sensors

MDPI

Article

Attack Detection Using Network Coding in IoT Environment

Yong Lee [1,*] **and Goo Yeon Lee** [2,*]

[1] Independent Researcher, Chuncheon 24341, Korea
[2] Department of Computer and Communications Engineering, Kangwon National University, Chuncheon 24341, Korea
* Correspondence: yleehyun@gmail.com (Y.L.); leegyeon@kangwon.ac.kr (G.Y.L.)

Received: 9 January 2020; Accepted: 18 February 2020; Published: 21 February 2020

Abstract: Network coding is a reasonable way to increase network efficiency in response to an increase of sensed data in the Internet of Things (IoT). In network coding, intermediate nodes combine packets received from neighboring nodes, transform, and transmit encoded packets that can be decoded at the destination. This scheme is based on trust among nodes. If any malicious node joins the network, it can act as an intermediate node that could fabricate encoded packets. It might be more difficult to identify the authenticity of such encoded packets since packets that are received at the destination might not originate from a single source, but be combined with several other packets originating from multiple sources. In this paper, we propose a scheme on how to detect attacked packets among the received packets at a destination and how to recover the original message from the packets including the attacked *"look-like-valid"* packets. This scheme shows that a destination could recover the valid message with just the received packets including some attacked packets and will result in a quite efficient performance in network coding.

Keywords: Internet of Things; network coding; attack detection; recovery

1. Introduction

With the provision of many applications based on Internet of Things (IoT), the amount of data processing in the network is rapidly increasing due to the increase in sensing information. Therefore, network coding is a good solution for improving network throughput and efficiency [1–3]. In network coding, an intermediate node combines packets that are received from neighboring nodes, then transforms them into encoded packets that can be decoded at destination [4,5]. This approach is particularly suited for IoT environments because it is primarily used in multi-hop multipath network architectures from source to sink to provide robustness and error tolerance for the network. In the IoT, wearable devices and sensor nodes collect, transmit, and relay data. Encoded packets are a combination of packets received from multiple sources, so that quantitatively combined information is transmitted as a result [1,3–5]. This method of mixing information and increasing the amount of information has the advantage of substantially increasing the transmission efficiency and allowing the network to be flexible in communication, hardware, or relay errors [1,3]. This scheme can only operate correctly if the network topology can be configured with trusted nodes. If a malicious node legitimately participates in the network configuration, it is possible to insert a forged encoded packet as an intermediate node. Since the packet received at the destination is not from one source but is a combination of packets from multiple sources, it is not easy to recognize whether the received packet is valid. Therefore, a network architecture using network coding has a high risk of information forgery by structurally malicious nodes.

In the IoT environment, the joining and identification of trusted nodes is a controversial topic, especially in the case of an autonomous network configuration by mobile nodes. Threats can be caused not only by the invading malicious nodes, but also by internal nodes that legally participate in the network configuration. In the case of threats by legitimate internal nodes, the attack will come from a trusted node, making it more difficult to determine the identity of the attacker. Cryptographic algorithms, such as digital signatures and encryption, can be applied to network coding, but if a malicious node that legitimately joins the network configuration deliberately manipulates, forges, and performs an internal attack with a legitimate digital signature or encryption, the other nodes will not be aware of it.

When the destination trusts the received packets, it performs a message recovery mechanism with those packets. Recovery by valid packets and recovery by packets disguised as valid packets by an internal attack will generate different messages. Let us consider an example. Assume that a node generates a message that is decomposed into b packets, transforms them into n packet combinations with redundancy, $m(n = b + m)$. If all n encoded packets would arrive at a destination but any intermediate node performs an internal attack, these packets all appear to be valid, but may include *"look-like-valid"* but actually *"attacked"* packets. Since packets can have a valid digital signature or encryption form and the destination cannot verify whether the packet is valid by digital signature verification or decoding, it is an important issue how to authenticate the packets received at the destination.

In this paper, we propose a valid message identification method of network coding considering the IoT environment where IoT devices can be added freely. This method detects the presence of an attacked packet among the packets received by the destination and makes it possible to determine a valid message when packets recover several different messages due to attacked but *"look-like-valid"* packets. Therefore, the burden of retransmission due to the attack can be reduced. This scheme shows that with a high probability, the destination can recover a valid message with received packets without requiring retransmissions, which will give quite throughput improvement.

The paper is organized as follows. We consider the previous work of network coding security in Section 2. We describe the system model and operations of our scheme in Section 3. Section 4 investigates characteristics of the model and the detailed algorithm. Section 5 shows performance analysis and the results. We conclude in Section 6.

2. Related Work

This section discusses work related to the security of network coding in IoT environment. Types of attacks that can occur in network coding include the Byzantine attack, impersonations, and pollution attacks [6]. To date, a lot of work has been researched to prevent these attacks or to detect them in network coding security.

One of the areas that have been most focused on is applying a cryptographic algorithm and a digital signature scheme to identify pollution packets. Peralta et al. proposed a homomorphic cryptography model for network coding to enhance end-to-end security, such as ensuring the confidentiality of data in the Internet of Things [1]. Boneh et al. have proposed a homomorphic signature scheme which could prevent any attacks by arbitrary nodes in network, and they have insisted that the destination could use this signature scheme to filter out corrupted packets, and even intermediate nodes could discard corrupted packets with computational overhead [4].

Yu et al. proposed an XOR network coding security scheme that can filter pollution attacks in a few hops using probabilistic key pre-distribution and message authentication codes (MACs) [7]. Shafagh et al. noted that the security of data is important when service providers and users access the cloud because the cloud is used to store data collected by an IoT application. To solve this problem, they proposed and evaluated a data sharing algorithm that applies homogeneous encryption algorithms as a data protection platform in the cloud [8].

Li et al. applied a network coding signature scheme to guarantee the authenticity of data according to the feature that it is difficult to share the key in the IoT where it is easy to add nodes, and proposed a signature method for data collected by various devices using their own authentication key [9]. In addition, Wu et al. [10] proposed a method that verifies whether a packet is valid by applying a digital signature algorithm in case a network coding is used in a vehicular area network (VANET) and a vehicle cannot recover a message due to a pollution attack. Cheng et al. showed that multi-generation pollution attacks are possible when homogeneous subspace signature schemes are used to cope with pollution attacks in network coding and described an algorithm that solves this problem by improving the key distribution method [11].

Chen et al. proposed a method of applying error detection and error correction techniques to encoded packets. This paper showed that the throughput can be improved by allowing the intermediate nodes to correct the packet [12]. Mamidwar et al. analyzed studies to prevent the rapid spread of pollution attacks from network coding throughout the network. These studies consisted of error correction, localization of malicious nodes, and pollution packet detection [13]. Ayday et al. have worked a security service called a location aware network coding scheme that provide data authenticity through node collaboration and data redundancy in any environment where nodes are dense enough such that an event can be sensed by multiple nodes [14].

