

논문 2026-3-19 <http://dx.doi.org/10.29056/jsf.2026.09.19>

정점 라벨 노출 및 간선 가중치 증가 방지를 위한 난수 마스킹을 활용한 별 그래프 기반 암호화에 대한 연구

윤성철*, 김수현**†, 조용준*** 이임영**†

A Study on Star Graph Encryption using Randomized Masking for Preventing Vertex-Label Exposure and Edge-Weight Growth

SeongCheol Yoon*, Su-Hyun Kim**†, YongJoon Joe*** Im-Yeong Lee**†

요 약

그래프 기반 암호화는 평문을 그래프의 정점 및 간선에 대응시켜 그래프의 라벨과 가중치를 암호문으로 사용하는 암호화 기법이다. 그래프 기반 암호화에는 Ni et al., Bokhary et al., Ali et al. 등 다양한 방법이 존재한다. 그중 Ali et al.의 별 그래프 기반 방식은 문자를 서로 다른 소수 쌍으로 매핑하고, 소수의 곱을 정점 라벨로 구성하여 정점 라벨에서 위치에 따라 특정 값을 차감해 간선 가중치를 생성한다. 그러나 이 방식은 동일 문자가 동일한 라벨로 표현되고, 간선 가중치의 비트 길이가 문자 위치에 따라 크게 증가한다. 이를 해결하고자 논문에선 새로운 별 그래프 기반 암호화 방식을 제안한다. 제안방식은 정점 라벨을 난수인 nonce 및 비밀키 등으로 난수화하고, 범위가 제한된 간선 가중치와 보조 라벨로 분리한다. 이로써 제안방식은 모든 간선 가중치를 일정 범위 내로 제한할 수 있고, 고정된 문자 집합 환경에서 전체 암호문 크기를 선형으로 구성할 수 있다. 또한, 정점 라벨의 반복 패턴 노출을 완화하며 암호문의 무결성 또한 제공할 수 있다.

Abstract

Graph-based encryption maps plaintext to graph vertices and edges, using their labels and weights as ciphertext. Various schemes have been proposed by Ni et al., Bokhary et al., and Ali et al. Ali et al.'s star graph-based scheme maps characters to distinct prime pairs and uses their products as vertex labels. Edge weights are generated by subtracting position-dependent values from these labels. However, identical characters yield identical labels, and edge-weight bit lengths grow substantially with character position. To address these limitations, this paper proposes a new star graph-based encryption scheme. It randomizes vertex labels using a random nonce and a secret key, then splits them into bounded edge weights and auxiliary labels. This bounds all edge weights and achieves linear ciphertext size for a fixed character set. The scheme also mitigates vertex-label pattern leakage and provides ciphertext integrity.

한글키워드 : 그래프 기반 암호화, 간선 가중치, 정점 라벨 난수화, 선형 크기 암호문, 마스킹

keywords : Graph-based encryption, edge weights, vertex label randomization, linear-size ciphertext, masking

* 순천향대학교 SW융합학과

** 순천향대학교 컴퓨터소프트웨어공학과

*** 엘에스웨어(주)

† 교신저자: 김수현, 이임영 (email: {kimsh, imylee}@sch.ac.kr)

접수일자: 2026.09.14. 심사완료: 2026.09.19.

게재확정: 2026.09.20.

1. 서론

현대 사회에서는 네트워크, 소셜 관계, 소스코드 연결 구조 및 지식 데이터와 같이 객체 사이의 복잡한 관계를 표현하기 위해 그래프를 활용한다[1]. 또한, 암호화는 공개된 통신 환경에서 평문을 인가되지 않은 객체가 해석하기 어려운 암호문으로 변환하기 위한 정보보호 기술이다[2, 3]. 그래프는 정점과 간선으로 객체와 객체 사이의 관계를 직관적으로 표현할 수 있으며, 그래프의 구조와 라벨을 활용하여 데이터를 변환하는 다양한 연구가 수행되고 있다. 이와 같은 연구 중 그래프 기반 암호화는 평문 데이터를 그래프의 정점 혹은 간선에 대응시키고, 그래프의 구조와 라벨을 암호문으로 활용하는 방식이다[4].

그래프 기반 암호화 방식에는 Ni et al.[5], Bokhary et al.[6], Ali et al.[4] 등 다양한 방식이 존재한다. 그중 Ali et al.[4]은 corona graph, complete bipartite graph 및 별 그래프를 활용한 그래프 기반 암호화 방식을 제안했다. 그중 별 그래프 기반 방식은 UFD(Unique Factorization Domain)의 유일 분해 성질을 활용하여 각 문자를 행 소수와 열 소수의 쌍으로 매핑한 뒤, 두 소수의 곱을 외곽 정점 라벨로 구성한다. 이후 각 정점 라벨 a_i 에서 증가하는 10^i 를 차감하여 간선 가중치 $w_i^{Ali} = a_i - 10^i$ 를 생성하고, 라벨이 부여된 별 그래프를 수신자에게 전송한다.

그러나 기존 방식은 문자 위치 i 가 증가할수록 10^i 의 크기가 증가하므로 간선 가중치의 절댓값과 이를 표현하기 위한 비트 길이가 증가한다. 특히 10^i 자체의 수치 크기는 지수적으로 증가하고, 해당 정수를 표현하기 위한 비트 길이는 $O(i)$ 가 되므로 전체 간선 가중치의 표현 크기는 $O(n^2)$ 이 된다. 또한, 수신자가 수행하는 복원 연산은 공격자도 동일하게 수행할 수 있어 간선

가중치만으로 정점 라벨이 직접 복원되는 문제가 존재한다. 동일한 문자는 동일한 소수 쌍과 정점 라벨로 표현되기 때문에 반복 문자와 빈도 정보가 노출될 가능성 또한 존재한다.

