Hindawi
Mobile Information Systems
Volume 2020, Article ID 8818374, 9 pages
https://doi.org/10.1155/2020/8818374

Research Article

Provenance Transmission through a Two-Dimensional Covert Timing Channel in WSNs

Qinbao Xu,[1] Li Liu,[1] Rizwan Akhtar,[2] Muhammad Asif Zahoor Raja,[3] and Changda Wang[1]

[1] *School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China*
[2] *School of Electronics and Information, Jiangsu University of Science and Technology, No. 2 Mengxi Road, Zhenjiang 212003, China*
[3] *Department of Electrical and Computer Engineering, COMSATS University Islamabad, Attock Campus, Attock, Pakistan*

Correspondence should be addressed to Changda Wang; changda@ujs.edu.cn

Received 15 March 2020; Revised 14 June 2020; Accepted 30 June 2020; Published 23 July 2020

Academic Editor: Ali Kashif Bashir

Provenances, which record the history of data acquisition and transmission, are hard to be transmitted in resource-tightened wireless sensor networks (WSNs) due to their drastic size expansion with the increase in packet transmission hops. To ease the burden caused by the provenance transmission, we first designed a two-dimensional covert timing channel (2dCTC) and then applied it to provenances transmission in WSNs. Based on Cantor Expansion, 2dCTC uses pseudo packet IDs permutation and packet sizes variation together to form a two-dimensional communication medium. Both theoretical analysis and experimental results show that 2dCTC not only has a much higher channel capacity than those of most of the known CTCs, but also conserves more energy for provenance transmission in WSNs. Furthermore, 2dCTC provides a new way to increase CTCs channel capacity and stealthiness through multi-dimensional approaches.

1. Introduction

In the context of wireless sensor networks (WSNs), the provenance of a data item refers to where the item is produced and how it is delivered, i.e., forwarded and/or aggregated to the base station (BS) [1]. Provenance plays an important role in data trust evaluations. Because the size of provenance grows rapidly when packet transmission hop increases, it is then critical to efficiently transmit provenance in resource-tightened WSNs [2]. As a result, several lightweight provenance schemes have been proposed [2–6].

Originally, in a multilevel security system, a covert channel is a mechanism by which a user with high security level can violate the system's security policy to leak sensitive information to a user with lower security level [7]. Now it has been extended to various communication networks and generally defined as the following: if a sender and a receiver use a medium that is not originally designed as the communication medium for the overt channel, it is a covert channel. As a result, a covert channel has two interesting characteristics: (1) as a side channel it can enlarge its overt channel's capacity without consuming extra energy on signals transmission; (2) its channel capacity is much smaller than that of its overt channel in general. Although the first characteristic is fascinating for provenance transmission through covert channel in WSNs, the second characteristic limits such a usage due to the fact that the channel capacity is too small.

In a packet-switched network, according to the applied communication mediums, covert channels can be roughly categorized as covert storage channels (CSCs) and covert timing channels (CTCs). CSC uses the shared storage in a packet as the communication mediums, e.g., the reserved bits in a packet head; CTC uses the timing characteristics relevant to packet transmissions as the communication mediums, e.g., packet sending frequencies, inter-packet delays, etc. Due to the mediums' deference, CSC can be eliminated by a network firewall through traffic normalization [8], whereas CTC is hardly to be removed thoroughly. Many CTC schemes such as [9–12] are then proposed.

The inspiration of the paper is to build a CTC which has much higher channel capacity for provenance transmission in WSNs. We then propose a two-dimensional CTC (2dCTC) scheme which uses pseudo packet IDs permutation and packet sizes variation together as the communication medium. Because the two-dimensional communication medium can carry more information, 2dCTC has a much higher channel capacity than the known traditional CTCs.

The main contributions of this paper are as follows:

(1) We propose a 2dCTC which encodes covert messages into multiple dimension spaces. 2dCTC overwhelms most of the known CTCs with respect to both channel capacity and channel stealthiness.

(2) We devise the message encoding and decoding algorithms for 2dCTC through Cantor Expansion, which is the key to build a two-dimensional communication medium.

(3) We apply 2dCTC to the provenance transmissions in resource-tightened WSNs, which saves both energy and channel capacity.

The remainder of this paper is arranged as follows: Section 2 provides the related works. Section 3 presents 2dCTC's design and implementation. Section 4 shows 2dCTC's performance and corresponding experimental results. Section 5 gives the practice of provenance transmission through 2dCTC. Section 6 concludes the paper.

2. Related Works

Generally, CTCs adopt the timing behaviour of an entity to transmit covert messages in overt network communication.