Wang et al. proposed a trust scheme of defining reliable nodes' behavior in order to prevent pollution attacks from spreading rapidly to the network [15]. Ji et al. proposed a distributed detection algorithm to prevent wormhole attack by malicious nodes using expected transmission counts in the wireless network coding systems [16].

In the IoT, when a lot of data is stored in a data collection center, network coding may be applied to the data for retrieval efficiency or security of information. Oliveira et al. applied network coding to the storage of data and optimized the added redundancy to ensure reliable data storage and to retrieve more packets at minimal cost [17]. Lei et al. showed that the network coding can be efficiently applied between data producers and consumers to handle large amounts of data transmission in the named data networking (NDN) model to provide IoT applications [18].

Cebe et al. proposed a method of applying network coding to transmit sensing data using blockchain technology in the IoT environment. In the proposed scheme, blockchain technology has a large amount of computation and blocks, and the overhead of a long time delay is involved, whereas network coding technology transmits data packets by combining them, which helps to solve the problem of blockchain by increasing throughput [19].

Lima et al. have explained the security vulnerabilities of network coding and compared the differences between attack scenarios in network coding combined with classical cryptography [6].

Dong et al. have described the framework with a focus on network coding systems designed for wireless mesh networks, identified potential security vulnerabilities that could seriously degrade system performance, and defined security goals and challenges [20].

Zhao et al. have studied network coding using real-world BitTorrent measurements called NCTorrent on a wireless body area network (WBAN) for reliable medical data transfer, in which data are transferred via relays from multiple wireless body sensors to the monitoring stations. The study concluded with a pessimistic conclusion that network coding might not be beneficial for real-world BitTorrent systems [21].

3. System Model and Descriptions

In this paper, we do not consider transmission errors. However, assuming errors in transmission, our scheme also applies in the case that error processing methods are applied. That means packets with some errors will be discarded at the destination by error detection methods which means no arrivals, and error corrected packets by error correction methods are considered as normal arrivals at the destination. In network system, a node transmits packets transformed by applying random linear combination to its received packets as random network coding. In this case, a message means data of any size generated by a source, and a packet means a fragment of a message divided into fixed size length for transmission. Figure 1 shows an example in which information sensed by an IoT device in IoT network is delivered to the IoT server through network coding.

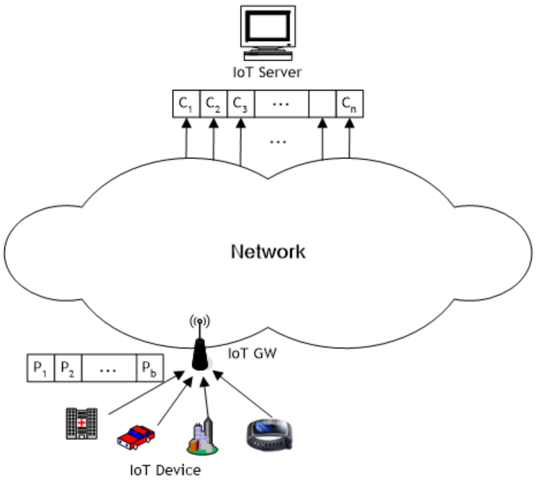

Figure 1. An example of network coding in an Internet of Things (IoT) network.

3.1. Basic Concepts

3.1.1. Operations at Source

A source decomposes a message to be transmitted into b packets. We define these packets as $P_i, i = 1, 2, \cdots, b$. The source transforms these packets into the encoded packets, $C_j, j = 1, 2, \cdots, n$, applying linear combination and transmits them to the network [22,23].

$$C_j = \sum_{i=1}^{b} r_{ji} P_i \tag{1}$$

From Equation (1), r_{ji} are composed of operations with addition and multiplication for Galois field, $GF(2^q)$ as randomly chosen coefficients. Encoding vector, $\overrightarrow{r_j} = (r_{j1}, r_{j2}, \cdots, r_{jb})$ are embedded in the header of packets C_j and this header is used for the packet reconstruction at destination [3,22,23].

3.1.2. Message Recovery at Destination

When a destination receives the encoded packets (namely, the combinations), it reconstructs them using the coding coefficient. As each encoding packet is represented with a linear equation of original packets, b, the destination can perform the decoding mechanism to the received packets using a linear equation and recover the original message from the packets. At this time, the b packets which are used in each recovery are subset of the n packets which are generated by the source and linearly independent [22,23].

$$\begin{pmatrix} r_{11} & r_{12} & \dots & r_{1b} \\ r_{21} & r_{22} & \dots & r_{2b} \\ & \dots\dots\dots & & \\ r_{n1} & r_{n2} & \dots & r_{nb} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_b \end{pmatrix} = \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{pmatrix} \tag{2}$$

From Equation (2), C_j are the encoding packets that are received at any destination node and the corresponding encoding vectors are $\overrightarrow{r_j} = (r_{j1}, r_{j2}, \cdots, r_{jb})$.

3.2. Attack Example

Let us assume an attack example in the model described in the previous section.

Example 1. *Assume that a source performs mod* 4 *operation in* $GF = \{0, 1, 2, 3\}$ *and generates packets. The source decomposes a message into two packets,* P_1,P_2, *calculates a linear equation to them with redundancies,* r_{ji}. *Then it generates and transmits the combinations,* C_1,C_2,C_3,C_4 *as follows.*

$$\begin{pmatrix} r_{11}P_1 + r_{12}P_2 = C_1 \\ r_{21}P_1 + r_{22}P_2 = C_2 \\ r_{31}P_1 + r_{32}P_2 = C_3 \\ r_{41}P_1 + r_{42}P_2 = C_4 \end{pmatrix} \tag{3}$$

In transmission, packet C_4 is attacked and modified to C_e as in Equation (4).

$$r_1^e P_1^* + r_2^e P_2^* = C_e. \tag{4}$$

Since this attacked packet, C_e seems to be a *"look-like-valid"* with a valid digital signature and a valid encrypted value, no node could recognize it until they arrived at the destination and performed the recovery operation. We assume that there are no errors due to transmission errors in the network and the destination can receive all the packets that are transmitted from the source. Eventually the destination receives all packets, C_1,C_2,C_3,C_e as follows.

$$\begin{pmatrix} r_{11}P_1 + r_{12}P_2 = C_1 \\ r_{21}P_1 + r_{22}P_2 = C_2 \\ r_{31}P_1 + r_{32}P_2 = C_3 \\ r_1^e P_1^* + r_2^e P_2^* = C_e \end{pmatrix} \tag{5}$$

When the destination performs the recovery operation to four packets of Equation (5), it can get six reconstruction results, as follows.

$$\begin{pmatrix} r_{11}P_1 + r_{12}P_2 = C_1 \\ r_{21}P_1 + r_{22}P_2 = C_2 \end{pmatrix} \begin{pmatrix} r_{11}P_1 + r_{12}P_2 = C_1 \\ r_{31}P_1 + r_{32}P_2 = C_3 \end{pmatrix} \begin{pmatrix} r_{21}P_1 + r_{22}P_2 = C_2 \\ r_{31}P_1 + r_{32}P_2 = C_3 \end{pmatrix} \Rightarrow (P_1, P_2)$$

$$\begin{pmatrix} r_{11}P_1 + r_{12}P_2 = C_1 \\ r_1^e P_1^* + r_2^e P_2^* = C_e \end{pmatrix} \Rightarrow (P_1', P_2')$$

$$\begin{pmatrix} r_{21}P_1 + r_{22}P_2 = C_2 \\ r_1^e P_1^* + r_2^e P_2^* = C_e \end{pmatrix} \Rightarrow (P_1'', P_2'')$$

$$\begin{pmatrix} r_{31}P_1 + r_{32}P_2 = C_3 \\ r_1^e P_1^* + r_2^e P_2^* = C_e \end{pmatrix} \Rightarrow (P_1''', P_2''')$$

If the four packets are all valid, these six reconstructions must show all identical results. As C_4 is forged, the reconstructions including C_e may show the incorrect results. When all reconstruction results are not identical, it means that there are errors among the received packets. However, it is still unknown which and how many packets are polluted.