본 논문에서는 이러한 문제를 해결하기 위해 정점 라벨을 먼저 난수화하고, 난수화된 라벨을 몫과 나머지로 분해하여 별 그래프에 배치하는 새로운 그래프 기반 암호화 방식을 제안한다. 제안방식은 PRF(Pseudorandom Function)[7]를 사용하여 암호화마다 생성되는 nonce와 문자 위치에 대응하는 마스크를 생성한다. 이후 정점 라벨과 마스크를 모듈러 덧셈하여 난수화된 라벨을 생성하고, 이를 범위가 제한된 간선 가중치와 보조 라벨로 분해한다. 또한 HKDF(HMAC-based Extract-and-Expand Key Derivation Function)[8, 9]를 통해 마스크 생성 키와 인증 키를 분리하고, HMAC(Keyed-Hash Message Authentication Code) 기반 MAC(Message Authentication Code)[10]을 적용하여 암호문의 무결성을 검증한다.

본 논문의 기여는 다음과 같다.

- 정점 라벨 난수화: 정점 라벨을 암호문에 직접 반영하지 않고, nonce와 문자 위치에 대응하는 PRF 기반 마스크를 이용하여 난수화한다. 이를 통해 동일한 문자가 반복되더라도 암호화 nonce와 위치가 다르므로 서로 다른 그래프 라벨이 생성되도록 한다.
- 간선 가중치 범위 제한: 난수화된 정점 라벨을 몫과 나머지로 분해하고, 나머지를 간선 가중치로 사용하며 몫을 보조 라벨로 구성한다. 따라서 모든 간선 가중치는 $0 \leq w_i < M$ 의 고정된 범위를 만족한다.
- 암호문 무결성 제공: nonce, 그래프 라벨, 공개 파라미터를 MAC의 입력으로 하여 공격자가 간선 가중치나 보조 라벨을 변경하면 복호화 이전에 이를 탐지할 수 있도록 한다.

- 선형 크기의 암호문 구성: 고정된 문자 집합과 공개 파라미터를 사용할 경우 문자별 그래프 라벨의 표현 크기가 상수로 제한되므로, 전체 암호문 크기를 $O(n)$ 으로 구성한다.

2. 관련 연구

본 장에서는 제안방식과 관련된 기존 그래프 기반 암호화 연구와 제안방식에서 사용하는 암호학적 난수화 및 인증 기법에 대해 설명한다. 그래프 기반 암호화 관련 연구로는 Ni et al.[5], Bokhary et al.[6] 및 Ali et al.[4]이 존재한다. 이에 대한 설명은 아래에서 자세히 설명한다.

2.1 Ni et al.

Ni et al.[5]은 그래프 이론과 대수적 성질을 활용하여 비밀 메시지를 라벨 그래프로 변환하는 세 가지 암호화 방식을 제안했다. 해당 연구에서 사용하는 그래프는 corona graph, bipartite graph 및 별 그래프이다. 송신자는 각 방식에서 평문 문자를 수치로 변환한 뒤, 그래프의 정점과 간선에 계산된 값을 라벨링하여 수신자에게 전달한다. 수신자는 전달받은 그래프의 라벨을 정렬하거나 역연산하여 원래 평문을 복원한다.

Corona graph 방식은 문자값과 서로소인 값을 선택한 뒤 모듈러 역원을 계산하여 정점 라벨을 구성한다. Bipartite graph 방식은 UFD(Unique Factorization Domain)의 유일 분해 성질과 소수 기반 인코딩 테이블을 사용하여 문자를 소수 쌍으로 변환한다. 별 그래프 방식은 문자값을 외곽 정점에 배치하고 문자 위치에 따라 증가하는 값을 차감하여 간선 가중치를 생성한다.

Ni et al.의 방식은 서로 다른 그래프 구조와 대수적 성질을 이용하여 평문을 그래프 라벨 형태로 변환할 수 있다는 특징이 존재한다. 그러나 별 그래프 방식은 문자 위치가 증가할수록 간선

가중치의 표현 크기가 증가하며, 간선 가중치와 문자 위치를 알고 있는 경우 정점 라벨을 다시 복원할 수 있다. 또한, 해당 연구에서는 공유 키의 필요성을 언급하지만, 별도의 암호문 인증 절차는 포함하지 않는다.

2.2 Bokhary et al.

Bokhary et al.[6]은 통신 네트워크에서의 보안성과 오류 저항성을 향상하기 위해 Cartesian product graph 기반 방식 한 가지와 complete bipartite graph 기반 방식 두 가지를 제안했다. 해당 연구는 그래프의 정점과 간선에 평문과 관련된 값을 배치한 뒤, 이를 행렬 또는 가중치 형태로 구성하여 암호값을 생성한다. 또한 각 알고리즘의 시간 복잡도와 공간 복잡도를 Big-O 표기법으로 분석했다는 특징이 존재한다.

Cartesian product graph 방식은 평문 길이에 따라 그래프를 구성하고, 문자값을 여러 정점과 간선에 분산한 뒤 계산된 정점 가중치를 행렬 형태로 구성한다. Complete bipartite graph 기반 방식은 문자값과 그래프 위치를 이용하여 정점과 간선 라벨을 생성하거나, 문자값을 몫과 나머지로 분해한 뒤 그래프 라벨에 배치한다. 일부 방식에서는 대칭키 행렬과의 연산을 통해 암호값을 생성하며, 수신자는 이에 대한 역연산을 수행하여 원래 문자를 복원한다.

