SCOPE: A LIGHTWEIGHT CRYPTOGRAPHIC APPROACH FOR PRIVATE COPE DATA CODING

Ngoc Hong Tran[1], Cao Vien Phung[2], Binh Quoc Nguyen[1], Leila Bahri[3] and Dung Hai Dinh[1]

[1]Vietnamese German University, Vietnam
[2]Technische Universität Carolo-Wilhelmina zu Braunschweig, Germany
[3]KTH, Sweden

ABSTRACT

In this paper, we investigate a simple but effective coding mechanism, namely Coding Opportunistically (COPE) [1], and the privacy violations which are likely to happen to COPE. The COPE data, from the source node through the network to its destination node, can be easily learned by the surrounding nodes, particularly, intersecting nodes and neighbour nodes. This leads to a serious consequence in leaking the node identity and its private data. In order to cope with the mentioned issues, we can apply cryptographic schemes. However, the security solutions often exploit the public key schemes which nowadays run more slowly and give the longer encrypted values as the effects of the key size increase to guarantee the algorithm complexity. Hence, while the coding mechanism aims to decrease the bandwidth consumption by aggregating (i.e., coding) multiple packets in the network, the security solutions increase the data quantity. However, a necessary needs of combining data coding mechanism and cryptographic algorithm is raised for both preserving privacy and optimizing bandwidth use at the same time as mentioned above. In this paper we thus propose SCOPE, a lightweight privacy preserving approach able to support nodes running the COPE protocol in a secret way, by adopting the Elliptic Curve Cryptography (ECC) homomorphic encryption algorithm. The proposal's effectiveness and efficiency are proved through a variety of experiments.

KEYWORDS

Network Coding, Peer-to-Peer, Homomorphic Encryption, Elliptic Curve Cryptography (ECC).

1. INTRODUCTION

In order to optimize the network performance, particularly reducing the bandwidth consumption in a wireless network, there exist several techniques, such as network load balancing [2], routing optimization [2] [3] [5]. Another approach is the *network coding* paradigm that aims at increasing the performance in the network. In network coding, intermediate nodes, in a transmission network, can combine multiple native packets from different sources using simple operations, such as the Exclusive-OR operator, into a coded packet and then broadcast the coded packet in a single transmission instead of simply forwarding each native packet one by one. Therefore, the number of transmissions is reduced, and the network capacity is increased.

One of the first practical network coding techniques proposed as an effective forwarding architecture for wireless networks is the Coding Opportunistically (COPE) model [1]. COPE uses a two-hop coding pattern in order to identify the packets that can be coded together. Even though COPE is simple and effective, there still exist some short comings that lead to security and privacy issues. COPE nodes can overhear packets from their neighbours. That is, each node can read any content of packets transmitted in its radio range. Moreover, the intermediate nodes which is in charge of coding the packets are mostly likely to misuse the retrieved information for the malicious purposes. As an example, see the sample network represented in Figure 2. We can see that N_1 can listen to and get packets which N_2 and N_4 send to N_5. If N_1 is a malicious node, it can replay data from N_2 and N_4 for the other transactions, or modify the data. Therefore, *there is a need for protecting packets against transmitting nodes that are able to listen to transmissions from their neighbours, both in terms of confidentiality as well as of integrity.* In addition to this, COPE relies on opportunistic listening by which transmitting nodes can be aware of the packet queue lists of their neighbours to increase the chances of finding packets that can be coded. This introduces more privacy issues when considering intruders or malicious transmission nodes.

In order to address the privacy issues related to COPE, we propose in this paper a secure COPE (i.e., SCOPE) protocol that exploits cryptographic techniques to address the privacy issues related to the plain COPE. More precisely, and not to affect the desired goal of increasing the network capacity aimed from COPE, we adopt the additive homomorphic Elliptic Curve Cryptography (ECC) [12] that, in addition to providing the needed security requirements, is also lightweight and fits our considered scenarios for secure and still good performing network coding using COPE. Our experimental results demonstrate that the performance cost of SCOPE is negligible.

The rest of this paper is organized as follows: in Section 2 we review the related literature and position our suggested work. In Section 3 we provide background information on the required concepts related to both COPE and to ECC encryption. In Section 4 we formalize our security model, and in Sections 5 and 6 we present the SCOPE architecture and protocol, respectively. In Section 7 we provide the experimental evaluation of SCOPE. The security property will be expressed in Section 8. We finally conclude the paper in Section 9.

2. RELATED WORKS

Today, coding opportunity is one of the hot topics in network coding technique. As such, after COPE, several other interesting research works focused on designing network coding conditions that can increase the coding opportunities, hence better improve the network capacity. For instance, we find BEND [6] (BEND is not an acronym) with non-intersecting two-hop flows and DCAR (Distributed Coding-Aware Routing in Wireless Networks) [7] with intersecting more than two-hop flows. DODE (Distributed Opportunistic and Diffused Coding in Multihop Wireless Networks) [8] has combined the advantages of COPE, BEND and DCAR to still increase the coding chances. DODEX (Distributed and Diffused Encoding with Multiple Decoders) [9] with multiple encoder has been developed from DODE. On the other hand, Re-encoding of a coded packet is agreed in DODEX+ (Distributed and Diffused Encoding with Multiple Encoders and Multiple Decoders) [13] to detect more codable flows. Besides that, some applications of linear programming to optimize the throughput of COPE [5] and BEND [3] to create more opportunities for coding and provide a better total network throughput. These issues have been addressed by considering the best paths which not only have more coding chances but also avoid wireless interference among network nodes in the network.

With the richness of the literature in terms of network coding schemes that aim at improving the coding opportunity rate, COPE, as well as most of its improved successors, suffer from privacy and security related issues.One of the first works providing an analysis to secure network coding is in [14]. In this work, an eavesdropper able to see information in a single source scenario has been considered. Subsequently, in order to allow for secure network coding different models have been proposed [15]. The early works started by proposing a *wiretap* model, where the main idea consists at collecting subsets of nodes in a network coding system in wiretap networks such that each wiretapper has access to only one of these subsets [16].After that, the focus has shifted to addressing network coding security and privacy issues using different cryptographic schemes. The challenge, however, is to ensure the security and privacy of a network coding protocol (such as COPE), without much sacrificing the initial goal of increasing network capacity. It is indeed well known that cryptographic schemes do mostly come with huge costs both in terms of size and processing time.

Of the works available, we find the model proposed in [17], where the authors focus on the privacy preservation issue in terms of preventing traffic analysis and tracing in multi-hope wireless networks. The authors deploy the Paillier [22] homomorphic encryption scheme, which is based in the usage of large primes. The related consequence is on performance as operations done on large prime numbers is quite costly.

In [18], the authors have focused on the shortcomings of the Homomorphic Message Authentication Code (H-MAC) in terms of its vulnerability to pollution attacks, and in the context of being used to secure communications in transmission networks. The authors have proposed an improvement to minimize both data pollution and tag pollution attacks related to H-MAC, by introducing two types of tags, one related to verifying the integrity of the packet and the other related to checking the integrity of the H-MAC itself. In their analysis, they have showed that their method not only decreases the probability of tag pollution but it also results in decreasing the related bandwidth overhead. However, the overhead remains considerable. Many other works, such as [20] [21], have also studied the usage of H-MAC based techniques to secure network coding; however, the overhead remains quite high and the problem of data and tag pollutions often remain an issue. Moreover, we can also find the work in [19] where the authors have identified a flaw in H-MAC in general and have provided a corresponding inaccuracy in its formal security proof. This keeps it to doubt whether H-MAC based solutions are worthwhile in practice or not.

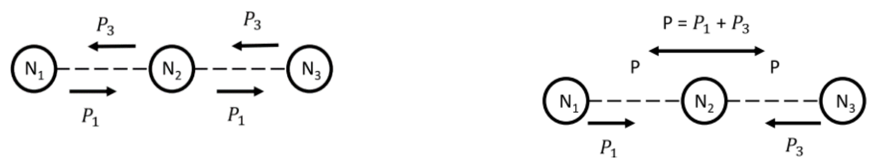
Differently from the available works in cryptographic secure network coding schemes, we proposed in this paper applying the lightweight ECC scheme to address the security and privacy issues in COPE.

3. BACKGROUND

COPE is a forwarding architecture for wireless mesh network. It is quite simple and effective in reducing the quantity of transmissions. Whereas, encryption algorithms are strong enough to secure data as well as to preserve the privacy against the curious attackers. We introduce herewith COPE and additive homomorphic encryption in the sense of highlighting their combination strength in tackling the security requirements mentioned in Sections 1 and 4.2.

