

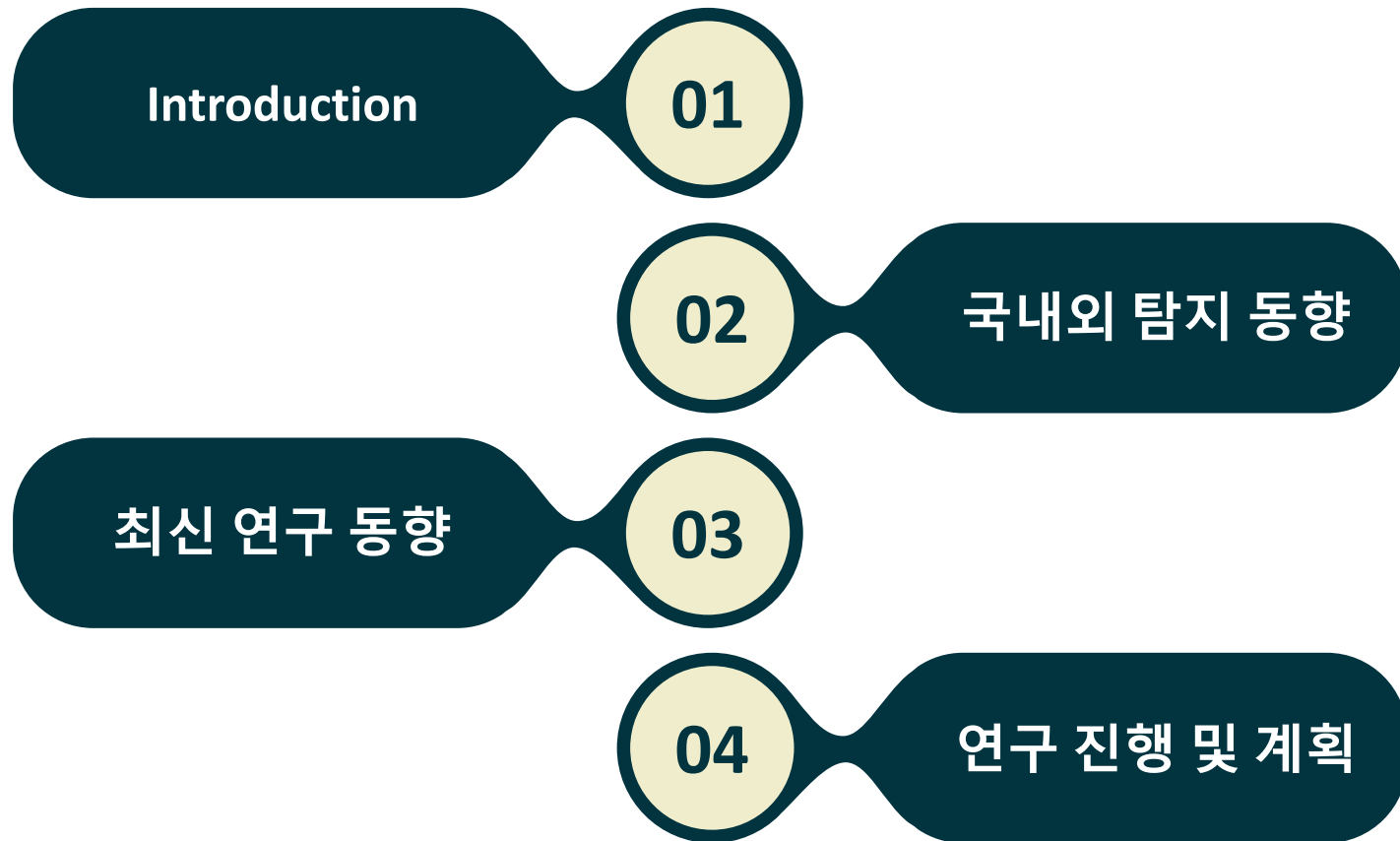


비양자내성암호 알고리즘 탐지 동향

2025.10.16

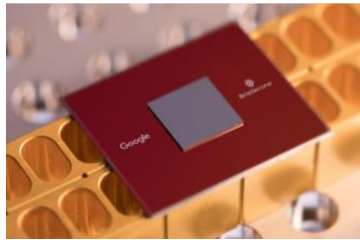
성신여자대학교 김수리

Contents

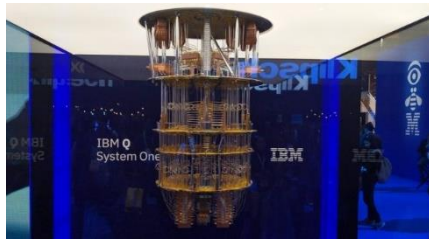


Introduction [1/10]

- 양자 컴퓨팅 기술 현황



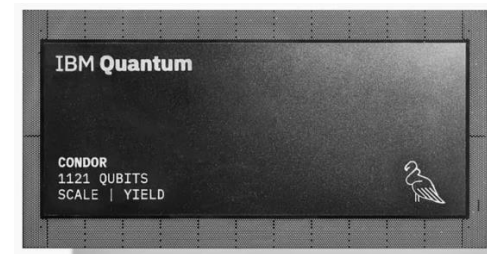
Google 72-qubit chip
"Bristlecone"
March 2018



IBM 20-qubit quantum computer
"Q System One"
January 2019
(53-qubit, September 2019)



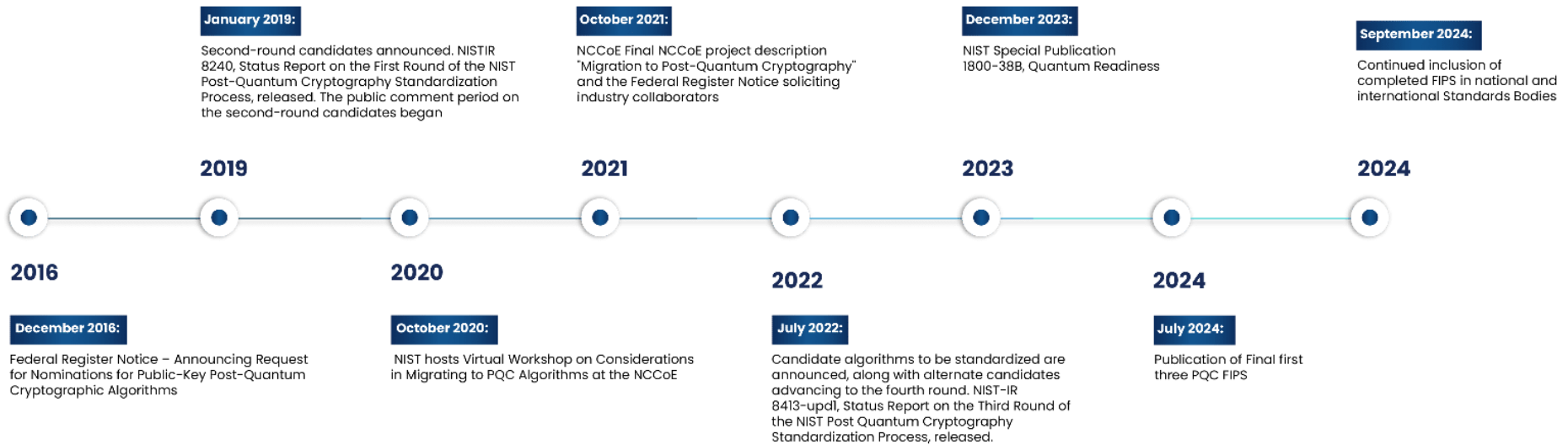
IBM 65-qubit chip
"Hummingbird"
August 2020