Bokhary et al.의 방식은 행렬 내부의 반복 정보를 전송 오류를 식별하는 데 활용할 수 있으며, 그래프 구조별 시간·공간 복잡도를 분석했다. 그러나 일부 방식은 길이 n 의 평문에 대해 n^2 개의 정점 또는 암호값을 생성하므로 메시지 길이가 증가할수록 그래프와 암호문의 크기가 증가할 수 있다. 또한, 별 그래프 기법에서 간선 가중치가 크게 증가하는 문제가 존재하며, 동일 정점 라벨의 위치별 난수화는 제공하지 않는다.

2.3 Ali et al.

Ali et al.[4]은 corona graph, complete bipartite graph 및 별 그래프를 활용한 그래프 기반 암호화 방식을 제안했다. Corona graph 방식은 평문에 shift cipher를 적용한 뒤 문자값과 서로소인 값을 선택하고, 이에 대한 모듈러 역원을 그래프에 배치한다. Complete bipartite graph 방식은 소수 기반 인코딩 테이블과 UFD의 유일 분해 성질을 사용하여 문자별 소수 쌍을 생성하고, 이를 정점과 간선 라벨로 구성한다.

별 그래프 방식은 각 문자를 행 소수 r_i 와 열 소수 c_i 의 쌍으로 매핑한 뒤 $a_i = r_i c_i$ 와 같이 정점 라벨을 생성한다. 이후, $w_i^{Ali} = a_i - 10^i$ 와 같이 각 정점 라벨에서 문자 위치에 따라 간선 가중치를 생성한다.

수신자는 간선 가중치에 10^i 를 다시 더해 정점 라벨을 복원하고, 소인수분해와 인코딩 테이블을 통해 원래 문자를 획득한다.

Ali et al.은 별 그래프의 구성 및 라벨링이 corona graph와 complete bipartite graph보다 단순하므로 암호화와 복호화를 효율적으로 수행할 수 있다고 설명한다. 또한, 별 그래프 구성의 시간 복잡도를 $O(n)$ 으로 제시했다. 그러나 해당 분석은 각 간선 가중치를 표현하기 위해 필요한

비트 길이를 별도로 고려하지 않는다. 정점 라벨의 크기가 고정된 범위에 있을 경우 문자 위치가 증가함에 따라 10^i 의 크기도 증가하므로 각 간선 가중치의 비트 길이는 문자 위치에 따라 증가한다. 이에 따라 전체 간선 가중치의 표현 크기는 $O(n^2)$ 이 된다. 또한, 간선 가중치에 10^i 를 다시 더하면 정점 라벨을 직접 복원할 수 있으며, 동일한 문자는 동일한 소수 쌍과 정점 라벨로 표현된다는 한계점이 존재한다.

2.4 기존 그래프 기반 암호 연구 비교

앞에서 설명한 세 연구는 그래프의 정점, 간선, 라벨 및 행렬 구조 등을 활용하여 평문을 변환한다는 공통점이 존재한다. 그러나 각 기법에서 활용되는 그래프 구조와 변환 연산이 상이하며, 기법별 한계점이 존재한다. 이에 대한 비교표는 표 1과 같다.

3. Preliminaries

본 장에선 제안방식의 이해에 필요한 암호학적 배경지식을 설명한다. 본 연구에선 위치별 마스크 생성을 위해 PRF를, 전송되는 암호문의 인증 및 무결성을 위해 MAC을 활용한다.

표 1. 기존 방식 비교표
Table 1. Comparison Table of Existing Schemes

연구	그래프 구조	주요 변환 방식	한계점
Ni et al.[5]	Corona, Bipartite, 별 그래프	Shift, 모듈러 역원, UFD 기반 라벨링	라벨 난수화·인증 부재, 가중치 증가
Bokhary et al.[6]	Cartesian product, Complete bipartite graph	라벨 합산, 행렬 연산, 몫·나머지 분해	일부 방식의 $O(n^2)$ 크기 증가
Ali et al.[4]	Corona, Complete bipartite, 별 그래프	소수 쌍 라벨링, 위치 기반 가중치	라벨 복원·반복 노출, $O(n^2)$ 크기

3.1 Pseudorandom Function

PRF는 무작위로 선택된 난수인 비밀키에 의해 결정된 함수가 동일한 정의역과 공역을 갖는 난수 함수와 계산적으로 구분하기 어려운 함수를 의미한다[7]. 일반적으로 PRF는 결정론적 알고리즘으로서 같은 값을 넣게 되면 동일한 값을 출력한다. 단, 비밀값을 모르는 공격자는 PRF로 생성된 난수와 무작위 난수값을 구분하기 어렵다.

3.2 Message Authentication Code

MAC은 서로 비밀키를 공유한 송신자 및 수신자 간 메시지의 무결성과 인증을 위해 사용하는 암호학적 기법이다. HMAC은 해시 함수와 비밀키를 결합하여 인증 태그를 생성하는 대표적인 MAC 구조이다[10]. 암호문에 MAC을 추가해 수신자에게 보낼 경우 공격자가 암호문을 변조하면 MAC을 검증하여 암호문의 변조 여부를 탐지할 수 있다. 특히 암호화된 값을 먼저 생성한 뒤 해당 암호문을 인증하는 Encrypt-then-MAC 구성은 암호화와 인증을 일반적으로 결합하기 위한 효과적인 구조이다[11, 12].