Among the entities, inter-packet delays (IPDs) are the most common one that are modulated to encode covert messages. Berk et al. [10] proposed encoding messages through the intervals between adjacent packet transmissions, which avoids the time synchronization requirement that may threat the channel's concealment. In [11], a CTC is built through mimicking the inter-packet delays (IPDs) of the normal packet traffic flow, by which to implement a detect-resisting CTC. In addition to the IPDs, packet order can also be used to establish CTC, in which the covert messages are represented as reorderings of packets. El-Atawy et al. [12] proposed a packet-reordering channel which uses the packet sequence disorder in transmission as the communication medium. Such a CTC simulates the phenomenon of naturally occurring packet reordering over networks, which has higher channel capacity than those of CTCs based on the fixed time windows and the IPDs. Zhang et al. [13] proposed a method for establishing a VoLTE CTC through packet re-orderings. To further improve the robustness of such a CTC, Gray code is employed to encode the covert message for the purpose of alleviating the packet loss and packet out-of-order. Liang et al. [14] proposed a payload-dependent packet rearranging CTC for mobile VoIP traffic. Such a CTC can deal with the traffic with more complicated packet distributions such as that in the mobile VoIP environments. In contrast to the aforementioned packet re-ordering methods, we use pseudo packet IDs permutation to encode messages, which can gain more flexibility. There are also some studies using packet length information to build CTC. Liang et al. [15] proposed a packet length covert channel for mobile VoIP traffics, in which the packet length distribution was partitioned and such partitions were mapped to data symbols. The main concept of such a CTC is to send covert messages through transmitting packets of corresponding size. Our method is inspired by such a concept. There is also a category of CTCs using the number of packets transmitted within a time slot to encode/decode messages. Cabuk et al. [9] proposed the Simple Timing Covert Channel (STC), in which the sender divides the timeline into a series of smaller time slots with fixed length; the binary number 1 or 0 is then encoded based on whether a packet is sent within a given time slot. However, such a method requires the clock synchronization between the sender and receiver, which is hard to achieve especially in large-scale networks.

Because each of the CTCs mentioned above uses only one communication medium, all of them are one-dimensional CTCs. To drastically raise the CTCs' capacity, in addition to applying any hardware-based methods, we propose the concept of multi-dimensional CTCs. As a first step for multi-dimensional CTCs' practice, we design and implement a two-dimensional CTC named 2dCTC in the paper.

Among the existing provenance schemes in WSNs, Probabilistic Provenance Flow (PPF) scheme [16] as a block provenance scheme probabilistically appends the node IDs on the packet path to the provenance, and therefore each packet only carries a block of the provenance, i.e., a connected subgraph of a packet transmission path, to the BS. Similarly, Probabilistic Provenance Mark (PPM) scheme [17] probabilistically incorporates node ID to the packet and each packet only contains one node ID. As to provenance transmission through covert channels, to the best of our knowledge, only one paper can be found; viz., in [18], Sultana et al. use the IPDs (inter-packets delays) based CTC for provenance transmission, in which the original purpose is to increase the concealment of the transmission, but objectively saves both energy and channel capacity in WSNs. As a one-dimensional CTC, the IPDs based CTC has very limited channel capacity; the steady packet flows are then required for provenance transmission in [18].

3. 2dCTC's Design and Implementation

The 2dCTC proposed in this paper uses pseudo packet IDs permutation and packet sizes variation together as the communication medium. Like the works in [18], the relatively stable data packets flow is required. To facilitate understanding our two-dimensional CTC scheme, we first provide the message encoding and decoding in two one-dimensional mediums, viz., messages encoding and decoding through pseudo packet IDs permutation and packet sizes variation, respectively.

3.1. Pseudo Packet IDs Permutation as the Medium. In packet-switched networks, the packet ID disorder rate in transmission is between 0.1% and 3% roughly [19], which provides few packets to form a CTC by the packet IDs permutation. We thus propose the concept of pseudo packet ID that is a data block with a unique value appended to a packet. Unlike packet ID that resided in packet-header, the pseudo packet ID resided in the payload area. Figure 1 shows the working principle of a CTC using the pseudo packet IDs permutation as the communication medium.

At the beginning, the message is divided into N binary blocks, i.e., $\{s_1, s_2, s_3, \ldots, s_i, \ldots, s_N\}$, and each block contains 8 bits. The corresponding decimal number of s_i is S_i. Let $\{sid_i \mid sid_n \in R^+, i = 1, 2, \ldots, n\}$ represent the set of pseudo packet IDs; the main steps of the message encoding through the pseudo packet IDs permutation are as follows.

(1) With the number of bits in s_i, the number of packets n that satisfies $2^L \leq n!$ is chosen. So, each s_i keeps 8 bits and $n = 6$.

(2) With the value of S_i, a pseudo packet IDs permutation generated from $\{sid_1, sid_2, sid_3, \ldots, sid_n\}$ is processed by Cantor Expansion inverse operation [20], which provides a bijection between a Cantor value X and a permutation. If there are n packets, a pseudo packet IDs permutation of $a[i]$ $(1 \leq i \leq n,)$, where a Cantor value X can be derived through the following equation:

$$X = a[n](n-1)! + a[n-1](n-2)! + \cdots + a[1]0!. \tag{1}$$

(3) Each generated pseudo packet ID is appended to the payload area of the sending packets in a stream manner.

Note that, compared to the message encoding and decoding through a mapping table whose time complexity is $O(n\lg n)$, the time complexity of our Cantor Expansion based scheme is $O(n)$.

After the CTC receiver filtrates the required packets, the pseudo packet IDs are rearranged according to the packet's arrival time and then the messages can be retrieved through Cantor Expansion by equation (1).

To better understand the approach in this subsection, we provide an example in here. Assume that s_i is 00001011 (the corresponding decimal number S_i is equal to 11). The Cantor value X is then equal to 11 and the pseudo packet IDs are $\{1, 2, 3, 4, 5, 6\}$. According to the inverse form of Cantor Expansion, the process is as follows: 11 divided by 5! equals 0 with reminder 11; therefore $a[6] = 0, sid_1 = 1$; 11 divided by 4! equals 0 with reminder 11; therefore $a[5] = 0, sid_2 = 2$. Following the same process, $a[4] = 1$, $sid_3 = 4$; $a[3] = 2$, $sid_4 = 6$; $a[2] = 1$, $sid_5 = 5$; $a[1] = 0$, $sid_6 = 3$. As a result, the order of the pseudo packet IDs of 11 is $\{1, 2, 4, 6, 5, 3\}$. We append $\{1, 2, 4, 6, 5, 3\}$ to the sending packets' payload areas. After the CTC receiver filtrates such packets, the pseudo packet IDs permutation $\{1, 2, 4, 6, 5, 3\}$ whose Cantor value $X = 11$ is retrieved; $s_i =$ 00001011 is then decoded.

