https://doi.org/10.7840/kics.2024.49.5.684

XOR-Embedded Random Linear Network Coding

Patrick Enenche◆, Dongho You°

ABSTRACT

Data confidentiality is a critical concern in modern communication systems. This paper introduces XOR-Embedded Network Coding (XORE-NC), a novel approach to enhance data security in network coding. XORE-NC combines an embedding key mechanism with Random Linear Network Coding (RLNC) to fortify data protection against eavesdroppers. The results indicate improved reliability, reduced encoding delay, and lower latency and throughput compared to traditional RLNC, making it an attractive solution for data confidentiality. However, challenges in decoding delay remain to be addressed. XORE-NC represents a promising avenue for bolstering data security while maintaining efficient data transmission in communication systems.

Key Words : data confidentiality, network coding, reliability, encoding delay, decoding delay, and overall latency

Ⅰ. Introduction

Data confidentiality is a significant burden for businesses of all sizes. However, not all data are open to the same level of risk[1]. For instance, low-risk environ- ments, publicly available files, non-sensitive data, and temporary or disposable files may not require a high degree of security protection[1,2]. As such, it is important for organizations to develop low-risk data security guidelines that balance the level of protection with the associated costs and resources required. Also, in situations where stronger encryption methods may not be practical to implement due to computational or bandwidth limitations, Network Coding can provide some level of confidentiality.

XOR network coding efficiently transmits data by applying the XOR operation to mix packets, enhancing multicast data transmission[3] and offering some protection against eavesdropping by complicating signal decryption[4]. Random linear network coding (RLNC) extends this by using random coefficients, increasing data confidentiality as eavesdroppers need exact coefficients to decode[5]. In RLNC, these random coefficients are elements chosen from a finite field, typically denoted as $GF(q)$, where 'q' represents the field size, equating to 2^h. For instance, in GF(2), 2 elements are available, any of which can be coefficients in the encoding process. RLNC divides data into n packets, encoding them with random finite field coefficients. It transmits at least n packets, formed by linearly combining original packets. The receiver, upon collecting $k(k \geq n)$ packets, constructs a matrix from the coefficients to decode data by solving linear equations. RLNC secures data through complex mathematics, using random coefficients and linear independence to deter unauthorized decoding and require eavesdroppers to capture many independent packets to decode the data. It also offers error resilience, enhancing network security and reducing overhead with minimal

※ This research was supported in part by 2023 Hannam University Research Fund, and in part by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE)(2021RIS-004).
◆ First Author : Hannam University, Department of Information and Communication Engineering; 20214233@gm.hannam.ac.kr, 학생회원
° Corresponding Author : Hannam University, Department of Information and Communication Engineering; dongho.you@hnu.kr, 정회원
논문번호 : 202312-173-B-RN, Received December 27, 2023; Revised February 7, 2024; Accepted February 7, 2024

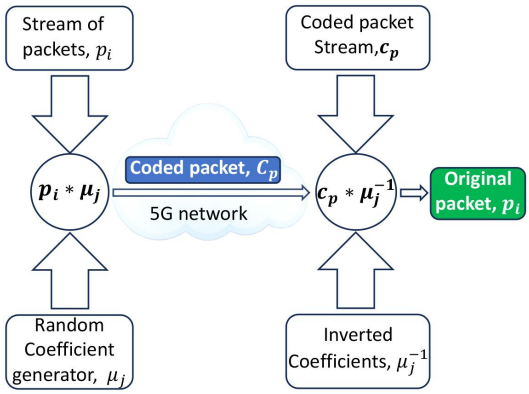


Fig. 1. Fundamental description of RLNC

spectrum usage. Refer to Fig. 1. for a basic description of RLNC.

Some reported works[6-9] collectively explore advanced network coding techniques like RLNC using a Vandermonde matrix to create the coding vectors (RL-NCV), Secure Practical Network Coding (SPOC), and Fulcrum network coding (FNC). These methods enhance data confidentiality and network performance by integrating cryptographic methods, optimizing encoding mechanisms, and reducing overhead, each contributing uniquely to improved security in network communications. Additionally, recent studies[10,11] have extensively compared RLNC and XOR network coding, with evidence favoring RLNC, especially in networks with packet losses or larger sizes. However, to the best of our knowledge, no existing scenarios have incorporated both RLNC and XOR network coding to enhance data confidentiality against eavesdroppers while maintaining reliability and minimizing encoding latency.