IBM 1121-qubit chip
"Condor"
December 2023

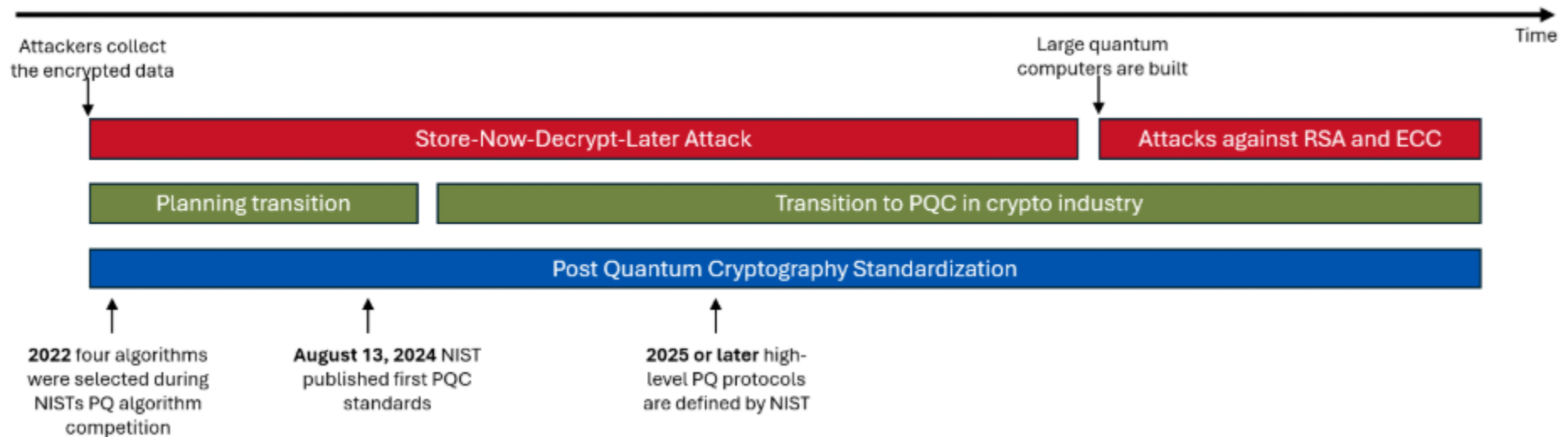
Introduction [2/10]

- NIST PQC Standardization project



Introduction [3/10]

- PQC로의 전환
 - Post-quantum cryptography timeline



Introduction [4/10]

- Quantum Computing Cybersecurity Preparedness ACT H.R. 7535
 - 2022년 미국 백악관에서 양자 보안에 대한 체계적 대응을 위해 법제화
 - 현재 사용중인 정보기술 자산 중 양자컴퓨팅에 취약한 요소 파악하고 이를 인벤토리화하여 지속적으로 관리해야함을 명시

(a) Findings.--Congress finds the following:

(1) Cryptography is essential for the national security of the United States and the functioning of the economy of the United States.

(2) The most widespread encryption protocols today rely on computational limits of classical computers to provide

양자컴퓨터 기반 암호해독에 취약한 정보기술을 파악하고 migration 하는 지침 필요

- 요구사항 : 취약한 정보기술 목록 수립
- 지침 추가 사항 : 우선순위 마련, 최대한 자동화 강조

(4) The rapid progress of quantum computing suggests the potential for adversaries of the United States to steal sensitive encrypted data today using classical computers, and wait until sufficiently powerful quantum systems are available to decrypt it.

(a) Inventory.--

(1) Establishment. <<NOTE: Deadline. Guidelines.>> --Not later than 180 days after the date of enactment of this Act, the Director of OMB, in coordination with the National Cyber

CISA, shall inventory information technology to a minimum--
ment for each
nt inventory of

information technology in use by the agency that is vulnerable to decryption by quantum computers, prioritized using the criteria described in subparagraph (B);

(B) <<NOTE: Criteria.>> criteria to allow agencies to prioritize their inventory efforts; and

(C) a description of the information required to be reported pursuant to subsection (b).

(2) Additional content in guidance.--In the guidance established by paragraph (1), the Director of OMB shall include, in addition to the requirements described in that paragraph--

(A) a description of information technology to be prioritized for migration to post-quantum cryptography; and

Introduction [5/10]

- NIST NCCoE (National Cybersecurity Center of Excellence)
 - NIST 산하에 설치된 사이버보안 전문연구소
 - 실제 산업계의 문제를 해결하기 위해 실용적이고 상용화 가능한 사이버보안 솔루션 개발을 목적으로 설립

The screenshot displays the NIST NCCoE website. At the top, the NIST logo and 'NATIONAL CYBERSECURITY CENTER OF EXCELLENCE' are on the left, while navigation links for 'SECURITY GUIDANCE', 'OUR APPROACH', 'NEWS & INSIGHTS', 'GET INVOLVED', and a 'SEARCH' button are on the right. The main heading 'Working Together for Cybersecurity' is partially visible. Below this, a grid of four resource categories is shown: 'Data Protection', 'Trusted Enterprise', 'Resilient Embedded Systems Security', and 'Frameworks Application'. Each category contains a list of specific resources. The 'Data Protection' list includes 'Mobile Driver's License (mDL)', 'Digital Identity Lab', 'Multifactor Authentication for Public Safety', 'Genomics Privacy Enhancing Technologies (PETs)', 'Genomics Threat Model', 'Privacy', 'Cryptographic Modernization Validation Program (CMVP)', and 'Migration to Post-Quantum Cryptography'. The 'Trusted Enterprise' list includes 'Secure Software Development (DevSecOps)', '5G', 'Secure AI Dioptra', 'Data Classification Practices', 'Transport Layer Security (TLS 1.3)', and 'Zero Trust'. The 'Resilient Embedded Systems Security' list includes 'Healthcare Cybersecurity', 'Manufacturing Cybersecurity', 'Water Cybersecurity', 'Manufacturing Training', 'Blockchain for Supply Chain', 'Smart Inverters', and 'Internet of Things (IoT) Onboarding'. The 'Frameworks Application' list includes 'Resources for Applying NIST Frameworks', 'Cyber AI Profile', 'Ransomware Profile', 'Transportation and Rail Profile', 'Semiconductor Profile', 'Positioning, Navigation, and Timing (PNT) Profile', 'Genomics Profile', and 'Natural Language Processing'. On the left side of the grid, there are buttons for 'VIEW OUR WORK', 'JOIN A COMMUNITY', and 'SUBSCRIBE TO UPDATES'.

NIST NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

SECURITY GUIDANCE OUR APPROACH NEWS & INSIGHTS GET INVOLVED SEARCH

Working Together for Cybersecurity

At the NCCoE, we bring together experts to address the real-world needs of securing the nation's critical infrastructure.