3.2. Packet Sizes Variation as the Medium. Using packet sizes variation to encode and decode messages has several obvious advantages. For instance, such a coding method cannot be easily affected by the channel noise such as packet transmission delays and jitters. The working principle of a packet sizes variation based CTC is illustrated in Figure 2. By adopting such a CTC, the message can be encoded through the following steps:

(1) A histogram model $\{M, B, X\}$ of packet size is established, in which M, B, and X denote the number of packets of each group, the group distance, and the sample data sequence, respectively. The statistical function $M = Hg_{B,R}(X)$, in which R sets the packet sizes range for each group, is used to calculate the value of M, i.e., the number of packets in each group.

(2) A mapping table is built to represent the correlation between the packet sizes barrel, i.e., a packet size group, and the corresponding binary blocks. Obviously, if a packet size barrel represents α bits, the number of packet size barrels will be equal to 2^α.

(3) The message s_i in a binary representation is encoded into the sending packets based on the mapping table built in the previous step.

After the receiver filtrates the corresponding packets, the messages can be retrieved by looking up the mapping table.

A simple example is provided here for better understanding such a coding method. Assume the message s_i to be sent is represented in binary as 00001011. There are 9 packets, i.e., $p_1, p_2, \ldots, p_9$ with different sizes, i.e., $l_1, l_2, \ldots, l_9$. We suppose to classify these 9 packets into two packet size barrels, B_1 and B_2 according to the packet size threshold l; i.e., packets whose sizes are less than l are associated with B_1; otherwise, B_2. Assume that p_1, p_3, p_4, p_5, p_6 and p_9 belong to B_1 and others belong to B_2. In this example, α is equal to 1 and the number of packet size barrel is 2. Then, s_i can be encoded into packet transmission order: $p_1, p_3, p_4, p_5, p_8, p_6, p_2, p_9$. After the receiver filtrates the packet size as $l_1, l_3, l_4, l_5, l_8, l_6, l_2, l_9$, it can decode s_i as 00001011 by looking up the mapping table.

3.3. Two Mediums Are Used Together. To transmit a message consisting of L bits, the message needs to be organized as two parts. The first part (K bits) is encoded through packet sizes variation and the second part ($L - K$ bits) is encoded through pseudo packet IDs permutation. Figure 3 shows the working principle of 2dCTC. The main steps are shown as follows.

(1) Calculate n, the number of packets needed in communication, by

$$\alpha n + \log n! \geq L, \tag{2}$$

where α denotes the number of bits represented by one packet size in the mapping table. As a result, K bits are the first αn bits of the message counting from the left.

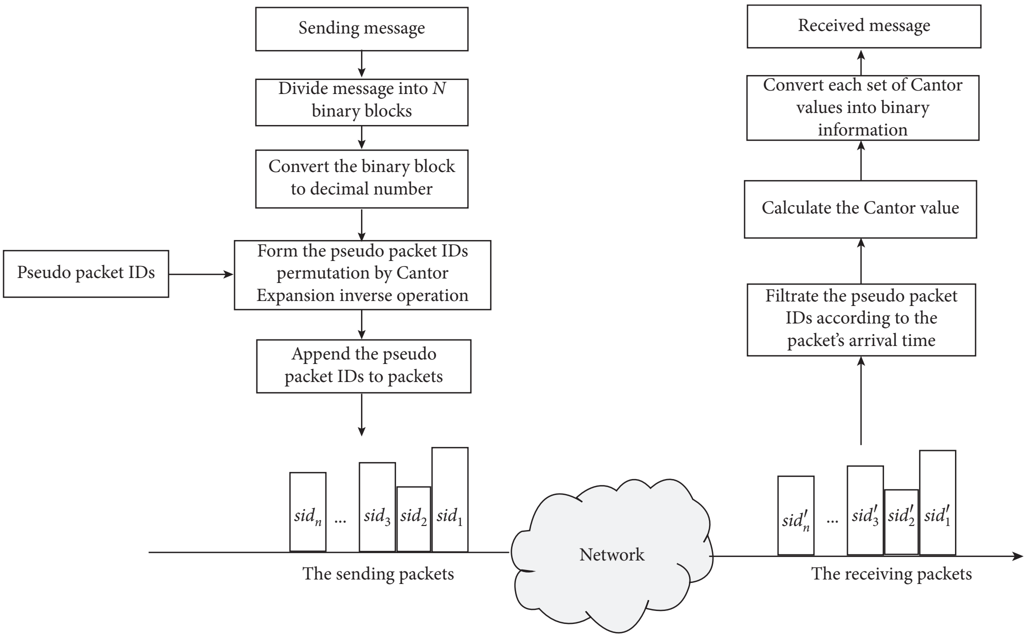
Sending message
Divide message into N binary blocks
Convert the binary block to decimal number
Pseudo packet IDs
Form the pseudo packet IDs permutation by Cantor Expansion inverse operation
Append the pseudo packet IDs to packets
sid_n ... sid_3 sid_2 sid_1
The sending packets
Network
Received message
Convert each set of Cantor values into binary information
Calculate the Cantor value
Filtrate the pseudo packet IDs according to the packet's arrival time
sid'_n ... sid'_3 sid'_2 sid'_1
The receiving packets

Figure 1: Message encoding and decoding through pseudo packet IDs permutation.