3.1. COPE - Coding Opportunistically

Let us briefly describe COPE protocol. COPE has been the first practical network coding mechanism applied in wireless networks, aiming at reducing a significant quantity of transmissions. Thus the wireless throughput consumption can be optimized by coding several packets into one are forwarding them through a single transmission.



a) Without COPE. b) With COPE.

Figure1. Simple data transmission model for three nodes.

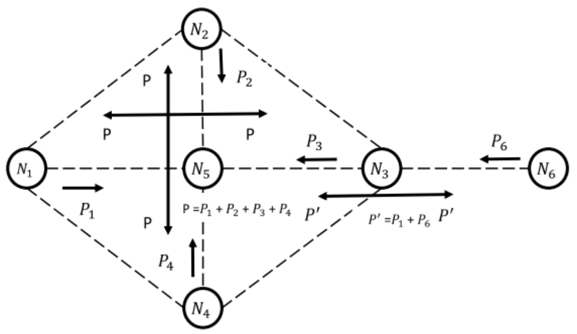


Figure 2. An example of network with COPE

3.1.1. COPE Protocol.Initially, with the standard COPE (see Figure 1), only 2-hop flows (i.e., including three nodes connected and involved in a transaction) are considered. Let consider 2-hop flows $N_1 \rightarrow N_2 \rightarrow N_3$ and $N_3 \rightarrow N_2 \rightarrow N_1$ in Figure 1. Two nodes N_1 and N_3 want to make a packet transfer to each other. But they locate out of the radio ranges of each other. Consequently, they cannot directly communicate. However, both of them can send their packets through node N_2 which places in the both of radio ranges. Then, N_2 becomes an intermediate node in charge of relaying packets between N_1 and N_3. Let us see Figure 1a which describes a protocol not applied with COPE. Let P_1, P_3 be packets sent, respectively, from N_1 to N_3 and vice versa. Hence, there are four unicast transmissions (i.e., P_1:$N_1 \rightarrow N_2$, P_3:$N_3 \rightarrow N_2$, P_3:$N_2 \rightarrow N_1$, P_1:$N_2 \rightarrow N_3$) in this protocol. In case of COPE protocol (see Figure 1b), the packets from N_1 and N_3 are considered as strings of bits, and aggregated by N_2, using the operator Exclusive--OR denoted as '+', to produce a coded packet $P=P_1+P_3$. Hence, N_2 is called ***encoder***, whereas N_1 and N_3 are ***decoders***. As a result, the number of transmissions drops down to three, including two unicast transmissions (i.e., P_1: $N_1 \rightarrow N_3$, P_3:$N_3 \rightarrow N_1$) and 1 broadcast transmission (i.e., P:$N_2 \rightarrow \{N_1, N_3\}$).

Example 1. .Let us give an example by considering Figure 2. A link connecting two nodes, depicted as a dash line, indicates that they are in radio ranges of each other, therefore they are so-called neighbours. Initially, nodes N_1, N_2,N_3 and N_4 send packets P_1, P_2,P_3 and P_4, respectively, to nodes N_6, N_4, $N_1$and N_2, via node N_5. N_5 has to wait to adequately get four native packets P_1, P_2, P_3, P_4, respectively, from N_1, N_2, N_3 and N_4. N_5 then aggregates them to a coded packet $P = P_1 + P_2 + P_3 + P_4$ and broadcasts P to N_1, N_2, N_3 and N_4 in a single transmission. Nodes N_1, N_2, N_3 and N_4 can decode to obtain its expected packet. They can do this, since each of nodes can catch the packets from its neighbours by overhearing their signals in the air. For example, N_3 can overhear P_2 and P_4 from N_2 and N_4. By XOR-ing P with its generated-and-sent packet P_3 and the overheard packets P_2 and P_4, N_3 obtains $P_1 = P + P_2 + P_3 + P_4$. Similarly, N_6 wants to send packet P_6 to N_5 via N_3. N_3 receives P_6, then, creates the aggregation $P' = P_1+P_6$, then broadcasts P' to its neighbours, that is, N_5 and N_6. N_6 receives and decodes P' to retrieve its expected packet $P_1=P'+P_6$. Similarly, N_5 decodes and retrieves its expected original packet P_6.

3.1.2. COPE header. A COPE header is inserted into the header of a packet, placed between the MAC and IP headers. The COPE header has a structure, as follows. A COPE header includes three blocks, that is, coding report, reception report, and ACK report. Coding report contains information of the XOR-ed native packets and their next hop. Reception report contains information of overlearned packets from neighbours including the source, the received last packet from that source, and a bitmap presenting the list of recently received packets from that source. ACK report contains the information of received or missed packets which the sending node has, including a neighbour IP, the last ACKed packet from that neighbour and a bit map of ACKed packet.

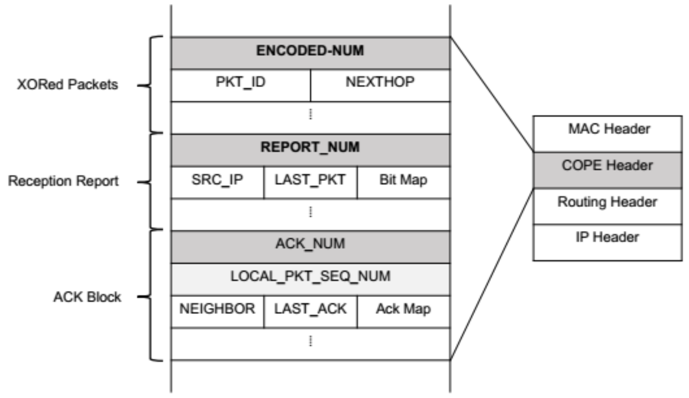


Figure 3. COPE Header Format[1]

3.2. Homomorphic Encryption

Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on ciphertexts and generate an encrypted result which, when decrypted, matches the result of operations performed on the plaintexts. Homomorphic encryption has three kinds, that is, additive homomorphic encryption, multiplicative encryption, and full homomorphic encryption having both of additive and multiplicative encryptions.

In this paper, the additive homomorphic encryption in [11] is considered as a brilliant candidate for solving the mentioned privacy issues of data coding as it can secure the coded data, as well as, preserve the result of executing the Exclusive-OR bit-wise operator. The property of the Exclusive-OR operator is “A+A=0”.Let P_1 and P_2 be two considered messages, k be public key, P be the coded data of P_1 and P_2, C be the cipher text of P encrypted with k, $Dec_k(C)$ or P be the plain text ofC decrypted with k. Let us consider P_1, P_2 and P be three strings of bits. Encryption and decryption of the two messages as follows:

- Encryption:
  $C=Enc_k(P)= Enc_k(P_1+P_2)=Enc_k(P_1)+Enc_k(P_2)$.
- Decryption:
  $P=Dec_k(Enc_k(C))=Dec_k(Enc_k(P_1+P_2))=P_1+P_2$.

In our paper, we adopt Elliptic Curve Cryptography (ECC) [12] based on the binary finite field $F^2{}_m$. Assume each node n_i in the network has a pair of keys (k_i, k_i.B) where B is a base parameter of ECC and public to all over the network, k_i is the private key, and k_i.B is the public key. With ECC, the encryption of a message P_1 with k_i is $C_1=Enc_{ki}(P)=(r_i.B, P_i+r_i.B.k_i)$, and the decryption with k_i is $P_i=Dec_{ki}(C_i)=P_i+r_i.B.k_i-(r_i.B).k_i$, where $r_i$is a random number generated by the encryptor.

Hence, the addition of two encryption gets
$C = Enc_{ki}(P_1)+Enc_{ki}(P_2)=(r_1.B+r_2.B, P_1+P_2+r_1.B.k\_i+r_2.B.k_i)$, and the responsive decryption is $Dec_{ki}(C)=P_1+P_2+r_1.B._{ki}+r_2.B.k_i-(r_1.B+r_2.B).k_i=P_1+P_2$.

4. Security Model and Requirement

4.1. Security Model

Although COPE is simple but effective in reducing the amount of transmissions, COPE still discloses the risks of being attacked. Particularly, COPE header, as presented in Section 3.1.3.,containsmuch sensitive information relating to identity of recipient and sender of the considered packet. In this paper, we consider the honest-but-curious attackers.

Honest-but-curious attack. They correctly operate the protocol without modifying data. However, they try to learn as much personal information of the other users as possible to satisfy their curiosity. They do not use the learned data for any harmful purpose. These adversaries do not cause serious consequences, but their act can leave a back door for the other attacks if they do not preserve well that data. In this work, each node may be considered as an honest-but-curious attacker. They can learn the private information from the COPE header as well as infer the path on which the packet moves through. If they are intersecting nodes, they can try on the received packets. In case they are surrounding nodes of the packet's path, they can try to overhear the packets from that path.

