. Rohatgi, "Towards sound approaches to counteract power-analysis attacks," in *Proc. Crypto*, 1999, pp. 398–412.

[38] L. Goubin and J. Patarin, "DES and differential power analysis," in *Proc. CHES*, 1999, pp. 158–172.

[39] P. Kocher, J. Jaffe, and B. Jun, "Using unpredictable information to minimize leakage from smartcards and other cryptosystems," USA Patent, Int. Publication Number WO 99/63696, Jun. 3, 1999.

[40] M. Joye, A. K. Lenstra, and J. J. Quisquater, "Chinese remaindering cryptosystems in the presence of faults," *J. Cryptol.*, vol. 12, no. 4, pp. 241–245, Sep. 1999.

[41] S. Fruhauf and L. Sourge, "Safety device against the unauthorized detection of protected data," U.S. Patent US 4 932 053 A, Nov. 10, 1988.

[42] D. May, L. H. Muller, and N. P. Smart, "Random register renaming to foil DPA," *Cryptogr. Hardware Embedded Syst.*, vol. 2162, pp. 28–38, 2001.

[43] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 1997.

[44] L. B. Oliveira, A. Kansal, B. Priyantha, M. Goraczko, and F. Zhao, "Secure-TWS: Authenticating node to multi-user communication in shared sensor networks," *Comput. J.*, vol. 55, no. 4, pp. 384–396, 2012.

[45] P. Czypek, S. Heyse, and E. Thomae, "Efficient implementations of MQPKS on constrained devices," *Cryptogr. Hardware Embedded Syst.*, vol. 7428, pp. 374–389, 2012.

[46] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.

[47] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, Apr. 1984.

[48] D. Dolev, D. Dwork, and M. Naor, "Non-malleable cryptography," *SIAM J. Comput.*, vol. 30, pp. 391–437, 2000.

[49] M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen Ciphertext attacks," in *Proc. 22nd Annu. ACM Symp. Theory Comput.*, 1990, pp. 427–437.

[50] C. Rackoff and D. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen Ciphertext attack," in *Proc. Crypto*, 1991, pp. 433–444.

[51] D. H. Phan and D. Pointcheval, "On the security notions for public-key encryption schemes," *Security Commun. Netw.*, vol. 3352, pp. 33–47, 2004.

[52] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes," in *Proc. Crypto*, 1998, pp. 26–45.

[53] M. Bellare and P. Rogaway, "Optimal asymmetric encryption: How to encrypt with RSA," in *Proc. Eurocrypt*, 1994, pp. 1–19.

[54] D. Bleichenbacher, "Chosen Ciphertext attacks against protocols based on the RSA encryption standard PKCS #1," in *Proc. Crypto*, 1998, pp. 1–12.

[55] "PKCS #1: RSA Encryption Standard," RSA Data Security Inc., Redwood City, CA, USA, Version 1.5, Nov. 1993.

[56] "PKCS #1 RSA Cryptography Standard," RSA Data Security Inc., Redwood City, CA, USA, Version 2.0, Oct. 1998.

[57] J. Manger, "A chosen Ciphertext attack on RSA Optimal Asymmetric Encryption Padding (OAEP) as standardized in PKCS # 1 v2.0," in *Proc. Crypto*, 2001, pp. 230–238.

[58] "PKCS #1 RSA Cryptography Standard," RSA Data Security Inc., Redwood City, CA, USA, Version 2.1, Jun. 2002.

[59] "PKCS #1 RSA Cryptography Standard," RSA Data Security Inc., Redwood City, CA, USA, Version 2.2, Oct. 2012.

[60] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.

[61] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen Ciphertext attack," in *Proc. Crypto*, 1998, pp. 13–25.

[62] M. Bellare and P. Rogaway, "The exact security of digital signatures-how to sign with RSA and rabin," in *Proc. Eurocrypt*, 1996, pp. 399–416.

[63] Public-Key Infrastructure (X.509), (pkix). [Online]. Available: http://www.ietf.org/proceedings/59/211.htm

[64] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Crypto*, 1984, pp. 47–53.

[65] Crossbow Technology, Inc. [Online]. Available: http://www.xbow.com/

[66] Sentilla Corporation. TMote Sky Datasheet. [Online]. Available: http://www.sentilla.com/pdf/eol/tmote-sky-datasheet.pdf

[67] V. S. Miller, "Use of elliptic curves in cryptography," in *Proc. CRYPTO*, 1986, pp. 417–426.

