


Precomputation Methods for UOV Signature on Energy-Harvesting Sensors

BO LV¹, ZHINIANG PENG², AND SHAOHUA TANG¹ 

¹School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China

²Core Security Department, Qihoo 360, Beijing 100016, China

Corresponding author: Shaohua Tang (shtang@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61632013, in part by the Guangdong Provincial Natural Science Foundation under Grant 2014A030308006, and in part by the Guangdong Provincial Project of Science and Technology under Grant 2016B090920081.

ABSTRACT Wireless sensor networks (WSNs) are increasingly gaining impact on our daily lives. They are finding a wide range of applications in various domains, such as health-care, environmental monitoring, and so on. In future, WSNs are expected to be integrated into the Internet of Things (IoT). The integrity of the sensed data is of primary importance in WSN and IoT applications. However, WSN platforms always have very limited resources in terms of battery power, computing, and memory. Therefore, it is a key challenge to design an energy-friendly, lightweight digital signature algorithm for WSN platforms. In this paper, we present precomputation methods for unbalanced oil-vinegar (UOV) signature scheme by exploiting the energy-harvesting capabilities of WSN to enhance the performance of UOV signature. In addition, we combine a circulant method with precomputation in the UOV signature to further reduce the energy cost. Meanwhile, the circulant method reduces the size and the memory overhead of the precomputation tuple. This increases the availability of precomputation and greatly enhances the availability of the overflow energy. By integrating the above-mentioned optimization methods, the cost of UOV signature in a WSN node can be reduced by 93%. This greatly enhances the performance of UOV signature, making it feasible for the practical deployment on resource-constrained wireless sensor platforms.

INDEX TERMS Energy harvesting, UOV, precomputation, IoT and WSNs security.

I. INTRODUCTION

Wireless sensor networks (WSNs) are spatially distributed autonomous devices using sensors to monitor physical or environmental conditions [1], such as temperature, sound, pressure, etc. It has been widely used in commercial and industrial applications [2]–[7], [8] due to its low cost and pervasive capability. In some WSN applications, the integrity of the sensed data is of primary importance. For example, in the social/health care systems, sensitive information about elderly people or patients emergent conditions are transmitted from sensors to base stations. Altered or modified data could induce serious consequences for people in critical condition.

However, due to the property of wireless communication, WSNs are more vulnerable to various attacks than wired networks [9]. Moreover, WSN platforms always have very limited resources in terms of battery power, computing and memory. Traditional digital signature schemes, such as ECDSA and RSA, are not suitable for WSN. Previous works [10]–[13] have presented ECDSA and RSA implementations for WSN, but the performance and energy

consumption of generating signatures are still unacceptable. It is a key challenge to design an energy-friendly, lightweight digital signature algorithm for WSN platforms.

Energy Harvesting Technologies [14] provide the possibility of reducing energy consumption and improving performance of digital signature in WSN platforms [15]. Wireless sensor nodes with energy harvesting capabilities (EH-WSN) are motes that are able to extract energy from surrounding environment and convert it into usable electrical power. The harvested energy varies depending on the environment of the motes. An energy peak occurs when the harvested energy exceeds the maximum capacity of the capacitor or a given charging level threshold, and a node is harvesting power at a rate that exceeds its current power consumption. The overflowed energy would be wasted if not immediately used. Precomputation techniques can be used to take advantage of these overflowed energy. With precomputation techniques, one can divide a signature scheme into online phase and offline phase. The offline phase is independent of the message to be signed, and the online phase depends on the message

to be signed. We can precompute the offline phase when an energy peak occurs and save the intermediate value in memory. When we get message to be signed in the online phase, we can get the intermediate value from memory and compute the signature. This will minimize the run-time energy and latency for signature generation.

Bianchi *et al.* [16] show that complex security mechanisms may become significantly less demanding when it is implemented in order to take advantage of energy harvesting opportunities. They propose AGREE, a framework that exploits energy harvesting opportunities to precompute Ciphertext Policy Attribute Based Encryption (CP-ABE), so as to minimize the run-time energy and latency of CP-ABE. Inspired by this work, the method of combining energy harvesting capabilities with precomputation methods to generate signature has been applied to many signature schemes, such as ECDSA [17], [18], hash based signature schemes [19], and lattice based signature scheme [20].

Multivariate Public Key Cryptography (MPKC) is one of the most promising candidates for Post-Quantum Cryptography which attracts the attention of scholars [21], [22]. MPKC appears to be a good solution to WSN platforms because it has reasonable performance and moderate resources requirement [23]. The Unbalanced Oil and Vinegar (UOV) scheme is one of them. It remains secure for two decades and none of the existing attacks can cause severe security threats to it. Many previous works implemented UOV in low-resource embedded systems [24]–[26]. However, as mentioned in [20], the method of combining energy harvesting capabilities with precomputation methods to generate signature seems not suitable for MPKC.

In this paper, we present precomputation methods for UOV signature on energy-harvesting platforms. Our contributions are the followings:

- 1) We present a precomputation method for UOV signature scheme by exploiting the energy harvesting capabilities of WSN to enhance the performance of UOV signature. It can reduce the energy consumption by 44%, making it feasible for the practical deployment on resource-constrained wireless sensor platforms.
- 2) We combine circulant method with precomputation in UOV signature which reduce the energy consumption of the basic precomputation for UOV by 87%. Meanwhile, circulant method reduces size and memory overhead of the precomputation tuple. Therefore, the number of precomputation tuples increases, and the availability of precomputation increases. This also greatly enhances the availability of overflow energy.
- 3) By integrating the above optimization methods, the cost of UOV signature in a WSN node can be reduced by 93%. Our experimental results show that the energy harvesting technique does not leave MPKC behind while improving other signature schemes for WSN platforms.

II. BACKGROUND

A. PRECOMPUTATION IN EH-WSN PLATFORMS

WSN platforms often have long idle periods before they are interrupted by an immediate request. Therefore, their nature allows to partition signature generation into offline and online phases. The offline phase includes the workload that can be handled at idle periods before a request comes. It corresponds to all computations that can be completed without knowledge of the message to be signed. The online phase includes the workload that depends on the message to be signed. The online phase is always very efficient and can be completed quickly. However, execution with precomputation is generally not used in WSN platforms because it requires more energy and storage than monolithic execution.