Example 2.The adversary can get the aggregate package, e.g., $P=P_1+P_2+P_3$, at the same time itreceives another aggregate packet, i.e., $P'=P_1+P_2$,so it can infer the content of the packet $P_3= P - P'$.

4.2. Security Requirement

Based on the risk analysis from the above attack models, to avoid serious consequences of attacks, some security requirements need to be guaranteed to be done on the aggregate packet moving through the network.

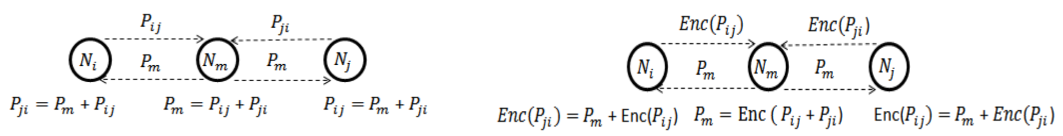
(1) Packet data security. The packet payload cannot be read by the intermediate nodes on the network path. It should be accessed by only its destination. To ensure this requirement, a solution should adopt cryptographic algorithms into the problem. The COPE packets should be encrypted with the public key of the destination node, and are aggregated in a secret way. There are two parts to be encrypted, that is, packet payload and fields in the COPE header for coding condition evaluation. This will be presented in further details in Section 6.

(2) Secure coding condition evaluation. The intermediate nodes only code the packets they receive if the packets satisfy the coding conditions of theirs. However, the coding condition evaluation process also leaks the packet flow information. To avoid the other party can see the evaluation process at the intermediate node, the security solution is needed. In this work, the cryptographic solutions are proposed to secure this coding condition evaluation process. This will be expressed in more details in Section 6.

(3) Performance optimization. As the network model is peer-to-peer, the peer devices own plenty of restrictions, that is, limited physical resource and performance. Hence there is a need of requirement, that is, to select a lightweight cryptographic algorithm to secure packets and make the protocol of encrypting packets more securely. In this work, the lightweight cryptography algorithm, i.e., ECC is adopted to reduce the performance cost, at the same time keep the security complexity of an encryption algorithm. The details will be expressed in Section 6.

5. SCOPE ARCHITECTURE

This section describes a secure COPE architecture, namely SCOPE.SCOPE has three sides, that is, the source, the destination and the intermediate nodes. As in Figure 4, the source is N_i in charge of sending a packet to a certain destination. Whereas, the destination is N_j receiving the packet sent from N_i. Moreover, the middle side is N_m, as considered as an intermediate node, which takes packets and checks the conditions of SCOPE to see if it can aggregate the received packets by using the operator Exclude-OR (see Section6for the aggregate SCOPE conditions), then propagate the aggregate to the next node. For example, as in Figure 4-a, it is assumed that N_i sends a packet P_{ij} to N_j through N_m, and N_j sends a packet P_{ji} to N_i through N_m. N_m invokes an Exclusive-ORoperator (denoted as "+") over the two received packets, and obtains an aggregate value, that is, $P_m = P_{ij} + P_{ji}$.



a - COPE: Data coding without encryption *b - SCOPE:Encrypting coding data.*

Figure 4. COPE Header Format COPE vs SCOPE

In order to make N_m able to successfully compute P_m but also keep the aggregate value in a secret without learning any information from the packets. To achieve this goal, the additive homomorphic encryption is investigated to be used. Let us consider the scenario when the additive homomorphic encryption is applied into data coding mechanism as follows:

- *At source node N_i:*Let us consider Figure 4-b, node N_i tends to send data P_{ij} to N_j through N_m. Before it transmits P_{ij} to N_m, it encrypts P_{ij} by N_j's public key k_j and obtains an encryption $Enc_{Kj}(P_{ij})$. It propagates this encryption to N_m. In the meantime, N_j also wants to send data P_{ji} to N_i through N_m. N_j encrypts P_{ji} by N_i's public key k_i and obtains an encryption $Enc_{ki}(P_{ji})$. Then N_j send this encryption to N_m as well.

- *At intermediate node N_m:* N_m aggregates the two received encryption from N_i and N_j by performing the operator Exclusive-OR on the two cipher text, to receiveC = $Enc_{kj}(P_{ij})$ + $Enc_{ki}(P_{ji})$=$Enc_{(ki,kj)}(P_{ji} + P_{ij})$ where C is the encryption of the coded data of P_{ij} and P_{ji}. Then, N_m transfers C to both N_i and N_j.

- *At destination node N_j:* N_j again adds its $Enc_{kj}(P_{ij})$ to C and obtains $Enc_{kj}(P_{ij})$ + ($Enc_{kj}(P_{ij})$ + $Enc_{ki}(P_{ji})$)=$Enc_{ki}(P_{ji})$. N_j then decrypts $Enc_{ki}(P_{ji})$ with its private key to achieve the data from N_i to it, that is, P_{ji}. The same steps are similarly performed at N_i.

6. SCOPE SECRET PROTOCOL

6.1. Secure Coding Condition

Let us see Figure 4-b,node N_i sends a packet P_{ij} to its target, that is, N_j, through N_m. To make P_{ij} able to reach N_j, the work at N_m is crucial as it decides to forward the packet to N_j. Forwarding the packet is not simply receiving and sending the received packet P_{ij} to the network. It does not also mean that packets move through the intermediate nodes to reach their destination, not all of intermediates will code (i.e., aggregate) the by-passed packets and propagate the result packets towards. As the objectives of coding protocol, to reduce the bandwidth cost, N_m needs to aggregate (i.e., code) several packets together and forwards only the aggregate packet. In order to support N_m in deciding the P_{ij} propagation, the coding conditions are built up for N_m to be evaluated. The necessary parameters for the coding condition evaluation are retrieved from the header of COPE packet. Only the packets satisfying the coding conditions will be aggregated into their suitable packet then sent towards to their destination. More specifically:

Definition 1. **Coding condition.** *Let F_i and F_j be two flows of packets crossing at node N_m, i.e., $Fi \cap Fj = \{N_m\}$.N_m codes packets received from F_i and F_j in case the next hops of the packet at N_m on flow F_i (or flow F_j) are the previous hops of the packet at node N_m on flow F_j (or flow F_i), or are the neighbours of that previous hop. More formally:*

$$X_j = PH[N_m, F_j]$$
$$X_i = PH[N_m, F_i]$$
$$C(N_m, F_i, F_j) = \begin{cases} true, & ((NH[N_m, F_i] \subset NB_{X_j}) \vee (NH[N_m, F_i] = PH[N_m, F_j])) \\ & \wedge((NH[N_m, F_j] \subset NB_{X_i}) \vee (NH[N_m, F_j] = PH[N_m, F_i])) \\ false, & otherwise. \end{cases}$$

Where

- *$C(N_m, F_i, F_j)$ is the coding condition.*
- *N_m is the intersecting node of two flows F_i and F_j.*
- *$NH[N_m,F_i]$ is the set of next hops of node N_m in flow F_i.*
- *$PH[N_m,F_i]$ or X_i is the set of previous hops of node N_m in flow F_i.*
- *NB_{Xi} is the set of all neighbours of nodes in X_i in flow F_i.*

Example 4.Let us see Figure 1-b. It is noted that in case of COPE, the set of neighbour nodes and the set of previous node contain only one element. It is assumed that there are two flows of packets, that is, F_1 and F_2 (F_1: $N_1 \rightarrow N_2 \rightarrow N_3$, and F_2: $N_1 \rightarrow N_2 \rightarrow N_3$). N_2 is clearly the intersecting node of the two flows, so it is also the intermediate node where the coding can cause. Hence, the set of previous hops of N_2 on F_1 is $PH[N_2, F_1] = \{N_1\}$, whereas the set of next hops of N_2 on F_1 is $NH[N_2, F_1] = \{N_3\}$.In the meanwhile, the set of previous hops of N_2 on F_2 is $PH[N_2, F_2] = \{N_3\}$, whereas the set of next hops of N_2 on F_2 is $NH[N_2,F_2] = \{N_1\}$. For each node in sets of previous nodes of N_2 in F_1 and F_2, N_1's neighbours is $NB_{N1} = \{N_2\}$, and the set of N_3's neighbours is $NB_{N3} = \{N_2\}$. Let us check the coding conditions in Definition 1,we see that the case $C(N_2, F_1, F_2)$ = true happens.