Thus, in this paper, we introduce XOR-embedded network coding (XORE-NC), an innovative technique. It uses a two-stage encoding process, first applying systematic XOR network coding to packets in specified generation-size blocks. At this stage, XOR-encoded packets from each generation are woven into a new combined set, forming the outer code. These combined packets are then prepared for the second stage-RLNC inner encoding with GF(2) coefficients-enhancing the packets for robust transmission. This stage can dynamically adjust the encoding to make up for any lost or corrupted packets. Summarily, our model merges XOR and RLNC encoding to boost data transmission robustness and efficiency, and its outer code enhances RLNC inherent privacy, the primary focus of this paper. Results show that XORE-NC enhances reliability, reduces encoding delay, and lowers latency and throughput versus traditional RLNC, offering a viable solution for data confidentiality. Despite challenges with decoding delay, XORE-NC remains a promising approach for enhancing data security with efficient transmission in communication systems.

Ⅱ. Related Studies

Additional techniques like permutation encryption can be used to increase data confidentiality, even in cases of packet loss or corruption[6] . While[7] primarily focuses on evaluating and enhancing network performance parameters like throughput and delay for RLNC and RLNCV compared to traditional Store and Forward techniques, RLNCV also has significant implications for security. Its unique encoding properties, particularly the encryption of coding coefficients and the need for only a single element for decoding[7] , make it a potent tool for enhancing data confidentiality and resistance to eavesdropping in network communications.

The SPOC security framework, as discussed in [8], aims to enhance network coding inherent security through integration with standard cryptographic techniques. It modifies RLNC-based protocols at the source and receiver nodes, using two types of coefficients: unlocked coefficients (derived from the identity matrix for each coded packet) and locked coefficients (used for encoding and decoding but encrypted with keys available at the destination). This method allows for encrypting only the locked coefficients, as opposed to encrypting the entire data, thereby optimizing security and encryption efficiency.

Fulcrum network coding (FNC) framework employs a two-stage encoding process that enhances efficiency and flexibility by initially transforming n source packets into $n + r$ outer coded packets for error

correction using coefficients from $GF(2^8)$ or $GF(2^{16})$, and then encoding them in GF (2) independently, enhances coding flexibility, optimizes network throughput, and adds complexity to encoded data for increased security in diverse communication environments[9].

FNC can contribute to security in the context of SPOC by offering a simpler way to implement SPOC's concepts[9]. Specifically, in Fulcrum, the mapping of the outer decoder can act as a secret key (or part of it) between the source and destinations. This key, unlike in traditional SPOC implementations, does not need to be transmitted over the network along with the coded packets. By avoiding the transmission of this key, FNC reduces the overhead typically associated with SPOC, which often involves sending additional coding coefficients with each packet. This makes FNC a more efficient and potentially more secure alternative in the realm of practical network coding. However, the extra r redundant packets incurred by FNC, also increases the packet overhead when compared to traditional RLNC.

Thus, we propose XORE-NC, which maintains packet overhead equal to RLNC. Consider the example of XORE-NC in Section III, as simplified in Fig. 2. Compared to traditional RLNC, which encodes N original packets directly without adding overhead, FNC introduces additional r packets, increasing the total to $N+r$ and thereby adding overhead. SPOC may also introduce overhead through the encryption of coefficients. In contrast, XORE-NC aligns with traditional RLNC in terms of overhead. XORE-NC's outer code, once RLNC encoded, keeps the original count N. Therefore, while FNC and SPOC offer enhanced security features, XORE-NC can provide security improvement without the additional overhead, making it potentially the most efficient scheme compared to traditional RLNC. Consequently, our XORE-NC model positively impacts transmission efficiency by reducing latency and encoding time while enhancing throughput. Table 1 gives an overview the above schemes.

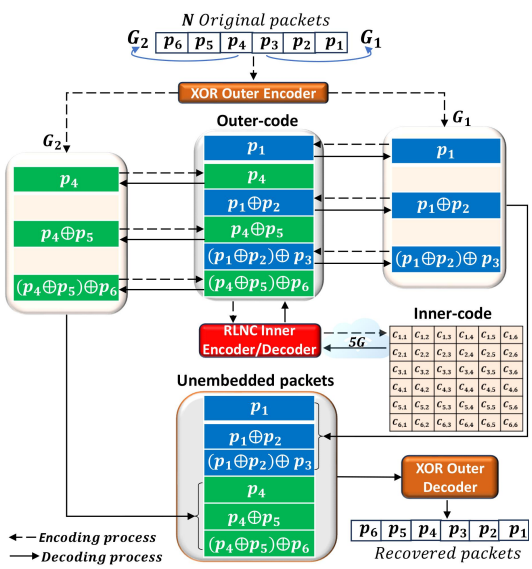


Fig. 2. XORE-NC encoding and decoding stages

Table 1. Overview of RLNC, FNC, SPOC, and XORE-NC Features

Scheme	Overhead	Security Features	Efficiency
Traditional RLNC	No additional overhead	Basic. Provides 1 security layer	High
Fulcrum Network Coding (FNC)	Increases packets to $N+r$	Enhanced security with additional packets, Provides 2 security layers	Reduces with increasing r packets
Secure Practical Network Coding (SPOC)	Add one extra encryption oflocked coefficients	Enhanced security with encrypted coefficients. Provides 2 security layers	Varies depending on encryption complexity
XORE-NC	No increase in packet count	Enhanced security without additional overhead. Provides 3 security layers	High, similar to traditional RLNC