[68] D. Coppersmith, "Fast evaluation of logarithms in fields of characteristic two," *IEEE Trans. Inf. Theory*, vol. 30, no. 4, pp. 587–594, Jul. 1984.

[69] "Public key cryptography for the financial services industry: The elliptic curve digital signature algorithm (ECDSA)," American Bankers Assoc., Washington, DC, USA, ANSI X9.62-1998, 1999.

[70] "Digital Signature standard," National Institute of Standards and Technology, Gaithersburg, MD, USA, FIPS Publication 186-2, Feb. 2000.

[71] SEC 1: Elliptic Curve Cryptography, Certicom Research, Version 2.0, May 21, 2009.

[72] N. Koblitz, "CM-curves with good crypto-graphic properties," in *Proc. Crypto*, 1991, vol. 576, pp. 279–287.

[73] J. A. Solinas, "Efficient arithmetic on Koblitz curves," *Des., Codes Cryptogr.*, vol. 19, no. 2/3, pp. 195–249, Mar. 2000.

[74] R. Gallant, R. Lambert, and S. Vanstone, "Faster point multiplication on elliptic curves with efficient endomorphisms," *Proc. Crypto*, 2001, pp. 190–200.

[75] D. Chu, J. Großschädl, and Z. Liu, "Twisted Edwards-form elliptic curve cryptography for 8-bit AVR-based sensor nodes, Proc. AsiaPKC, Rep. 2012/730, 2012.

[76] Sec 2: Recommended Elliptic Curve Domain Parameters, Certicom Research, Jan. 27, 2010, Version 2.0.

[77] FIPS 186-3: Digital signature standard (DSS), Nat. Inst. Std. Technol., Gaithersburg, MD, USA, 2009. [Online]. Available: http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf

[78] C. P. L. Gouvêa and J. López, "Efficient software implementation of public-key cryptography on sensor networks using the MSP430X microcontroller," *J. Cryptogr. Eng.*, vol. 2, no. 1, pp. 19–29, 2012.

[79] F. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," *Public Key Cryptogr.*, vol. 2947, pp. 277–290, 2004.

[80] P. S. L. M. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," *Sel. Areas Cryptogr.*, vol. 3897, 2006, pp. 319–331.

[81] F. Vercauteren, "Optimal pairings," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 455–461, Jan. 2010.

[82] P. S. L. M. Barreto, S. Galbraith, C. Ó' hÉigeartaigh, and M. Scott, "Efficient pairing computation on Supersingular Abelian varieties," *Des. Codes Cryptogr.*, vol. 42, no. 3, pp. 239–271, 2007.

[83] T. Iijima, K. Matsuo, J. Chao, and S. Tsujii, "Construction of Frobenius maps of twist elliptic curves and its application to elliptic scalar multiplication," in *Proc. Symp. Cryptogr. Inf. Security*, 2002, pp. 699–702.

[84] S. D. Galbraith, X. Lin, and M. Scott, "Endomorphisms for faster elliptic curve cryptography on a large class of curves," in *Proc. Eurocrypt*, 2009, pp. 518–535.

[85] P. Longa and F. Sica, "Four-dimensional Gallant-Lambert-Vanstone scalar multiplication," in *Proc. Asiacrypt*, 2012, pp. 718–739.

[86] S. Morozov, C. Tergino, and P. Schaumont, "System integration of elliptic curve cryptography on an OMAP platform," in *Proc. SASP*, 2011, pp. 52–57.

[87] D. Brown, "Sec 2: Recommended elliptic curve domain parameters," Certicom, Mississauga, ON, USA, Certicom Res., 2010.

[88] D. Câmara, C. P. L. Gouvêa, J. López, and R. Dahab, "Fast software polynomial multiplication on ARM processors using the NEON engine," *Security Eng. Intell. Informat.*, vol. 8128, pp. 137–154, 2013.

[89] A. Faz-Hernandez, P. Longa, and A. H. Sanchez, "Efficient and secure algorithms for GLV-based scalar multiplication and their implementation on GLV-GLS curves." [Online]. Available: http://eprint.iacr.org/2013/158

[90] P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS and OTHER SYSTEMS," in *Proc. Crypto*, 1996, pp. 104–113.