4. 요구사항 정의

본 장에서는 기존 방식에서 발생하는 문제점을 방지하기 위해 만족해야 하는 요구사항에 대해 설명한다. 만족해야 하는 요구사항은 정점 라벨 기밀성, 간선 가중치 범위 제한, 암호문 무결성, 연산 및 저장 효율성이다. 이에 대한 요구사항 정의는 다음과 같다.

- 정점 라벨 기밀성: 기존 방식의 경우 간선 가중치와 문자 위치를 활용해 공격자가 a_i 를 직접 획득할 수 있었다. 이를 방지하기 위해 정점 라벨을 그래프에 배치하기 전, 비밀키 기반의 난수 마스킹을 적용해야 한다. 또한, 동일 문자가 반복되어 마스킹돼도 nonce와

위치의 차이가 다른 서로 다른 난수 라벨이 생성되도록 해야 한다.

- 간선 가중치 범위 제한: 기존 방식의 간선 가중치는 문자 위치가 증가할수록 표현 크기가 증가한다. 즉, 모든 간선 가중치를 사전에 정의한 범위인 $[0, M-1]$ 로 제한해야 한다. 이를 통해 메시지 길이와 관계없이 각 간선 가중치의 최대 비트 길이를 일정하게 유지할 수 있도록 해야 한다.
- 암호문 무결성: 공격자가 암호문 전송 과정에서 간선 가중치, 보조 라벨, nonce 등을 변경할 경우 검증자가 이를 알 수 있어야 한다.
- 효율성: 메시지 길이 n 에 대해 암호화와 복호화를 선형 시간에 수행해야 하며, 고정된 문자 집합에서는 암호문 크기 또한 $O(n)$ 을 만족해야 한다.

5. 제안방식

본 장에서는 본 논문의 제안방식에 대해 설명한다. 제안방식은 기존 방식과 달리 정점 라벨을 PRF로 마스킹하여 난수화를 하며, 난수화된 값을 범위가 제한된 간선 가중치와 보조 라벨로 분리한다. 또한, 암호문에 MAC을 포함해 변조된 그래프가 복호화되는 것을 방지한다.

제안방식의 동작과정은 Setup, KeyGen, Enc 및 Dec의 네 가지 알고리즘으로 구성된다. 주요 파라미터는 표 2와 같다.

5.1 시스템 모델과 그래프 암호문

송신자와 수신자는 사전에 동일한 비밀키 sk 를 안전하게 공유한다고 가정하며, 공개 파라미터 pp 에는 문자 집합, 소수 집합, 인코딩 테이블 및 모듈러 분해 파라미터가 포함된다. nonce ρ 는 매 암호화마다 새로 생성하며 동일한 비밀키에서 재사용하지 않는다. nonce는 NIST의

DRBG(Deterministic Random Bit Generator) 권고와 같이 암호학적으로 안전한 난수 생성기를 사용해 생성할 수 있다[13].

표 2. 주요 파라미터
Table 2. Key Parameters

기호	설명
λ	보안 파라미터
Σ	평균 문자 집합
$\mathbb{P}_R, \mathbb{P}_C$	행 소수 집합과 열 소수 집합
T	문자-소수 쌍 인코딩 테이블
a_i	문자 p_i 의 정점 라벨
$B = 2^\ell$	난수화 라벨의 모듈러 상한
$M = 2^m$	간선 가중치 분해 파라미터
ρ	암호화마다 생성되는 nonce
s_i	위치 i 에 대한 PRF 마스크
x_i	난수화된 정점 라벨
q_i, w_i	x_i 의 보조 라벨 및 간선 가중치
τ	암호문 인증 태그

평균 $P = p_1 p_2 \dots p_n$ 에 대응되는 별 그래프는 $S_{n+1} = (V, E)$, $V = \{v_0, v_1, \dots, v_n\}$, $E = \{(v_0, v_i)\}_{i=1}^n$ 와 같다. 외곽 정점은 구조적으로 서로 대칭이므로 문자 순서를 보존하기 위해 위치 라벨 $f_I(v_i) = i$ 를 함께 사용한다. 간선 가중치 함수는 $f_E(v_0, v_i) = w_i$ 이고, 보조 라벨 함수는 $f_Q(v_i) = q_i$ 로 정의한다.

최종적으로 그래프 암호문은 $G_C = (S_{n+1}, f_I, f_E, f_Q)$ 로 정의한다. 단, 정점 라벨 a_i 는 그래프 암호문에 직접 포함하지 않는다.

5.2 Setup

Setup은 보안 파라미터와 문자 집합을 입력받아 문자 인코딩, 난수화 및 인증에 필요한 공개 파라미터 pp 를 생성하는 단계이다. 전체 과정은 알고리즘 1과 같다.

알고리즘 1: Setup

입력: $1^\lambda, \Sigma$
출력: pp
1 $\alpha, \beta \leftarrow \{(\alpha, \beta) : |\Sigma| \leq \alpha\beta\};$
2 $\mathbb{P}_R \leftarrow \{r_1, \dots, r_\alpha\}; \mathbb{P}_C \leftarrow \{c_1, \dots, c_\beta\}, \mathbb{P}_R \cap \mathbb{P}_C = \emptyset;$
3 $T : \Sigma \hookrightarrow \mathbb{P}_R \times \mathbb{P}_C;$
4 $a_{\max} \leftarrow \max_{p \in \Sigma} \{r(p)c(p)\};$
5 $\ell \leftarrow \lceil \log_2(a_{\max} + 1) \rceil; B \leftarrow 2^\ell;$
6 $m \leftarrow \{m : 1 \leq m < \ell\}; M \leftarrow 2^m;$
7 $F : \{0, 1\}^\lambda \times \{0, 1\}^* \rightarrow \{0, 1\}^\ell;$
8 $\text{MAC} : \{0, 1\}^\lambda \times \{0, 1\}^* \rightarrow \{0, 1\}^t;$
9 $pp \leftarrow (\lambda, \Sigma, \mathbb{P}_R, \mathbb{P}_C, T, B, M, F, \text{MAC});$
10 반환 $pp;$

문자 인코딩을 위한 소수 집합과 인코딩 함수를 구성한 후, 최대 정점 라벨 $a_{\max}$ 를 기준으로 난수화 공간 B 를 정의한다. 또한 간선 가중치의 범위를 제한하기 위한 분해 파라미터 M 을 설정하고 PRF와 MAC을 정의한다.