Energy harvesting capabilities provide possibility of using precomputation in EH-WSN platforms. The amount of harvested energy depends on the environment and has a very large variance in different periods. Figure 1 shows the trace of energy harvested by IXOLAR XOB17-04x3 micro solar cells with a 1F Maxwell HC series capacitor.

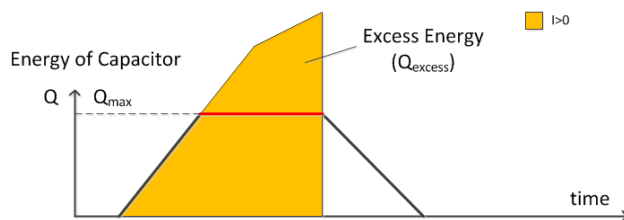


FIGURE 1. Trace of energy harvested by IXOLAR XOB17-04x3 micro solar cells with a 1F Maxwell HC series capacitor.

The red line in Figure 1 stands for the maximum capacity of the capacitor. Those overflowed energy will be wasted if we do not use it immediately. Precomputation techniques can be used to take advantage of these overflowed energy. It can reduce the total power consumption and run-time latency of signature generation.

B. UOV SIGNATURE SCHEME

UOV [27] is a modified version of the Oil and Vinegar scheme designed by J. Patarin. It poses a strong security and none of the existing attacks can cause severe security threats to it.

To figure out what UOV is, first of all, we would like to introduce the concept of Oil-Vinegar polynomial with the following form:

$$f = \sum_{i=1}^v \sum_{j=1}^v a_{ij}x'_i x'_j + \sum_{i=1}^o \sum_{j=1}^v b_{ij}\hat{x}_i x'_j + \sum_{j=1}^v \beta_j x'_j + \sum_{i=1}^o \alpha_i \hat{x}_i + c.$$

Variables are divided into two kinds in the above polynomial: Oil variables (\hat{x}_i) and Vinegar variables (x'_j). The number of Oil variables is o and the number of the Vinegar variables is v .

Central map F can be composed of o Oil-Vinegar polynomials. The map $F = (f_1, f_2, \dots, f_o)$ can be easily inverted. The invertibility of the central map comes from the fact that once random values are assigned to the Vinegar variables set, it becomes a set of linear equations of Oil variables and can be solved by Gauss Elimination.

Once the central map F is determined, the public key can be calculated as:

$$P = F \circ T,$$

where T is an affine transformation. The inverse of P can be computed as follows:

Step 1

Randomly choose $v_1, \dots, v_v \in (K^v)$.

Step 2

Substitute (x'_1, \dots, x'_v) with (v_1, \dots, v_v) , we will get o linear equations of o variables. Solve the system and obtain a solution $\hat{x}_1, \dots, \hat{x}_o$ (If the system is not regular, go back to Step 1). Let $(x_1, \dots, x_n) = (x'_1, \dots, x'_v, \hat{x}_1, \dots, \hat{x}_o)$.

Step 3

Apply inverse map of T to (x_1, \dots, x_n) .

Define $d = v - o$. When $d = 0$, it's called balanced Oil-Vinegar scheme. When $d > 0$, it's known as UOV [27].

C. CIRCULANT UOV

Like other MPKC schemes, UOV has a large key size. In [28], Circulant UOV is proposed with shorter private key and faster signature generation.

The basic idea underlying circulant method is to speed up Step 2 of UOV signing process, which is the slowest part in the signing algorithm. In Step 2 of UOV signing process, we need to solve a linear $L\mathbf{o} = \mathbf{u}$. By introducing some rotation relations into UOV's private key, it can make L become a circulant matrix. The inverse of a circulant matrix can be computed very efficiently by using extended Euclidean algorithm. The inverse of P can be computed as follows:

Step 1

Randomly choose $v_1, \dots, v_v \in (K^v)$.

Step 2

Substitute (x'_1, \dots, x'_v) with (v_1, \dots, v_v) , we will get o linear equations of o variables $L\mathbf{o} = \mathbf{u}$ where matrix L is an $o * o$ circulant matrix. Use extend Euclidean algorithm to find an inverse of L and get a solution for $(\hat{x}_1, \dots, \hat{x}_o)$. (If the system is not regular, go back to Step 1). Let $(x_1, \dots, x_n) = (x'_1, \dots, x'_v, \hat{x}_1, \dots, \hat{x}_o)$.

Step 3

Apply inverse map of T to (x_1, \dots, x_n) .

Circulant UOV stands against all known attacks for UOV if we choose the parameter properly. Experimental results in [28] show that the private key size is 45% smaller than that of UOV and its signing speed is more than 14 times faster than that of UOV.

III. APPLYING PRECOMPUTATION METHODS TO UOV ON EH-WSN

In this section, we are going to describe our precomputation methods for UOV on EH-WSN.

A. PRECOMPUTATION METHOD FOR UOV

1) CONSTRUCTION

The most expensive part in signing process of UOV is Step 2. In Step 2, we need to choose a random vinegar vector \mathbf{v} to set up a linear system $L\mathbf{o} = \mathbf{u}$, and then solve this linear system using Gauss Elimination. Here we are going to divide Step 2 of signing process of UOV into an online phase and an offline phase.

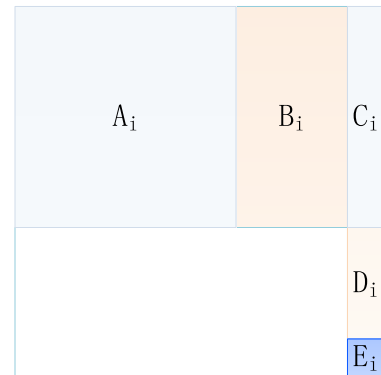


FIGURE 2. Central matrix of Circulant UOV.

First, we show the matrix representation of UOV central polynomials. We keep the constant and linear parts so that our central matrices are $(n + 1) * (n + 1)$ matrices of the form in Figure 2. The white area stands for zero elements. Submatrix A_i is a $v * v$ matrix standing for Vinegar-Vinegar cross-terms coefficients, B_i is a $v * o$ matrix standing for Oil-Vinegar cross-terms. C_i is the linear coefficients of Vinegar variables. D_i in the last column is the linear coefficients of Oil variables, and E_i is the constant term.