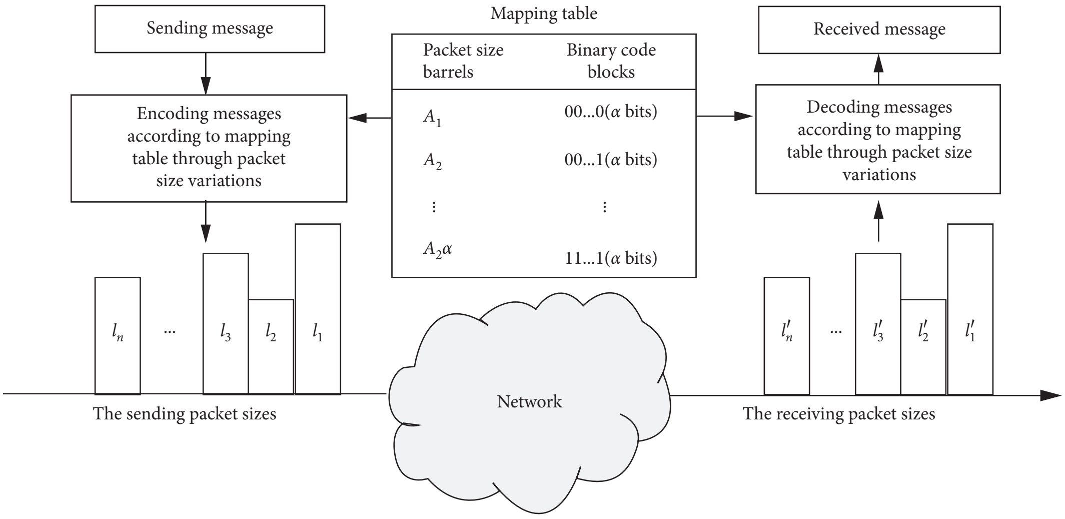
Sending message
Encoding messages according to mapping table through packet size variations
Mapping table
Packet size barrels
Binary code blocks
A_1 00...0(α bits)
A_2 00...1(α bits)
⋮ ⋮
$A_2\alpha$ 11...1(α bits)
Received message
Decoding messages according to mapping table through packet size variations
l_n ... l_3 l_2 l_1
The sending packet sizes
Network
l'_n ... l'_3 l'_2 l'_1
The receiving packet sizes

Figure 2: Message encoding and decoding through packet size variations.

(2) Encode K bits through packet sizes variation and $(L-K)$ bits through pseudo packet IDs permutation.

Algorithms 1 and 2 are messages encoding and decoding, respectively.

To better understand the approach in this subsection, we provide an example in here. Assume that s_i is equal to 00001011; α is equal to 1; the packet size variation satisfies $l_1 < l_3 < l_4 < l_5 < l < l_2$; and the set of the pseudo packet IDs is $\{1, 2, 3, 4\}$. According to equation (2), $n = 4$, $K = 0000$, and $L - K = 1011$. The first part K bits are encoded as the packet sending order as follows: 1^{st}, 3^{rd}, 4^{th}, 5^{th}, and the second part $L - K$ bits are encoded as the pseudo packet IDs permutation $\{2, 4, 3, 1\}$. Therefore, the pseudo packet IDs, viz., 2, 4, 3, 1, are appended to the sending packets. At the receiver, the packet sizes variation $l_1 < l_3 < l_4 < l_5 < l$ and the pseudo packet IDs permutation $\{2, 4, 3, 1\}$ can be retrieved. Thereafter, $K = 0000$ can be decoded by looking up the mapping table. Furthermore, $L - K = 1011$ can be decoded through Cantor Expansion. s_i is then successfully decoded as 00001011.

4. Provenance Transmission through 2dCTC

To transmit provenance through 2dCTC, a new provenance scheme 2dCTCP (2dCTC provenance scheme) is devised.

4.1. Provenance Encoding. In the continuous data flow environment of WSNs, it is assumed that the network topology is relatively stable, which is the basis for the provenance

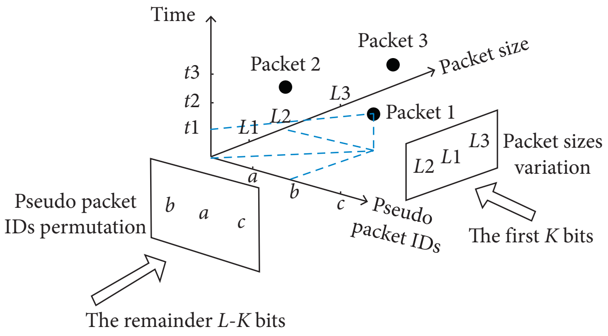

Figure 3: The working principle of 2dCTC.

```
Input: packets, message
Output: packet flow
packet size ⟵ $\{l_i | l_i \in R, i = 1, 2 \ldots n\}$
pseudo packet IDs ⟵ $\{sid_i | sid_n \in R, i = 1, 2 \ldots n\}$
message ⟵ $\{s_1, s_2, s_3, \ldots s_i, \ldots, s_N\}$
FOR each $s_i$ DO
  calculate the number of packet $n$
  $L$ ⟵ the number of bits in $s_i$
  $K$ ⟵ the first $\alpha n$ bits of message from $s_i$
  cache packets before sending
  $L - K$ ⟵ the number of remainder bits
  $D_{L-K}$ ⟵ the decimal number of the $L - K$ bits
  new packet sequence $(M_i)$ = encoding through packet sizes variation $(K)$
  send pseudo packet ID $(sid_1, sid_2, \ldots, sid_n)$ = uncantor$(n, D_{L-K}$, pseudo package IDs$)$
  FOR each packet $(M_n)$ DO
  append pseudo packet IDs $(sid_1, sid_2, \ldots, sid_n)$ into $(M_1, M_2, \ldots M_n)$
  END FOR
END FOR
send packets in a stream manner
```