5.3 KeyGen

KeyGen은 공개 파라미터를 입력받아 비밀키 sk 를 생성한다. 전체 과정은 알고리즘 2와 같다. 먼저 마스터 키 msk 를 무작위로 선택하고, HKDF의 extract 및 expand를 통해 마스크 생성용 키 K_E 와 인증용 키 K_A 를 생성하여 최종 비밀키 sk 를 생성한다.

알고리즘 2: KeyGen

입력: pp
출력: $sk = (K_E, K_A)$
1 $msk \xleftarrow{\$} \{0, 1\}^\lambda;$
2 $PRK \leftarrow \text{HKDF-Extract}(0^\lambda, msk);$
3 $K_E \leftarrow \text{HKDF-Expand}(PRK, \text{"mask"}, \lambda);$
4 $K_A \leftarrow \text{HKDF-Expand}(PRK, \text{"auth"}, \lambda);$
5 $sk \leftarrow (K_E, K_A);$
6 반환 $sk;$

5.4 Enc

Enc는 공개 파라미터, 비밀키 및 평문 P 를 입력받아 그래프 암호문을 생성하는 단계다. 이에 대한 전체적인 동작 과정은 알고리즘 3과 같다.

여기서 각 문자 p_i 는 인코딩 테이블을 통해

$(r_i, c_i) = T(p_i)$, $a_i = r_i c_i$ 와 같이 소수 쌍과 정점 라벨로 변환된다. 이후, $s_i = \text{OS2IP}(F_{K_E}(\rho \| i \| n))$ 와 같이 위치별 마스크를 생성한다. 여기서 OS2IP는 ℓ 비트 문자열을 $[0, B-1]$ 의 정수로 변환하는 함수이다.

알고리즘 3: Enc

입력: $pp, sk = (K_E, K_A), P = p_1 \cdots p_n$
출력: $C = (\rho, \mathcal{G}_C, \tau)$

- 1 $\rho \xleftarrow{\$} \{0, 1\}^\lambda; S_{n+1} \leftarrow K_1 \odot \overline{K_n};$
- 2 **for** $i = 1$ **to** n **do**
- 3 $(r_i, c_i) \leftarrow \mathcal{T}(p_i);$
- 4 $a_i \leftarrow r_i c_i;$
- 5 $s_i \leftarrow \text{OS2IP}(F_{K_E}(\text{"mask"} \| \rho \| i \| n));$
- 6 $x_i \leftarrow (a_i + s_i) \bmod B;$
- 7 $q_i \leftarrow \lfloor x_i / M \rfloor; w_i \leftarrow x_i \bmod M;$
- 8 $f_I(v_i) \leftarrow i; f_E(v_0, v_i) \leftarrow w_i; f_Q(v_i) \leftarrow q_i;$
- 9 $\mathcal{G}_C \leftarrow (S_{n+1}, f_I, f_E, f_Q);$
- 10 $\tau \leftarrow \text{MAC}_{K_A}(\rho \| n \| B \| M \| \text{ser}(\mathcal{G}_C));$
- 11 **반환** $C = (\rho, \mathcal{G}_C, \tau);$

또한, $x_i = (a_i + s_i) \bmod B$ 와 같이 정점 라벨을 난수화하고, 이를 $q_i = \lfloor \frac{x_i}{M} \rfloor$ 및 $w_i = x_i \bmod M$ 와 같이 몫과 나머지로 분해한다.

5.5 Dec

Dec는 그래프 암호문을 입력받아 인증 태그를 검증하고 평문을 복원하는 과정이다. 이에 대한 전체 동작 과정은 알고리즘 4와 같다.

수신자는 $\rho, n, B, M, \text{ser}(\mathcal{G}_C)$ 를 입력으로 인증 태그를 다시 계산한 후, 암호문에 포함된 τ 와 비교한다. 검증 실패 시 $\perp$ 을 반환하고 중단한다. 인증 성공 시 외곽 정점을 위치 라벨 f_I 에 따라 정렬하고, 각 위치에서 보조 라벨 q_i 와 간선 가중치 w_i 를 이용하여 난수화된 라벨 x_i 를 복원한다. 이후 암호화에 사용한 것과 동일한 입력을 활용해 위치별 마스크 s_i 를 다시 생성하고, 이를 제

거하여 원래 정점 라벨 a_i 를 획득한다.