Assuming the value to be inverted is \mathbf{m} . We randomly choose a Vinegar vector \mathbf{v} . Substituting (x'_1, \dots, x'_v) with $\mathbf{v} = (v_1, \dots, v_v)$, we will get a linear equation system of o variables. For each central polynomial f_k we get equation:

$$\underbrace{\mathbf{v}^T * A_k * \mathbf{v} + \mathbf{v}^T \cdot \alpha_k + c_k}_{\text{constant}} + \underbrace{\mathbf{v}^T * B_k * \mathbf{o} + \beta_k \cdot \mathbf{o}}_{\text{linear in } \mathbf{o}} = m_k.$$

Vector $\mathbf{o}=(o_1, \dots, o_o)$ stands for Oil variables vector $(x_{v+1}, \dots, x_{v+o})$. Let $\mathbf{y} = (y_1, y_2, \dots, y_o)$, $y_k = (\mathbf{v}^T * A_k * \mathbf{v} + \mathbf{v}^T \cdot \alpha_k + c_k)$ for $k \in [1, \dots, o]$. Let $\mathbf{u} = \mathbf{m} - \mathbf{y}$. Then we get the linear system $L\mathbf{o} = \mathbf{u}$:

$$\underbrace{\begin{pmatrix} \mathbf{v}^T * B_1 + \beta_1 \\ \mathbf{v}^T * B_2 + \beta_2 \\ \vdots \\ \mathbf{v}^T * B_{o-1} + \beta_{o-1} \\ \mathbf{v}^T * B_o + \beta_o \end{pmatrix}}_L \begin{pmatrix} o_1 \\ o_2 \\ \vdots \\ o_{o-1} \\ o_o \end{pmatrix} = \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_{o-1} \\ m_o \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{o-1} \\ y_o \end{pmatrix}.$$

In the linear system, matrix L and vector \mathbf{y} are independent of message \mathbf{m} . We can compute them before we know the message to be signed. So we can divide it into offline phase and online phase. We propose the basic precomputation method for UOV as follows.

Offline phase:

Randomly choose $v_1, \dots, v_v \in (K^v)$. Substitute (x'_1, \dots, x'_v) with (v_1, \dots, v_v) , we can get an $o * o$ matrix L and a vector \mathbf{y} with size o , $y_k = (\mathbf{v}^T * A_k * \mathbf{v} + \mathbf{v}^T \cdot \alpha_k + c_k)$ ($k \in [1, \dots, o]$). If the L is not invertible, choose another vinegar vector \mathbf{v} . Compute the inverse of the matrix L . Store the precomputation tuple $(\mathbf{v}, \mathbf{y}, L^{-1})$ into memory.

Online phase:

Compute $\mathbf{u} = \mathbf{m} - \mathbf{y}$ and solve the system $\mathbf{o} = L^{-1}\mathbf{u}$. Then the signature can be computed as $\mathbf{s} = T^{-1}(\mathbf{v}||\mathbf{o})$. The $||$ operator is the concatenation operator which joins vectors together.

2) PERFORMANCE ANALYSIS

The total computational complexity of execution with precomputation is slightly larger than monolithic execution. In addition, execution with precomputation needs more memory to store the precomputation tuple $(\mathbf{v}, \mathbf{y}, L^{-1})$. Table 1 gives the computational complexity of UOV with precomputation. From Table 1, we can observe that the total complexity for UOV with precomputation is not reduced though, the runtime latency is much smaller. When the message arrives, it only needs to do two simple vector matrix multiplication operations. For $v = 2o$, this can reduce the latency of signature generator by a factor of o .

TABLE 1. Computational complexity of UOV with precomputation.

	Offline	Online
Computational complexity	$\Omega(o^3 + v^2o)$	$\Omega(o^2 + n^2)$

Although our UOV signature scheme with precomputation can reduce energy cost and accelerate the speed of signature generation in EH-WSN, its experimental performance is not as good as expected. The reason for this problem is that the size of precomputation tuple $(\mathbf{v}, \mathbf{y}, L^{-1})$ is too large and available storage is not enough to store enough tuples. Since the supercapacitor suffers from leakage, energy which is harvested and not used progressively leaks and is wasted. It is better to precompute tuples as much as possible when harvested energy is available. However, suppose we have 10KB RAM and 1024KB flash and take (GF(31), $o = 33$, $v = 66$) as parameters of UOV. We need about 743 bytes space to store a tuple, and we have to store them in a flash memory. Access to flash memory brings about an extra time/energy cost. Another problem is that we can only store 1200 tuples in the daytime. This will limit the availability of precomputation greatly. To solve this problem, we have to reduce the size of the precomputation tuple.

B. PRECOMPUTATION METHOD FOR CIRCULANT UOV

1) CONSTRUCTION

In this section, We propose Circulant UOV with precomputation. Compared with UOV, Circulant UOV has faster signature generation and smaller private key.

Circulant UOV has some rotating relations among parts of submatrix of different central matrices. B_i and D_i ($i \in [1, \dots, o]$) in Figure 2 have the following rotating relations:

$$\begin{aligned}
 B_1 &= (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_o) \\
 B_2 &= (\mathbf{b}_o, \mathbf{b}_1, \dots, \mathbf{b}_{o-1}) \\
 &\vdots \\
 B_o &= (\mathbf{b}_2, \mathbf{b}_3, \dots, \mathbf{b}_1) \\
 D_i &= (d_1, d_2, \dots, d_o) \\
 D_2 &= (d_o, d_1, \dots, d_{o-1}) \\
 &\vdots \\
 D_o &= (d_2, d_3, \dots, d_1)
 \end{aligned}$$

Those rotating relations will make the coefficient matrix L of the linear equations be a circulant matrix. Therefore, only the first row of L needs to be calculated. The inverse of L can be computed using Extended Euclidean algorithm.

Precomputation method for Circulant UOV is similar to that for UOV. The difference is that Circulant UOV only needs to save the first row of L^{-1} , while UOV needs to save the entire matrix L^{-1} .