Ⅲ. XORE-NC: Methodology

Our approach allows for the transmission of multiple generations of packets in pairs when the total number of packets to be transmitted is even while increasing data confidentiality. Otherwise, when there is an odd number of packets, the last block of packets that cannot be paired with another is sent using conventional RLNC. This approach, under small generation sizes, performs better in terms of minimizing latency, encoding latency, and data

reliability.

As illustrated in Fig. 3, two instances highlight the advantages of XOR-embedded packets in thwarting eavesdroppers. In the first case, decoding the original data proves challenging for eavesdroppers. Even if they decode the RLNC-encoded data, in the second instance, the XOR packet embedding adds a layer of protection that necessitates the embedding key for successful decoding. However, it is crucial to consider additional security measures to protect highly sensitive data. Therefore, the goal is to apply RLNC on embedded XOR-encoded packets to enhance performance. We simplify our approach by breaking it down into distinct steps for clarity as shown in Fig. 2.

1. We start with a message consisting of N packets, for example, $P1$ to $P6$. We divide the packets into two generations: generation one ($G1$) consists of the initial n packets, and generation two ($G2$) consists of the remaining ($N - n$) packets.
   Using XOR network coding, we perform systematic XOR operations within each generation separately. This process creates two sets of XOR-encoded packets: one for $G1$ and another for $G2$ using a field size of GF(2).
2. This method supports transmitting n generations simultaneously for an even number of packets, but switches to a single generation for the last block of packets when the total count is odd. To simplify, we use $n = 2$.
3. To enhance security, we embed $G1$ and $G2$ into each other, resulting in a fresh methodically embedded packets (outer-code). The embedding follows this pattern: $P1$, $P4$, $P1 \oplus P2$, $P4 \oplus P5$, $(P1 \oplus P2) \oplus P3$, $(P4 \oplus P5) \oplus P6$.
4. Subsequently, we apply RLNC to the embedded outer code, using a field size of GF(2) with its two elements, 0 and 1. This simplifies computations and reduces complexity in the RLNC inner coding stage.
5. At the receiver, the decoder collects at least k coded packets and employs progressive Gaussian elimination for outer-code recovery.
6. As illustrated in Fig. 2, the outer-code is subsequently disentangled and decoded via a straightforward XOR operation. This procedure comprises two distinct stages:
   - We extract all odd indexed packets from the decoding matrix and by performing a simple XOR operation, the first generation of packets $P1$ to $P3$ (i.e., $G1$) is recovered.
   - Similarly, all the indexed even packets are extracted and decoded, recovering the second generation of packets $P4$ to $P6$ (i.e., $G2$).
7. Finally, we pass the packets to the application layer in the original order, i.e., from $P1$ to $P6$. Find the whole process described in Fig. 4

In XORE-NC, the data encryption key mechanism includes systematic XOR network coding, interweaving XOR-coded packets from two generations to form an embedded outer code, and

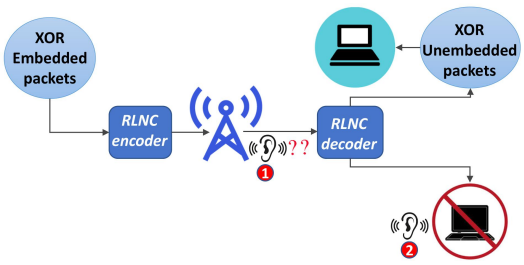


Fig. 3. XORE-NC architecture

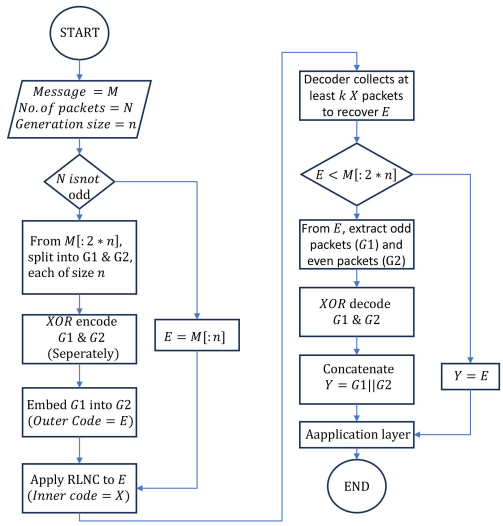


Fig. 4. Flow Chart representation of XORE-NC.

applying RLNC. The receiver decodes this outer code, uses an unembedding key to recover systematic XOR-coded packets, and then decodes these to obtain the original packets. These combined layers effectively enhance data security by complicating unauthorized access and thwarting eavesdroppers.