The above conditions are readable and stored in the header of each node. However, the coding condition evaluation process is done by the intersecting node (i.e., the intermediate node) N_m. Node N_m also needs the information from its surrounding nodes, especially the nodes are the sender and the recipients of the packets through it, as in Figure 4-b, that is, N_i and N_j. Hence, in case that N_m is not an honest and benign node, this assessment process can leak the packet flow information to the intermediate, and cause a serious consequence to the data security and privacy as presented in Section 4.2. This process thus should be done in a secure way. In this work, we adopt the homomorphic encryption as an effective way to secure this process, particularly, ECC is used for securing data. More specifically, all information owned by N_m are encrypted with a public key K_m. All data at N_i and N_j are respectively encrypted with the public keys of N_i and N_j, that is, K_i and K_j. It is noted that the atomic data which is encrypted is the ID of node's neighbours or previous hops or next hops. The encrypted atomic data is specified as in Definition 2.

Definition 2. Atomic Data Cipher*. Let K_2 be the public key of node N_2. Let D_2 is the atomic data to be encrypted with K_2. Therefore, $Enc_{K2}(D_2)$is the encryption (i.e., cipher) of the atomic data D_2 encrypted with the key K_2 of node N_2.*

Example 5.Let us continue Example 4. Let K_m be the public key of N_m, N_i be the previous hop of N_m on flow F_1. Hence, Enc_{Km} (N_i) is the encryption of N_i's ID. It is noted that N_i is also considered as its own ID.

Each of nodes in the network keeps one previous hop list, one next hop list, and one neighbour list. Hence, for evaluating the coding condition, the intersecting node, i.e., N_m, exploits its previous hop list and next hop list, at the same time, requests each of its previous nodes for sending it their neighbour list. These lists are also the input parameters of the coding evaluation process. The coding condition parameters all are lists of atomic data ciphers, and defined in Definition 3.The lists of encryption defined in Definition 4 are also stored in the SCOPE header instead of the plain text as in COPE header.

Definition 4. Coding Condition Parameter Cipher. *Let N_t, $F_i$be respectively the considered node and the consider flow of data. Let $NH[N_t, F_i]$, $PN[N_t, F_i]$, $NB_{PN[Nt,Fi]}$respectively N_t's previous hop list, N_t's next hop list and the neighbour node lists of N_t's previous nodes. From Definition 2 of atomic data cipher, for each element of the lists $NH[N_t, F_i]$, $PN[N_t, F_i]$, $NB_{PN[Nt,Fi]}$, the respective cipher of the lists are denoted as $Enc_{Kt}(NH[N_i, F_i]))$, $Enc_{Kt}(PN[N_t, F_i])$, $ENB_{PN[Nt,Fi]}$ and defined as follows:*

$$
\begin{aligned}
Enc_{K_t}(NH[N_t,F_i]) &= \{Enc_{K_t}(N_{(0,F_i)}), Enc_{K_t}(N_{(1,F_i)}), \cdots, Enc_{K_t}(N_{(n,F_i)})\} \\
Enc_{K_t}(PH[N_t,F_i]) &= \{Enc_{K_t}(N_{(n+1,F_i)}), Enc_{K_t}(N_{(n+2,F_i)}), \cdots, Enc_{K_t}(N_{(n+m,F_i)})\} \\
ENB_{PH[N_t,F_i]} &= Enc_{(K_{(n+1)},K_{(n+2)},\cdots,K_{(n+m)})}(NB_{\{N_{(n+1,F_i)},N_{(n+2,F_i)},\cdots,N_{(n+m,F_i)}\}}) \\
&= \bigcup_{u=n+1}^{n+m} Enc_{K_u}(NB_{N_{(u,F_i)}}) \\
&= \bigcup_{u=n+1}^{n+m} \{Enc_{K_u}(N_{(u,F_i,1)}), Enc_{K_u}(N_{(u,F_i,2)}), \cdots\}
\end{aligned}
$$

Example 7.Let us continue Example 4 and 5. Let N_2, K_2 be respectively the considered node and its own public key. Let F_1 be the considered flow. N_2's lists of next hops, previous hops, and its previous nodes' neighbour node lists on flow F_1as follows:$Enc_{K2}(NH[N_2, F_1]) = \{Enc_{K2}(N_{(3,1)})\}$; $Enc_{K2}(PH[N_2, F_1]) = \{Enc_{K2}(N_1)\}$; $Enc_{K2}(NB_{PH[N2, F1]}) = \{Enc_{K2}(N_1), Enc_{K2}(N_3)\}$ as N_2 on F_1 has two neighbours, that is, N_1 and N_3.

The coding condition evaluation is processed at the intersecting node but it needs a collaboration among multiple parties (i.e., the intersecting node, and its previous hop and next hop on the same flow) to support this evaluation process. For example, as in Figure 4-b, the coding condition evaluation is done by N_m but it needs a collaboration among N_i, N_m and N_j. However, as presented in Section 4.2., to prevent the risk of information violation, this collaboration needs to be processed secretly to avoid leaking anode's private information to the others. In this situation, the information of N_i and N_j must be kept against the reading of N_m. Moreover, the nature of each coding condition evaluation is a comparison among elements of the two lists. Hence, to meet the security requirement in Section 4.2., this comparison is securely processed among encryptions of elements of the lists. For example, as in Definition 1, one of coding condition is the comparison between the lists $NH[N_m, F_i]$ and $PH[N_m, F_j]$, whereas, the comparison between $NH[N_m, F_i]$ and $NB_{PH[Nm, Fi]}$ is a series of comparisons between the list $NH[N_m, F_i]$ and each of lists in $NB_{PH[Nm, Fi]}$ since $NB_{PH[Nm, Fi]}$contains many lists, each of lists contains the neighbour nodes relating to each of

N_m's previous nodes. As in Definition 4, the coding conditions contain the lists of encrypted elements. The comparison operators used for assessing the coding conditions are executed on the lists of encryptions. Thus, let us present one secure comparison between NH[N_m, F_i] and PH[N_m, F_j] done at N_m. The other comparisons in the coding conditions are similarly performed.
Let us consider the two original lists of NH[N_m, F_i] and PH[N_m, F_j] as follows:

$$NH[N_m, F_i] = \{N_{(0,F_i)}, N_{(1,F_i)}, \cdots, N_{(n,F_i)}\}$$
$$PH[N_m, F_j] = \{N_{(n+1,F_j)}, N_{(n+2,F_j)}, \cdots, N_{(n+m,F_j)}\}$$

As in steps in Table 1, first N_i is in charge of generating the encryption of NH[N_m, F_i] using its destination node's public key, that is N_j's public key (i.e., K_j), to obtain $Enc_{K_i}(NH[N_m, F_i]) = \{Enc_{K_i}(N_{(0,F_i)}), Enc_{K_i}(N_{(1,F_i)}), \cdots, Enc_{K_i}(N_{(n,F_i)})\}$ (step 1). In the meanwhile, N_j is in charge of generating the encryption of PH[N_m, F_j] with its destination node's public key, that is Ni's public key (i.e., K_i), to obtain $Enc_{K_j}(PH[N_m, F_i]) = \{Enc_{K_j}(N_{(n+1,F_j)}), Enc_{K_j}(N_{(n+2,F_j)}), \cdots, Enc_{K_j}(N_{(n+m,F_j)})\}$ (step 2). After generating the encryption lists, N_i sends the encryptions to N_m (Step 3), while N_j sends the encryptions to N_m (step 4).Hence, N_m can help transfer the received encryption lists to their destination nodes, that is, N_m sends Enc_{Ki}(NH[N_m, F_i]) to N_j(step 5), and forwards Enc_{Kj}(PH[N_m, F_i]) to N_i (step 6).At N_i, N_i continues to uses the its public key, i.e., K_i, to encrypt the encryption list from Nm, to obtain a twice-encrypted list, that is,

$Enc_{(K_i,K_j)}(PH[N_m, F_i]) = \{Enc_{(K_i,K_j)}(N_{(n+1,F_j)}), Enc_{(K_i,K_j)}(N_{(n+2,F_j)}), \cdots, Enc_{(K_i,K_j)}(N_{(n+m,F_j)})\}$ (step 7). Similarly, $N_j$again encrypts the received list of encryptions with its public key, i.e., K_j, and obtains the twice-encrypted list, that is,
$Enc_{(K_j,K_i)}(NH[N_m, F_i]) = \{Enc_{(K_j,K_i)}(N_{(0,F_i)}), Enc_{(K_j,K_i)}(N_{(1,F_i)}), \cdots, Enc_{(K_j,K_i)}(N_{(n,F_i)})\}$ (step 8). After that, N_i and N_j transfer the twice-encrypted lists to N_m (steps 9, 10).N_m then invokes the function *EqualList ()* as in Algorithm 1 (step 11). *EqualList()*evaluates if two lists are equal to each other. It inputs two lists, that is, $Enc_{(K_j,K_i)}(NH[N_m, F_i])$ and $Enc_{(K_i,K_j)}(PH[N_m, F_i])$, and returns a boolean result, that is, true or false. True is returned as the two encryption lists are equal, and false as the two encryption lists are unequal.