알고리즘 4: Dec

입력: $pp, sk = (K_E, K_A), C = (\rho, \mathcal{G}_C, \tau)$
출력: P 또는 $\perp$

- 1 $\tau' \leftarrow \text{MAC}_{K_A}(\rho \| n \| B \| M \| \text{ser}(\mathcal{G}_C));$
- 2 **if** $\tau' \neq \tau$ **then**
- 3 **반환** $\perp$
- 4 외곽 정점을 $f_I(v_i) = i$ 에 따라 정렬;
- 5 **for** $i = 1$ **to** n **do**
- 6 $x_i \leftarrow f_Q(v_i)M + f_E(v_0, v_i);$
- 7 **if** $x_i \notin [0, B-1]$ **then**
- 8 **반환** $\perp$
- 9 $s_i \leftarrow \text{OS2IP}(F_{K_E}(\text{"mask"} \| \rho \| i \| n));$
- 10 $a_i \leftarrow (x_i - s_i) \bmod B;$
- 11 **if** $a_i \notin \text{Im}(\mathcal{T})$ **then**
- 12 **반환** $\perp$
- 13 $(r_i, c_i) \leftarrow \text{Factor}_{\mathbb{P}_R, \mathbb{P}_C}(a_i);$
- 14 $p_i \leftarrow \mathcal{T}^{-1}(r_i, c_i);$
- 15 **반환** $P = p_1 p_2 \cdots p_n;$

복원된 a_i 가 유효한 인코딩 값이면 행 소수와 열 소수의 쌍 (r_i, c_i) 를 복원하고, 역인코딩 테이블 T^{-1} 을 통해 각 문자 p_i 를 획득한다. 최종적으로 이를 통해 모든 문자를 순서대로 결합해 평문 P 를 복원할 수 있다.

6. 제안방식 분석 및 검증

본 장에서는 제안방식의 복호화 정확성, 정점 라벨 기밀성, 암호문 무결성 및 효율성을 분석한다. 위 분석은 표준 암호 프리미티브의 안전성과 nonce를 재사용하지 않는 것을 가정한다.

6.1 복호화 정확성

정상 암호문에 대해 복호화를 수행 시 정점 라벨과 문자가 복원된다. 몫과 나머지의 정의에 따

라 $q_i M + w_i = \lfloor \frac{x_i}{M} \rfloor M + (x_i \bmod M) = x_i$ 가 성립한다. 암호화와 복호화에서 동일한 K_E, ρ, i, n 를 사용하므로 동일한 s_i 가 생성되어 기존 라벨값을 복원할 수 있다.

$$\begin{aligned} x_i &\equiv a_i + s_i \pmod{B} \\ x_i - s_i &\equiv a_i \pmod{B} \\ (x_i - s_i) \bmod B &= a_i \end{aligned}$$

또한, $P_R \cap P_C = \emptyset$ 이므로 복원된 두 소수 중 행 소수와 열 소수를 구분할 수 있다. 인코딩 테이블이 단사 함수이므로 $T^{-1}(T(p_i)) = p_i$ 가 성립한다. 따라서 모든 위치에서 원래 문자가 복원되어 전체 평문을 정확히 얻을 수 있다.

6.2 정점 라벨 기밀성

기존 방식에서는 w_i^{Ali} 와 문자 위치 i 를 알면 $a_i = w_i^{Ali} + 10^i$ 와 같이 정점 라벨을 복원할 수 있었다. 반면 제안방식에서 공개값은 x_i 의 가역 표현인 (q_i, w_i) 이다. 공격자가 정점 라벨을 복원하기 위해선 $a_i = (q_i M + w_i - s_i) \bmod B$ 의 s_i 를 알아야 한다.

PRF 사용 시 nonce가 재사용되지 않으면 각 s_i 는 $\mathbb{Z}_B$ 에서 독립된 균등 난수이다. 이때 $x_i = (a_i + s_i) \bmod B$ 또한 a_i 와 관계없이 균등하게 분포한다. (q_i, w_i) 는 x_i 와 일대일 대응하므로 몫, 나머지 분해는 추가적인 정점 라벨 정보를 노출하지 않는다. 실제 환경에서는 안전한 PRF와 난수 함수를 다항 시간 내에 구분하기 어렵다는 가정에 따라 이 성질이 유지된다[7].

6.3 암호문 무결성

제안방식은 $\rho || n || B || M || ser(G_C)$ 와 같은 값을 MAC 입력에 포함한다. 따라서 공격자가 nonce, 위치 라벨, 간선 가중치, 보조 라벨 중 하나를 변

경하면 기존 태그 τ 와 새로 계산한 태그가 달라진다. MAC이 SUF-CMA(Strong Unforgeability under Chosen-Message Attack)를 만족한다고 가정하면 공격자가 검증을 통과하는 새로운 암호문을 생성할 확률은 무시할 수 있다.

제안방식은 암호화 결과를 생성한 후 해당 결과 전체를 인증하므로 Encrypt-then-MAC 구조에 해당한다[11, 12].

6.4 간선 가중치 및 암호문 크기

제안방식의 간선 가중치는 $w_i = x_i \bmod M$ 과 같으므로 모든 i 에 대해 $0 \leq w_i < M = 2^m$ 가 성립한다. 각 간선 가중치는 m 비트 이내로 표현할 수 있으며, 보조 라벨은 $0 \leq q_i < 2^{\ell-m}$ 이므로 $\ell - m$ 비트 이내로 표현할 수 있다. 즉, 문자 하나에 필요한 전체 라벨 크기는 $|w_i|_{bit} + |q_i|_{bit} \leq m + (\ell - m) = \ell$ 와 같다.

고정된 문자 집합에서 ℓ 은 상수이므로 전체 그래프 라벨 크기는 $|C|_{bit} = n\ell + |\rho| + |\tau| + O(n) = O(n)$ 와 같다. 여기서 $O(n)$ 은 위치 라벨과 그래프 직렬화에 필요한 부가 정보를 의미한다. 따라서 제안방식은 간선 가중치와 보조 라벨을 모두 포함하더라도 고정 문자 집합에서 전체 표현 크기를 $O(n)$ 으로 감소시킬 수 있다.