[VIEW OUR WORK](#) [JOIN A COMMUNITY](#) [SUBSCRIBE TO UPDATES](#)

- Data Protection**
 - [Mobile Driver's License \(mDL\)](#)
 - [Digital Identity Lab](#)
 - [Multifactor Authentication for Public Safety](#)
 - [Genomics Privacy Enhancing Technologies \(PETs\)](#)
 - [Genomics Threat Model](#)
 - [Privacy](#)
 - [Cryptographic Modernization Validation Program \(CMVP\)](#)
 - [Migration to Post-Quantum Cryptography](#)
- Trusted Enterprise**
 - [Secure Software Development \(DevSecOps\)](#)
 - [5G](#)
 - [Secure AI Dioptra](#)
 - [Data Classification Practices](#)
 - [Transport Layer Security \(TLS 1.3\)](#)
 - [Zero Trust](#)
- Resilient Embedded Systems Security**
 - [Healthcare Cybersecurity](#)
 - [Manufacturing Cybersecurity](#)
 - [Water Cybersecurity](#)
 - [Manufacturing Training](#)
 - [Blockchain for Supply Chain](#)
 - [Smart Inverters](#)
 - [Internet of Things \(IoT\) Onboarding](#)
- Frameworks Application**
 - [Resources for Applying NIST Frameworks](#)
 - [Cyber AI Profile](#)
 - [Ransomware Profile](#)
 - [Transportation and Rail Profile](#)
 - [Semiconductor Profile](#)
 - [Positioning, Navigation, and Timing \(PNT\) Profile](#)
 - [Genomics Profile](#)
 - [Natural Language Processing](#)

Introduction [6/10]

- NIST SP 1800-30B : Approach, Architecture, and Security Characteristics of Public Key Application Discovery Tools
 - 조직들이 자신의 IT 시스템 내에서 양자 취약암호가 어디에, 어떻게 사용되는지 자동으로 탐지하여 전환계획을 수립할 수 있도록 하는 도구와 방법론을 제안
 - 도구 설계 방식
 - 입력 소스
 - 개발 파이프라인
 - 운영시스템 (실행파일, 암호 라이브러리, 인증서 등등)
 - 네트워크 트래픽
- NIST SP 1800-38C : Testing Draft Standards for Interoperability and Performance
 - 양자 내성 알고리즘 간의 호환성 문제 식별
 - 각 조직이 자체 PQC 전환을 위해 유사한 상호 운용성 테스트 방안 제안
 - PQC 교체 후 성능이나 호환성 검증 방안 제시

Introduction [7/10]

- CISA (Cybersecurity & Infrastructure Security Agency)
 - 양자내성암호로의 전환을 위해 비양자내성암호를 자동 식별하는 Automated Cryptography Discovery and Inventory (ACDI) tool 개발 강조
 - CISA/NSA/NIST 와 작업 timeline 제시 (~2035년 완료)



The screenshot displays the official website of the Cybersecurity & Infrastructure Security Agency (CISA). At the top left is the CISA seal, which features an eagle and the text 'CYBERSECURITY & INFRASTRUCTURE SECURITY AGENCY'. To the right of the seal is the agency's name, 'America's Cyber Defense Agency', in a large blue font, with the subtitle 'NATIONAL COORDINATOR FOR CRITICAL INFRASTRUCTURE SECURITY AND RESILIENCE' below it. A search bar is located on the top right. A dark blue navigation bar contains links for 'Topics', 'Spotlight', 'Resources & Tools', 'News & Events', 'Careers', and 'About'. Below this bar, a breadcrumb trail reads: 'Home / Resources & Tools / Resources / Strategy for Migrating to Automated Post-Quantum Cryptography Discovery and Inventory Tools'. The main content area is titled 'PUBLICATION' in small letters, followed by the large, bold title 'Strategy for Migrating to Automated Post-Quantum Cryptography Discovery and Inventory Tools'.

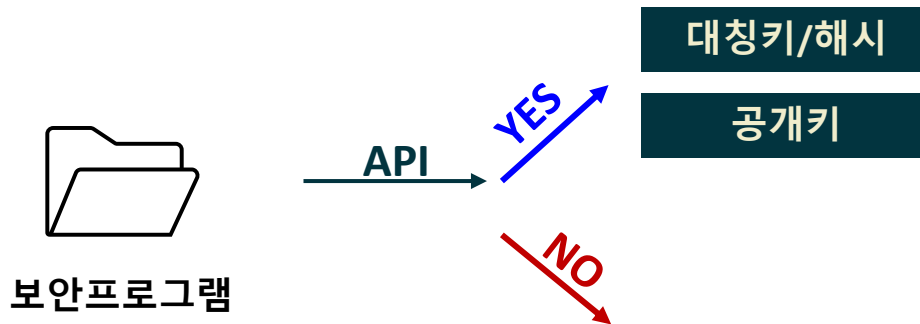
Introduction [8/10]

- PQC로의 전환
 - 양자 컴퓨팅 환경으로 인해 기존의 암호 알고리즘 사용에 변화가 필요
 - 대칭키/해시 → 키 길이 증가, 출력값 증가
 - 공개키 → PQC 암호로 전환
 - 현재 정보 시스템은 다양한 플랫폼 위에서 운영되고 있으며, 이 시스템 내에는 양자 컴퓨터에 취약한 고전 암호 사용
 - 웹 브라우저, 운영체제, 펌웨어, IoT 기기, 네트워크 장비 등 ...
 - 이를 수동적으로 탐지하는데 한계가 존재

실제 운영환경에서 비양자내성암호를 효율적으로 탐지할 수 있는 기술 및 도구 개발 필요

Introduction [9/10]

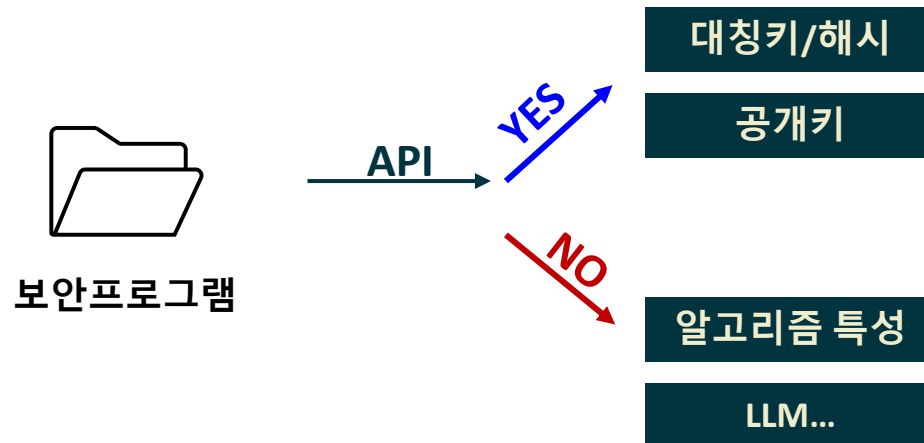
- 탐지 방법



```
1 void B_CAVP_Test()  
9  
10 char *dl_error;  
11 void *XXX_LIBRARY;  
12  
13 XXX_LIBRARY = dlopen("./libXXX.so", RTLD_LAZY);  
14  
15 if(!XXX_LIBRARY)  
16 {  
17     fprintf(stderr, "ERROR LOADING LIBRARY \n");  
18     exit(1);  
19 }  
20  
21 *(void**)&K_BLOCKTEST = dlsym(XXX_LIBRARY, "XXX_CAVP_BlockTest");  
22 *(void**)&K_BLOCKTEST_DEC = dlsym(XXX_LIBRARY, "XXX_CAVP_BlockTest_Dec");  
23 *(void**)&K_B_ENC = dlsym(XXX_LIBRARY, "XXX_CAVP_BEncrypt");  
24 *(void**)&K_CCM_GCM_ENC = dlsym(XXX_LIBRARY, "XXX_CAVP_CCM_GCM_ENC");  
25 *(void**)&K_CCM_GCM_DEC = dlsym(XXX_LIBRARY, "XXX_CAVP_CCM_GCM_DEC");
```