We divide the signing process of Circulant UOV into two phases as follows:

Offline phase:

Randomly choose $v_1, \dots, v_v \in (K^v)$. Compute $\mathbf{v} * B_1 + \beta_1$ to get the first row of L and associate it with the polynomial $f(x) = \sum_{i=0}^{o-1} l_i x^i$. Use the extend Euclidean algorithm to find a polynomial $g(x)$ in $K[x]$ such that $f(x) * g(x) = 1 \pmod{x^o - 1}$. If there is no such $g(x)$, choose another vinegar vector \mathbf{v} . Substitute (x'_1, \dots, x'_v) with (v_1, \dots, v_v) to get the vector \mathbf{y} , $y_k = (\mathbf{v}^T * A_k * \mathbf{v} + \mathbf{v}^T \cdot \alpha_k + c_k)$ ($k \in [1, \dots, o]$). Store the precomputation tuple $(\mathbf{v}, \mathbf{y}, g(x))$ into memory.

Online phase:

Compute $\mathbf{u} = \mathbf{m} - \mathbf{y}$ and associate $g(x)$ with vector \mathbf{b} to get the first row of L^{-1} . We can see that L^{-1} is a circulant matrix too. Solve the system $\mathbf{o} = L^{-1}\mathbf{u}$. Then the signature can be computed as $\mathbf{s} = T^{-1}(\mathbf{v}||\mathbf{o})$.

2) PERFORMANCE ANALYSIS

Table 2 gives the computational complexity of Circulant UOV with precomputation. Comparing Table 2 with Table 1, we can observe that Circulant UOV has a smaller computation cost in total than that of UOV. Precomputation is perfectly

TABLE 2. Computational complexity of Circulant UOV with precomputation.

	Offline	Online
Computational complexity	$O(v^2o)$	$O(o^2 + n^2)$

suitable for Circulant UOV. Because the precomputation tuple $(\mathbf{v}, \mathbf{y}, g(x))$ is much smaller than that of UOV. For Circulant UOV ($GF(31)$, $o = 33$, $v = 66$), we only need about 80 bytes space to store a tuple which is 10 times smaller than that of UOV. This will increase the availability of precomputation. Smaller tuple also reduces extra time/energy cost for reading flash memory.

Circulant Rainbow [29] has better performance than Circulant UOV. However, Rainbow is a multilayer variant of UOV. The inverse of the central map depends on the message to be signed. Therefore, the performance improvement of precomputation on Circulant Rainbow is not significant.

IV. PERFORMANCE EVALUATION

In this section, we will systematically evaluate the performance of our precomputation in a testbed of energy harvesting wireless motes by using simulations-based experiments and actual deployment with real-life energy traces.

A. EXPERIMENTAL SCENARIO

In our experiments, the signature scheme is implemented on the widely used wireless sensor mote TelosB. TelosB is a low power wireless sensor module developed and initially distributed to research community by UC Berkeley. It uses the low power MSP430 microcontroller as the core processor, supports TinyOS operating system, provides 48kB ROM, 10kB RAM, 1024kB flash, and adopts CC2420 wireless radio frequency transceiver chip. CC2420 radio frequency transceiver is the first single-chip 2.4 GHz IEEE 802.15.4 compliant and ZigBee ready RF Transceiver. Its performance is stable and the power consumption is very low. The data transmission rate of wireless communication equipment developed using this chip can reach 250kbps. In addition, we have configured an energy harvesting subsystem on TelosB, including an energy harvester, an energy management module, and an energy storage module. The energy obtained by the harvester can be used directly by the node. When the node is harvesting power at a rate that exceeds its current power consumption, the remaining energy can be stored in the energy storage device. Generally, rechargeable battery or supercapacitor can be used as energy storage device. This paper uses IXOLAR XOB17-04x3 solar cells as energy harvester, 1F Maxwell HC series supercapacitor as energy storage device. In order to reduce the overhead of idle monitor of main transceiver, we have also integrated the present advanced low power RF wake-up receiver for low delay asynchronous communication.

The availability of relatively large storage flash memory chips embedded in modern sensor nodes plays a crucial role

in permitting memory/performance trade-offs, which were not possible in previous generation platforms due to memory constraints. Flash chip is a specific type of EEPROM (Electrically Erasable Programmable Read-Only Memory) that enables access to n -bytes blocks in a single operation, instead of one operation per byte. This memory is non-volatile, which means it doesn't need energy to maintain the information stored in the chip. TelosB uses the ST M25P80 40MHz Serial flash for external data and code storage. The flash memory holds 1024 KB of data and is decomposed into 16 segments, each 64 KB in size. It enables the random access for readings and shares SPI communication bus with the CC2420 transceiver. The minimum unit to be erased in flash is a block. This means all cells in a block must be erased together. Writing is performed on a per-byte basis, but it requires the block to be erased before writing on it.

In order to manage the data stored in the flash, we rely on TinyOS 2.x primitives [30]. Specifically, TinyOS 2.x provides three basic storage abstractions: small objects, circular logs, and large objects. TinyOS 2.x divides the flash chip into one or more fixed-size volumes. Each volume provides a single type of storage abstraction (e.g., configuration, log, or block storage). The abstraction type defines the physical layout of data on flash memory. We use the LogStorage and ConfigStorage abstractions to write and read data.

We focus on a climate monitoring scenario, in which temperature and humidity of environment are measured twice per minute. We equipped TelosB with an on-board Sensirion SHT1x sensors that perform temperature and humidity measurements twice per minute. Measured data will be signed by UOV signature scheme and the signed data will be delivered by the IEEE 802.15.4 compliant wireless transceiver. During the rest of time, MCU is in sleep mode to reduce power consumption.

B. PERFORMANCE EVALUATION OF OUR PRECOMPUTATION

We implement our online/offline signature scheme by using nesC language on TinyOS. TinyOS is an open source operating system designed for low-energy wireless devices. The component based on framework has greatly facilitated the development of embedded applications.

1) PARAMETER CHOSEN

We choose $GF(31)$ as the base field of UOV for faster basic operations. According to the current conclusion in [28] and [31]. We choose $(GF(31), v = 38, o = 19)$ for 64-bit security and $(GF(31), v = 66, o = 33)$ for 80-bit security.

2) HYBRID REPRESENTATION

As MSP430 is a 16-bit RISC processor, we use 16-bit integer to represent a field element. Lazy modular reduction can be used to speed up the basic field operations. However, storing elements in 16-bit representation will cause redundant storage. A better idea is to pack three 5-bit elements in 16-bit and

TABLE 3. Overview comparison among different schemes on TelosB.