[91] W. Schindler, "A timing attack against RSA with the chinese remainder theorem," in *Proc. CHES*, 2000, pp. 110–125.

[92] W. Schindler, "A combined timing and power attack," *Public Key Cryptogr.*, vol. 2274, pp. 263–279, 2002.

[93] B. Boer, K. Lemke, and G. Wicke, "A DPA attack against the modular reduction within a CRT implementation of RSA," in *Proc. CHES*, 2003, pp. 228–243.

[94] "RSAES-OAEP encryption scheme-algorithm specification and supporting documentation," RSA Lab., RSA Security Inc., Hopkinton, MA, USA.

[95] B. Kaliski, "Timing attacks on cryptosystems," RSA Lab., Hopkinton, MA, USA, RSA Lab. Bulletin No. 2, Jan. 1996. [Online]. Available: www.rsalabs.com

[96] J. J. Quisquater and C. Couvreur, "Fast decipherment algorithm for RSA public-key cryptosystem," *Electron. Lett.*, vol. 18, no. 21, pp. 905–907, 1982.

[97] P. Q. Nguyen and I. Shparlinski, "The insecurity of the elliptic curve digital signature algorithm with partially known Nonces," *Des. Codes Cryptogr.*, vol. 30, no. 2, pp. 201–217, 2003.

[98] J. Fan and I. Verbauwhede, "An updated survey on secure ECC implementations: Attacks, countermeasures and cost," *Cryptogr. Security: Theory Appl.*, vol. 6805, pp. 265–282, 2012.

[99] X. Gio, "Secure and efficient implementations of cryptographic primitives," Ph.D. dissertation, Virginia Tech, Blacksburg, VA, USA, 2012.

[100] V. Shoup, "OAEP reconsidered," *J. Cryptology*, vol. 15, pp. 223–249, 2002.

[101] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern, "RSA-OAEP is secure under the RSA assumption," *J. Cryptology*, vol. 17, no. 2, pp. 260–274, Mar. 2001.

[102] V. Klíma and T. Rosa, "Further results and considerations on side channel attacks on RSA," in *Proc. CHES*, 2002, pp. 244–259.

[103] V. Klíma, O. Pokorný, and T. Rosa, "Attacking RSA-based sessions in SSL/TLS," in *Proc. CHES*, 2003, pp. 426–440.

[104] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.

[105] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *J. Cryptology*, vol. 17, pp. 297–319, 2004.

[106] A. Joux, "A one round protocol for tripartite Diffie-Hellman," in *Proc. 4th ANTS*, 2000, pp. 385–394.

[107] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairings," in *Proc. Symp. Cryptogr. Inf. Security*, Okinawa, Japan, 2000, pp. 26–28.

[108] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short Ciphertexts and private keys," in *Proc. Crypto*, 2005, pp. 258–275.

[109] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search," in *Proc. Eurocrypt*, 2004, pp. 506–522.

[110] National Institute of Standards and Technology: "Recommendation for Key Management," 2007. [Online]. Available: http://www.itl.nist.gov

[111] V. S. Miller, "The weil pairing, and its efficient calculation," *J. Cryptology*, vol. 17, pp. 235–261, 2004.

[112] L. B. Oliveira *et al.*, "TinyTate: Computing the tate pairing in resource-constrained nodes," in *Proc. NCA*, 2007, pp. 318–323.

[113] P. Szczechowiak, L. B. Oliveira, M. Scott, M. Collier, and R. Dahab, "NanoECC: Testing the limits of elliptic curve cryptography in sensor networks," in *Proc. EWSN*, 2008, pp. 305–320.

[114] T. Ishiguro, M. Shirase, and T. Takagi, "Efficient implementation of pairings on sensor nodes," National Institute of Standards and Technology (NIST), Boulder, CO, USA, Identity Based Encryption Workshop, 2008. [Online]. Available: http://csrc.nist.gov/groups/ST/IBE/documents/June08/Takagi.pdf

[115] P. Szczechowiak, A. Kargl, M. Scott, and M. Collier, "On the application of pairing-based cryptography to wireless sensor networks," in *Proc. WISE*, 2009, pp. 1–12.

[116] L. B. Oliveira *et al.*, "TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks," *Comput. Commun.*, vol. 34, no. 3, pp. 485–493, Mar. 2011.