Introduction [10/10]

- 탐지 방법

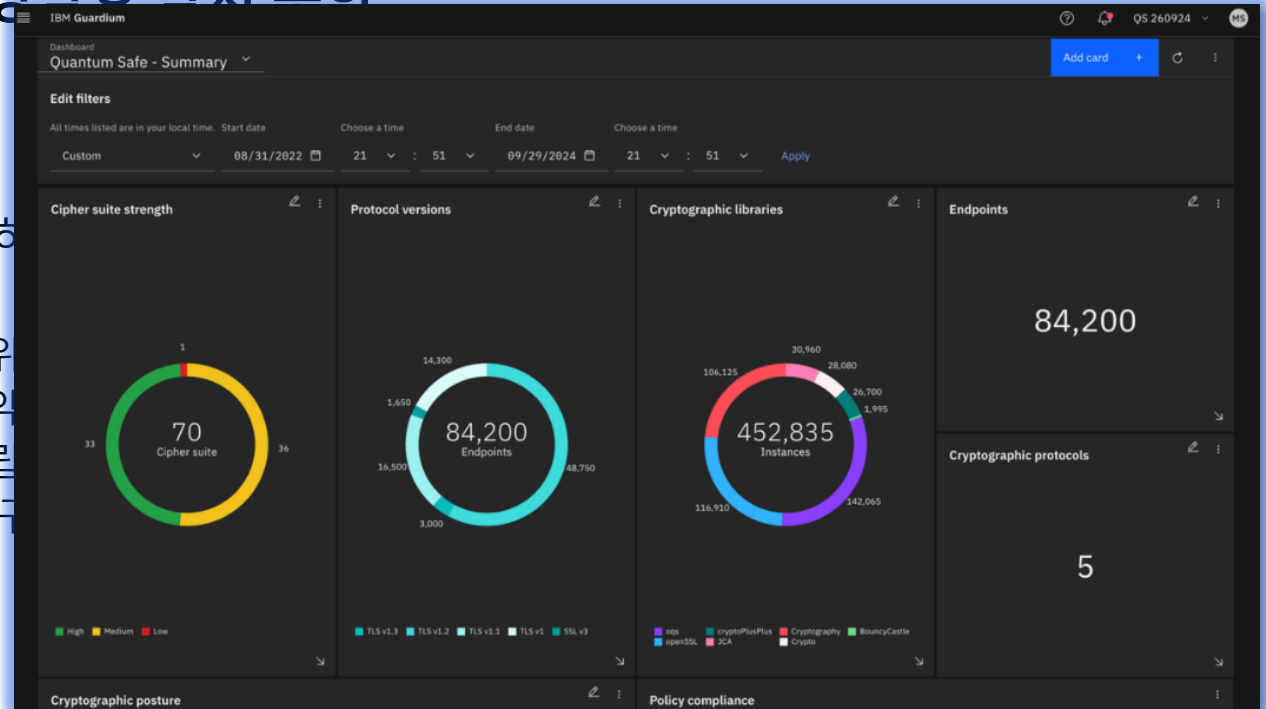


국외기업 비양자내성암호 탐지 동향

- IBM Quantum Safe
 - PQC 암호로 전환하기 위한 기술,서비스, 인프라 포괄하는 솔루션
 - 주요 기능
 - 암호화 사용 위치 탐색 및 관찰
 - 취약 지점 파악 및 위험 분석
 - 전환 전략 수립
 - Crypto Agility 구현

국외기업 비양자내성암호성 타지 도하

- IBM Quantum Safe
 - PQC 암호로 전환하
 - 주요 기능
 - 암호화 사용 우
 - 취약 지점 파악
 - 전환 전략 수립
 - Crypto Agility



Cryptographic inventory - Endpoints

Total rows: 84200

Scan ID	Country	Host	Port	Protocol type	Protocol version	Cipher suite name	Cipher suite strength
15	Bahrain	172.16.104.33	443	TLS	1.3	TLS13-AES-128-GCM-SHA256	MEDIUM
15	Italy	172.16.104.32	3389	TLS		AES128-CCM-8	MEDIUM
15	Italy	172.16.104.32	3389	TLS	1.2	AES256-CCM	HIGH
15	Italy	172.16.104.32	3389	TLS	1.2	DHE-RSA-AES256-CCM	HIGH
15	Italy	172.16.104.32	3389	TLS	1.2	ECDHE-ECDSA-AES256-GCM-SHA384	HIGH