Scheme	Security	Private key size	offline	online
UOV	64 bit	18.8 KB	88ms/0.48mJ	
UOV	80 bit	96.5 KB	409ms/2.21mJ	
UOV with precomputation	64 bit	18.8 KB	112ms/0.60mJ	7ms/0.03mJ
UOV with precomputation	80 bit	96.5 KB	508ms/2.74mJ	19ms/0.10mJ
Circulant UOV	64 bit	12.5 KB	59ms/0.31mJ	
Circulant UOV	80 bit	53.9 KB	244ms/1.32mJ	
Circulant UOV with precomputation	64 bit	12.5 KB	50.4ms/0.27mJ	8ms/0.04mJ
Circulant UOV with precomputation	80 bit	53.9 KB	223ms/1.21mJ	21ms/0.11mJ

convert them to 16-bit representation during computing. This simple strategy can reduce 65% of the memory overhead.

3) LAZY MODULAR REDUCTION

To compute $a + b$ over the base field, we first do integer addition: $int(a) + int(b)$, then reduce the result into $[0, 30]$. Shift-and-add can be applied to speed up the reduction operations. As we use 16-bit integer to represent a field element when doing computation, $int(a)$ and $int(b)$ is far less than 2^{16} . No overflow will occur during integer addition. $int(a)*int(b)$ is strictly less than $(2^5 - 1)^2$, we can do only 1 modular reduction for 64 mul-and-add operations, which will save a lot of time.

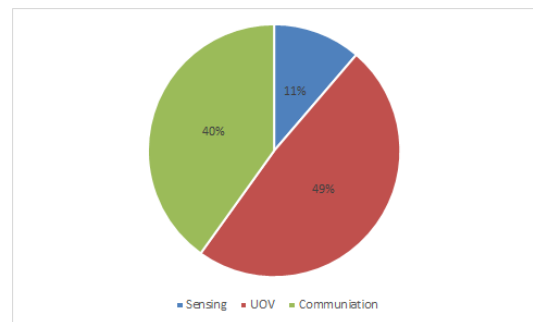
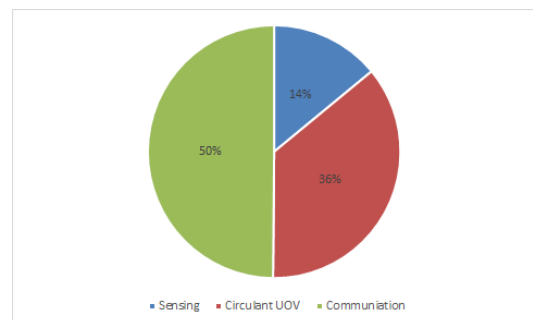
We implement UOV, UOV with precomputation, Circulant UOV and Circulant UOV with precomputation on our experimental platforms. We carried 1000 experiments for each scheme and recorded their average performance. The energy consumption can be calculated by the voltage, current and working time of MCU. Table 3 gives an overview comparison among them.

From Table 3, we can observe that energy overhead and latency of signing process of Circulant UOV is approximately 60% of that of UOV, and the private key size of Circulant UOV is approximately 55% of that of UOV. At the meantime we can observe that precomputation can significantly reduce the energy cost and accelerate the signing speed of EH-WSN. For UOV, precomputation can accelerate the run-time signing and reduce the run-time energy cost by about 95%. For Circulant UOV, precomputation can reduce the run-time signing speed and the run-time energy cost by about 91%.

C. PERFORMANCE EVALUATION WITH HARVESTING-ENABLED OPTIMIZATIONS

In order to evaluate the performance optimization of the energy harvesting and precomputation method for the system, we simulate UOV, UOV with precomputation, Circulant UOV and Circulant UOV with precomputation in GreenCastalia [32], which is an open-source energy-harvesting simulation framework developed for the Castalia simulator. GreenCastalia supports multi-source, multi-storage energy harvesting framework. In the simulations, we consider the same scenario as in Section IV-A. We include a realistic model of MSP430 to match the energy cost of TelosB motes. Temperature and humidity of environment are measured

twice per minute. We set the power consumption to 3mW and the working time to 171ms for each measurement by using the on-board Sensirion SHT1x. The collected data is sent to the aggregation node after being signed. We use the default settings of GreenCastalia for communication. The channel data rate is set to 250kbps. We run the simulation for seven days. The energy harvesting system uses the IXOLAR XOB17-04x3 solar cells as the energy harvester, the 1F Maxwell HC series supercapacitor as the energy storage device, the solar charging battery as the main battery, and a non-rechargeable battery as the standby battery.

**FIGURE 3.** Percentage of per-day energy consumption associated with sensing, signing, and communication using UOV without precomputation.**FIGURE 4.** Percentage of per-day energy consumption associated with sensing, signing, and communication using Circulant UOV without precomputation.

First, we test the overhead of wireless sensor network nodes in three aspects: data sensing, signing, and communication. Figure 3 and Figure 4 give the percentage of per-day energy consumption associated with sensing, signing,

and communication using UOV or Circulant UOV without precomputation.

Here, we use a state-of-the-art RF Wake-Up Receiver with nano ampere current consumption rather than duty-cycle-based communication. By eliminating idle listening in main transceiver, it can reduce the energy consumption of communication. From Figure 3 and Figure 4, we can see that the energy consumption of generating signatures is a very expensive part in our systems. It accounts for 49% of total overhead when signing with UOV and it accounts for 36% of the total overhead when signing with Circulant UOV. And the energy consumption of sensing is the least. Therefore, optimization of signing process can reduce the total energy cost of the system.

To simulate energy harvesting, we can collect solar energy from real life as input for simulation. Here, we use the energy-harvesting datasets published by the National Renewable Energy Laboratory. We randomly choose the solar energy data in Denver for seven consecutive days in June as our energy supply. Figure 5 presents the solar irradiance in Denver for a day in June. Suppose I is the radiant energy incident onto surface, we calculate the power P_h harvested by a solar cell of size S and efficiency η as: $P_h = S \cdot \eta \cdot I$.