Table 1. Private condition evaluation between two lists NH[N_t, F_i], PH[N_t, F_i].

1	N_i	Creates $Enc_{K_i}(NH[N_m, F_i]) = \{Enc_{K_i}(N_{(0,F_i)}), Enc_{K_i}(N_{(1,F_i)}), \cdots, Enc_{K_i}(N_{(n,F_i)})\}$
2	N_j	Creates $Enc_{K_j}(PH[N_m, F_i]) = \{Enc_{K_j}(N_{(n+1,F_j)}), Enc_{K_j}(N_{(n+2,F_j)}), \cdots, Enc_{K_j}(N_{(n+m,F_j)})\}$
3	$N_i \rightarrow N_m$	N_i sends Enc_{Ki}(NH[N_m, F_i]) to N_m
4	$N_j \rightarrow N_m$	N_j sends Enc_{Kj}(PH[N_m, F_j]) to N_m
5	$N_m \rightarrow N_i$	N_m forwards Enc_{Ki}(NH[N_m, F_i]) to N_j
6	$N_m \rightarrow N_j$	N_m forwards Enc_{Kj}(PH[N_m, F_j]) to N_i
7	N_i	N_i encrypts Enc_{Kj}(PH[N_m, F_j]) and obtains the new encryption list, that is, $Enc_{(K_i,K_j)}(PH[N_m, F_i])$
8	N_j	Nj encrypted Enc_{Ki}(NH[N_m, F_i]) and obtains the result, that is,

		$Enc_{(K_j,K_i)}(NH[N_m,F_i])$
9	$N_i \rightarrow N_m$	Ni transfers the $Enc_{(K_i,K_j)}(PH[N_m,F_i])$ to N_m
10	$N_j \rightarrow N_m$	N_j transfers the $Enc_{(K_j,K_i)}(NH[N_m,F_i])$ to N_m
11	N_m	• Evaluate the equality of two lists $Enc_{(K_i,K_j)}(PH[N_m,F_i])$ and $Enc_{(K_j,K_i)}(NH[N_m,F_i])$ by calling the function *EqualList()* as in Algorithm 1. • If the two lists are totally equal, the conditions is met.

More specifically, in the Algorithm 1, $N_m$traverses the two lists $Enc_{(K_i,K_j)}(PH[N_m,F_i])$ and $Enc_{(K_j,K_i)}(NH[N_m,F_i])$(lines 2,3) to evaluate the equality of elements of two lists by subtracting (or Exclusive-ORing)each element $\overline{x}$of $Enc_{(K_i,K_j)}(PH[N_m,F_i])$ by each element $\overline{y}$of $Enc_{(K_j,K_i)}(NH[N_m,F_i])$ (line 4).Let *count* be a temporary integer. If the subtractive result is equal to an encryption of 0 generated with the public key of N_i and N_j, i.e., $K_i$and K_j, that means $\overline{x}$ is equal to $\overline{y}$, *count* is increased by 1 (line 5). Then, if *count* is equal to the sizes of two lists, that is, sizeNH and sizePH (lines 9, 10, 11), the functions return true (line 12), it means two lists are equal, otherwise false is returned (line 14). In case the two lists are equal, it also indicates that the coding condition is met. Then, the other coding condition can be continued to be evaluated.

Algorithm 1 *EqualList()*

Input:

$$Enc_{(K_i,K_j)}(NH[N_m,F_i]) = \{Enc_{(K_i,K_j)}(N_{(0,F_i)}), Enc_{(K_i,K_j)}(N_{(1,F_i)}), \cdots, Enc_{(K_i,K_j)}(N_{(n,F_i)})\}$$
$$Enc_{(K_i,K_j)}(PH[N_m,F_j]) = \{Enc_{(K_i,K_j)}(N_{(n+1,F_i)}), Enc_{(K_i,K_j)}(N_{(n+2,F_i)}), \cdots, Enc_{(K_i,K_j)}(N_{(n+m,F_i)})\}$$

Output: true | false

```
1: $count = 0$;
2: for $\overline{x} \in Enc_{(K_i,K_j)}(NH[N_m,F_i])$ do
3:     for $\overline{y} \in Enc_{(K_i,K_j)}(PH[N_m,F_j])$ do
4:         if $(\overline{x} + \overline{y} == Enc_{(K_i,K_j)}(0))$ then
5:             $count++$;
6:         end if
7:     end for
8: end for
9: $sizeNH = sizeof(Enc_{(K_i,K_j)}(NH[N_m,F_i]))$;
10: $sizePH = sizeof(Enc_{(K_i,K_j)}(PH[N_m,F_j]))$;
11: if $(count == sizeNH == sizePH)$ then
12:     return true;
13: end if
14: return false;
```

6.2. Secure Payload Coding

The fact that COPE header are protected against attacks of observing the flow of packet and intervening the packets' routines is protecting coding conditions and operations on them as presented in Section 6.1.Even though that is a meaningful and important security strategy, securing data payload also play a substantial role since the payload contains several sensitive information of users. Especially, the coding is done at the intersecting node. As discussed in Section 4.2., the intersecting node can be an adversary and the plain data can reveal the personal information to the intersecting node. Hence, the payload should be secured. In this work, ECC algorithm is used to encrypt data into the cipher. This solution makes the intersecting node unable to read the data but at the same time still work on the encryption only. Hence, the sending node needs to encrypt the data before propagating the encryption to the intersecting node.

Definition 5. Coded Payload Cipher. *Let $K_0, K_1, \ldots, K_n$ be public keys of nodes $N_0, N_1, \ldots, N_n$. Let $Enc_{K0}(P_0)$, $Enc_{K1}(P_1)$, …, $Enc_{Kn}(P_n)$ be the n payload encryption of packets:$P_0$with K_0, P_1 with K_1,…, $P_n$with K_n, where packetsare sent through the intermediate node N_m. It is assumed that the coding condition as in Definition 1 are met at N_m. The coded payload cipher made at $N_m$is formulated as follows:*

$$\begin{aligned} Enc_{(K_0,K_1,,K_n)}P &= Enc_{(K_0,K_1,,K_n)}\sum_{i=0}^{n}(P_i) \\ &= \sum_{i=0}^{n}(Enc_{K_i}(P_i)) = \sum_{i=0}^{n}(r_i.B, P_i + r_i.B.K_i)) \\ &= (\sum_{i=0}^{n}(r_i.B), \sum_{i=0}^{n}(P_i + r_i.B.K_i)) \end{aligned}$$

where $r_0, r_1, \ldots, r_n$ are random numbers generated at nodes creating the partial encryptions.

Example 8. Let us continue Example 7. It is assumed that N_i wants to send the packet P_{ij} to N_j through N_m on flow F_1, so it encrypts the payload of P_{ij} into $Enc_{Kj}(P_{ij})$ and forwards this encryption to N_m. In the meanwhile, N_j wants to send the packet P_{ji} to N_i through N_m on flow F_2, so it encrypts the payload of P_{ji} into $Enc_{Ki}(P_{ji})$ and forwards this encryption to N_m. At node N_m, after evaluating the coding condition as in Example 4, N_m code the two encryptions $Enc_{Kj}(P_{ij})$ and $Enc_{Ki}(P_{ji})$ by aggregating them, and get $Enc_{(Ki,Kj)}(P_{ij}+P_{ji})=(r_i.B+r_j.B,(P_{ij}+P_{ji})+(r_1.B.K_1+r_2.B.K_2))$ as the coded payload cipher.

Receiving packets from different neighbour nodes, after evaluating the coding condition, $N_m$detaches the encrypted payloads of all packets and code them. Then N_m put them into a new packet. Then, N_m propagates it towards the network. As the receiving node gets the coded packet from N_m, it can assess the coded payload cipher, then decodes and decrypts the cipher to obtain the data for it. This process is concerned as the coded payload assessment. The decoded payload is defined as in Definition 6.