In the following section, we present a mathematical model to support this approach.

Ⅳ XORE-NC: Mathematical Model

To create a mathematical model for the approach, we break down the process into key components and describe them mathematically. The approach involves message splitting into generations, XOR operations, and Random linear network coding operations.

4.1 System Parameters

We define some variables:

ε : A range of erasure probabilities, $\varepsilon \in [0.1, 0.8]$

G : Generation size.

$\mathbb{P}$: Packet size in bits.

N : Number of packets.

M : Original message

$f\,lag$: Indicates whether the last message block is of size G (flag = 1) or $2G$ (flag = 0).

4.2 XORE-NC: Encoding Process

Consider a message M consisting of N packets, each with a size of P bits per packet.

$$M \in \{0,1\}^{N \times \mathbb{P}} \tag{1}$$

Equation (1) denotes that M is a binary matrix with dimensions $N \times$ P. Based on the generation size, G, for each block in M starting at index i:

$$\begin{aligned} G1[0] &= M[i] \\ G2[0] &= M[i+G] \end{aligned} \tag{2}$$

Subsequently, for each index j from 1 to $G-1$:

$$\begin{aligned} G1[j] &= G1[j-1] \oplus M[i+j] \\ G2[j] &= G2[j-1] \oplus M[i+G+j] \end{aligned} \tag{3}$$

The embedded outer coded packet, E, is constructed by interleaving $G1$ and $G2$:

$$E[k] = \begin{cases} G1[k/2] & \text{if } k \text{ is even} \\ G2[(k-1)/2] & \text{if } k \text{ is odd} \end{cases} \tag{4}$$
$$\text{for } k = 1, 2, \ldots, 2G.$$

Then, the flag type is set to zero, $f\,lag = 0$ since the number of packets in $E[k]$ is a multiple of $2G$.

4.2.1 Handling the Last Block

If the number of packets, N, is not a multiple of $2G$, the last block will be of size G. In this situation, the embedded outer code, E, is constructed by just assigning the last message block to E given by equation (5). This is a unique scenario where there is no need for disentangling during the decoding process.

$$E[k_l] = M[l] \tag{5}$$

Where l is the last unpairable packet of the message block. Then, the flag is set to one, $f\,lag = 1$. Note that in this scenario, the embedded packet E[kl] lacks a pair. From equations (4 and 5), equations (6 gives a general expression for the embedded outer code.

$$E = \begin{cases} E[k] & \text{if flag=0} \\ E[k_l] & \text{if flag=1} \end{cases} \tag{6}$$

4.2.2 Applying RLNC encoding over the embedded message

Random Linear Network Coding involves generating linear combinations of packets with coefficients chosen randomly from a finite field $\mathbb{F}$. Coefficients, denoted as C, are generated and the size of C depends on the generation size being encoded (i.e., $2G$ or G). The RLNC inner code, X, is then computed by performing matrix-vector multiplication using the coefficients C and the embedded outer code E :

$$X_i = \sum_{j=1}^{R} C_{ij} \cdot E_j \tag{7}$$

Where R is either G or $2G$. X_i is the i-th encoded packet. C_{ij} represents the coefficient of packet E_j in encoded packet X_i. The summation limit extends up to $2G$ when E_j is a multiple of $2G$, and it extends up to G when E_j is not a multiple of $2G$.

4.3 XORE-NC: Decoding process

The encoded packet X obtained in equation (7) is passed to the RLNC decoder, which calculates the extended row-reduced echelon form (Ext_{rref}) of the augmented matrix given by equation (8).

$$Ext_{rref} = \text{RowReducedForm}([X|E]) \tag{8}$$

Based on equation (7), the embedded outer code, E_j, can be solved with equation (9) and extracted as presented by equation (10).

$$E_j = \sum_{j=1}^{R} C_{ij}^{-1} \cdot X_i \tag{9}$$

However, initially, the decoding process is carried out based on the flag type of the received RLNC inner code, X. To extract the outer code $E\,j$ computed by equation (9) from the $E\ xtrre\ f$ in equation (8), we use equation (10). The extraction starts from the G-th position for $f\,lag$ = 1, and from the 2G-th position for $flag$ = 0.

$$E = \begin{cases} Ext_{rref}[2G:] & \text{if flag=0} \\ Ext_{rref}[G:] & \text{if flag=1} \end{cases} \tag{10}$$