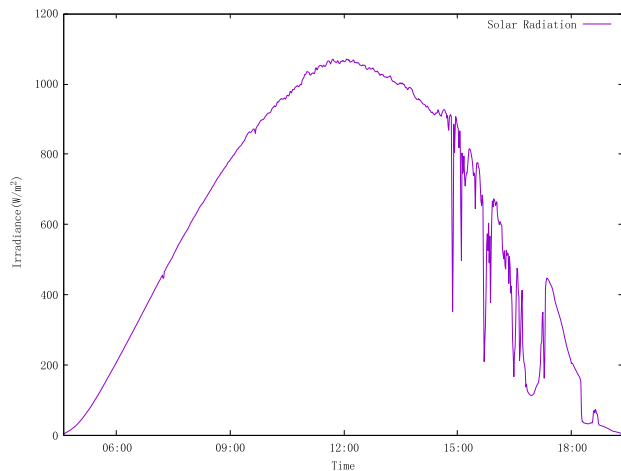


FIGURE 5. Solar irradiance in Denver for a day in June.

The energy harvested by the nodes varies with weather conditions. This causes energy to be either sparsely used, or there may be an excess of energy. The overflowed energy would be wasted if not immediately used. Therefore, when the current energy storage exceeds a given charging level threshold and a node is harvesting power at a rate that exceeds its current power consumption, we precompute tuples as much as possible. The precomputed result is stored in the flash memory of the nodes. Figure 6 shows the average energy spent per day to sign messages with different signing schemes. The average energy spent per day to sign messages is 6.3J for UOV without precomputation and around 3.54J with precomputation, resulting in a 44% reduction in energy consumption. For Circulant UOV, precomputation can reduce

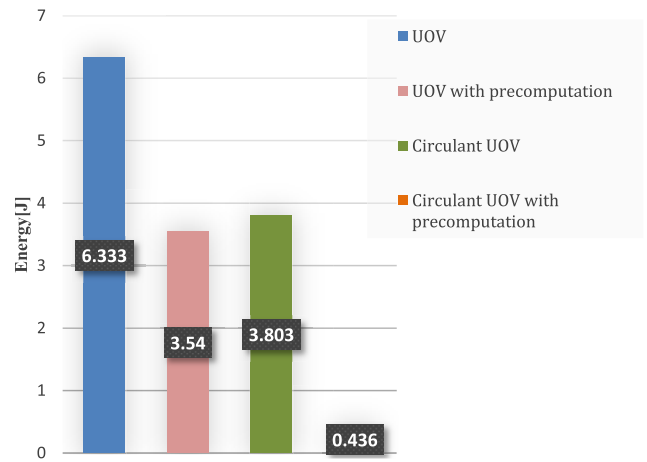


FIGURE 6. Average energy spent per day to sign messages with different signing schemes.

energy cost by about 88%. Also, compared with the basic precomputation for UOV, the average energy spent per day to sign messages is 0.436J for Circulant UOV with precomputation, resulting in a 87% reduction in energy consumption. That means, circulant method can further reduce the energy cost. Overall, by combining circulant method with precomputation in UOV signature, the cost of UOV signature in a WSN node can be reduced by 93%. This greatly improves the performance of the signature and makes it more suitable for resource-limited wireless sensor network environment.

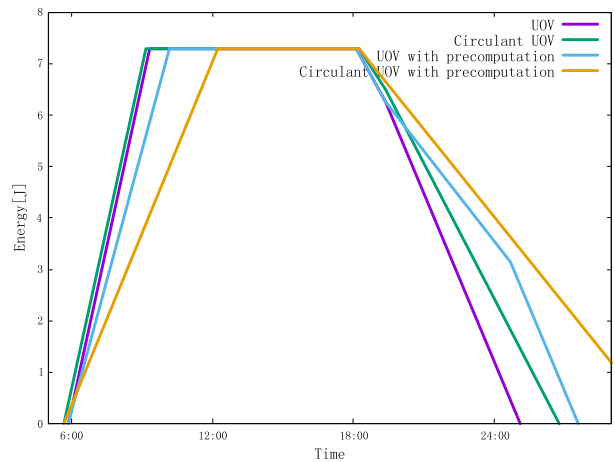


FIGURE 7. Energy stored in the supercapacitor in the first day with different signing schemes.

In order to further estimate the impact of precomputation on availability, we consider TelosB with an on-board Sensirion SHT1x sensors that performs temperature and humidity measurements four times per minute. Figure 7 shows snapshot of energy stored in supercapacitor in the first day with different signing schemes. The power stored in supercapacitor will run out in 7 hours when generating signature using UOV. Precomputation can make UOV stay a little longer,

TABLE 4. The daily average battery energy consumption of each scheme when the solar energy is first supplied and the non rechargeable battery is used as backup power.

Energy Consumption Type (battery)	UOV	Circulant UOV	UOV with Precomputation	Circulant UOV with Precomputation
Daily Energy Consumption of Communication	1.70J	1.08J	0.77J	0.10J
Daily Energy Consumption of Sensor	0.48J	0.30J	0.22J	0.03J
Daily Energy Consumption of Signing	2.06J	0.78J	0.94J	0.01J
Total Daily Energy Consumption	4.24J	2.16J	1.93J	0.14J

TABLE 5. The total number of signatures generated within three days when nodes are powered by a harvesting subsystem.

Scheme	UOV	Circulant UOV	UOV with Precomputation	Circulant UOV with Precomputation
Number of Signatures	14481	15507	16005	17112

TABLE 6. Comparison with other signature schemes.

Scheme	Security	Private key size	offline	online
UOV [33]	80bit	77.3 kB	-	136ms/2.45mJ
UOV	80 bit	96.5 kB	409ms/2.21mJ	
Circulant UOV	80 bit	53.9 kB	244ms/1.32mJ	
UOV with precomputation	80 bit	96.5 kB	508ms/2.74mJ	19ms/0.10mJ
Circulant UOV with precomputation	80 bit	53.9 kB	223ms/1.21mJ	21ms/0.11mJ

but the impact is limited because of the limited number of precomputation tuple. For UOV with precomputation, we can see that there exists significant slope change in polylines of energy stored in supercapacitor during the discharging phase. The reason for this problem is the precomputation tuples are used up. Circulant UOV has a faster signing process, which will make it more economical. Circulant UOV with precomputation is the cheapest of these four solutions. When precomputation tuples are sufficient, the supercapacitor can support it for 12.7 hours. It can be seen that precomputation enhances the availability of overflow energy and reduces energy consumption in the online phase, which prolongs lives of batteries. For a frequent computing system, the smaller the precomputation result is, the more stored precomputation pairs can be, and the better the availability of the system will be.