Definition 6. Decoded Payload. *Let N_n be the destination node receiving the encrypted coded payload as defined in Definition 5. Let $K_0, K_1, \ldots, K_{n-1}$ be public keys of N_n's neighbour nodes $N_0, N_1, \ldots, N_{n-1}$. Let $Enc_{(K_0,K_1,,K_n)}\sum_{i=0}^{n}(P_i) = (\sum_{i=0}^{n}(r_i.B), \sum_{i=0}^{n}(P_i + r_i.B.K_i))$ be the coded payload cipher of packets from $N_0, N_1, \ldots, N_n$ with the random number $r_0, r_1, \ldots, r_n$ generated by nodes generating the partial encryptions.More formally, the decoded payload by N_n, that is $Enc_{Kn}(P_n)$, is defined as follows:*

$$\begin{aligned} Enc_{K_n}(P_n) &= Enc_{(K_0,K_1,,K_n)}\sum_{i=0}^{n}(P_i)) + Enc_{(K_0,K_1,,K_{(n-1)})}\sum_{i=0}^{(n-1)}(P_i)) \\ &= (\sum_{i=0}^{n}(r_i.B), \sum_{i=0}^{n}(P_i + r_i.B.K_i)) + (\sum_{i=0}^{(n-1)}(r_i.B), \sum_{i=0}^{(n-1)}(P_i + r_i.B.K_i)) \\ &= ((\sum_{i=0}^{n}(r_i.B) + (\sum_{i=0}^{(n-1)}(r_i.B)), (\sum_{i=0}^{n}(P_i + r_i.B.K_i) + \sum_{i=0}^{(n-1)}(P_i + r_i.B.K_i))) \\ &= (r_n.B, P_n + r_n.B.K_n) \end{aligned}$$

$$Dec_{K_n}(Enc_{K_n}(P_n)) = (r_n.B).K_n + P_n + r_n.B.K_n = P_n$$

Example 9. Let us continue Example 8. N_j receives the coded payload cipher for it, that is, $Enc_{(Ki,Kj)}(P_{ij} + P_{ji}) = (r_i.B + r_j.B,\ (P_{ij} + P_{ji}) + (r_i.B.K_i + r_j.B.K_j))$. N_j still keeps the coded payload cipher of the packet it wants to send to Ni, that is, $Enc_{Ki}(P_{ji}) = (r_i.B,\ P_{ji} + r_i.B.K_i)$ where r_i is a random number generated by N_j. Hence, the decoded payload is $Enc_{Kj}(P_{ij}) = Enc_{(Ki,Kj)}(P_{ij} + P_{ji}) + Enc_{Ki}(P_{ji}) = (r_i.B + r_j.B,\ (P_{ij} + P_{ji}) + (r_i.B.K_i + r_j.B.K_j)) + (r_i.B,\ P_{ji} + r_i.B.K_i) = ((r_i.B + r_j.B + r_i.B),\ ((P_{ij} + P_{ji} + (r_i.B.K_i + r_j.B.K_j + P_{ji} + r_i.B.K_i)) = (r_j.B,\ P_{ij} + r_j.B.K_j)$. Then, the decryption is executed at N_j with the private key of N_j (i.e., K_j), to get the data needed for N_j, i.e., $Dec_{Kj}(Enc_{Kj}(P_{ij})) = (r_j.B).K_j + P_{ij} + r_j.B.K_j = P_{ij}$.

7. SCOPE EVALUATION

In this work, to prove for effectiveness and efficiency of the proposed secure protocol, experiments on different number of nodes and different key sizes of ECC encryption are done. These experiments are operated on PC with the physical resources in terms of CPU 1.8GHz Intel Duo-Core, RAM 4GB, HDD 16GB.

7.1. Throughput

We use NS-2 as a simulator for the experiment. We use *4* topologies in Figure 1, Figure 6a, Figure 6b, and Figure 6c. Flows in test scenarios are shown in the Table 2. Each topology has been generated in a flat area *1000m X 1000m*. The data traffic in the network are all CBR (Constant Bit Rate) flows sent over UDP (User Datagram Protocol) using *1000*-byte datagrams with an arrival interval of *0.01s* and traffic generation duration at source of *150s*. The routing protocol used is DSDV (Destination-Sequenced Distance-Vector) [10]. The results are collected with a confident interval of *95%*.

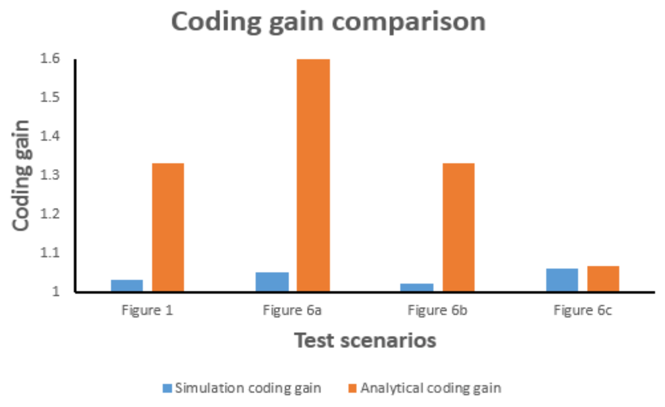
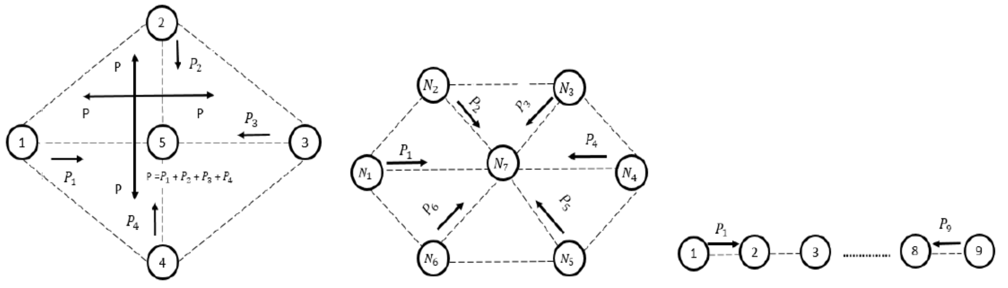


Figure 5.Comparison between analytical coding gain and simulation coding gain.

Results are presented in Figure 5, which compares between the coding gains obtained by simulation and the ones obtained by theoretical analysis [5]. We observe that the coding gains obtained by theoretical analysis are always greater than what obtained by simulation. For instance, the test case of Figure 1, the simulated coding gain is equal to *1.033* while the theoretical coding gain is$\frac{4}{3}$= *1.333*. For the test scenario of Figure 6a, the simulated and analytical coding gains are *1.050* and $\frac{8}{5}$= *1.600*, respectively. For the test case of Figure 6b, the coding gain is *1.020* for the simulation while the coding gain is $\frac{12}{9}$= *1.333* for the theoretical analysis. For the test scenario of Figure 6c, the simulated coding gain and the theoretical coding gain equal *1.060* and $\frac{16}{15}$= *1.067*, respectively. These deviations are because the theoretical analysis cannot take into account the collision in wireless network environment. The collision can lead to the delay increase that a packet sent from source node to destination node and so, some coding opportunities are missed. Besides that, frame loss, frame retransmission, or framing error are also one of the problems derived from the collision, affecting to the coding gain results.

7.2. Time overhead

In this experiment, to assess the computing performance of SCOPE. We make a diversity of experimentson different parameters. Particularly, we create fourSCOPE scenarios (as in Figures 1, 6a, 6b, 6c), and change the key sizes of ECC additive homomorphic encryption algorithm. Each calculated value in the experimentsis the average of 20 times running the same experiments.



(a) COPE with 5 nodes, packets XOR-ed at node 5 (b) COPE with 7 nodes, packets XOR-ed at node 7 (c) COPE with 9 nodes, packets XOR-ed at node 2, 3, 4, 5, 6, 7 or 8.

Figure 6.SCOPE scenarios.

Table 2. Flows in test scenarios.

Scenario	Figure	Flows
1	1	F1: N1 → N2 → N3; F2: N3 → N2 → N1
2	6a	F1: N1 → N5 → N3; F2: N3 → N5 → N1; F3: N2 → N5 → N4; F4: N4 → N5 → N2

3	6b	F1: N1 → N7 → N4; F2: N4 → N7 → N1;
		F3: N2 → N7 → N5; F4: N5 → N7 → N2;
		F5: N3 → N7 → N6; F6: N6 → N7 → N3

4	6c	F1: N1 → N2 → N3 → ... → N9; F2: N9 → N8 → N7 → ... → N1

Figures 1 and 6 describes four scenarios, and table 2 presents the number of flows w.r.t. the scenarios. Figure 1 involves 3 nodes and 2 flows. Figure 6a involves 5 nodes and 4 flows. Figure 6b involves 7 nodes and 6 flows. Figure 6c involves 9 nodes and 2 flows. To evaluate the computing performance of SCOPE applied with ECC encryption algorithms, the selected ECC key sizes are varied in {163, 283, 409, 571} bits. These key sizes are guaranteed to be still secure by NIST.