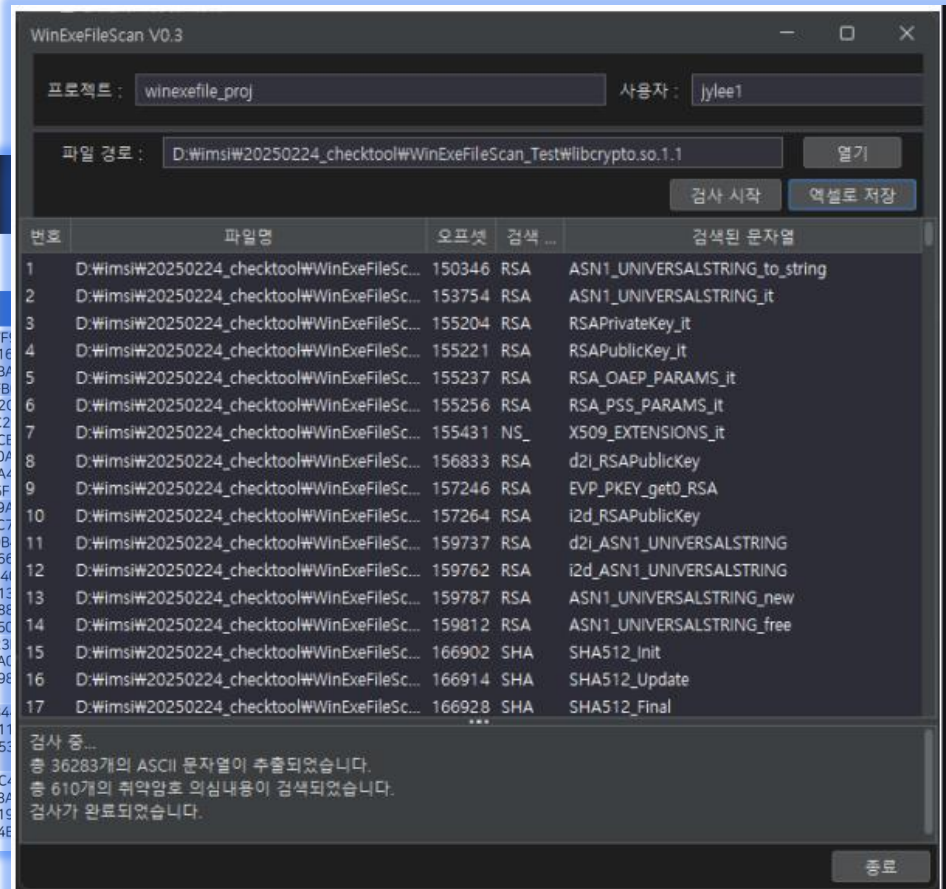
국내기업 비양자내성암호 탐지 동향

- NSHC

- 실행파일 대상으로 비양자내성암호 분석
- 취약암호 관련 문자열 포함 여부 분석
- 결과와 위치 분류하여 엑셀에 저장

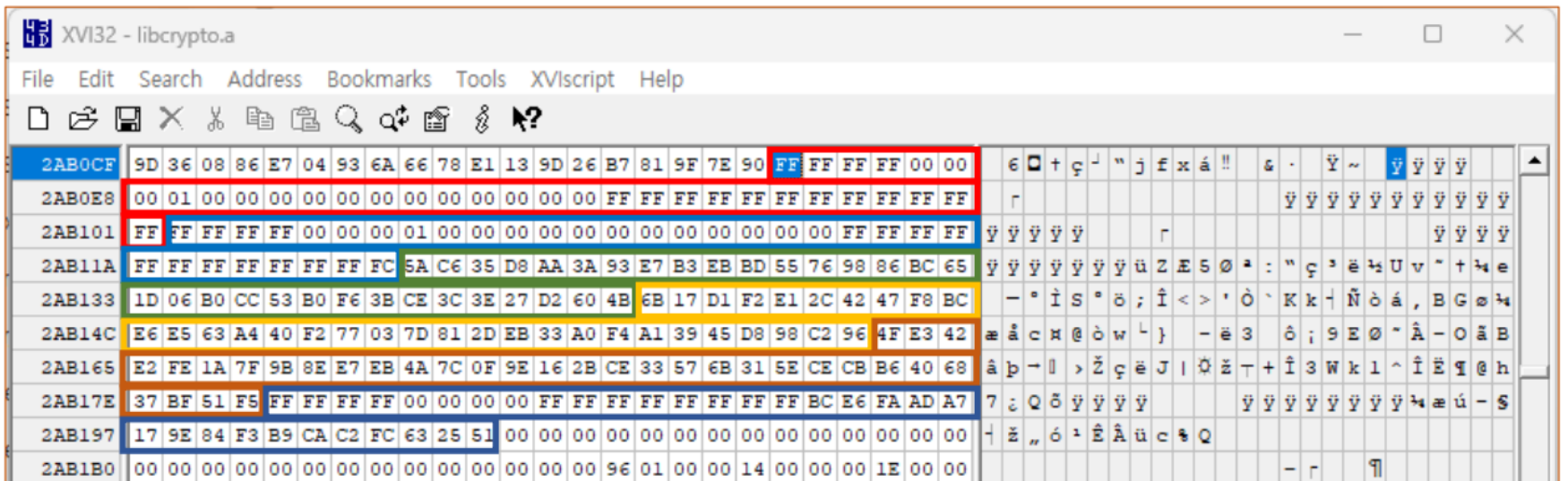
취약암호 파라미터 탐지

파라미터

[illegible]

국내기업 비양자내성암호 탐지 동향

- NSHC
 - Example : 바이너리 파일 상에서의 파라미터 검색

[illegible]

libcrypto.a in openssl_3.4.1

국내기업 비양자내성암호 탐지 동향

- NSHC
 - 실행파일 대상으로 비양자내성암호 분석
 - 취약암호 관련 문자열 포함 여부 분석
 - 결과와 위치 분류하여 엑셀에 저장

알고리즘	키워드
RSA	RSA, RSA-1024, RSA-2048, RSA-4096, PKCS#1, PKCS#8, modulus, public exponent, private exponent
ECC	ECC, ECDSA, ECDH, secp256r1, secp384r1, secp521r1, P-256, P-384, P-521, Curve25519, Ed25519
DSA	DSA, DSS, Digital Signature Algorithm
Diffie-Hellman	DH, Diffie-Hellman, DH-1024, DH-2048, DH-4096, modexp, prime modulus
ElGamal	ElGamal, ElGamal Encryption, ElGamal Signature
AES-128	AES, AES-128, Rijndael, Advanced Encryption Standard, AES-CBC, AES-GCM, AES-ECB, AES-OFB, AES-CTR
3DES (Triple DES)	3DES, TripleDES, DES, Data Encryption Standard
Blowfish	Blowfish, bcrypt
RC4	RC4, ARC4, Rivest Cipher 4
SHA	SHA-1, Secure Hash Algorithm 1, SHA1, sha1sumSHA-256, SHA-512, SHA-384, SHA-224, Secure Hash Algorithm 2, sha256sum, sha512sum
MD5	MD5, Message Digest 5, md5sum

국내기업 비양자내성암호 탐지 동향

- NSHC
 - Example : RSA 관련 키워드 및 파라미터
 - **PKCS #1 : RSA Cryptography Standard**
 - RSA 알고리즘의 구조와 파라미터 정의
 - **X.509 / RFC 5280**
 - 인증서에 RSA 키가 사용될 때 구조 정의
 - RSA public key를 인증서에 어떻게 넣는지 정의
 - **FIPS 186-4 / FIPS 186-5**
 - NIST에서 제공하는 디지털 서명 표준
 - RSA 키 생성 방법 및 조건 명시
 - **ASN.1 DER 인코딩**
 - RSA 키나 인증서를 바이너리 구조로 파싱할 때 참고
 - OpenSSL, Java Key Tool 등에서 RSA 파라미터를 추출할 때 필요

국내기업 비양자내성암호 탐지 동향

- NSHC

- Example : OID ↔ DER 인코딩 활용

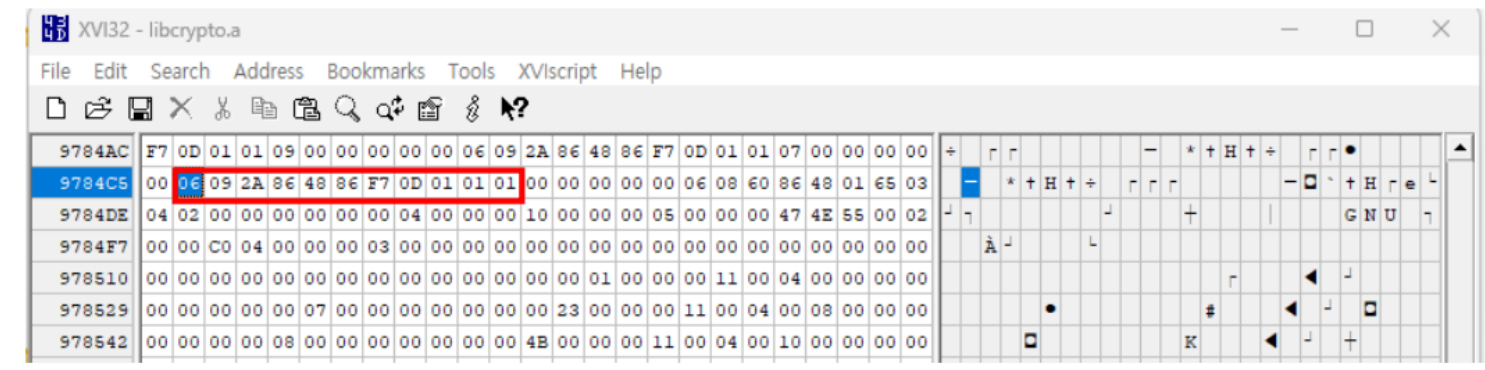
- DER 인코딩 (Distinguished Encoding Rules)