[117] P. Szczechowiak and M. Collier, "Practical identity-based key agreement for secure communication in sensor networks," in *Proc. 18th ICCCN*, 2009, pp. 1–6.

[118] "MIRACL Cryptographic Library: Multiprecision Integer and Rational Arithmetic C/C++ Library." [Online]. Available: https://certivox.org/display/EXT/MIRACL

[119] G. Adj, A. Menezes, T. Oliveira, and F. Rodriguez-Henriquez, "Weakness of $F^{36\cdot 509}$ for discrete logarithm cryptography," *Proc. Pairing*, 2014, pp. 19–43.

[120] T. Hayashi, T. Shimoyama, N. Shinohara, and T. Takagi, "Breaking pairing-based cryptosystems using η_T pairing over $GF(397)$," *Adv. Cryptology*, vol. 7658, pp. 43–60, 2012.

[121] A. Joux, "Discrete logarithms in $GF(2^{6168})[= GF((2^{257})^{24})]$. NMBRTHRY list, May 21, 2013. [Online]. Available: https://listserv.nodak.edu/cgi-bin/wa.exe?A2=NMBRTHRY;49bb494e.1305

[122] S. Galbraith, New discrete logarithm records, and the death of Type 1 pairings Posted on Feb. 1, 2014 by ellipticnews.

[123] S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for cryptographers," *Discrete Appl. Math.*, vol. 156, no. 16, pp. 3113–3121, Sep. 2008.

[124] D. Page and F. Vercauteren, "Fault and side-channel attacks on pairing based cryptography," Cryptology ePrint Archive, Rep. 2004/283, 2005.

[125] M. Scott, "Computing the tate pairing," in *Proc. CT-RSA*, 2005, pp. 293–304.

[126] C. Whelan and M. Scott, "Side channel analysis of practical pairing implementations: Which path is more secure?" in *Proc. Vieycrypt*, 2006, pp. 99–114.

[127] T. H. Kim, T. Takagi, D.-G. Han, H. W. Kim, and J. Lim, "Power analysis attacks and countermeasures on η_T pairing over binary fields," *Electron. Telecommun. Res. Inst. J.*, vol. 30, no. 1, pp. 68–80, 2008.

[128] M. Shirase, T. Takagi, and E. Okamoto, "An efficient countermeasure against side channel attacks for pairing computation," *Inf. Security Practice Exp.*, vol. 4991, pp. 290–303, 2008.

[129] S. C. Seo, D.-G. Han, and S. Hong, "An efficient DPA countermeasure for the EtaT pairing algorithm over GF(2^n) based on random value addition," *Electron. Telecommun. Res. Inst. J.*, vol. 33, no. 5, pp. 780–790, 2011.

[130] P. Nguyen, "Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from Crypto'97," in *Proc. Crypto*, 1999, pp. 288–304.

[131] P. Nguyen and J. Stern, "Lattice reduction in cryptology: An update," *Algorithmic Number Theory*, vol. 1838, pp. 85–112, 2000.

[132] A. Shamir, "A polynomial-time algorithm for breaking the basic Merkel-Hellman cryptosystem," in *Proc. 23rd IEEE Symp. Found. Comput. Sci.*, 1982, pp. 145–152.

[133] M. Ajtai and C. Dwork, "A public-key cryptosystem with worst case/average case equivalence," in *Proc. 29th ACM Symp. Theory Comput.*, 1997, pp. 284–293.

[134] O. Goldreich, S. Goldwasser, and S. Halevi, "Public-key cryptography from lattice reduction problems," in *Proc. Crypto*, 1997, pp. 112–131.

[135] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," *Algorithmic Number Theory*, vol. 1423, pp. 267–288, 1998.

[136] M. Ajtai, "The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract)," in *Proc. STOC*, 1998, pp. 10–19.

[137] R. Kannan, "Improved algorithms for integer programming and related lattice problems," in *Proc. STOC*, 1983, pp. 193–206.

[138] I. Dinur, G. Kindler, and S. Safra, "Approximating-cvp to within almost-polynomial factors is NP-hard," in *Proc. FOCS*, 1998, pp. 99–112.

[139] D. Micciancio, "The shortest vector in a lattice is hard to approximate to within some constant," in *Proc. FOCS*, 1998, pp. 92–103.

[140] R. M. Karp, "Reducibility among combinatorial problems," *Complexity Comput. Comput.*, pp. 85–103, 1972.