Where $Ext_{rre\ f}[2G:]$ denotes a subset of $Ext_{rre\ f}$ from index 2G to the end, and $E\ xtrre\ f\,[G:]$ denotes a subset from index G to the end. We first consider the case for which $flag$ = 0. Given an array E with n elements, where E = [e_0, e_1, e_2, ⋯; e_{n-1}], and n is even (i.e., $2G$), you can distribute this array as:

$$\begin{aligned} L_1 &= [e_0, e_2, e_4, \ldots, e_{n-2}], \\ L_2 &= [e_1, e_3, e_5, \ldots, e_{n-1}] \end{aligned} \tag{11}$$

Where L_1 and L_2 consists of elements at even indices and at odd indices respectively. Then, to reconstruct the unembedded coded packet, a concatenated list (L_C) is formed by concatenating L_1 and L_2 given by:

$$L_C = [e_0, e_2, e_4, \cdots, e_{n-2}, e_1, e_3, e_5, \cdots, e_{n-1}] \tag{12}$$

Summarily, the operation can be expressed as:

$$L_C = Distribute(E) \tag{13}$$

To reconstruct the original message (i.e., M), we initialize a matrix for decoded message y of size ($2G$, $\mathbb{P}$) with all elements initialized to zeros:

$$y[i, j] = 0, \text{ for } 1 \le i \le 2G \text{ and } 1 \le j \le \mathbb{P} \tag{14}$$

Matrix wise, equation (14) is expressed thus,

$$y = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix} \tag{15}$$

Set the first row of y to the values in the first row of unembadded packets, L_C :

$$y[1, j] = L_C[1, j], \text{ for } 1 \le j \le \mathbb{P} \tag{16}$$

Set the G-th row of the decoded message, y to the values in the G-th row of L_C :

$$y[G, j] = L_C[G, j], \text{ for } 1 \le j \le \mathbb{P} \tag{17}$$

For each i from 1 to G - 1:

$$G1_{rx} = y[i - 1] \oplus L_C[i] \tag{18}$$

$$y[i] = G1_{rx} \tag{19}$$

$$G2_{rx} = y[G + i - 1] \oplus L_C[G + i] \tag{20}$$

$$y[G + i] = G2_{rx} \tag{21}$$

Let Y_{2G} = [y_1, y_2, y_3, ⋯; y_N], where Y_i represents the i-th decoded block, N is the total number of decoded blocks and N is a multiple of $2G$. The complete

decoded message, denoted as Y_{2G}, is obtained by concatenating all the decoded blocks Y_{2Gi} as seen in equation (22).

$$Y_{2G} = y_1 \| y_2 \| y_3 \| \cdots \| y_N \tag{22}$$

In this expression, $\|$ represents the concatenation operation. The resulting decoded message (Y_{2G}), contains all the decoded information from the individual blocks and represents the final reconstructed message.

Secondly, consider the case for which $flag$ = 1. In this case the generation of G packets has no pair and was sent unembedded. Hence, its decoding process is completed after the initial decoding process. Thus, from equation (10), $E = Ext_{tref}[G:]$ and

$$L_C = E \tag{23}$$

As a result, in cases where the last block of message is not a multiple of $2G$ (i.e., the number of packets, N, is odd), the decoded packet, denoted as Y_G, can be explicitly represented by equation (24).

$$Y_G = L_C \tag{24}$$

The complete decoded message Y for this scenario can be expressed as the concatenation of Y_{2G} and Y_G as defined in equation (25);

$$Y = Y_{2G} \| Y_G \tag{25}$$

Otherwise, if the number of packets N is a multiple of $2G$, then Y is given by;

$$Y = Y_{2G} \tag{26}$$

4.4 Data confidentiality optimization

Data confidentiality is vital in modern communication systems, particularly when safeguarding sensitive information from eavesdroppers is crucial. XORE-NC offers strategies and techniques to enhance data confidentiality in network coding, supported by relevant mathematical expressions.

4.4.1 Embedding Key Mechanism

To enhance data confidentiality in XORE-NC, we employ an embedding key mechanism, adding an extra layer of protection to the encoded data, making it harder for unauthorized access. This key is integrated during XOR-embedded packet generation, creating complex interdependencies among packets as in equations (4), (5), and (6). For example, when XORing packets $P1$ and $P2$, it involves additional XOR operations with $P3$ and $P4$. Decoding this data requires not only retrieving XOR-encoded packets but also possessing the embedding key to navigate these intricate dependencies. This embedding key significantly heightens the challenge for eavesdroppers in deciphering the original data. Even if they decode the RLNC-encoded data, this approach adds security by obscuring packet relationships and requiring knowledge of the embedding key for successful decoding.