Algorithm 1: 2dCTC encoding.

```
Input: packet flow
Output: message
assigne n packets into a group
FOR each group DO
  add pseudo packet IDs to generate permutation
  $D_{L-K}$←cantor (pseudo packet IDs permutation)
  $L - K$←the binary of the $L - K$ bits
  FOR each packet size DO
    $K$←looking up mapping table
    decode the first part $K$ bits
  END FOR
  decode the second part $L - K$ bits
END FOR
```

Algorithm 2: 2dCTC decoding.

transmission method based on 2dCTC proposed in this paper. 2dCTCP is a segmented scheme, which probabilistically incorporates the provenance at each node on the packet path into a series of packets provenance blocks.

In this paper, we consider a node-level provenance; i.e., the node IDs on the path the packet traversed are encoded as provenance. For the formal network model of the WSN we considered and provenance model, one can refer to [3–5].

The main steps of provenance transmission by 2dCTC are as follows.

(1) Set the hash value to group the provenance blocks.

In order to identify the packets that have the same provenance, we calculate the hash value for the packet path at each node through

$$H(n_i) = H(H(n_{i-1}) + n_i), \tag{3}$$

where n_i and $H(n_{i-1})$ denote i^{th} node's ID and the hash value on the $(i-1)^{\text{th}}$ node, respectively. Therefore, the packets that encoded the different part of the same provenance share the same hash value.

(2) Determine the number of packets needed to encode provenance.

Assume that the length of the maximum ID is L bits; the number of packets n then satisfies

$$\alpha n + \text{logn}! \geq L. \tag{4}$$

(3) Update the provenance.

If the random probability p_i generated at the current node is larger than the preset probability threshold P, the provenance and hash value will be updated; otherwise, only the hash value is updated.

(4) Encode the provenance to the sending packets.

Algorithm 3 shows provenance encoding through 2dCTCP.

```
Input: packets, pseudo packet IDs
Output: provenance
FOR each packet DO
    hash_value = hash(hash_value + $n_i$)
END FOR
choose n packets with the same hash value
IF $p_i < P$ THEN
    encode provenance with Algorithm 1
END IF
send packets in a stream manner
```

Algorithm 3: 2dCTCP provenance encoding.

4.2. Provenance Decoding. When the BS receives the packets, the main steps of provenance decoding are as follows:

(1) The BS classifies these packets according to the hash values and assigns n packets into a group

(2) In each group, the BS gets the packet sizes and decodes partial provenance through looking up the mapping table; thereafter, the BS retrieves the reminder provenance part according to the Cantor value formed by the pseudo packet IDs permutation

Algorithm 4 shows the provenance decoding through 2dCTCP. In the related works, the only known provenance transmission through CTC uses the IPDs based one-dimensional CTC [18], which was designed mainly to improve the concealment of provenance transmission. Compared to such a method, our 2dCTC provenance scheme can conserve more energy and channel capacity in WSNs.

5. Evaluation

5.1. 2dCTC Performance Analysis. The performance of 2dCTC is analysed and the corresponding experimental results are provided.

5.1.1. Channel Capacity. Note that n packets can represent (1) $n!$ bits through pseudo packet IDs permutation and (2) m^n bits through packet sizes variation, where m is the number of packet size differences. If L bits are encoded by n packets, L, n, and m should satisfy the following equation:

$$L = \log n! + n \log m. \tag{5}$$

As a result, the upper bound of the channel capacity is as follows:

$$C = \frac{L}{(n-1)T} = \frac{\log n! + n \log m}{(n-1)T}. \tag{6}$$

5.1.2. Channel Error Rate. The 2dCTC's channel error rate can be caused: (1) the noise that spoils the order of packets in transmission, e.g., packet transmission jitters and delays; (2) the noise that spoils the number of packets in transmission, i.e., packet loss, packets aggregation, packet division, and dummy packet padding.

In our previous work [21], the negative influence of those noises has been thoroughly discussed for one-dimensional CTCs. Here, we used part of the conclusions from [21] to derive 2dCTC's channel error rate.

As to the error rate caused by the packet transmission delays and jitters, the inter-packet delay T_r at the receiver can be calculated by

$$\begin{aligned} T_r &= t_{k+1} + T_d + j_{k+1} - (t_k + T_d + j_k) \\ &= T + (j_{k+1} - j_k) \\ &= T + j_k^{(1)}, \end{aligned} \tag{7}$$

where t_k and t_{k+1} denote the sending moments of the k^{th} and $(k+1)^{\text{th}}$ packets, respectively; T_d denotes the transmission expectation time; j_k and j_{k+1} denote the transmission jitters of the k^{th} and $(k+1)^{\text{th}}$ packets, respectively; and j_k and j_{k+1} are normal distribution random variables.