[141] K.-P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Program.*, vol. 66, no. 1–3, pp. 181–199, Aug. 1994.

[142] M. Ajtai, R. Kumar, and D. Sivakumar, "A sieve algorithm for the shortest lattice vector problem," in *Proc. STOC*, 2001, pp. 601–610.

[143] P. Q. Nguyen and T. Vidick, "Sieve algorithms for the shortest vector problem are practical," *J. Math. Cryptology*, vol. 2, no. 2, pp. 181–207, Jul. 2008.

[144] D. Micciancio and P. Voulgaris, "Faster exponential time algorithms for the shortest vector problem," in *Proc. SODA*, 2010, pp. 1468–1480.

[145] X. Wang, M. Liu, C. Tian, and J. Bi, "Improved Nguyen-Vidick heuristic sieve algorithm for shortest vector problem," in *Proc. ASIACCS*, 2011, pp. 1–9.

[146] D. Micciancio and P. Voulgaris, "A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations," in *Proc. STOC*, 2010, pp. 351–358.

[147] J. Hoffstein, N. A. H. Graham, J. Pipher, J. H. Silverman, and W. Whyte, "NTRUSIGN: Digital signatures using the NTRU lattice," in *Proc. CT-RSA*, 2003, pp. 122–140.

[148] J. Hoffstein, D. Lieman, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," Algorithmic Number Theory, vol. 1423, pp. 267–288, 1988.

[149] A. May and J. H. Silverman, "Dimension reduction methods for convolution modular lattices," *Proc. Cryptogr. Lattices Conf.*, 2001, pp. 110–125.

[150] M. Abdalla, M. Bellare, and P. Rogaway, "DHIES: An encryption scheme based on the Diffie-Hellman Problem, 2001, unpublished. [Online]. Available: http://www.cs.ucdavis.edu/rogaway/papers/dhies.pdf

[151] M. W. Schab, "Extremely low-overhead security for wireless sensor networks: Algorithms and implementation," Ph.D. dissertation, Rochester Inst. Technol., Rochester, NY, USA, [submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Engineering], 2009.

[152] P. Q. Nguyen and O. Regev, "Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures," *J. Cryptology*, vol. 22, no. 2, pp. 139–160, 2009.

[153] Consortium for Efficient Embedded Security, "Efficient embedded security standards #1: Implementation aspects of NTRUencrypt and NTRUsign, Version 2.0," 2003. [Online]. Available: http://grouper.ieee.org/groups/1363/lattPK/index.html

[154] *Public-Key Cryptographic Techniques Based on Hard Problems over Lattices*, Std. IEEE P1363.1, 2003. [Online]. Available: http://grouper.ieee.org/groups/1363/lattPK/index.html

[155] Y. Hu, B. Wang, and W. He, "NTRUSIGN with a new perturbation," *IEEE Trans. Inf. Theory*, vol. 54, no. 7, pp. 3216–3221, 2008.

[156] L. Ducas and P. Q. Nguyen, "Learning a Zonotope and more: Cryptanalysis of NTRUSIGN countermeasures," in *Proc. Asiacrypt*, 2012, pp. 433–450.

[157] J. Hoffstein, N. Howgrave-Graham, J. Pipher, and W. Whyte, "Practical lattice-based cryptography: NTRUEncrypt and NTRUSIGN," *LLL Algorithm*, pp. 349–390, 2010.

[158] *Standard Specifications for Public Key Cryptography, Institute of Electrical and Electronics Engineers*, IEEE Std. 1363, 2000.

[159] N. Howgrave-Graham, J. H. Silverman, A. Singer, and W. Whyte, "NAEP: Provable security in the presence of decryption failures," Submissions and Contributions to IEEE P1363.1. [Online]. Available: http://eprint.iacr.org/2003/172

[160] Consortium for Efficient Embedded Security, "Efficient Embedded Security Standards (EESS #1: Implementations Aspects of ETRUEncrypt and NTRUSign", Version 2, 2003. [Online]. Available: http://grouper.ieee.org/groups/1363/lattPK/submissions/EESS1v2.pdf

[161] N. Howgrave-Graham, J. H. Silverman, and W. Whyte, "Choosing parameter sets for NTRUEncrypt with NAEP and SVES-3," in *Proc. CT-RSA*, 2005, pp. 118–135.