4.4.2 Use of Random Linear Network Coding (RLNC)

In addition to the embedding key mechanism, XORE-NC utilizes RLNC to enhance data confidentiality. RLNC adds a layer of complexity by creating linear combinations of packets using coefficients selected randomly from a finite field (as illustrated in equations 7 and 9). These coefficients, denoted as $Ci\,j$, are applied during the encoding process to the XOR-embedded packets, as shown in equation 7. By introducing randomness and complexity, these coefficients make it even more challenging for eavesdroppers to reverse-engineer the original content. Consequently, data confidentiality is improved by increasing the difficulty of unauthorized decryption.

Ⅴ. Performance Evaluation

In network communication systems, optimizing data transfer is critical. XORE-NC transmits twice as many packets as RLNC for generation sizes $G \in 5$, 10 and packet sizes $\mathbb{P} \in$ 1B, 2B, suggesting a potential for increased throughput.

In Fig. 5 with G = 5, while RLNC may seem to

require less added transmission for 1-byte packets than XORE-NC, the latter compensates by transmitting twice the number of generations, effectively doubling the secured data per transmission. This increased data protection does not proportionately raise the added transmission, affirming XORE-NC efficiency. When G is increased to 10, although the efficiency gains for 1-byte and 2-byte packets are comparable for both schemes, XORE-NC dual-generation transmission provides enhanced confidentiality without significantly impacting overhead, especially at higher erasure probabilities - the likelihood that a transmitted packet will be lost and not received by the destination. This

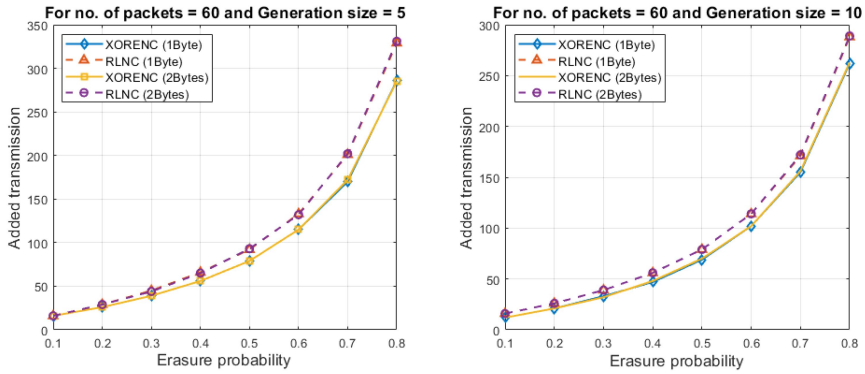


Fig. 5. XORE-NC overhead compared to traditional RLNC.

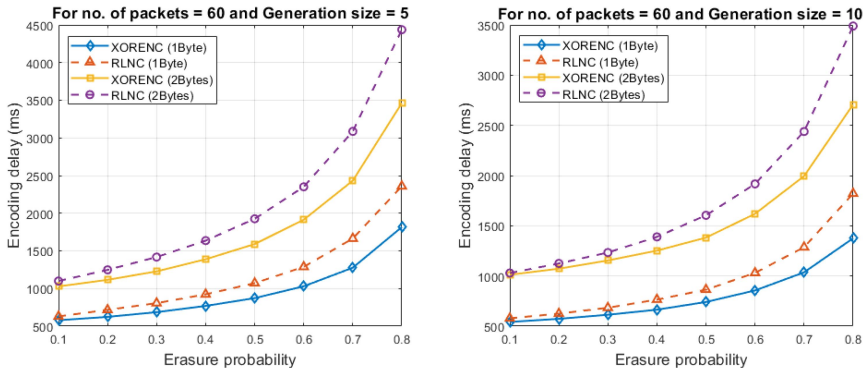


Fig. 6. XORE-NC encoding delay compared to traditional RLNC.

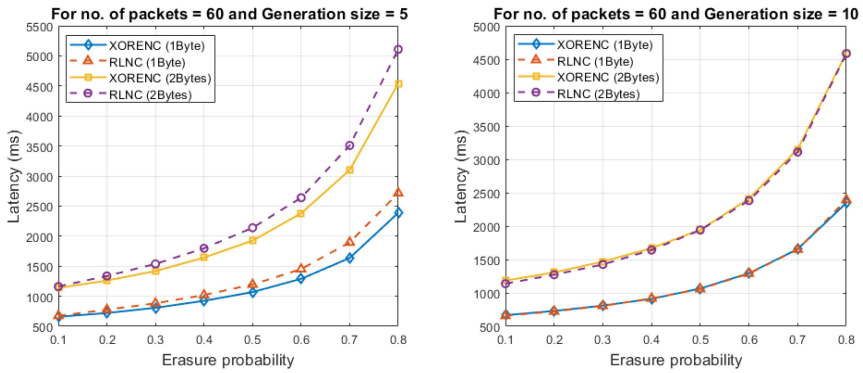


Fig. 7. XORE-NC latency compared to traditional RLNC.