Except for the reconstructed attacked packet, C_e, the other reconstructions are generated from all the correct packets and produce an identical solution. These packets can be grouped together. If all reconstruction results are not identical, we can recognize the existence of erroneous packets. If there is one attacked packet, as in Example 1, the reconstructions generated from the remaining valid packets all have the identical solution, and if the identical solution is the majority, a valid message can be found. If the reconstruction solutions include the attacked packet, C_e, it will fortunately give a different value, it is easy to apply majority rule. Although some reconstruction results including attacked packets show the identical result, majority rules can be applied if the number of these reconstructions is less than the reconstruction performed only with the correct packets. From the above example, if $(P'_1, P'_2) \neq (P''_1, P''_2) \neq (P'''_1, P'''_2) \neq (P_1, P_2)$, it is possible to determine (P_1, P_2) that is calculated from each (C_1, C_2), (C_1, C_3), and (C_2, C_3) combinations by applying the majority rule. However, when the number of attacked packets is much more or all of the reconstructions including attacked packets might result in one identical value, they obstruct that the destination could identify the correct reconstruction solution by applying the majority rule.

In this paper, we propose an algorithm that detects the presence of the attacked packets among the packets received at the destination and determine the valid message despite the presence of the attacked packets.

4. Attack Detection and Correction Algorithm Using Network Coding

In IoT architecture with network coding, the destination may not be able to reconstruct the original message correctly when any pollution exists in packets due to attacks. Sometimes destination may not be aware of the existence of attacks and can misinterpret it by reconstructing the wrong message. If the destination can detect the existence of an attack among the received packets, it could make it possible to recover the original message, even if there exists any attacked packet. Such a scheme could reduce the possibilities of retransmission due to attack and contribute to improve network efficiency.

4.1. Notations and Assumptions

We have some assumptions for the proposed algorithm.

- At source, a message is decomposed into b plain packets for transmission. Then they are transformed into n encoded packets with m redundancies using network coding.
- Any b packets out of the encoded n packets are required to recover the original message at destination.
- We assume that each packet is independently transferred with other packets.
- A malicious node can forge packets that appear to be *"look-like-valid"* combinations and send these *"look-like-valid"* packets instead of correct packets. We call it an attacked packet.
- It is assumed that there is no transmission error in the network, and all the packets transmitted by the source node are received by the destination node.

We have the following notations in this paper.

- b : Number of plain packets which are decomposed from a message.
- n : Number of combination packets which are encoded with the plain packets and redundancies. $n = b + m$.
- e : Number of attacked packets among the n combination packets.
- r : Number of non-attacked, correct packets, $n = e + r$.
- $(P_1, P_2, ..., P_b)$: Valid original packets which are decomposed from the original message.
- C_i : Combination packets which are encoded from valid original packets and redundancies.
- C^e_j : Erroneous combination packets which are fabricated by any attack.
- $coeff$: Encoding coefficient.
- $\{C_1, C_2, \ldots, C_i, i > b\}$: a group.
- $size\{\cdot\}$: group size, that is number of packets in a group $\{\cdot\}$.
- $expectedR\{\cdot\}$: Expected number of reconstructions that a group of size i can recover.
- $actualR\{\cdot\}$: Actual number of reconstructions that a group of size i makes an identical result.

4.2. Characteristics of Encoded Packets

4.2.1. Group

If a destination decodes the receiving packets including any attacked packets to recover the original message, more than one reconstruction result may exist. (If all received packets are valid, there will be one valid reconstruction result.) We can classify the received packets by the reconstruction solution. We classify the reconstruction according to the result, and the packets involved in generating a result are called groups. The condition of being a group is that the number of packets in the group must be greater than b. That is $size\{group\} > b$.

Example 2. *Assume that we have encoded packets C_1, C_2, C_3, and C_4^e, C_5^e, when $b = 2$, $m = 3$. Here, C_4^e and C_5^e are attacked packets and destination could not recognize it. Let us assume that the reconstructions show the following ten results.*

$$coeff^{-1}\begin{pmatrix}C_1\\C_2\end{pmatrix} = (P_1, P_2), \quad coeff^{-1}\begin{pmatrix}C_2\\C_3\end{pmatrix} = (P_1, P_2), \quad coeff^{-1}\begin{pmatrix}C_1\\C_3\end{pmatrix} = (P_1, P_2)$$

$$coeff^{-1}\begin{pmatrix}C_1\\C_4^e\end{pmatrix} = (P_1^e, P_2^e), \quad coeff^{-1}\begin{pmatrix}C_2\\C_4^e\end{pmatrix} = (P_1^e, P_2^e),$$

$$coeff^{-1}\begin{pmatrix}C_1\\C_5^e\end{pmatrix} = (P_1^e, P_2^e), \quad coeff^{-1}\begin{pmatrix}C_2\\C_5^e\end{pmatrix} = (P_1^e, P_2^e)$$

$$coeff^{-1}\begin{pmatrix}C_3\\C_4^e\end{pmatrix} = (P_1^{e'}, P_2^{e'}), \quad coeff^{-1}\begin{pmatrix}C_3\\C_5^e\end{pmatrix} = (P_1^{e''}, P_2^{e''}), \quad coeff^{-1}\begin{pmatrix}C_4^e\\C_5^e\end{pmatrix} = (P_1^{e'''}, P_2^{e'''})$$

From the above results, three reconstructions generated from (C_1, C_2), (C_2, C_3), and (C_1, C_3) give one identical solution, (P_1, P_2). So C_1, C_2, and C_3 form one group, and the group size is $size\{C_1, C_2, C_3\} = 3$. Four reconstructions generated from (C_1, C_4^e), (C_2, C_4^e), (C_1, C_5^e), and (C_2, C_5^e) also give one identical solution, (P_1^e, P_2^e) and C_1, C_2, C_4^e and C_5^e form another group, and $size\{C_1, C_2, C_4^e, C_5^e\}$ is 4. The remaining (C_3, C_4^e), (C_3, C_5^e), and (C_4^e, C_5^e) have solutions $(P_1^{e'}, P_2^{e'})$,$(P_1^{e''}, P_2^{e''})$, and $(P_1^{e'''}, P_2^{e'''})$, respectively. However C_3 and C_4^e cannot form a group because they result in only one reconstruction and $size\{C_3, C_4^e\} = 2 \not> b$. The same applies to (C_3, C_5^e) and (C_4^e, C_5^e).