- ASN.1 의 BER/DER 인코딩 형식 중 하나
 - 이진 바이너리 형식으로 실제 파일에 저장되는 형태
 - DER 인코딩은 네트워크 전송과 저장을 위해 MSB-first (big endian) 형식만을 사용

OID	Name	Description	Source
1.2.840.113549.1.1.1	rsaEncryption	RSA publickey encryption	PKCS#1 (RFC 8017)



06 09 2A 86 48 86 F7 0D 01 01 01



비양자내성암호 탐지 연구 동향

비양자내성암호 탐지 연구 동향

- A Toolchain for Assisting Migration of Software Executables Towards Post-quantum Cryptography
 - IEEE Access, 2024
 - 양자 컴퓨팅 환경에 대응하기 위한 비양자내성암호 탐지 기술 관련한 논문
 - QED (Quantum-vulnerable Executable Detection) toolchain 제안
 - API level 에서 비양자내성암호 탐지
 - Real-world dataset 사용
 - 200 개의 software executables
 - 실제환경에서는 90% 이상의 탐지율

비양자내성암호 탐지 연구 동향

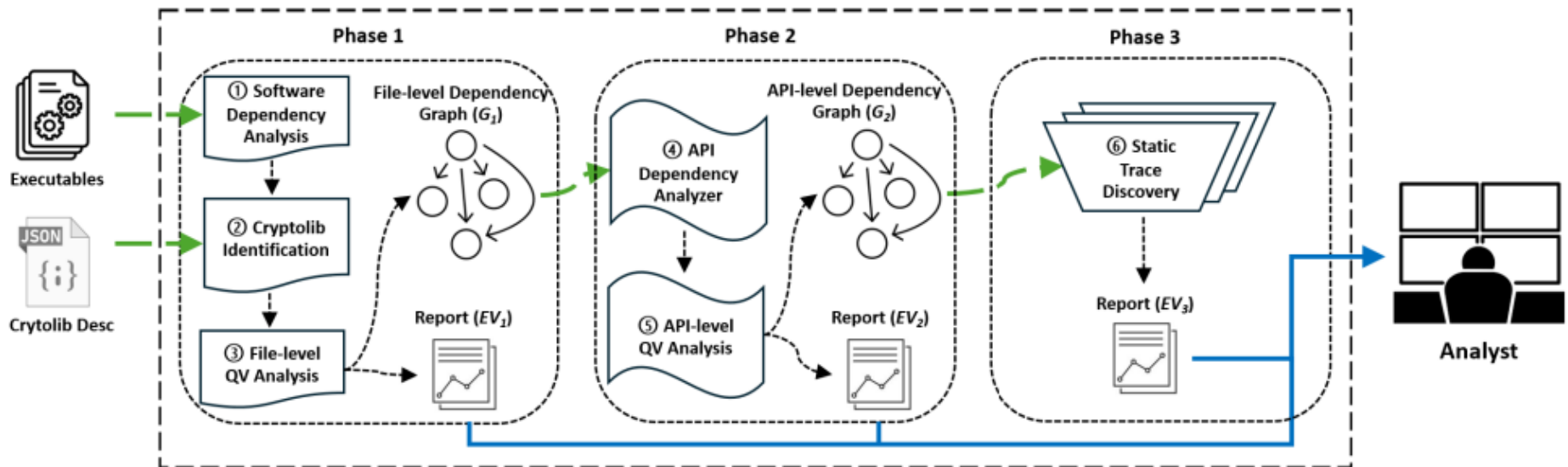
- A Toolchain for Assisting Migration of Software Executables Towards Post-quantum Cryptography
 - Target
 - 리눅스 운영체제
 - C, C++ 로 쓰여진 software
 - Linux executable, Linkable Format (ELF)
 - 암호학적 라이브리를 Dynamic linking 으로 사용하는 환경만 고려 (.so)
 - 자체 라이브러리 구현이나 정적으로 링크한 실행파일은 고려하지 않음 (.a)
 - Implementation
 - Python
 - Pyelftools library 로 ELF 파일 확인
 - QED의 그래프는 NetworkX 라이브러리 사용
 - 코드 공개
 - 전체적인 내부와 외부 함수 및 라이브러리를 그래프 형태로 모델링하고 실제로 연결되는지를 그래프 탐색을 통해 분석

비양자내성암호 탐지 연구 동향

- A Toolchain for Assisting Migration of Software Executables Towards Post-quantum Cryptography
 - 설계 방법
 - Phase 1 : 파일 수준 종속성 분석
 - 실행파일이 사용하는 모든 shared library 탐색
 - 이 중 비양자내성암호 (**Quantum Vulnerable, QV**)를 포함하는 라이브러리 (OpenSSL, wolfSSL, MbedTLS) 확인
 - libcrypto.so
 - Phase 2 : API 수준 분석
 - 실제 호출되는 외부 API 조사하여 비양자내성암호를 사용하는 API 사용 여부 확인
 - 비양자내성암호를 직접 호출하지 않는 경우 제거 → false positive 감소
 - Phase 3 : 정적 추적 분석
 - Main 함수부터 비양자내성암호 API 까지 호출경로가 실제 존재하는지 검증
 - 실제 호출 가능성을 정적으로 입증

비양자내성암호 탐지 연구 동향

- A Toolchain for Assisting Migration of Software Executables Towards Post-quantum Cryptography
 - 설계 방법



비양자내성암호 탐지 연구 동향

- A Toolchain for Assisting Migration of Software Executables Towards Post-quantum Cryptography
 - Example of report

```
"EV_1": [  
  {  
    "path": [  
      "/usr/bin/sftp",  
      "/usr/lib/libcrypto.so.1.1"  
    ]  
  },  
  {  
    "path": [  
      "/usr/bin/dig",  
      "/usr/lib/libdns.so",  
      "/usr/lib/libcrypto.so.1.1"  
    ]  
  },  
  {  
    "path": [  
      "/usr/bin/nmap",  
      "/usr/lib/libssl.so.1.1",  
      "/usr/lib/libcrypto.so.1.1"  
    ]  
  },  
  ...  
  {  
    "path": [  
      "/usr/bin/curl",  
      "/usr/lib/libcurl.so.4",  
      "/usr/lib/libcrypto.so.1.1"  
    ]  
  }  
]
```

```
"EV_2": [  
  {  
    "path": [  
      "/usr/bin/nmap",  
      "/usr/lib/libssl.so.1.1",  
      "/usr/lib/libcrypto.so.1.1"  
    ],  
    "QV_apis": [  
      "DSA_do_sign",  
      "DSA_do_verify",  
      "EVP_PKEY_get1_DSA",  
      ...  
      "RSA_verify"  
    ]  
  },  
  ...  
  {  
    "path": [  
      "/usr/bin/curl",  
      "/usr/lib/libcurl.so.4",  
      "/usr/lib/libcrypto.so.1.1"  
    ],  
    "QV_apis": [  
      "DH_get0_key",  
      "DSA_get0_key",  
      "DSA_get0_pqg",  
      "EVP_PKEY_get0_DH",  
      ...  
      "RSA_get0_key"  
    ]  
  }  
]
```

```
"EV_3": [  
  {  
    "static-trace": [  
      [  
        "/usr/bin/nmap",  
        "main"  
      ],  
      [  
        "/usr/bin/nmap",  
        "sub_3f340"  
      ],  
      ...  
      [  
        "/usr/bin/nmap",  
        "SSL_CTX_new"  
      ],  
      [  
        "/usr/lib/libssl.so.1.1",  
        "SSL_CTX_new"  
      ],  
      ...  
      [  
        "/usr/lib/libssl.so.1.1",  
        "EVP_PKEY_get0_RSA"  
      ],  
      [  
        "/usr/lib/libcrypto.so.1.1",  
        "EVP_PKEY_get0_RSA"  
      ]  
    ]  
  }  
]
```