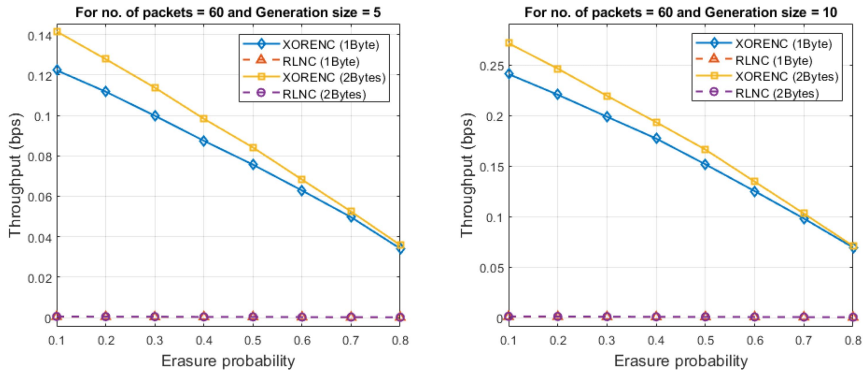


Fig. 8. XORE-NC throughput compared to traditional RLNC.

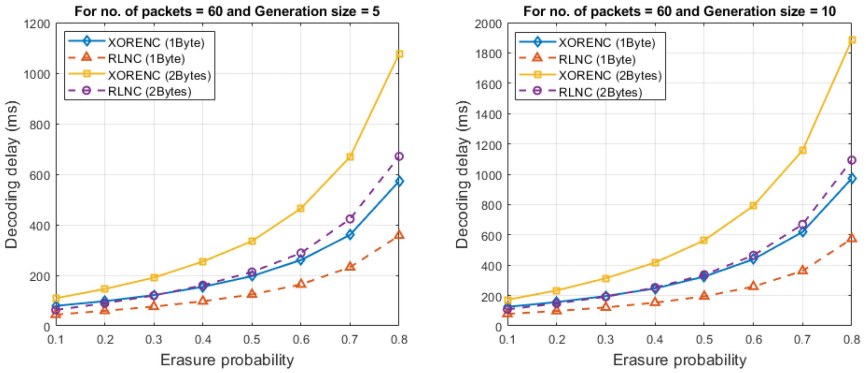


Fig. 9. XORE-NC decoding delay compared to traditional RLNC.

balance of security and efficiency positions XORE-NC as a superior choice for secure, high-throughput communications under variable network conditions

As depicted in Fig. 6, XORE-NC exhibits the lowest encoding delay for 1-byte packets across all erasure probabilities when G = 5, underscoring its superior efficiency, which encompasses not only speed but also the heightened data confidentiality integral to its design. Despite transmitting secured data that requires more complex encoding operations, XORE-NC maintains a lower delay than RLNC for both 1-byte and 2-byte packets. When the generation size increases to G = 10, we observe an expected rise in encoding delay for both schemes due to larger data volumes and the additional security computations. Nevertheless, XORE-NC continues to demonstrate the quickest encoding times, especially for smaller packets. This trend suggests that XORE-NC's approach to data confidentiality, while introducing additional processing steps, still allows it to outperform RLNC in terms of encoding speed, making it highly suitable for scenarios demanding both swift and secure data transmission.

Fig. 7 shows overall latency, crucial from encoding to decoding, under various scenarios. For G = 5, the latency trends upward steadily with the rise in erasure probability. Notably, XOR-NC shows commendable efficiency, as it does not proportionally increase latency despite transmitting twice as many packets compared to RLNC with the same byte size. This suggests that the efficient encoding utilized by XOR-NC does not significantly burden transmission time. Moving to G = 10, latency rises for both schemes, but XOR-NC and RLNC show comparable latency, regardless of payload size. Despite larger payloads increasing latency, the graph shows XOR-NC and RLNC have similar latency rises,

indicating no significant advantage for XOR-NC encoding over RLNC in this case. However, Fig. 8 reveals XOR-NC capacity to handle larger data volumes and more challenging erasure conditions effectively, underscoring its potential for secure, high-throughput environments. The data thus underscores XOR-NC as a suitable candidate for scenarios demanding robust security without compromising on transmission efficiency. XORE-NC's latency performance may decrease with fewer packets, e.g., from 60 to 20, indicating its suitability for high data rate scenarios.

Fig. 9 shows that the decoding delay for both XORENC and RLNC increases with erasure probability for 1 and 2-byte packets and generation sizes of 5 and 10. The sharper rise in XORE-NC decoding delay, particularly for larger packets, might be attributed to the more complex data unscrambling necessitated by its enhanced confidentiality features. Although XORE-NC encoding and transmission are efficient, the intricacies of its secure encoding impact decoding performance under higher loss conditions. Future enhancements, such as integrating a sliding window protocol, could potentially mitigate this by streamlining the decoding process, even in scenarios with high erasure probabilities, thus further optimizing XORE-NC robustness.