Suppose that n combinations are composed of r valid packets and e attacked packets when they arrive at a destination, $n = m + b = r + e$. Let the combinations that a destination receives be $\{C_1, \ldots, C_r\}\{C_{(r+1)}^e, \ldots, C_{r+e}^e\}$. The total number of possible reconstructions which are made by the received packets is $\binom{n}{b}$. The number of reconstructions which are made by only valid packets is $\binom{r}{b}$ when $r \geq b$. The number of reconstructions which contain at least one attacked packet, C_i^e, is $\binom{n}{b} - \binom{r}{b}$.

If each reconstruction which contains at least one attacked packet generates different solutions and $(C_1, \ldots, C_r)$ becomes a only group, one identical result that is recovered by the group is the only valid message. If two or more reconstructions which contain at least one attacked packet generate one identical solution, the packets which are included in these reconstructions also make a group. When we have two or more groups, we should identify a correct solution among several groups. From the above example, we should identify whether the correct solution is the first group (C_1, C_2, C_3) or the second group (C_1, C_2, C_4^e, C_5^e).

4.2.2. Consistency

In the previous section, a group is defined as encoded packets included in the reconstruction to recover the identical solution. This section describes how to configure attack detection and determine a valid message using group. Let x be the total number of encoded packets in one group and y be the number of the packets to be required to recover the original message. $\binom{x}{y}$ is the total number of reconstructions that x packets can generate.

When the actual number of reconstructions included in a group is identical as $\binom{x}{y}$, we says that the group has consistency. The expected number of reconstructions that a group of size x can generate is $\binom{x}{y}$, let it be $expectedR\{\}$. Let $actualR\{\}$ be the actual number of reconstructions that a group of size

x makes an identical result. If $expectedR\{\}$ of a group is equal to $actualR\{\}$ of this group, the group has consistency. The conditions of consistency are shown in Algorithm 1.

Algorithm 1: The conditions of consistency.

```
if $size\{\cdot\} > b$
    $\{\cdot\}$ is group
if $expectedR\{\cdot\} == actualR\{\cdot\}$
    group $\{\cdot\}$ has consistency
```

From Example 2, the first group, $\{C_1, C_2, C_3\}$ has $size\{C_1, C_2, C_3\} = 3$. This group has consistency because $expectedR\{C_1, C_2, C_3\} = \binom{3}{2} = 3$ and $actualR\{C_1, C_2, C_3\}$ is 3 since there are three cases of (C_1, C_2),(C_2, C_3) and (C_1, C_3) producing the identical solution. The size of second group, $size\{C_1, C_2, C_4^e, C_5^e\}$ is 4, so $expectedR\{C_1, C_2, C_4^e, C_5^e\} = \binom{4}{2} = 6$. However, $actualR\{C_1, C_2, C_4^e, C_5^e\}$ is 4 because there are 4 cases of (C_1, C_4^e), (C_2, C_4^e), (C_1, C_5^e), and (C_2, C_5^e) producing the identical solution. Thus, this group does not have consistency since $expectedR\{C_1, C_2, C_4^e, C_5^e\} \neq actualR\{C_1, C_2, C_4^e, C_5^e\}$.

Assume that a group has r valid packets and $r > b$. This group has consistency because they all generate an identical result through $\binom{r}{b}$ reconstructions. Assume that a group has attacked packets and $e > b$. If the actual number of reconstruction in this group is $\binom{e}{b}$ and they make one identical solution, this group has also consistency.

Suppose that a group, G has r' valid packets and e' attacked packets. Let $r' \geq b$ and $e' \geq 1$. $size\{G\} = r' + e'$, so $expectedR\{G\} = \binom{r'+e'}{b}$. $expectedR\{G\} = actualR\{G\}$ must be satisfied for this group to be consistent. However, since $r' > b$, the reconstructions produced by these valid packets are already included in $\binom{r}{b}$ and the $actualR\{G\}$ is $\binom{r'+e'}{b} - \binom{r'}{b}$. We have $actualR\{G\} < expectedR\{G\}$, so this group cannot have consistency.

Consider that a group has r' valid packets and e attacked packets, and $r' \leq b - 1$ and $r' + e > b$. This group can be consistent if all the reconstructions generated by the $r' + e$ encoded packets makes the identical solution.

Example 3. *Suppose that a destination receives encoded packets with $b = 3$, $m = 5$, $n = 8$, $r = 5$, and $e = 3$, and the existence of attack packets is unknown. The encoded packets can be represented with $\{C_1, C_2, C_3, C_4, C_5\}\{C_6^e, C_7^e, C_8^e\}$. Since $\{C_1, C_2, C_3, C_4, C_5\}$ are valid packets, we have $\binom{5}{3} = 10$ reconstructions that generate an identical solution and the packets form a group, g_1 with $size\{C_1, C_2, C_3, C_4, C_5\} = 5$. $expectedR\{C_1, C_2, C_3, C_4, C_5\} = actualR\{C_1, C_2, C_3, C_4, C_5\} = 10$, so this group g_1 has consistency.*

Assume that another group, g_2 has valid packets with $r' = 3$ and an attacked packet with $e' = 1$, and include $\{C_1, C_2, C_3\}\{C_6^e\}$. Since $size\{C_1, C_2, C_3, C_6^e\} = 4$, $expectedR\{C_1, C_2, C_3, C_6^e\}$ is $\binom{4}{3} = 4$. However, the reconstruction calculated by $\{C_1, C_2, C_3\}$ generates the correct solution, which is included in group, g_1. Therefore the group g_2 has $actualR\{C_1, C_2, C_3, C_6^e\} = \binom{4}{3} - 1 = 3$ and cannot have consistency.

Suppose the third group has $r' = 2$ valid packets and e attacked packets, and $r' + e$ encoded packets are $\{C_1, C_2\}\{C_6^e, C_7^e, C_8^e\}$. Suppose that reconstructions of these encoded packets generate one identical solution. $size\{C_1, C_2, C_6^e, C_7^e, C_8^e\} = 5$, so $expectedR\{C_1, C_2, C_6^e, C_7^e, C_8^e\}$ is $\binom{5}{3} = 10$. The $actualR\{C_1, C_2, C_6^e, C_7^e, C_8^e\}$ of this group is also 10 and this group has consistency because $\{C_1, C_2\}$ packets cannot generate a solution.

4.3. Algorithm

With group and consistency characteristics of encoded packets, we obtain the following Algorithm 2 for detecting attacks and identifying a valid message at the destination.