[162] P. Nguyen and D. Pointcheval, "Analysis and improvements of NTRU encryption paddings," in *Proc. Crypto*, 2002, pp. 210–225.

[163] N. Howgrave-Graham, P. Q. Nguyen, D. Pointcheval, A. Singer, and W. Whyte, "Padding schemes and decryption failures in NTRU Encryption, 2003.

[164] J.-S. Coron, M. Joye, D. Naccache, and P. Paillier, "Universal padding schemes for RSA," in *Proc. Crypto*, 2002, pp. 226–241.

[165] *The IEEE P1363 Study Group for Future Public-Key Cryptography Standards, Draft Standard for Public-Key Cryptographic Techniques Based on Hard Problems over Lattices*, Std. IEEE P1363.1, Jan. 2007. [Online]. Available: http://grouper.ieee.org/groups/1363/lattPK/draft.html

[166] J. Buchmann and M. Döring, *Efficiency Improvement for NTRU*. Darmstadt, Germany: Tech. Univ. Darmstadt, 2007.

[167] D. Stehlé and R. Steinfeld, "Making NTRU as secure as worst-case problems over ideal lattices," in *Proc. Eurocrypt*, 2011, pp. 27–47.

[168] R. Steinfeld, S. Ling, J. Pieprzyk, C. Tartary, and H. Wang, "NTRUCCA: How to strengthen NTRUEncrypt to chosen-ciphertext security in the standard model," *Public Key Cryptography*, vol. 7293, pp. 353–371, 2012.

[169] J. H. Silverman and W. Whyte, "Timing attacks on NTRUEncrypt via variation in the number of hash calls," in *Proc. CT-RSA*, 2007, pp. 208–224.

[170] J. Hoffstein and J. H. Silverman, "Optimizations for NTRU," *Public Key Cryptogr. Comput. Number Theory*, pp. 77–88, 2001.

[171] A. Kipnis, J. Patarin, and L. Goubin, "Unbalanced oil and vinegar signature schemes," in *Proc. Eurocrypt*, 1999, pp. 206–222.

[172] N. T. Courtois, A. B. Klimov, J. Patarin, and A. Shamir, "Efficient algorithms for solving overdefined systems of multivariate polynomial equations," in *Proc. Eurocrypt*, 2000, pp. 392–407.

[173] J. C. Faugére, "A new efficient algorithm for computing Gröbner bases (F4)," *J. Pure Appl. Algebra*, vol. 139, pp. 61–88, 1999.

[174] J. C. Faugére, "A new efficient algorithm for computing Gröbner bases without reduction to zero (F5)," *Proc. ISSAC*, 2002, pp. 75–83.

[175] M. Bardet, J. C. Faugére, B. Salvy, and B. Y. Yang, "Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems," in *Proc. MEGA*, 2005, pp. 1–16.

[176] L. Bettale, J. C. Faugére, and L. Perret, "Hybrid approach for solving multivariate systems over finite fields," *J. Math. Cryptology*, vol. 3, no. 3, pp. 177–197, Jan. 2009.

[177] N. T. Courtois, L. Goubin, W. Meier, and J. D. Tacier, "Solving underdefined systems of multivariate quadratic equations," *Public Key Cryptogr.*, vol. 2274, pp. 211–227, 2002.

[178] E. Thomae and C. Wolf, "Solving underdetermined systems of multivariate quadratic equations revisited," *Public Key Cryptogr.*, vol. 7293, pp. 156–171, 2012.

[179] H. Miura1, Y. Hashimoto, and T. Takagi, "Extended algorithm for solving underdefined multivariate quadratic equations," in *Proc. PQCrypto*, 2013, pp. 118–135.

[180] L. C. Wang, Y. H. Hu, F. Lai, C. Yen Chou, and B. Y. Yang, "Tractable rational map signature," *Public Key Cryptogr.*, vol. 3386, pp. 244–257, 2005.

[181] J. Ding, C. Wolf, and B. Y. Yang, "*l*-invertible cycles for multivariate quadratic public key cryptography," *Public Key Cryptogr.*, vol. 4450, pp. 266–281, 2007.

[182] D. Gligoroski, S. Markovski, and S. J. Knapskog, "Multivariate quadratic trapdoor functions based on multivariate quadratic quasigroups," in *Proc. WSEAS*, 2008, pp. 44–49.