Ⅵ. Conclusions

Based on the presented findings, XOR-Embedded Network Coding (XORE-NC) offers a promising approach to enhance data confidentiality in network coding. By employing an embedding key mechanism and Random Linear Network Coding (RLNC), it adds layers of protection, making unauthorized access and decryption more challenging. XORE-NC demonstrates improved reliability, reduced encoding delay, and lower latency and throughput compared to traditional RLNC. However, it requires further research to address higher decoding delay. In conclusion, XORE-NC presents a robust solution to bolster data confidentiality in communication systems while maintaining efficient data transmission.

References

[1] Stanford University IT, *Risk classifications*, Retrieved Jun. 16, 2023, from https://uit.stanford.edu/guide/riskclassifications

[2] Havartd University, *By data security level*, Retrieved Jun. 16, 2023, from https://policy.security.harvard.edu/view-data-security-level

[3] D. T. Hai, "Re-designing dedicated protection in transparent WDM optical networks with XOR network coding," in *2018 Advances in Wireless and Optical Commun.* (*RTUWO*), pp. 118-123, 2018. (https://doi.org/10.1109/RTUWO.2018.8587873)

[4] G. Savva, K. Manousakis, and G. Ellinas, "Network coding-based routing and spectrum allocation in elastic optical networks for enhanced physical layer security," *Photonic Netw. Commun.*, vol. 40, pp. 160-174, Aug. 2020. (https://doi.org/10.1007/s11107-020-00893-w)

[5] H. Noura, S. Martin, and K. A. Agha, "An efficient lightweight security algorithm for random linear network coding," in *2014 11th SECRYPT*, pp. 1-7, 2014.

[6] S. Yao, J. Chen, R. Du, L. Deng, and C. Wang, "A survey of security network coding toward various attacks," in *2014 IEEE 13th Int. Conf. Trust, Secur. and Privacy in Comput. and Commun.*, pp. 252-259, 2014. (https://doi.org/10.1109/TrustCom.2014.35)

[7] P. R. Mane, S. G. Adiga, and M. S. Kumar, "Performance evaluation of random linear network coding using a Vandermonde matrix," *Phys. Commun.*, vol. 10, pp. 24-30, 2014. (https://doi.org/10.1016/j.phycom.2013.11.008)

[8] J. P. Vilela, L. Lima, and J. Barros, "Lightweight security for network coding," in *2008 IEEE Int. Conf. Commun.*, pp. 1750-1754, 2008. (https://doi.org/10.1109/ICC.2008.336)

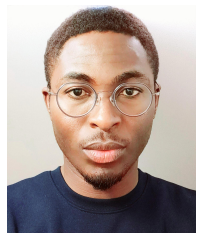
[9] D. E. Lucani, M. V. Pedersen, and D. Ruano, et al., "Fulcrum: Flexible network coding for heterogeneous devices," *IEEE Access*, vol. 6,

pp. 77890-77910, 2018.
(https://doi.org/10.1109/ACCESS.2018.2884408)

[10] L. Wang, Y. Liu, J. Xu, J. Yin, L. Xu, and Y. Yang, "Network coding for reliable video distribution in multi-hop device-to-device communications," *EURASIP J. Wireless Commun. and Netw.*, vol. 2020, pp. 1-21, 2020.
(https://doi.org/10.1186/s13638-020-01869-0)

[11] P. Enenche, D. H. Kim, and D. You, "Network coding as enabler for achieving URLLC under TCP and UDP environments: A survey," *IEEE Access*, vol. 13, pp. 76647-76674, 2023.
(https://doi.org/10.1109/ACCESS.2023.3297137)

Patrick Enenche



Feb. 2013 : B.Eng. Federal University of Technology Minna
Aug. 2018 : M.Eng. Federal University of Technology Minna
Sep. 2021~Current : Ph.D. Student, Hannam University

<Research Interest> Ultra-Reliable and Low Latency Commuications, Network Coding.
[ORCID:0000-0003-0895-3159]

Dongho You

Feb. 2012 : B.Eng. Seoul National University of Science and Technology
Feb. 2014 : M.Eng. Seoul National University of Science and Technology
Aug. 2018 : Ph.D. Seoul National University of Science and Technology
Sep. 2018~Feb. 2021 : Senior Researcher, Technische Universität Dresden
Mar. 2021~Current : Assistant Professor, Hannam University

<Research Interest> Immersive Media, Communication Networks.
[ORCID:0000-0003-3724-3244]