Algorithm 2: Attack detection and correction algorithm at destination.

```
For all $n$ received encoding packets
    Calculate all $\binom{n}{b}$ reconstructions
For all $\binom{n}{b}$ reconstructions
    Classify and group identical reconstruction result
For all group $i$
    if $size\{group\ i\} > b$, set group $i$ to true
For all $group\ i == true$
    if $actualR\{group\ i\} == expectedR\{group\ i\}$
        set group $i$ to consistency
For all groups with consistency
    Find the group with the largest size
```

Let us consider the detailed algorithm. First let us discuss the conditions under which valid reconstruction exists. When the reconstruction forms a consistent group, the message generated by this group are candidate for valid reconstruction. When $e < m$, we have $r > b$ and $\binom{r}{b} > 1$. These r valid packets participate in two or more reconstructions to generate the identical solution and form a consistent group. Therefore the condition that there exists a valid solution is $e < m$. Even if the other reconstructions, $\binom{n}{b} - \binom{r}{b}$ form a group with one identical solution, the group will not be consistent. This is because the size of this group is n and $expectedR\{this\ group\} = \binom{n}{b} > actualR\{this\ group\} = \binom{n}{b} - \binom{r}{b}$.

Example 4. *Suppose that encoded packets with b = 2, m = 2, and n = 4 are transmitted and a destination receives these packets with an attacked packet with e = 1. This example is the case of $m > e$, so the condition that a valid reconstruction exists is satisfied. Since there are valid packets with $r = 3$, the destination can get $\binom{r}{b}$ = 3 reconstructions, which is both $actualR$ and $expectedR$ of this group, and verify the consistency of this group. The other reconstructions are $\binom{4}{2} - \binom{3}{2}$ = 3. Even if we get an identical result, $size\{C_1, C_2, C_3, C_4^e\}$ = 4 and $expectedR\{C_1, C_2, C_3, C_4^e\}$ = 6 makes it inconsistent. Thus, we can find the valid solution group.*

If the destination receives packets with $e = 2$ and $r = 2$, only one valid reconstruction can be obtained through $r = 2$, so it is not possible to determine if it is a valid message. In this case, the condition of $m > e$ is not satisfied.

Next, consider the conditions under which majority rule can be applied to find a group that generates a valid message when there are multiple consistent groups. We have two conditions as follows.

1. Consider the case where $r > e\,(r \geq b, e \geq b)$. Two groups g_r with $group\ size\ r$ and g_e with $group\ size\ e$ can be formed, and the $expectedR$ of each group becomes $\binom{r}{b}$ and $\binom{e}{b}$. We have $expected\{g_r\} > expected\{g_e\}$, so it is possible to apply majority rule to identify the valid message.
2. Suppose that a group g_r' consists of $\{C_1, \ldots, C_{r'}\}\{C_1^e, \ldots, C_e^e\}$ packets, and $r' \leq b - 1$ and $r' + e > b$. Assume that the reconstructions of $r' + e$ encoded packets generate one identical result. Since $r' \leq b - 1$, r' encoded packets could not generate a solution. If $r' + e < r$, $expected\{g_r'\} = \binom{r'+e}{b} < \binom{r}{b}$, and this group g_r' is not subject to majority rule and the group g_r can be selected by majority rule. For this group to be consistent, the maximum value of r' would be $b - 1$. Now let be $r' = b - 1$. With $r' + e < r$ and $n = b + m = r + e$, we have $e \leq \lfloor \frac{m}{2} \rfloor$. When $e \leq \lfloor \frac{m}{2} \rfloor$, even if this group with attacked packets has one identical solution and it is consistent, $expectedR\{g_r'\} < expectedR\{g_r\}$, therefore the group g_r' is dropped by majority rule.

Example 5. *A destination that receives encoded packets with $b = 3$, $m = 5$, $n = 8$, $e = 4$, and $r = 4$ are represented with $\{C_1, C_2, C_3, C_4\}\{C_5^e, C_6^e, C_7^e, C_8^e\}$. Here we have $m > e$, $r = e$, and $e > \lfloor \frac{m}{2} \rfloor$, these satisfy the condition that a valid recovery exists, but do not satisfy the conditions (1) and (2) to find a valid recovery. Let us consider the detail. From valid packet group $\{C_1, C_2, C_3, C_4\}$ with $size\{C_1, C_2, C_3, C_4\} = 4$, we have $expectedR\{C_1, C_2, C_3, C_4\} = actualR\{C_1, C_2, C_3, C_4\} = \binom{r}{b} = 4$.*

If four attacked packets generate one identical solution, group $\{C_5^e, C_6^e, C_7^e, C_8^e\}$ *with* $size\{C_5^e, C_6^e, C_7^e, C_8^e\} = 4$, *would have* $actualR\{C_5^e, C_6^e, C_7^e, C_8^e\} = 4$ *which is the same as* $expectedR\{C_5^e, C_6^e, C_7^e, C_8^e\} = \binom{e}{b} = 4$ *and has consistency. The r valid packet group is consistent, and the e attacked packet group is also consistent. Then we will not be able to identify the correct solution.*

From the above considerations, we have conditions, $e < m$, $r > e$, and $e \leq \lfloor \frac{m}{2} \rfloor$. Finally, the conditions for identifying a valid solution is $r > e$ and $e \leq \lfloor \frac{m}{2} \rfloor$. In the first case of Example 4, since $e < m$, $r > e$, and $e \leq \lfloor \frac{m}{2} \rfloor$ with $b = 2$, $m = 2$, $r = 3$, and $e = 1$, we can see that the condition to find a valid message is satisfied.

5. Performance Analysis and Results

5.1. Performance Analysis

Suppose that the probability that an encoded packet is attacked in transmission is p. The probability that the number of attacked packets e is equal to i is as follows.

$$P(e = i) = \binom{n}{i} p^i (1-p)^{n-i}. \tag{6}$$

Let us consider all the conditions to find a valid message. The probability that $e \leq \lfloor \frac{m}{2} \rfloor$ is,

$$P(e \leq \lfloor \tfrac{m}{2} \rfloor) = \sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} \binom{n}{i} p^i (1-p)^{n-i}. \tag{7}$$

We have the probability that $e < r$,

$$P(e < r) = \sum_{j=0}^{r-1} \binom{n}{j} p^j (1-p)^{n-j}. \tag{8}$$

The probability to find a valid message P_g is equal to the probability that $e \leq \lfloor \frac{m}{2} \rfloor$ and $e < r$, and can be expressed as shown in Equation (9),

$$P_g = \sum_{k=0}^{min(\lfloor \frac{m}{2} \rfloor, r-1)} \binom{n}{k} p^k (1-p)^{n-k}. \tag{9}$$

Here, we define a cost function related to the transmission proportional to the number of transmitted packets. Assume that the cost of transmitting one packet is C_{tx}. If the message is successfully recovered at the destination with probability of P_g, the cost is $n \cdot C_{tx}$. We have $1 - P_g$ probability that a message cannot be successfully recovered at its destination, in which case the message will be re-transmitted until it is successfully recovered. Therefore, the average number of attempts to deliver a message to its destination is calculated as $\frac{1}{P_g}$, and the total cost can be expressed as

$$\begin{aligned} Total\ Cost &= \frac{1}{P_g} \cdot n \cdot C_{tx} \\ &= \frac{n \cdot C_{tx}}{\sum_{k=0}^{min(\lfloor \frac{m}{2} \rfloor, r-1)} \binom{n}{k} p^k (1-p)^{n-k}}. \end{aligned} \tag{10}$$