As a result, to keep the order of packets in transmission, $\Delta + j_k^{(1)} > 0$ must be satisfied. Since n packets in transmission form $n-1$ delays, the channel error rate is then as the following [22]:

$$P_e = 1 - \left[P_{\Delta + j_k^{(1)} > 0}\right]^{n-1} = 1 - \left[1 - \frac{1}{2} erfc\left(\frac{\Delta}{2\sigma}\right)\right]^{n-1}, \tag{8}$$

where $erfc(x) = 1 - erf(x) = (2/\sqrt{\pi}) \int_x^{+\infty} \exp(-y^2) d_y$.

To decrease the channel error rate caused by packet transmission jitters and delays, the interval between adjacent packets sending should be enlarged.

As to the channel error rate caused by packet loss, packets aggregation, packet division, and dummy packet

```
Input: packets, pseudo packet IDs, packet sizes
Output: provenance
FOR each packet DO
  IF packets have the same hash value THEN
    every n packets are assigned into a group
  END IF
  FOR each group DO
    get packet sizes and pseudo packet IDs
    $L - K$ = cantor (pseudo packet IDs)
    provenance_remainder = the binary of the $L - K$ bits
    provenance = message corresponding to packet sizes obtained from the mapping table
  END FOR
END FOR
```

Algorithm 4: 2dCTCP provenance decoding.

padding, without loss of generality, assuming λ denotes the probability of packet loss, μ denotes the probability of a packet aggregated with its following packet, v denotes the probability of a dummy packet insertion, and ω denotes the probability of a packet division. The expectation for the channel error rate under those kinds of noise is then

$$\varphi = 1 - (1 - \lambda)(1 - \mu)(1 - v)(1 - \omega). \tag{9}$$

The physical meaning of φ is that the probability of at least one of those kinds of noise has happened.

To mitigate the negative influence caused by packet loss, packets aggregation, packet division, and dummy packet padding, the redundant information should be added, i.e., sending the same message K times under a noisy 2dCTC, where $K \geq 1$ and $k \in \mathbb{N}^{+}$.

5.2. 2dCTC Experiments. In order to verify the correctness and effectiveness of 2dCTC, we used Python to implement the covert communication between two hosts. The IP addresses of the two hosts were 112.24.29.117 and 10.3.11.180, respectively, where TCP is used as the communication protocol. In the experiment, packets are generated through the Scapy library. A 400-byte text file is selected as the message. The intervals between packets are selected from 5 ms to 40 ms. We compare the total time consumption and capacity of 2dCTC with those of two one-dimensional CTCs, where the unit of capacity is Bps, i.e., the number of bytes transmitted in 1 s. The first one-dimensional CTC is packet rearrangement CTC, which uses different packet IDs permutation to represent the message. The other one-dimensional CTC is packet rearrangement CTC that applies the packet sizes variation to represent the message. Packet rearrangement CTC represents 8 bits by 6 packets, and the other packet rearrangement CTC uses each different packet size to represent 1 bit, viz., 8 packets bearing 8 bits. The 2dCTC uses 4 packets to represent 8 bits. The experimental results are shown in Figures 4(a) and 4(b), respectively, in which 2dCTC has the smallest time consumption and the higher channel capacity than those of the two one-dimensional CTCs.

5.3. 2dCTCP Simulations. We used TinyOS 2.1.2 TOSSIM as the simulator to evaluate the performance of the 2dCTCP scheme. The energy consumption is measured by POWERTOSSIMz [23]. We compared the performance of our scheme with those of segment based provenance schemes, i.e., Probabilistic Provenance Mark (PPM) scheme [17] and Probabilistic Provenance Flow (PPF) scheme [16]. The sensor network of 121 nodes with IDs 0 through 120 is deployed. The node with ID 0 is set as the BS. The maximum network diameter is 12, the communication protocol is CTP (Collection Tree Protocol) [24], and the data stream was generated by TinyOS through setting the packets sending interval.

5.3.1. Performance Metrics. The main performance metrics are as follows:

(A) Average Provenance Size (APS). The APS is defined as follows [4]:

$$\text{APS} = \frac{\sum_{i=1}^{m} PS_i}{m}, \tag{10}$$

where PS_i is the provenance length of the i^{th} packet and m is the total number of packets received by the BS.

(B) Total Energy Consumption (TEC). The TEC is defined as follows [4]:

$$\text{TEC} = \sum_{i=1}^{N} EC_{n_i}, \tag{11}$$

where EC_{n_i} is the energy consumed by the node n_i and N is the total number of nodes in the WSN.

5.3.2. Simulation Results. We simulated the PPM, PPF, and 2dCTCP schemes under the same simulation environment and the results are shown in Figures 5(a) and 5(b), respectively.