비양자내성암호 탐지 연구 동향

- A Toolchain for Assisting Migration of Software Executables Towards Post-quantum Cryptography
 - 실험 결과
 - Synthetic Dataset (40개 파일)
 - OpenSSLv1.1.1 OpenSSLv3.3.1, MbedTLS v2.28.9, wolfSSLv5.7.2
 - SHA-512, AES-256 (Non QV) , DH, RSA, ECDSA (QV)
 - Direct Dependency Set
 - 각 라이브러리마다 5개의 예제 프로그램 작성
 - 각 프로그램은 해당 라이브러리의 API를 직접 호출
 - Indirect Dependency Set
 - 중간 shared library 경유하여 사용하도록 설계
 - 각 실행파일은 이 wrapper 라이브러리를 동적으로 링크

QED's Phases (→) Synthetic Dataset (↓)	P1		P1 +P2		P1 +P2 +P3	
	TP/FN (TPR)	TN/FP (TNR)	TP/FN (TPR)	TN/FP (TNR)	TP/FN (TPR)	TN/FP (TNR)
Direct Dependency	12/0 (100%)	0/8 (0%)	12/0 (100%)	8/0 (100%)	12/0 (100%)	8/0 (100%)
Indirect Dependency	12/0 (100%)	0/8 (0%)	12/0 (100%)	0/8 (0%)	12/0 (100%)	8/0 (100%)
Total	24/0 (100%)	0/16 (0%)	24/0 (100%)	8/8 (50%)	24/0 (100%)	16/0 (100%)

비양자내성암호 탐지 연구 동향

- A Toolchain for Assisting Migration of Software Executables Towards Post-quantum Cryptography
 - 실험 결과
 - Real-World Dataset (226개 실행파일)
 - Coreutils, UnixBench, curl/ssh 등 네트워크 도구, TPM 도구 포함
 - 평균분석시간 : 4초/ 실행파일
 - Dataset
 - Coreutils, UnixBench
 - 암호학적 프로그램이 아님
 - 정확히 탐지할 경우 non-QV로 분류되어야 함
 - Network
 - Curl, ssh, sftp, sshd, telnet, tracepath, wget, ping, scp
 - 7 프로그램이 OpenSNI v1l1. 사용

Phases (→) Set (↓)	P1		P1 +P2		P1 +P2 +P3	
	TP/FN (TPR)	TN/FP (TNR)	TP/FN (TPR)	TN/FP (TNR)	TP/FN (TPR)	TN/FP (TNR)
Coreutils	0/0 (100%)	109/0 (100%)	n/a	n/a	n/a	n/a
UnixBench	0/0 (100%)	18/0 (100%)	n/a	n/a	n/a	n/a
Network	7/0 (100%)	4/2 (67%)	7/0 (100%)	6/0 (100%)	6/1 (86%)	6/0 (100%)

비양자내성암호 탐지 연구 동향

- A Toolchain for Assisting Migration of Software Executables Towards Post-quantum Cryptography

```
suhrikim@aex-desk:~/qed$ ls
APIAnalysis.py          dataset-make.sh
BaseAnalysis.py         datasets
crypto_desc.py          FileDepende
dataset-install.sh      install.sh
suhrikim@aex-desk:~/qed$
```

```
mbedtls-install.sh  __pycache__  test.txt
openssl-install.sh  qed.py      tpm-install.sh
```

```
#openssl_APIs = list_dynsym("/usr/lib/x86_64-linux-gnu/libcrypto.so.1.1", openssl)
openssl11_APIs = ['DH_security_bits', 'i2d_RSAPublicKey', 'PEM_read_RSAPublicKey', 'EC_KEY_OpenSSL', 'RSA_print', 'RSA_padding_check_PKCS1_OAEP_mgf1', 'DH_get0_g', 'DHparams_print_fp', 'd2i_RSAPublicKey', 'DH_get0_p', 'EC_POINT_copy', 'EC_POINT_free', 'i2d_RSAPrivateKey', 'EC_KEY_MakeKeyPair', 'RSA_padding_add_PKCS1_PSS', 'DSAParams_print_fp', 'DH_check_parameters', 'DH_compute_key_padded', 'DH_get_default_method', 'd2i_EC_PUBKEY', 'DHparams_print', 'PKCS7_RECIP_INFO_get0_alg', 'EC_KEY_get0_privkey', 'RSA_size', 'RSA_X931_generate_key_ex', 'PEM_read_DSA_PUBKEY', 'PEM_read_bio_DHparams', 'RSA_meth_set_sign', 'EC_POINT_oct2point', 'RSA_get0_multi_prime_crt_params', 'RSA_meth_get_verify', 'RSA_generate_key', 'DH_get_2048_224', 'DH_meth_new', 'EVP_PKEY_get1_RSA', 'DSA_new', 'EC_KEY_up_ref', 'd2i_ECPParameters', 'DH_get_2048_256', 'RSA_meth_get_encrypt_flags', 'PEM_read_bio_DSAParams', 'RSA_padding_check_X931', 'ECDSA_sign', 'DH_set_flags', 'DH_generate_key', 'BN_RECP_CTX_set', 'd2i_DSA_SIGNATURE', 'PEM_read_ECPKParameters', 'd2i_RSA_PUBKEY', 'DSA_meth_get_mod_exp_callback', 'EC_KEY_priv2buf', 'DSA_meth_get_finish', 'EC_GFp_simple_method', 'EC_POINT_set_compressed_coordinates_GFp', 'd2i_RSAPrivateKey']
```

탐지할 API 목록 라이브러리별 정의

비양자내성암호 탐지 연구 동향

- A Toolchain for Assisting Migration of Software Executables Towards Post-quantum Cryptography

```
suhrikim@ex-desk:~/qed$ ls
APIAnalysis.py      dataset-make.sh
BaseAnalysis.py     datasets
crypto_desc.py      FileDependencyAnalysis.py
dataset-install.sh  install.sh
suhrikim@ex-desk:~/qed$
```

```
mbedtls-install.sh  __pycache__  test.txt
openssl-install.sh  qed.py       tpm-install.sh
```

```
out-rw #include <openssl/ec.h>
out-syn#include <openssl/obj_mac.h>
#include <openssl/err.h>
#include <stdio.h>
#include <stdlib.h>

// Function to create a new EC key pair and print the public key
void generate_ec_key(FILE *out) {
    EC_KEY *ec_key = NULL;
    const EC_POINT *pub_key = NULL;
    char *pub_key_hex = NULL;
    size_t key_size;

    // Create a new EC key pair
    ec_key = EC_KEY_new_by_curve_name(NID_X9_62_prime256v1);
    if (ec_key == NULL) {
        fprintf(out, "Error creating EC key\n");
        ERR_print_errors_fp(out);
        return;
    }

    // Generate the EC key pair
    if (EC_KEY_generate_key(ec_key) != 1) {
        fprintf(out, "Error generating EC key\n");
        ERR_print_errors_fp(out);
        EC_KEY_free(ec_key);
        return;
    }
}
```