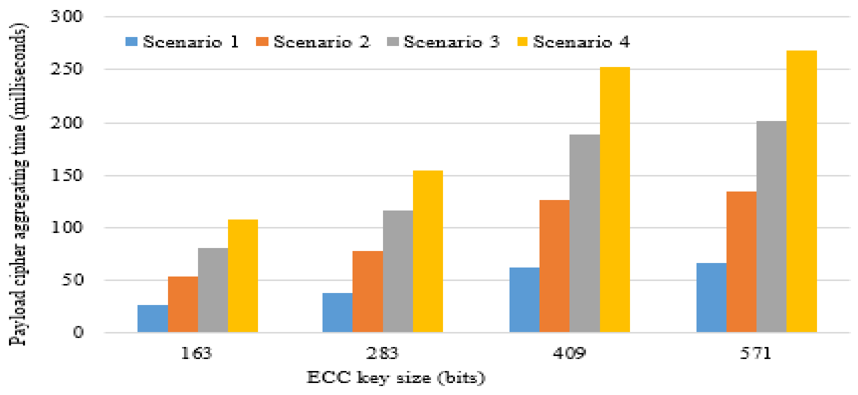


Figure 7. Time on aggregating the payload cipher at the intersecting node (milliseconds or ms) vs ECC key size (bits)

First, the time on aggregating the payload ciphers in the four scenarios. These payload ciphers are aggregated using the additive property of ECC homomorphic encryption. The number of payload cipher aggregation at the intersecting node in the four scenarios are respectively 2, 4, 6, 8 for each flow. In Figure 7, with the smallest ECC key size (i.e., 163 bits), the time on aggregating two payload ciphers of Scenario 1 is 26,8ms. With the 283-bit key size, the time on aggregating 8 payload ciphers of Scenario 4 is 107,2ms. In the worst case, that is, the largest key size (i.e., 571 bit), the time computed for aggregating 8 payload ciphers of Scenario 4 is 268,8ms. The times on different scenarios and key sizes are reasonable in the peer-to-peer environment.

Figure 8 is another experiment to compute the time cost for SCOPE transmissions. These time are measured to evaluate the time which a packet moves through a flow from the source node to the destination node. Hence, these times include the payload cipher aggregating times (as in the experiment of Figure 7) and transmission times. In this experiment, the number of payload cipher aggregations at the intersecting nodes are similar to the previous experiment. The ECC key sizes are also varied in {163, 283, 409, 571} bits. In the case of smallest ECC key size, that is, 163 bits, in scenario 1 involving 2 payload cipher aggregations, the time cost is 260,8ms. Whereas, in the case of largest ECC key size (i.e., 571 bits) and Scenario 4 involving 8 payload cipher aggregations, the time costs is 2.5s. The time overheads in this experiment in both cases are reasonable and prove that SCOPE is effective and efficient.

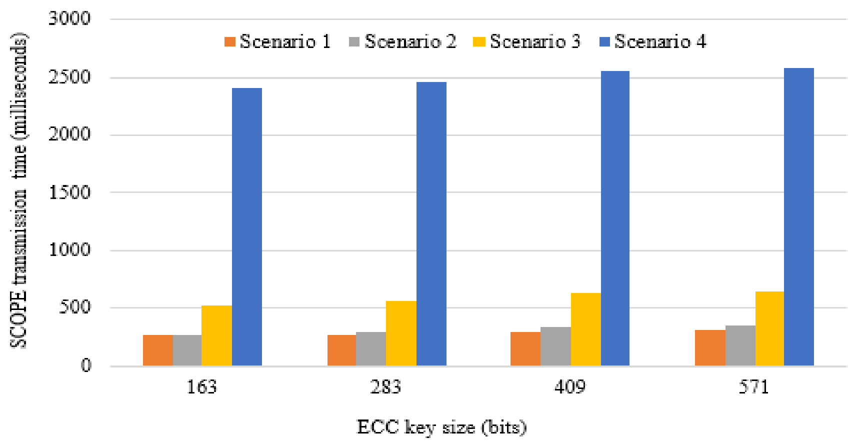


Figure 8. Time on SCOPE transmission including aggregation time (milliseconds or ms) vs ECC key size (bits)

In order to evaluate the cryptographic process of evaluating the coding conditions of SCOPE, we measure the time SCOPE spent on this evaluation (Figure 9). The key sizes are selected in the range {163, 283, 409, 571} bits. The number of coding conditions for Scenarios 1, 2, 3, and 4 are respectively 4, 20, 30, 32 conditions. The time on evaluating the coding conditions in Scenario 1 (i.e., with the lowest number of conditions, that is, 4) with the smallest key size (i.e., 163 bits) is 6.8ms. In the meanwhile, the time on coding conditions evaluation in Scenario 4 (i.e., the highest number of condition, that is, 32) with the largest ECC key size (i.e., 571 bits) is 115,2 ms. In the both cases of the smallest parameters and the largest parameters, the time overheads are still reasonable and prove the effectiveness and efficiency of SCOPE.

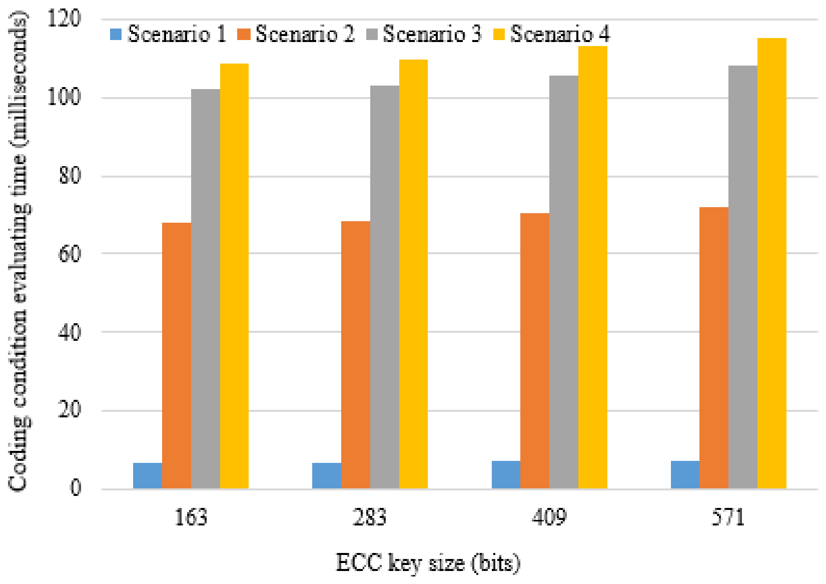


Figure 9. Time on evaluating coding condition at intersecting node (milliseconds or ms) vs ECC key size (bits)

8. Security Property

In this section, a security proof is presented. More specifically, the expression how the proposal can cope with risks of honest-but-curious attack as in Section 4.1., as well as how the proposal meets the security requirements as in Section 4.2. As introduced in Section 4.1, this attack does not aim to poison or misuse the data for any dangerous purposes, but the adversaries try to learn or infer as much personal information as possible only to satisfy their curiosity. However, this attack possibly gets more dangerous as its consequence can leave a backdoor for another attack.

Packet data security. To avoid this risk, ECC homomorphic encryption is adopted to cipher the content of the data payload of packets. The payload then gets into a secret writing, i.e., unreadable. As a result, this carries the COPE packets a shield to deteradversaries from inferring any information inside the payload. Particularly, the ECC public key used for encrypting the payload is kept by only the destination node of the packet, so the other nodes cannot read the packet anyway. Only the destination node of the packet has the private key which can be used for decrypting the payload, then the payload can be read.

Information inferring protection. In this work, the addition property of the homomorphic encryption ECC is exploited for coding multiple packets into one packet at the intermediate node before the aggregate packet is propagated to the next hop of the intermediate node. More specifically, the public key of the destination node of the packet is used for this secret aggregation phase. The intermediate nodes just follow up with the protocol steps, and aggregates the encrypted packets without being aware of the packets' content. This point helps the data safe from the intermediate node. They cannot read the data inside and cannot infer the information as they do not have the private key of the destination node.

Coding condition evaluation security. The coding conditions involve encryptions of the node IDs as presented in Section 6.1. The comparisons are executed on these encryptions. Thus the intermediate nodes cannot learn the node IDs inside the thresholds and operands. Additionally, we also protect the neighbour nodes by encrypting their IDs, and only their direct neighbours can know their IDs, but the two-hop nodes cannot know their IDs. The comparative results are also not recognized by the nodes. Hence, the coding conditions are secured during the evaluation phase.

Performance optimization. In this work, we empower our proposal's security with the ECC encryption algorithm. The ECC encryption algorithm is invoked based on the binary field with the binary operators. This reduces much the time consumption, and meets the computing performance requirements. Additionally, the ECC is still guaranteed to be secure by NIST. So, the proposal can ensure the computing performance to be optimized.