5.2. Results and Discussion

Figure 2 shows that as the probability of a packet being attacked increases, the probability of finding a valid message decreases. Also, as the redundancy increases, the probability P_g also increases. Looking at the example of $b = 4$, $m = 4$ in the graph, it can be seen that according to

Equation (9), the value of e becomes the minimum value between $\lfloor \frac{m}{2} \rfloor$ and maximum value of $r-1$ with $e < r$, so that even if two attacked packets occur, the recovery result can be found with high probability. Here we see that the probability of identifying a valid message, P_g when redundancy m is even, is higher than when m is odd. The reason is as follows. Let m_1 be an even denoted by $m_1 = 2q,\ q = 1, 2, \cdots$. We have $\lfloor \frac{m_1}{2} \rfloor = \lfloor \frac{2q}{2} \rfloor = q$, and $n_1 = b + 2q$. Now, let m_2 be an odd denoted by $m_2 = 2q+1,\ q = 1, 2, \cdots$. We have $\lfloor \frac{m_2}{2} \rfloor = \lfloor \frac{2q+1}{2} \rfloor = q$, and $n_2 = b + 2q + 1$. Then, $n_1 < n_2$, but $\lfloor \frac{m_1}{2} \rfloor = \lfloor \frac{m_2}{2} \rfloor = q$. In Equation (9), k has the identical value in both n_1 and n_2 even though $n_2 > n_1$. Then we have $(1-p)^{(n_1-q)} > (1-p)^{(n_2-q)}$. Therefore, the probability P_g shows lower when $m = 2q+1,\ q = 1, 2, \cdots$. However, since $(1-p) \simeq 1$ when p is small enough, the probability P_g is not affected by this characteristic of redundancy.

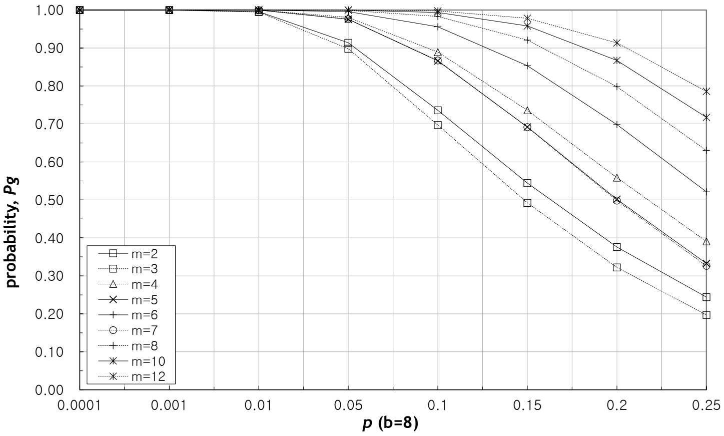

Figure 2. The probability of finding a valid solution as the probability p increases when $b = 8$.

We look into the probability P_g for the case where m is even in Figures 3 and 4. Figures 3 and 4 show the change of the probability P_g as the redundancy m increases when $b = 4$ and $b = 8$. In these figures, the greater the redundancy, the higher the probability P_g. The probability P_g does not have been affected by p as redundancy gets larger. Hence the probability P_g is stabilized at $m = 8$.

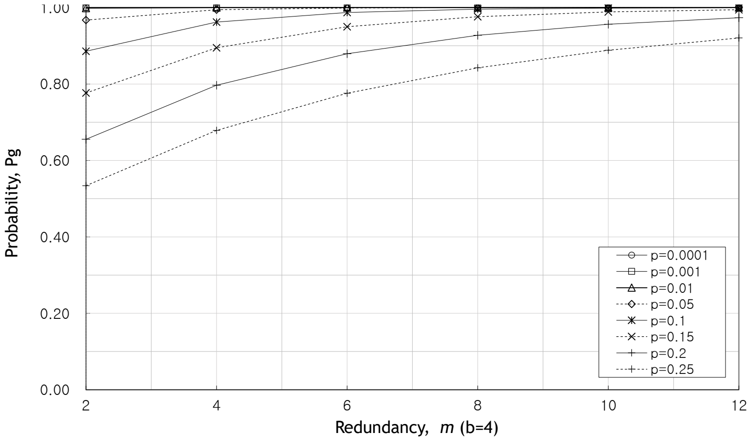

Figure 3. P_g as the redundancy, m, increases when $b = 4$.

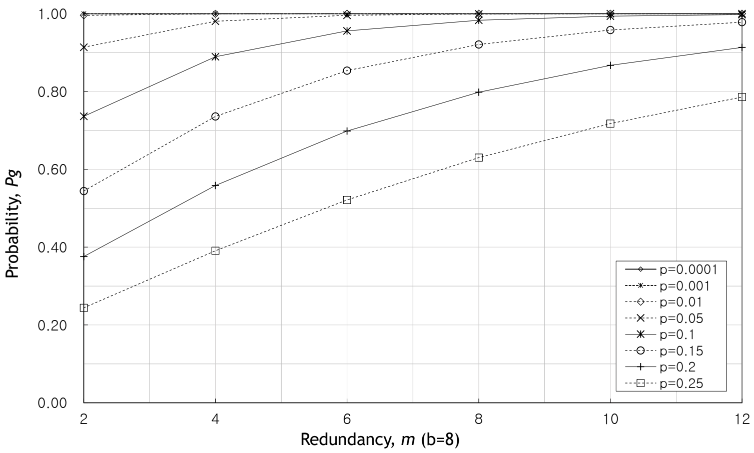

Figure 4. P_g as the redundancy, m, increases when $b = 8$.

Figure 5 shows the change in probability P_g with increasing p when $m = 6$. We can see that the probability P_g remains almost one until $p = 0.05$ regardless of the value of b. It can be seen that when the number of source packets, b is larger, P_g decreases rapidly due to the increase in p. Figures 2–5, fewer packets and greater redundancies give higher probability of obtaining a valid message at the destination.

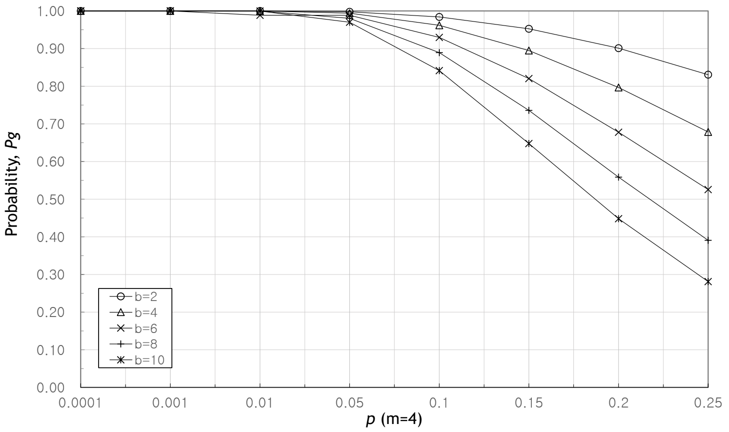

Figure 5. P_g as the number of packets increases when $m = 4$.