By using the harvest energy as the main energy supply and the battery as a backup energy, we can reduce the average battery consumption per day. When enough energy is collected from the environment, the system uses the collected energy to supply electricity. When the energy collected from the environment is not enough to support the operation of the system, the system will consume the energy of the backup battery. Table 4 gives the average energy consumption of battery per day for signing messages. Daily consumption of battery energy for generating signature reduces to 4.24J for UOV and 2.16J for Circulant UOV. It is clear that energy harvesting technologies can reduce the pre-day energy consumption of battery and prolong the life cycle of system. The impact of precomputation is also very obvious. The signing overhead of UOV with precomputation is about 45% of that of UOV, and the signing overhead of Circulant UOV with precomputation is about 6.5% of that of Circulant UOV. As can be seen from Figure 7 and Table 4, the longer the rechargeable

battery life is, the less battery consumption of the backup battery will be. After using the energy harvesting technology, the daily energy consumption of the backup battery is reduced to 0.14J when the Circulant UOV with precomputation signature scheme is used for signature. When the sensor node uses up the power of capacitor, it will use the non-rechargeable battery for power supply. At this time, the precomputation pairs generated in the offline phase are not used up. Therefore, it only needs to consume 0.11mJ for each signature. It can be seen from table 4 that using precomputation and energy harvesting technology can prolong the life of battery and the total available time of system.

Finally, we tested the total number of signatures generated by different signature schemes within 3 days in a practical experimental scenario. We consider the same experimental scenario as the Section IV-A, which only collects solar energy from the environment to power the system. We choose the parameters of UOV with 80-bit security. Each mote performs temperature and humidity measurements four times per minute.

We placed the sensor node by the window of our lab and recorded the total number of signatures generated by each signature scheme from August 25, 2017 to August 28, 2017, which is shown in Table 5. From Table 5, we can observe that using precomputation techniques increases total number of signatures. The faster the power consumption of the capacitor in Figure 7 is, the less the total signature number in Table 5 will be, and vice versa. Using the precomputation method, Circulant UOV signature scheme can make system run longer and get more signatures.

D. COMPARISON WITH OTHER SIGNATURE SCHEMES

We compare the performance of our signature schemes with other online/offline signature schemes on EH-WSN in terms

of time and energy consumption on a TelsoB node. The comparison result is shown in Table 6.

As is expected in Table 6, our signature scheme is about seven times faster than the work of [33]. And the energy consumption is much smaller than that of the compared work. It clearly shows that precomputation permits our scheme to significantly outperform other schemes reported in the table.

V. CONCLUSION

In this paper, we mainly focus on a concrete online/offline implementation of UOV signature scheme in wireless sensor networks with energy harvesting capabilities. We propose precomputation methods for UOV signature in wireless sensor networks. Through simulations and real-life experimentation, we showed that precomputations permit one to significantly reduce the energy cost and improve the performance of system. By combining circulant method with precomputation techniques and energy-harvesting capabilities of modern sensor nodes, the cost of UOV signature in a WSN node can be reduced by 93%. This greatly enhances the performance of UOV signature, making it feasible for the practical deployment on resource-constrained wireless sensor platforms. Besides, circulant method reduces size and memory overhead of the precomputation tuple. This increases the availability of precomputation and greatly enhances the availability of the overflow energy. By moving the computation of most resource-demanding operations to times when the energy is at peak, it can also greatly enhance the availability of the overflow energy.

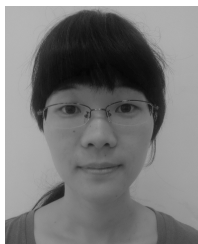
Our experimental results further show that the energy harvesting technique does not leave MPKC behind while improving other signature schemes for WSN platforms.

ACKNOWLEDGMENT

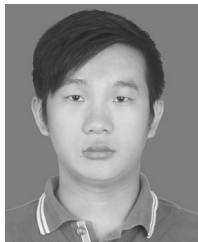
Z. Peng was with the South China University of Technology, and this work was done during his Ph.D. degree at SCUT.