Figure 5(a) shows the APS for the PPM, PPF, and 2dCTCP schemes with respect to packet transmission hops. The APS in our scheme does not increase as the number of

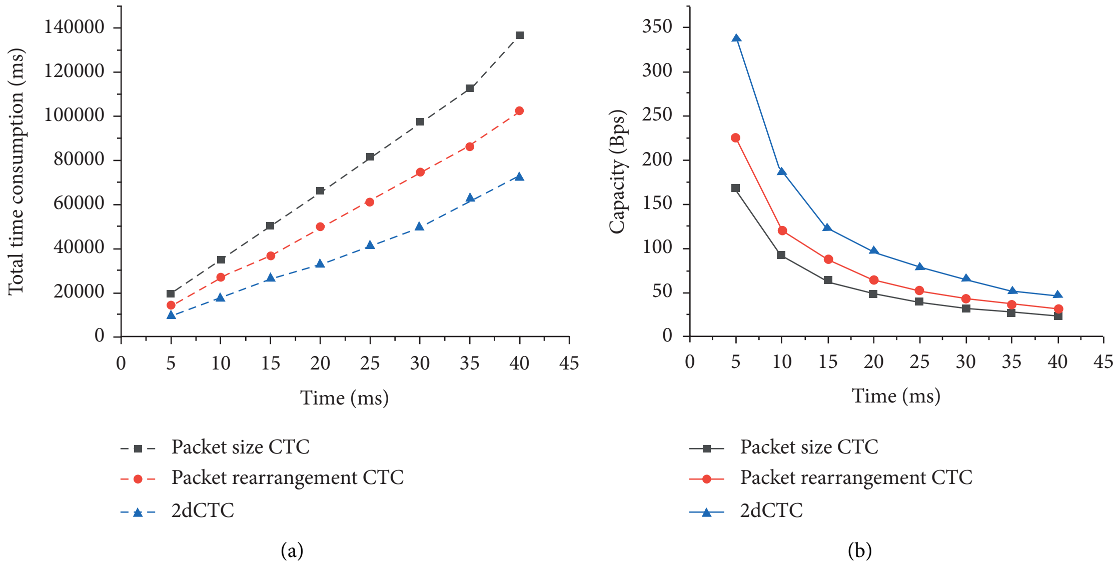

Figure 4: (a) Total time consumption for sending a 400-byte file with different time intervals. (b) Channel capacities with different time intervals.

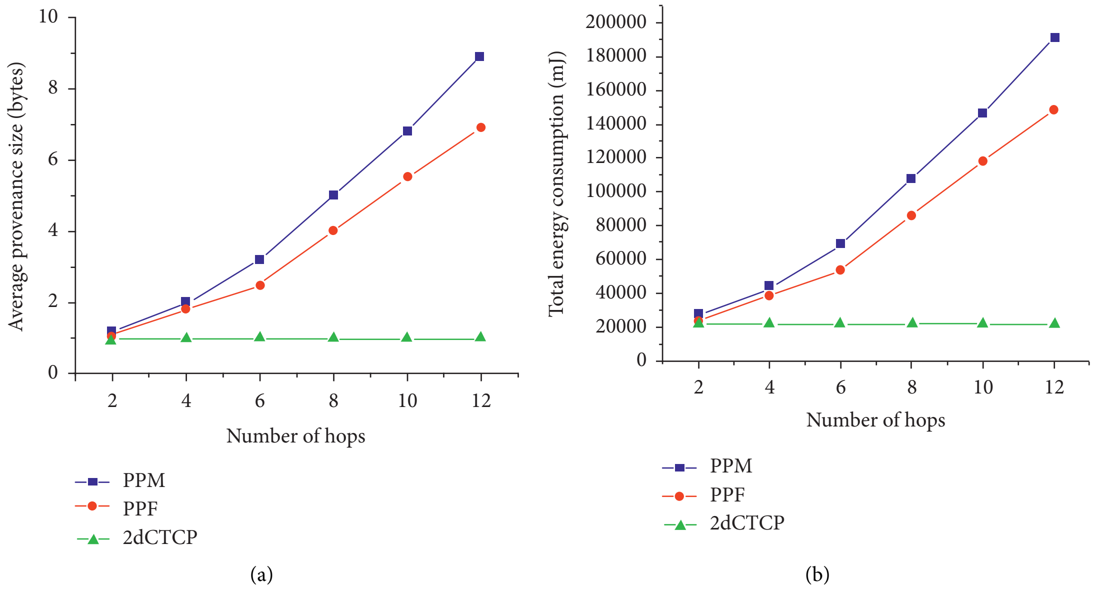

Figure 5: (a) The APS generated from a WSN with 121 nodes in which each source node generates about 360 packets. (b) The TEC generated from a WSN with 121 nodes in which each source node generates about 360 packets.

hops increases and remains constant at around 1 byte, whereas for PPM and PPF schemes, APS increases with the increases of packet transmission hops. In the 2dCTCP scheme, the provenances were encoded and transmitted in the timing channel but not in the packets. Although the packets are required to carry pseudo packet IDs, the size of packets is not expanded further according to the provenance's expansion. Hence, our scheme has much better performance than the PPM and PPF schemes with respect to provenance size.

Figure 5(b) shows the relationship between the number of packet transmission hops and TEC of the PPM, PPF, and 2dCTCP schemes. The trend of the curves in Figure 5(b) is closely consistent with that of the curves in Figure 5(a). As a result, under the same condition, the 2dCTCP scheme is more efficient than that of the PPM and PPF schemes regarding energy consumption.

6. Conclusion

In the paper, we propose 2dCTC, a two-dimensional CTC. By using both pseudo packet IDs permutation and packet sizes variation as the communication medium, 2dCTC can dramatically increase the channel capacity compared to the one-dimensional CTC. To ease the burden of provenance transmission, we apply 2dCTC to provenance transmission in resource constrained WSNs. We analysed the performance of the 2dCTC and validated the benefits of our method through experiments. The simulation results show that using 2dCTC for provenance transmission can conserve more energy than that of PPM and PPF, which further confirms the efficiencies of our method.