[183] P. A. Fouque, G. Macario-Rat, and J. Stern, "Key recovery on hidden monomial multivariate schemes," in *Proc. Euroctypt*, 2008, pp. 19–30.

[184] J. C. Faugére, A. Joux, L. Perret, and J. Treger, "Cryptanalysis of the hidden matrix cryptosystem," in *Proc. LatinCrypt*, 2010, pp. 241–254.

[185] V. Dubois, P. A. Fouque, and J. Stern, "Cryptanalysis of SFLASH with slightly modified parameters," in *Proc. Eurocrypt*, 2007, pp. 264–275.

[186] V. Dubois, P. A. Fouque, A. Shamir, and J. Stern, "Practical cryptanalysis of SFLASH," in *Proc. Crypto*, 2007, pp. 1–12.

[187] A. Joux, S. Kunz-Jacques, F. Muller, and P. M. Ricordel, "Cryptanalysis of the tractable rational map cryptosystem," *Public Key Cryptogr.*, vol. 3386, pp. 258–274, 2005.

[188] P. A. Fouque, G. Macario-Rat, L. Perret, and J. Stern, "Total break of the ′*l*-ic signature scheme," *Public Key Cryptogr.*, vol. 4939, pp. 1–17, 2008.

[189] M. Mohamed, J. Ding, J. Buchmann, and F. Werner, "Algebraic attack on the mqq public key cryptosystem," *Cryptology Netw. Security*, vol. 5888, pp. 392–401, 2009.

[190] J. Patarin, L. Goubin, and N. Courtois, "Improved algorithms for isomorphisms of polynomials," in *Proc. Eurocrypt*, 1998, pp. 184–200.

[191] C. Bouillaguet, P.-A. Fouque and A. Véber, "Graph-theoretic algorithms for the isomorphism of polynomials problem." [Online]. Available: http://eprint.iacr.org/2012/607.pdf

[192] A. Kipnis and A. Shamir, "Cryptanalysis of the oil and vinegar signature scheme," in *Proc. Crypto*, 1998, pp. 257–266.

[193] C. Bouillaguet, P. A. Fouque, and G. Macario-Rat, "Practical key-recovery for all possible parameters of SFLASH," in *Proc. Asiacrypt*, 2011, pp. 667–685.

[194] A. Kipnis and A. Shamir, "Cryptanalysis of the HFE public key cryptosystem by relinearization," in *Proc. ACrypto*, 1999, pp. 19–30.

[195] J. Patarin, "Cryptanalysis of the matsumoto and imai public key scheme of Eurocrypt'88," in *Proc. Crypto*, 1995, pp. 248–261.

[196] L. Goubin and N. T. Courtois, "Cryptanalysis of the TTM cryptosystem," in *Proc. Asiacrypt*, 2000, pp. 44–57.

[197] J. C. Faugére and A. Joux, "Algebraic Cryptanalysis of Hidden Field Equations (HFE) using Gröbner bases," in *Proc. Crypto*, 2003, pp. 44–60.

[198] N. T. Courtois, M. Daum, and P. Felke, "On the security of HFE, HFEv- and quartz," *Public Key Cryptogr.*, vol. 2567, pp. 337–350, 2002.

[199] C. Wolf, A. Braeken, and B. Preneel, "Efficient cryptanalysis of RSE(2)PKC and RSSE(2)PKC," in *Proc. SCN*, 2005, pp. 294–309.

[200] T. Matsumoto and H. Imai, "Public quadratic polynomial-tuples for efficient signature-verification and message-encryption," in *Proc. Eurocrypt*, 1988, pp. 419–453.

[201] J. Patarin, "Hidden field equations (HFE) and isomorphism of polynomials (IP): Two new families of asymmetric algorithms," in *Proc. Eurocrypt*, 1996, pp. 33–48.

[202] J. Patarin, L. Goubin, and N. T. Courtois, "Quartz, 128-bit long digital signatures," in *Proc. CT-RSA*, 2001, pp. 282–297.

[203] J. Patarin, N. Courtois, and L. Goubin, "SFLASH, a fast multivariate signature algorithm," 2003. [Online]. Available: http://eprint.iacr.org/

[204] A. Kipnis and A. Shamir, "Cryptanalysis of the HFE public key cryptosystem," in *Proc. Crypto*, 1999, pp. 19–33.

[205] A. Petzoldt, E. Thomae, S. Bulygin, and C. Wolf, "Small public keys and fast verification for multivariate quadratic public key systems," in *Proc. CHES*, 2011, pp. 475–490.