Figure 6 shows the change in the probability P_g of recovering a valid message as b increases when 2, 3, and 4 attacked packets occur, respectively. It can be seen that m must be increased to cope with the high p value and the increase of the attacked packet e. Despite the high probability of $p = 0.2$ at $b = 8$, $P_g = 0.558$ at $m = 4$, and $P_g = 0.798$ at $m = 8$, and we have a much higher recovery probability P_g at $m = 8$.

Figure 7 shows the probability P_g with the change of b and m when fixing n. Even with a high probability of $p = 0.2$, if b maintains more than half of n, it can be seen that a recovery probability of 0.8 or more can be obtained. In the case of $b = 10$, when $m = 6$ and $p = 0.1$, a higher recovery probability is obtained in spite of higher p than when $n = 12$ and $p = 0.05$.

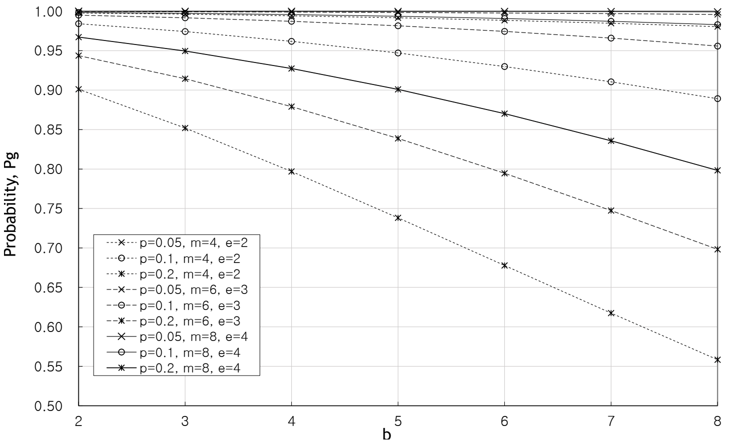

Figure 6. P_g as b increases when $e = 2, 3, 4$.

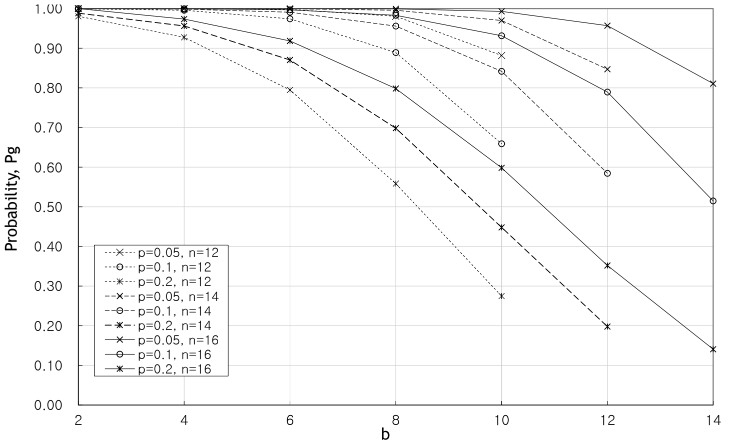

Figure 7. P_g with the change of b and m when fixing n.

Figure 8 shows the total cost as the probability p increases when $b = 8$. Until $p = 0.1$, it can be seen that the total cost remains stable regardless of the value of m. Increasing the value of m increases the total number of packets n, thus affecting the transmission cost, while increasing the probability P_g so that the total cost can be stabilized.

Figure 9 show the total cost as the number of original packets b and the probability p increase when $m = 4$. This graph also shows that until $p = 0.1$, the total cost remains stable regardless of the value of b. From Figures 8 and 9, we can see that when $p \geq 0.1$, the larger m and the larger b, the larger P_g, so the total cost can be reduced.

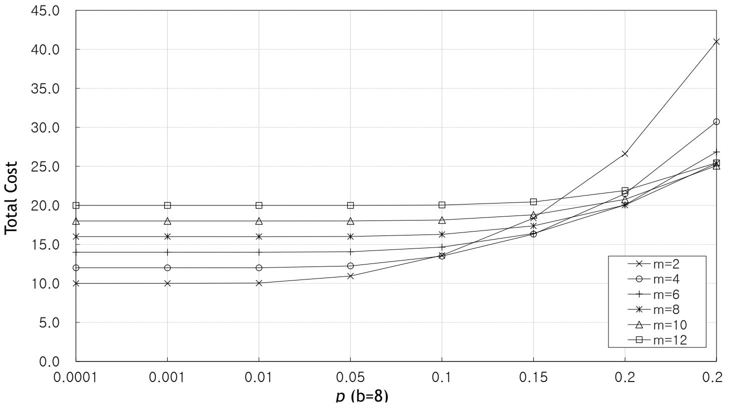

Figure 8. Total cost as the probability p increases when $b = 4$.

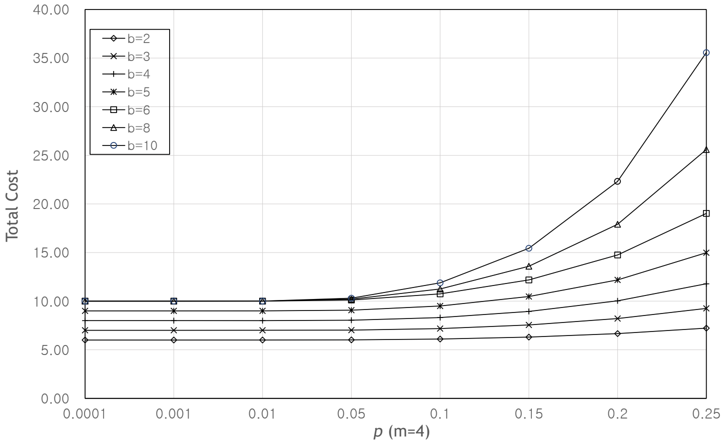

Figure 9. Total cost as the probability p increases when $m = 4$.

The study to ensure the security of network coding discussed in chapter 2 includes the method of applying cryptographic techniques such as encryption algorithms and digital signatures in [1,4,8–11], the method of secure network coding proposed in [7,9,17], and the prevention of attacks by distinguishing the trusted nodes in [13,15]. In the method of applying a cryptographic technique, there is overhead such as key exchange, and if an internal attacker modifies the packet and encrypts it with a correct key, the destination decrypts it and trusts the packet and executes the recovery for network coding. In this process, the destination cannot recognize the forgery of the packet. Secure network coding is also aimed at preventing the occurrence of polluted packets and cannot cope if polluted packets occur. In the method based on trust, internal attacks by trusted nodes cannot be prevented. While the existing methods focus on preventing the occurrence of attacked packets, the proposed method can be a solution that can detect the presence of attack and recover messages even if a packet is attacked by an internal attack.