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] S. H. Seo, J. Won, S. Sultana, and E. Bertino, "Effective key management in dynamic wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 371–383, Feb. 2015.
- [3] A. E. Kouche, L. Al-Awami, H. Hassanein, and K. Obaia, "WSN application in the harsh industrial environment of the oil sands," in *Proc. 7th Int. Wireless Commun. Mobile Comput. Conf.*, Jul. 2011, pp. 613–618.
- [4] M. A. Hussain, P. Khan, and K. K. Sup, "WSN research activities for military application," in *Proc. 11th Int. Conf. Adv. Commun. Technol.*, vol. 1, Feb. 2009, pp. 271–274.
- [5] C. Torres and P. Glösekötter, "Reliable and energy optimized WSN design for a train application," *J. Syst. Archit.*, vol. 57, no. 10, pp. 896–904, 2011.
- [6] M. Zennaro et al., "On the design of a water quality wireless sensor network (WQWSN): An application to water quality monitoring in malawi," in *Proc. Int. Conf. Parallel Process. Workshops (ICPPW)*, Washington, DC, USA, 2009, pp. 330–336.
- [7] P. Zhang, J. Wang, K. Guo, F. Wu, and G. Min, "Multi-functional secure data aggregation schemes for WSNs," *Ad Hoc Netw.*, vol. 69, pp. 86–99, Feb. 2018.
- [8] K. Moara-Nkwe, Q. Shi, G. M. Lee, and M. H. Eiza, "A novel physical layer secure key generation and refreshment scheme for wireless sensor networks," *IEEE Access*, vol. 6, pp. 11374–11387, 2018.
- [9] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Commun. ACM*, vol. 47, no. 6, pp. 53–57, Jun. 2004.
- [10] T. T. Chakavarika, B. K. Chaurasia, and S. K. Gupta, "Performance evaluation of a polynomial based key management scheme in wireless sensor networks," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2016, pp. 2114–2118.
- [11] G. Leelavathi, K. Shaila, and K. R. Venugopal, "RSA processor design with vedic multiplier for nodes in wireless sensor networks," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2017, pp. 1254–1257.
- [12] A. Caposelle, V. Cervo, G. De Cicco, and C. Petrioli, "Security as a CoAP resource: An optimized DTLS implementation for the IoT," in *Proc. ICC*, Jun. 2015, pp. 549–554.
- [13] B. Driessen, A. Poschmann, and C. Paar, "Comparison of innovative signature algorithms for WSNs," in *Proc. 1st ACM Conf. Wireless Netw. Secur. (WiSec)*, New York, NY, USA, 2008, pp. 30–35.
- [14] S. Priya and D. J. Inman, *Energy Harvesting Technologies*, vol. 21. Boston, MA, USA: Springer, 2009.
- [15] L. Tan and S. Tang, "Energy harvesting wireless sensor node with temporal death: Novel models and analyses," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 896–909, Apr. 2017.
- [16] G. Bianchi, A. T. Caposelle, C. Petrioli, and D. Spenza, "AGREE: Exploiting energy harvesting to support data-centric access control in WSNs," *Ad Hoc Netw.*, vol. 11, no. 8, pp. 2625–2636, 2013.
- [17] G. Ateniese, G. Bianchi, A. T. Caposelle, C. Petrioli, and D. Spenza, "Low-cost standard signatures for energy-harvesting wireless sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 3, 2017, Art. no. 64.
- [18] G. Ateniese, G. Bianchi, A. Caposelle, and C. Petrioli, "Low-cost standard signatures in wireless sensor networks: A case for reviving precomputation techniques?" in *Proc. NDSS*, 2013, pp. 1–12.
- [19] A. Aysu and P. Schaumont, "Precomputation methods for hash-based signatures on energy-harvesting platforms," *IEEE Trans. Comput.*, vol. 65, no. 9, pp. 2925–2931, Sep. 2016.
- [20] A. Aysu and P. Schaumont, "Precomputation methods for faster and greener post-quantum cryptography on emerging embedded platforms," *IACR Cryptol. ePrint Arch.*, Bellevue, WA, USA, Tech. Rep. 2015/288, 2015.
- [21] K. Sakumoto, T. Shirai, and H. Hiwatari, "Public-key identification schemes based on multivariate quadratic polynomials," in *Advances in Cryptology—CRYPTO*. Berlin, Germany: Springer, 2011, pp. 706–723.
- [22] S. Tang, L. Xu, N. Liu, X. Huang, J. Ding, and Z. Yang, "Provably secure group key management approach based upon hyper-sphere," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3253–3263, Dec. 2014.
- [23] S. Tang, B. Lv, and W. Shen, "Hybrid MQ signature for embedded device," in *Information Security and Privacy*. Melbourne, VIC, Australia: Springer, 2016, pp. 281–290.
- [24] C. Berbain, O. Billet, and H. Gilbert, "Efficient implementations of multivariate quadratic systems," in *Selected Areas in Cryptography*, vol. 4356. Berlin, Germany: Springer, 2006, pp. 174–187.
- [25] H. Seo, J. Kim, J. Choi, T. Park, Z. Liu, and H. Kim, "Small private key MQPKS on an embedded microprocessor," *Sensors*, vol. 14, no. 3, pp. 5441–5458, 2014.
- [26] P. Czypek, S. Heyse, and E. Thomae, "Efficient implementations of MQPKS on constrained devices," in *Cryptographic Hardware and Embedded Systems*. Berlin, Germany: Springer, 2012, pp. 374–389.
- [27] A. Kipnis, J. Patarin, and L. Goubin, "Unbalanced Oil and Vinegar signature schemes," in *Advances in Cryptology—EUROCRYPT*. Berlin, Germany: Springer, 1999, pp. 206–222.
- [28] Z. Peng and S. Tang, "Circulant UOV: A new UOV variant with shorter private key and faster signature generation," *KSI Trans. Internet Inf. Syst.*, vol. 12, no. 3, pp. 1376–1395, 2018.
- [29] Z. Peng and S. Tang, "Circulant rainbow: A new rainbow variant with shorter private key and faster signature generation," *IEEE Access*, vol. 5, pp. 11877–11886, 2017.
- [30] P. Levis et al., "TinyOS: An operating system for sensor networks," in *Ambient Intelligence*. Berlin, Germany: Springer, 2004, pp. 115–148.
- [31] A. Petzoldt, *Selecting and Reducing Key Sizes for Multivariate Cryptography*. Darmstadt, Germany: TUprints, 2013.
- [32] D. Benedetti, C. Petrioli, and D. Spenza, "GreenCastalia: An energy-harvesting-enabled framework for the Castalia simulator," in *Proc. 1st Int. Workshop Energy Neutral Sens. Syst. (ENSSys)*, New York, NY, USA, 2013, Art. no. 7.
- [33] J. Chen, S. Tang, D. He, and Y. Tan, "Online/offline signature based on UOV in wireless sensor networks," *Wireless Netw.*, vol. 23, no. 6, pp. 1719–1730, Aug. 2017.



BO LV received the B.E. degree from the South China University of Technology in 2012, where she is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering. Her current research interests include multivariate public key cryptography, crypto chip design, and applied cryptography.



ZHINIANG PENG received the B.E. degree in materials science and the Ph.D. degree in cryptography from the South China University of Technology in 2013 and 2017, respectively. He is currently a Cybersecurity Researcher with Qihoo 360. His current research interests include blockchain technology, binary exploitation, and applied cryptography.



SHAOHUA TANG received the B.Sc. and M.Sc. degrees in applied mathematics and the Ph.D. degree in communication and information system from the South China University of Technology, China, in 1991, 1994, and 1998, respectively. He was a Visiting Scholar with North Carolina State University, USA, and a Visiting Professor with the University of Cincinnati, USA. He has been a Full Professor with the School of Computer Science and Engineering, South China University of Technology, since 2004. He has authored or co-authored over 100 technical papers in journals and conference proceedings. His current research interests include information security, data security, and privacy preserving in cloud computing and big data. He is a member of the IEEE Computer Society.

...