9. CONCLUSION

In this work, we propose a cryptographic approach, namely SCOPE, which is able to support nodes secretly executing the COPE protocol, by applying the lightweight homomorphic encryption ECC. Hence, the packets in SCOPE can move through the intersecting nodes without leaking any private information of the packets. Moreover, SCOPE can be also against the honest-but-curious attack at the intersecting nodes. SCOPE can keep all operations in evaluating the coding conditions or in aggregating the payload work securely. The proposal is also proved to be effective and efficient through the different experiments on a variety of ECC key sizes and different scenarios. This work will be improved to fit with more network coding protocols, such as, BEND, DCAR, etc. and to be immune to the malicious attack in the future work.

REFERENCES

[1] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Mdard, and J. Crowcroft, XORs in the air: Practical Wireless Network Coding, Proc. ACM SIGCOMM, pp. 243-254, 2006.

[2] C. V. Phung, Q. T. Minh, and M. Toulouse, Routing Optimization Model in Multihop Wireless Access Networks for Disaster Recovery, International Conference on Advanced Computing and Applications (ACOMP2016), Can Tho, Vietnam, Nov. 23-25, 2016.

[3] C. V. Phung, V. H. Nguyen, and T. M. T. Nguyen, BEND-Aware Routing Optimization in Wireless Mesh Networks, IEEE International Conference on Advanced Technologies for Communications (ATC), Ho Chi Minh, Vietnam, 2015.

[4] C. V. Phung, T. V. Vu, and T. M. T. Nguyen, DCAR Coding Gain Modeling and Analysis, NoF 2013 - Fourth International Conference on the Network of the Future, Pohang, South Korea, pp. 1-3, (IEEE) (2013).

[5] Sudipta Sengupta, Shravan Rayanchu, and Suman Banerjee, Network Coding-Aware Routing in Wireless Networks, IEEE/ACM Transactions on Networking, vol. 18, no. 4, Agust 2010.

[6] J. Zhang, Y.B. Chen, and I. Marsic, MAC-layer proactive mixing for network coding in multi-hop wireless networks, Computer Networks, Elsevier, vol. 54, pp. 196-207, 2010.

[7] J. Le, J.C.S. Lui, and D.M. Chiu, DCAR: Distributed Coding-Aware Routing in Wireless Networks, IEEE Transactions on Mobile Computing, vol. 9, no. 4, pp. 596-608, April 2010.

[8] T.V Vu, T.M.T. Nguyen, and G. Pujolle, Distributed Opportunistic and Diffused Coding in Multi-hop Wireless Networks, IEEE ICC workshop on Cooperative and Cognitive Mobile Networks (COCONET), Ottawa, Canada, June 2012.

[9] T.V Vu, T.M.T. Nguyen, and G. Pujolle, Distributed Opportunistic and Diffused Coding with Multiple Decoders in Wireless Mesh Networks, ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Cyprus (2012).

[10] C.E. Perkins, and P. Bhagwat, Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, SIGCOMM, London, England UK, 1994.

[11] X. Yi, R. Paulet, and E. Bertino, Homomorphic Encryption and Applications, Springer Publishing Company, Incorporated, 2014.

[12] V. Katiyar, K. Dutta and S. Gupta, A Survey on Elliptic Curve Cryptography for Pervasive Computing Environment, International Journal of Computer Applications, vol. 11, no. 10, pp. 41–46, December 2010.

[13] T. V. Vu, T. M. T. Nguyen, G. Pujolle and N. Boukhatem, DODEX+: A new network coding scheme for mesh networks in mobile cloud computing, Wireless Days (WD), Rio de Janeiro, Brazil, Nov. 12-14 , 2014.

[14] N. Cai, R.W. Yeung, Network coding and error correction, in: Proceedings of the 2002 IEEE Information Theory Workshop, 2002, 2002, pp. 119–122.

[15] Talooki, V.N., Bassoli, R., Lucani, D.E., Rodriguez, J., Fitzek, F.H., Marques, H. and Tafazolli, R., 2015. Security concerns and countermeasures in network coding based communication systems: A survey. Computer Networks, 83, pp.422-445.

[16] N. Cai, R.W. Yeung, Secure network coding, in: Proceedings. 2002 IEEE International Symposium on Information Theory, 2002, 2002, p. 323.

[17] Fan, Y., Jiang, Y., Zhu, H., Chen, J. and Shen, X.S., 2011. Network coding based privacy preservation against traffic analysis in multi-hop wireless networks. IEEE Transactions on Wireless Communications, 10(3), pp.834-843.

[18] Esfahani, A., Mantas, G., Monteiro, V., Ramantasy, K., Datsikay, E. and Rodriguez, J., 2015, September. Analysis of a Homomorphic MAC-based scheme against tag pollution in RLNC-enabled wireless networks. In Computer Aided Modelling and Design of Communication Links and Networks (CAMAD), 2015 IEEE 20th International Workshop on (pp. 156-160). IEEE.

[19] Li, C., Chen, L., Lu, R. and Li, H., 2015. Comment on “An Efficient Homomorphic MAC with Small Key Size for Authentication in Network Coding”. IEEE Transactions on Computers, 64(3), pp.882-883.

[20] Li, X., Fu, F.W., Zhao, X. and Wang, G., 2015, August. Two improved homomorphic MAC schemes in network coding. In Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on (pp. 2214-2219). IEEE.

[21] Esfahani, A., Mantas, G., Rodriguez, J., Nascimento, A. and Neves, J.C., 2015, June. A null space-based MAC scheme against pollution attacks to Random linear Network Coding. In Communication Workshop (ICCW), 2015 IEEE International Conference on (pp. 1521-1526). IEEE.

[22] Paillier, P., 1999, May. Public-key cryptosystems based on composite degree residuosity classes. In Eurocrypt (Vol. 99, pp. 223-238)

AUTHORS

Ngoc Hong Tran has been a full-time lecturer at the Vietnamese German University, Vietnam, since 2016. She obtained a Bachelor Degree in Information Technology, and a Master Degree in Computer Science, from University of Science, Vietnam National University - Ho Chi Minh City. She was awarded a PhD Diploma in Computer Science from University of Insubria, Italy. Moreover, she had been a full-time lecturer at the University of Science, VNU-HCM, from 2004 to 2016. She also worked in National Institute of Informatics, Tokyo, Japan, in 2009, and was an exchange scholar in Portland State University (PSU), Portland City, Oregon State, USA, during the Fall term in 2010. She worked in LAAS-CNRS, Toulouse, France, in 2012, and was an invited researcher in Singapore University of Technology and Design, Singapore, from March to April 2017.Her research interests are applied-cryptography, secure collaboration among multiple parties in distributed and centralized networks, mobile/MANET social networks, web service composition, network coding; privacy preserving data mining.

Cao Vien Phung is currently a Ph.D student in Information System Technology at Braunschweig University of Technology, Germany. He was born in Phu Yen, Vietnam. He attended Luong Van Chanh high school for the gifted in Chemistry, Tuy Hoa - Phu Yen, Vietnam in 2007. He received the Engineer Degree in Electronics and Telecommunications from the Posts and Telecommunications Institute of Technology (PTIT), Ho Chi Minh, Vietnam in 2012. He obtained the Master Degree in Computer Science from Paris 6 University, Paris, France in 2015.

Binh Quoc Nguyen is currently working as Research and Teaching Assistance of Computer Science study program of Vietnam German University. He attended Le Hong Phong high school for the gifted in 2003. He obtained the Bachelor Degree in Information Technology in 2009 and Master Degree in 2012 at University of Science, VNU of Ho Chi Minh City.

Leila Bahri is currently a postdoc in KTH, Sweden. She obtained her bachelor's degree in Computer Science from the School of Science and Engineering at Al Akhawayn University in Ifarne - AUI, Morocco (with highest honor / Summa Cum Laude). In 2011 she received her master's degree in software engineering from AUI with a thesis on the implementation and adoption of ITIL in an academic IT department. In 2009, she received a scholarship for short term studies in Japan from the Japan Student Services Organization (JASSO) by which she performed research for one year at a Robotics laboratory in Meijo University, Nagoya. In 2010, she received the Google University Excellence Award for being selected as best computer science student at her university for that year. Right after that she fulfilled the position of IT Service Desk Manager at the IT department of AUI until May, 2013. She was awarded the Ph.D diploma from University of Insubria, Italy in 2016. She worked as a posdoc in Koc University, Turkey, in 2016.

Dung Hai Dinh has workedat the Vietnamese German University since 2015, and is now currently a full-time Lecturer cum Academic Coordinator of the M.Sc. study program Business Information Systems. He began his study and academic career in Germany since 2004 and got his degree at University of Applied Sciences Gelsenkirchen, major in Business Engineering. He holds a Ph.D. Degree granted by Saarland University, Saarbrücken, Germany. Since 2009, he worked as a researcher at Chair of Operations Research and Business Informatics, with focus on quantitative methods and optimization algorithms for business decision-making problems.