6. Conclusions

Network coding has contributed to improving the throughput of a network in IoT environment. In the network coding in which the intermediate node of the routing combines and encodes the packets received from the neighboring nodes, and transmits the combinations, it is inevitable that a combination of forged or corrupted packets by the malicious node occurs. It also allows for the presence of *"look-like-valid"* attacked packets that appear to have valid signatures and valid encryption by malicious nodes that hide their identity.

We focused on the case where a destination does not recognize the existence of this kind of attacked packets. When the destination recovers a message with packets containing the attacked packets, all recovery results may not match and the destination will get the wrong result. This paper has proposed an algorithm that detects the presence of attacked packets among received packets and can identify a valid message even if attacked packets exist. We also analyzed the conditions under which the proposed algorithm can operate. This algorithm has shown that even with a high probability of attack, a valid message can be identified with a high probability. The results showed a high recovery probability when the number of redundancies is about half of the total packets even under high attack probability. Even if attacks occur, the message can be recovered without re-transmitting the packet, which shows that the recovery cost gradually increases even with the increase of redundancy or attack.

Author Contributions: Conceptualization, Y.L. and G.Y.L.; methodology, Y.L.; formal analysis, Y.L.; investigation, G.Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L. and G.Y.L.; project administration, G.Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (No: 2018-0-00261, GDPR Compliant Personally Identifiable Information Management Technology for IoT Environment).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Peralta, G.; C-Fuentes, R.G.; Bilbao, J; Crespo, P.M. Homomorphic Encryption and Network Coding in IoT Architectures: Advantages and Future Challenges. *Electronics* **2019**, *8*, 827. [CrossRef]
2. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [CrossRef]
3. Li, J.; Liu, Y.; Zhang, Z.; Ren, J.; Zhao, N. Towards Green IoT Networking: Performance Optimization of Network Coding Based Communication and Reliable Storage. *IEEE Access* **2017**, *5*, 8780–8791. [CrossRef]
4. Boneh, D.; Freeman, D.; Waters, B.; Katz, J. Signing a Linear Subspace: Signatures for Network Coding. In *International Workshop on Public Key Cryptography*; Springer, Berlin/Heidelberg, Germany, 2009.
5. Ahlswede, R.; Cai, N.; Li, S.-Y.R.; Yeung, W. Network Information Flow. *IEEE Trans. Inf. Theory* **2000**, *46*, 1204–1216. [CrossRef]
6. Lima, L.; Vilela, J.P.; Oliveira, P.F.; Barros, J. Network Coding Security : Attacks and Countermeasures. *arXiv* **2008**, arXiv:0809.1366.
7. Yu, Z.; Wei, Y.; Ramkumar, B.; Guan, Y. An Efficient Scheme for Securing XOR Network Coding against Pollution Attacks. In Proceedings of the IEEE INFOCOM 2009, Rio de Janeiro, Brazil, 19–25 April 2009.
8. Shafagh, H.; Hithnawi, A.; Burkhalter, L.; Fischli, P.; Duquennoy, S. Secure Sharing of Partially Homomorphic Encrypted IoT Data. In Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems ACM SenSys'17, Delft, The Netherlands, 5–8 November 2017.
9. Li, T.; Chen, W.; Tang, Y.; Yan, H. A Homomorphic Network Coding Signature Scheme for Multiple Sources and its Application in IoT. *Secur. Commun. Netw.* **2018**, *2018*, 6. [CrossRef]
10. Wu, G.; Wang, J.; Wang, Y.; Yao, L. Pollution Attack Resistance Dissemination in VANETs Based on Network Coding. *Procedia Comput. Sci.* **2016** *83* 131–138. [CrossRef]
11. Cheng, C.; Lee, J.; Jiang, T.; Takagi, T. Security Analysis and Improvements on Two Homomorphic Authentication Schemes for Network Coding. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 993–1002. [CrossRef]
12. Li, J.; Li, T.; Ren, J.; Chao, H. Enjoy the Benefit of Network Coding: Combat Pollution Attacks in 5G Multihop Networks. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 13. [CrossRef]
13. Mamidwar, N.V.; Gothawal, D. Schemes against Pollution Attack in Network Coding: A Survey. *Int. J. Comput. Sci. Inf. Technol.* **2015**, *6*, 5085–5089.
14. Ayday, E.; Delgosha, F.; Fekri. F. Location-Aware Security Services for Wireless Sensor Networks using Network Coding. In Proceedings of the 26th IEEE International Conference on Computer Communications, Barcelona, Spain, 6–12 May 2007.
15. Cheng, W.; Yu, L.; Wang, W. Trusted Network Coding in Wireless Ad Hoc Networks. In Proceedings of the 2010 IEEE Global Telecommunications Conference (GLOBECOM), Miami, FL, USA, 6–10 December 2010.
16. Ji, S.; Chen, T.; Zhong, S. Wormhole Attack Detection Algorithms in Wireless Network Coding Systems. *IEEE Trans. Mob. Comput.* **2015**, *14*, 660–674. [CrossRef]
17. Oliveira, C.H.S.; G-Doudane, Y. ; Brito, C.E.F.; Lohier, S. Optimal Network Coding-Based In-Network Data Storage and Data Retrieval for IoT /WSNs. In Proceedings of the IEEE 14th International Symposium on Network Computing and Applications (NCA2015), Cambridge, MA, USA, 28–30 September 2015.
18. Lei, K.; Zhong, S.; Zhu, F.; Xu, K.; Zhang, H. An NDN IoT Content Distribution Model With Network Coding Enhanced Forwarding Strategy for 5G. *IEEE Trans. Ind. Inform.* **2018**, *14*, 2725–2735. [CrossRef]
19. Cebe, M.; Kaplan, B.; Akkaya, K. A Network Coding based Information Spreading Approach for Permissioned Blockchain in IoT Settings. In Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Service (Mobiquitous'18), New York, NY, USA, 5–7 November 2018; pp. 470–475.
20. Dong, J.; Curtmolab, R.; Nita-Rotarua, C. Secure Network Coding for Wireless Mesh Networks: Threats, Challenges, and Directions. In *4th Workshop on Secure Network Protocols 2008(NPSec 2008)*; Elsevier: Amsterdam, The Netherlands, 2008; pp. 33–38.

21. Wang, S.; Zhao, J.; Wang, X. Is Network Coding Helpful for BitTorrent: From a Practitioner's Perspective. In Proceedings of the 19th International Conference on Computer Communications and Networks (ICCCN), Zurich, Switzerland, 2–5 August 2010, pp. 1–6.
22. Rabin, M. Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance. *J. Assoc. Comput. Mach.* **1989**, *36*, 335–348. [CrossRef]
23. Barros, J. Mixing Packets: Pros and Cons of Network Coding. In Proceedings of the 11th International Symposium on (WPMC'08), Lapland, Finland, 8–11 September 2008.