[206] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *Proc. ACNS*, 2005, pp. 164–175.

[207] N. T. Courtois, "The Minrank problem, MinRank, a new zero-knowledge Scheme based on the NP-complete problem," presented at the Rump Session of Crypto, 2000. [Online]. Available: http://www.minrank.org

[208] B. Y. Yang and J. M. Chen, "Building secure tame-like multivariate public-key cryptosystems: The new TTS," in *Proc. ACISP*, 2005, pp. 518–531.

[209] B. Y. Yang, J. M. Chen, and Y. H. Chen, "TTS: High-speed signatures on a low-cost smart card," in *Proc. CHES*, 2004, pp. 371–385.

[210] B. Driessen, A. Poschmann, and C. Paar, "Comparison of innovative signature algorithms for WSNs," in *Proc. WiSec*, 2008, pp. 30–35.

[211] M. Bellare, C. Namprempre, and G. Neven, "Security proofs for identity-based identification and signature schemes," in *Proc. Eurocrypt*, 2004, pp. 268–286.

[212] T. B. R. Steinwandt and W. Geiselmann, "A theoretical DPA based cryptanalysis of the NESSIE candidates FLASH and SFLASH," in *Proc. ISC*, 2001, pp. 280–293.

[213] K. Okeya, T. Takagi, and C. Vuillaume, "On the importance of protecting △ in SFLASH against side channel attacks," *Inst. Electron., Inf. Commun. Eng. Trans.*, vol. 88, pp. 123–131, 2005.

[214] Y. Hashimoto, T. Takagi, and K. Sakurai, "General fault attacks on multivariate public key cryptosystems," in *Proc. PQCrypto*, 2011, pp. 1–18.

[215] M. L. Akkar, N. Courtois, L. Goubin, and R. Duteuil, "A fast and secure implementation of Sflash," *Public Key Cryptogr.*, vol. 2567, pp. 267–278, 2003.

[216] S. Tsujii, K. Kurosawa, T. Itoh, A. Fujioka, and T. Matsumoto, "A public-key cryptosystem based on the difficulty of solving a system of non-linear equations," *IEICE Trans. Inf. Syst.*, vol. J69-D, pp. 1963–1970, 1986.

[217] A. Shamir, "Efficient signature schemes based on birational permutations," in *Proc. Crypto*, 1994, pp. 1–12.

[218] T. Moh, "A public key system with signature and master key functions," *Commun. Algebra*, vol. 27, pp. 2207–2222, 1999.

[219] E. Barker and A. Roginsky, "Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths," National Inst. Std. Technol., Gaithersburg, MD, USA, NIST Special Pub. 800-131A, Comput. Security Div. Inf. Technol. Lab., 2011.

[220] A. Lenstra and E. Verheul, "Selecting cryptographic key sizes," *J. Cryptology*, vol. 14, no. 4, pp. 255–293, 2001.

[221] A. K. Lenstra, *Key Lengths, The Handbook of Information Security*, pp. 617–635, 2004.

[222] "BlueKrypt cryptographic key length recommendation." [Online]. Available: http://www.keylength.com/

[223] "Approved algorithms for block Ciphers," National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA.

[224] A. Petzoldt, S. Bulygin, and J. Buchmann, "A multivariate signature scheme with a partially cyclic public key," in *Proc. SCC*, 2010, pp. 229–235.

[225] A. Petzoldt, S. Bulygin, and J. Buchmann, "CyclicRainbow-A multivariate signature scheme with a partially cyclic public key," in *Proc. Indocrypt*, 2010, pp. 33–48.

[226] A. Petzoldt, S. Bulygin, and J. Buchmann, "Linear recurring sequences for the UOV key generation," *Public Key Cryptogr.*, vol. 6571, pp. 335–350, 2011.

Kyung-Ah Shim received the M.S. and Ph.D. degrees in mathematics from Ewha Womans University, Seoul, Korea, in 1994 and 1999, respectively. From 2000 to 2004, she was a Senior Researcher with the Korea Information Security Agency. From 2004 to 2008, she was a Research Professor with the Department of Mathematics, Ewha Womans University. In 2008, she joined the National Institute for Mathematical Sciences, Daejeon, Korea, as a Senior Researcher. Her research interests include cryptography and information security.