6.5 효율성 측정

제안방식의 효율성을 확인하기 위해 Python 3.12.14와 Intel Core i5-13400F 환경에서 암호화 시간과 암호문 크기를 측정하였다. 메시지 길이 $n = 10, 100, 1000$ 에 대해 각 알고리즘을 5회 실행한 평균값을 사용하였으며, 일회성 Setup과 KeyGen 시간은 제외하였다.

표 3과 같이 제안방식은 위치별 PRF와 MAC 검증을 수행하므로 기존 방식보다 암호·복호화 시

간이 증가했다. 그러나 $n = 1000$ 에서도 암호/복호화가 각각 약 4 ms에 수행되며, 메시지 길이에 대해 선형적인 증가를 보였다. 기존 방식에는 정점 라벨 난수화와 암호문 무결성 검증이 포함되지 않으므로, 측정된 차이에는 제안방식이 제공하는 추가 보안 기능의 연산 비용이 포함된다.

표 3. 암호·복호화 시간
Table 3. Encryption/Decryption Times

n	방식	암호화 시간 (ms)	복호화 시간 (ms)
10	Ali et al. 별 그래프	0.0032	0.0019
10	제안방식	0.0453	0.0446
100	Ali et al. 별 그래프	0.0343	0.0256
100	제안방식	0.4876	0.4414
1,000	Ali et al. 별 그래프	1.2028	1.1004
1,000	제안방식	4.0254	3.9137

간선 가중치 측정 결과, 그림 1과 같이 Ali et al. 방식의 전체 간선 크기는 $n = 10$ 에서 202 bits, $n = 100$ 에서 16,930 bits, $n = 1000$ 에서 1,664,129 bits로 증가했다. 그러나 본 제안방식은 40, 400, 4,000 bits로 선형적으로 증가하는 모습을 볼 수 있다.

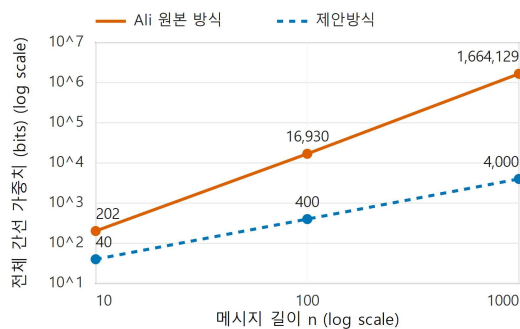


그림 1. 간선 가중치 크기 비교
Figure 1. Comparison of Edge Weight Sizes

암호문 크기는 각 논문에서 정의한 원래 구성 요소를 기준으로 계산하였다. 그림 2와 같이 $n = 10$ 에서는 Ali et al. 방식이 29.25 bytes, 제안방식이 86 bytes로 제안방식이 더 컸다.

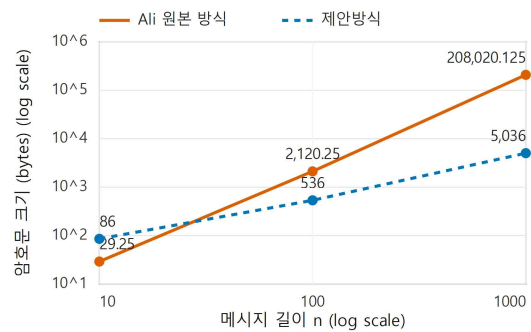


그림 2. 암호문 크기 비교
Figure 2. Comparison of Ciphertext Sizes

이는 nonce, 인증 태그, 보조 라벨 및 위치 라벨에 따른 고정 오버헤드 때문이다. 그러나 $n = 1000$ 에서는 기존 방식이 약 208,020 bytes, 제안방식이 5,036 bytes로 나타나 제안방식이 약 41배 작았다. 따라서 제안방식은 짧은 메시지에서 추가 보안정보로 인한 오버헤드가 발생하지만, 간선 가중치의 크기를 제한하여 메시지 길이가 증가할 때 전체 암호문 크기를 $O(n)$ 으로 유지한다.

7. 결론

Ali et al.의 별 그래프 기반 암호화 방식은 문자 위치에 따라 증가하는 10^i 를 사용하여 간선 가중치를 생성하므로, 메시지 길이가 증가할수록 가중치의 표현 크기가 증가하는 한계점이 존재한다. 또한 간선 가중치에 10^i 를 더하면 정점 라벨을 직접 복원할 수 있으며, 동일 문자에 대한 반복 정점 라벨이 노출되는 문제가 존재한다.

본 논문에서는 이를 해결하고자 정점 라벨에 PRF 기반 위치별 마스크를 적용하고, 난수화된 라벨을 뺀 나머지 분해하는 새로운 별 그래프 기반 암호화 방식을 제안했다. 제안방식은 나머지를 간선 가중치로 사용하여 모든 간선 가중치를 $[0, M-1]$ 의 범위로 제한하고, 뺀 외곽 정점의 보조 라벨로 구성하여 정보 손실 없이 난수화 라벨을 복원한다. 또한 HKDF를 통해 마스크 키와 인증 키를 분리하고, MAC을 적용해 변조된 그래프 암호문 복호화를 방지한다.

분석 결과 제안방식은 고정된 문자 집합에서 전체 암호문 크기를 선형으로 구성할 수 있으며, nonce가 재사용되지 않고 PRF와 MAC이 안전하다는 가정하에 정점 라벨의 반복 패턴 노출을 완화하고 암호문 무결성을 제공할 수 있다. 그러나 별 그래프의 정점 수를 통해 메시지 길이가 노출되며, nonce 재사용 시 정점 라벨 사이의 관계가 노출될 수 있다는 한계점이 존재한다.