Data Availability

No data are associated with this study.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Qinbao Xu and Li Liu contributed equally to the paper.

Acknowledgments

This work was supported by the National Science Foundation of China under grant 61672269, the National Key Research Project under grant 2017YFB1400703, and the Jiangsu Provincial Science and Technology Projects under grant BK20180860.

References

[1] C. Wang, W. Zheng, and E. Bertino, "Provenance for wireless sensor networks: a survey," *Data Science and Engineering*, vol. 1, no. 3, pp. 189–200, 2016.

[2] S. Sultana, G. Ghinita, E. Bertino, and M. Shehab, "A lightweight secure scheme for detecting provenance forgery and packet DropAttacks in wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 3, pp. 256–269, 2015.

[3] S. R. Hussain, C. Wang, S. Sultana, and E. Bertino, "Secure data provenance compression using arithmetic coding in wireless sensor networks," in *Proceedings of the 2014 IEEE International Performance Computing and Communications Conference (IPCCC)*, pp. 1–10, Austin, TX, USA, December 2014.

[4] C. Wang and E. Bertino, "Sensor network provenance compressionusing dynamic bayesian networks," *ACM Transactions on Sensor Networks*, vol. 13, no. 1, pp. 1–32, 2017.

[5] C. Wang, S. R. Hussain, and E. Bertino, "Dictionary based secure provenance compression for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 405–418, 2016.

[6] B. Shebaro, S. Sultana, S. R. Gopavaram, and E. Bertino, "Demonstrating a lightweight data provenance for sensor networks," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, pp. 1022–1024, Raleigh, NC, USA, October 2012.

[7] S. Cabuk, "Network covert channels: design, analysis, detection, and elimination," Ph D. thesis, Purdue University, West Lafayette, Indiana, 2006.

[8] L. Yao, X. Zi, L. Pan, and J. Li, "A study of on/off timing channel based on packet delay distribution," *Computers & Security*, vol. 28, no. 8, pp. 785–794, 2009.

[9] S. Cabuk, C. E. Brodley, and C. Shields, "Ip covert timing channels: design and detection," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, pp. 178–187, Washington, DC, USA, October 2004.

[10] V. Berk, A. Giani, G. Cybenko, and N. Hanover, "Detection of covert channel encoding in network packet delays," Tech. Rep. TR536, Université de Dartmouth, Hanover, New Hampshire, 2005.

[11] S. Gianvecchio, H. Wang, D. Wijesekera, and S. Jajodia, "Model-based covert timing channels: automated modeling and evasion," in *Lecture Notes in Computer Science*, pp. 211–230, Springer, Berlin, Germany, 2008.

[12] A. El-Atawy, Q. Duan, and E. Al-Shaer, "A novel class of robust covert channels using out-of-order packets," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 2, pp. 116–129, 2017.

[13] X. Zhang, L. Zhu, X. Wang, C. Zhang, H. Zhu, and Y. Tan, "A packet-reordering covert channel over volte voice and video traffics," *Journal of Network and Computer Applications*, vol. 126, pp. 29–38, 2019.

[14] C. Liang, X. Wang, X. Zhang, Y. Zhang, K. Sharif, and Y. Tan, "A payload-dependent packet rearranging covert channel for mobile voip traffic," *Information Sciences*, vol. 465, pp. 162–173, 2018.

[15] C. Liang, Y. Tan, X. Zhang, X. Wang, J. Zheng, and Q. Zhang, "Building packet length covert channel over mobile voip traffics," *Journal of Network and Computer Applications*, vol. 118, pp. 144–153, 2018.

[16] S. M. I. Alam and S. Fahmy, "A practical approach for provenance transmission in wireless sensor networks," *Ad Hoc Networks*, vol. 16, pp. 28–45, 2014.

[17] M. T. Goodrich, "Probabilistic packetmarking for large-scale IP traceback," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 15–24, 2008.

[18] S. Sultana, M. Shehab, and E. Bertino, "Secure provenance transmission for streaming data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 8, pp. 1890–1903, August 2013.

[19] J. C. R. Bennett, C. Partridge, and N. Shectman, "Packet reordering is not pathological network behavior," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 789–798, 1999.

[20] C. S. Calude, L. Staiger, and K. Svozil, "Randomness relative to cantor expansions," *Communications in Nonlinear Science and Numerical Simulation*, vol. 10, no. 8, pp. 921–930, 2005.

[21] H. Jin and C. Wang, "Robustness of the packet delay channels," in *Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA*, pp. 260–267, Tianjin, China, August 2016.

[22] C. Wang, Y. Yuan, and L. Huang, "Base communication model of ip covert timing channels," *Frontiers of Computer Science*, vol. 10, no. 6, pp. 1130–1141, 2016.

[23] E. Perla, A. O. Cathain, R. S. Carbajo, M. Huggard, and C. M. Goldrick, "PowerTOSSIM z: realistic energy modelling for wireless sensor network environments," in *Proceedings of the 3rd ACM International Workshop on Performance Monitoring, Measurement, and Evaluation of Heterogeneous Wireless and Wired Networks (PM2HW2N'08)*, pp. 35–42, Vancouver, Canada, October 2008.

[24] O. Gnawali, R. Fonseca, K. Jamieson, M. Kazandjieva, D. Moss, and P. Levis, "CTP: an efficient, robust, and reliable collection tree protocol for wireless sensor networks," *ACM ransactions on Sensor Networks*, vol. 10, no. 1, pp. 1–49, 2013.