향후 표준 AEAD(Authenticated Encryption with Associated Data) 구조와의 정량적 비교, nonce 오용 저항성 및 메시지 길이 은닉을 추가하여 실제 통신 환경에서 활용 가능한 그래프 기반 암호화 방식으로 개선하는 연구가 필요할 것으로 사료된다.

본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2026년도 SW저작권 생태계 조성 기술 개발 사업 (과제명 : 클라우드 서비스 활용 구축 형태별 대규모 소프트웨어 라이선스 검증 기술개발, 과제번호 : RS-2023-00224818, 기여율: 100%)

참 고 문 헌

- [1] D. B. West, "Introduction to Graph Theory", 2nd ed., Prentice Hall, 2001, ISBN 978-0-13-014400-3.
- [2] C. E. Shannon, "Communication theory of secrecy systems", Bell System Technical Journal, 28(4), pp.656-715, 1949, DOI : 10.1002/j.1538-7305.1949.tb00928.x.
- [3] D. R. Stinson, M. B. Paterson, "Cryptography: Theory and Practice", 4th ed., CRC Press, 2018, ISBN 978-1-315-28248-0.
- [4] N. Ali et al., "Secure communication in the digital age: A new paradigm with graph-based encryption algorithms", Frontiers in Computer Science, 6, Art. no. 1454094, 2024, DOI : 10.3389/fcomp.2024.1454094.
- [5] B. Ni, R. Qazi, S. U. Rehman, G. Farid, "Some graph-based encryption schemes", Journal of Mathematics, 2021, Art. no. 6614172, 2021, DOI : 10.1155/2021/6614172.
- [6] S. A. U. H. Bokhary, A. Kharal, F. M. A. Samman, M. E. E. Dalam, A. Gargouri, "Efficient graph algorithms in securing communication networks", Symmetry, 16(10), Art. no. 1269, 2024, DOI : 10.3390/sym16101269.
- [7] O. Goldreich, S. Goldwasser, S. Micali, "How to construct random functions", Journal of the ACM, 33(4), pp.792-807, 1986, DOI : 10.1145/6490.6503.
- [8] H. Krawczyk, P. Eronen, "HMAC-based extract-and-expand key derivation function (HKDF)", RFC 5869, RFC Editor, 2010.5, DOI : 10.17487/RFC5869.
- [9] L. Chen, "Recommendation for key derivation using pseudorandom functions", NIST Special Publication 800-108 Rev. 1 Update 1, National Institute of Standards and Technology, 2024, DOI : 10.6028/NIST.SP.800-108r1-upd1.
- [10] H. Krawczyk, M. Bellare, R. Canetti, "HMAC: Keyed-hashing for message authentication", RFC 2104, RFC Editor, 1997.2, DOI : 10.17487/RFC2104.
- [11] M. Bellare, C. Namprepmpre, "Authenticated

encryption: Relations among notions and analysis of the generic composition paradigm”, Journal of Cryptology, 21(4), pp.469-491, 2008,

DOI : 10.1007/s00145-008-9026-x.

- [12] H. Krawczyk, “The order of encryption and authentication for protecting communications (or: How secure is SSL?)”, Advances in Cryptology – CRYPTO 2001, Lecture Notes in Computer Science, 2139, pp.310-331, Springer, 2001, DOI : 10.1007/3-540-44647-8_19.

- [13] E. Barker, J. Kelsey, “Recommendation for random number generation using deterministic random bit generators”, NIST Special Publication 800-90A Rev. 1, National Institute of Standards and Technology, 2015,

DOI : 10.6028/NIST.SP.800-90Ar1.

저 자 소 개



윤성철(SeongCheol Yoon)

2023.2 순천향대학교 컴퓨터소프트웨어공학과 졸업

2025.2 순천향대학교 소프트웨어융합학과 석사

2025.3-현재 순천향대학교 소프트웨어융합학과 박사과정

<주관심분야> 정보보호, 암호학, 그래프 암호화



김수현(Su-Hyun Kim)

2010.2 순천향대학교 정보기술공학부 졸업

2012.2 순천향대학교 컴퓨터학과 석사

2016.2 순천향대학교 컴퓨터학과 박사

2016.3-2018.9 순천향대학교

IoT보안연구센터 연구교수

2018.10-2023.2 정보통신산업진흥원 책임

2023.3-현재 : 순천향대학교

컴퓨터소프트웨어공학과 교수

<주관심분야> 암호 프로토콜, IoT 보안, 클라우드 컴퓨팅 보안

— 저 자 소 개 —



조용준(YongJoon Joe)

2011.3 큐슈대학교 전기정보공학과 졸업
2013.3 큐슈대학교 정보학부 석사
2016.3 큐슈대학교 정보학부 박사과정 수료
2013.4-2016.3 일본 학술진흥원 특별연구원
2016-현재 엘에스웨어(주) 이사
<주관심분야> 병렬·분산 컴퓨팅, 게임이론, 분산 제약 최적화 문제



이임영(Im-Yeong Lee)

1981.2 홍익대학교 전자공학과 졸업
1986.2 오사카대학 통신공학전공 석사
1989.2 오사카대학 통신공학전공 박사
1989-1994 한국전자통신연구원 선임연구원
1994-현재 순천향대학교
컴퓨터소프트웨어공학과 교수
<주관심분야> 암호이론, 암호 프로토콜, 컴퓨터보안