테스트용 데이터셋 생성 가능

비양자내성암호 탐지 연구 동향

- A Toolchain for Assisting Migration of Software Executables Towards Post-quantum Cryptography

```
"RSA_generate_key_ex",
"RSA_get0_crt_params",
"RSA_get0_factors",
"RSA_get0_key",
"RSA_get_default_method",
"RSA_get_ex_data",
"RSA_meth_dup",
"RSA_meth_set1_name",
"RSA_meth_set_priv_dec",
"RSA_meth_set_priv_enc",
"RSA_new",
"RSA_public_decrypt",
"RSA_set0_crt_params",
"RSA_set0_factors",
"RSA_set0_key",
"RSA_set_ex_data",
"RSA_set_method",
"RSA_sign",
"RSA_size",
"d2i_ECPKParameters",
"o2i_ECPublicKey"
],
"path": [
    "./datasets/real-world/network/ssh",
    "/lib/x86_64-linux-gnu/libcrypto.so.1.1"
],
"type": "leaf"
}
```

```
},
{
    "elf": "./datasets/real-world/network/ssh",
    "shortest path": [
        "./datasets/real-world/network/ssh",
        "/lib/x86_64-linux-gnu/libcrypto.so.1.1"
    ],
    "type": "leaf"
},
{
    "elf": "./datasets/real-world/network/scp",
    "shortest path": [
        "./datasets/real-world/network/scp",
        "/lib/x86_64-linux-gnu/libcrypto.so.1.1"
    ],
    "type": "leaf"
}
]
}subrikim@ex-desk:~/ged/out-pw$
```

비양자내성암호 탐지 연구 동향

- A Toolchain for Assisting Migration of Software Executables Towards Post-quantum Cryptography
 - Example : 해시 함수 사용 경우

```
suhrikim@ex-desk:~/qed/fibsout$ cat api.txt
{
  "metadata": {
    "num_apps_before": 5,
    "num_total_before": 6,
    "num_apps_after": 0,
    "num_total_after": 1
  },
  "QV_apps": [],
  "report": [
    {
      "elf": "/lib/x86_64-linux-gnu/libcrypto.so.1.1",
      "api": [],
      "path": [
        "/lib/x86_64-linux-gnu/libcrypto.so.1.1"
      ],
      "type": "root"
    }
  ]
}
```

비양자내성암호 탐지 연구 동향

- A Toolchain for Assisting Migration of Software to quantum Cryptography
 - Example : openvpn

```

"metadata": {
  "num_apps_before": 1,
  "num_total_before": 14,
  "num_apps_after": 1,
  "num_total_after": 4
},
"QV_apps": [
  "./testm/openssl"
],
"report": [
  {
    "elf": "/lib/x86_64-linux-gnu/libcrypto.so.1.1",
    "path": [
      "/lib/x86_64-linux-gnu/libcrypto.so.1.1"
    ],
    "type": "root"
  },
  {
    "elf": "./testm/openssl",
    "shortest path": [
      "./testm/openssl",
      "/lib/x86_64-linux-gnu/libcrypto.so.1.1"
    ],
    "type": "leaf"
  }
]

```

```

"metadata": {
  "num_apps_before": 1,
  "num_total_before": 4,
  "num_apps_after": 1,
  "num_total_after": 4
},
"QV_apps": [
  "./testm/openssl"
],
"report": [
  {
    "elf": "/lib/x86_64-linux-gnu/libcrypto.so.1.1",
    "api": [],
    "path": [
      "/lib/x86_64-linux-gnu/libcrypto.so.1.1"
    ],
    "type": "root"
  },
  {
    "elf": "./testm/openssl",
    "api": [
      "DH_free",
      "DH_size",
      "DSA_bits",
      "EC_GROUP_get_curve_name",
      "EC_GROUP_order_bits",
      "EC_KEY_free",
      "EC_KEY_get0_group",
      "EC_KEY_new_by_curve_name",
      "EC_get_builtin_curves",
      "EVP_PKEY_get0_DSA",
      "EVP_PKEY_get0_EC_KEY",
      "EVP_PKEY_get0_RSA",
      "PEM_read_bio_DHparams",
      "RSA_bits",
      "RSA_flags",
      "RSA_free",
      "RSA_get0_key",
      "RSA_get_method",
      "RSA_meth_free",
      "RSA_meth_new",
      "RSA_meth_set0_app_data",

```

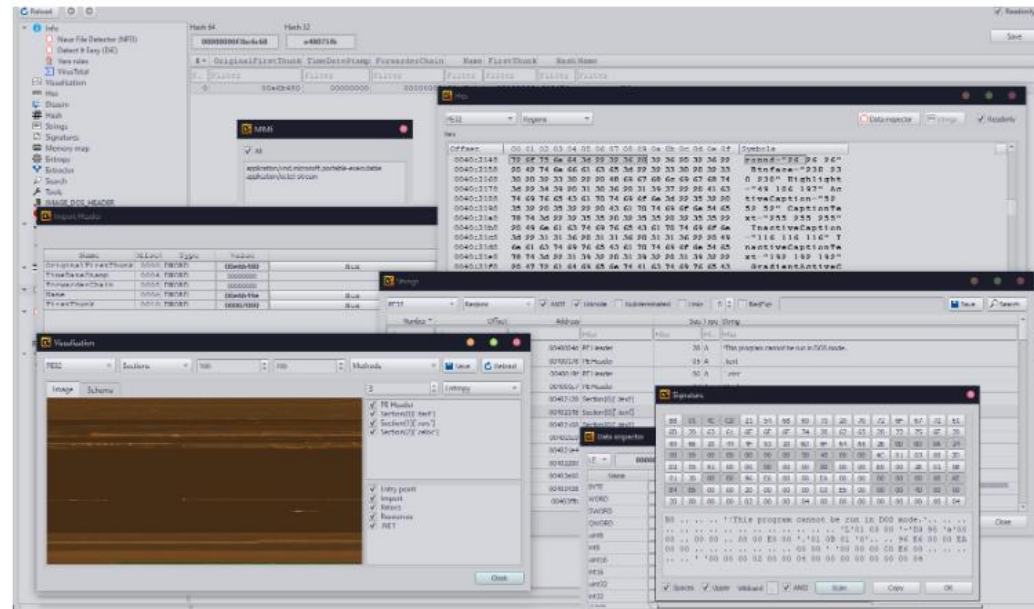

비양자내성암호 탐지 연구 동향

- A Toolchain for Assisting Migration of Software Executables Towards Post-quantum Cryptography
 - 실험 결과
 - OpenSSL 사용하는 경우 높은 정확도로 탐지
 - 자체 구현하거나 지정된 API 이외의 API 사용하는 경우 탐지 어려움

연구 진행 상황

비양자내성암호 탐지

- Detect It Easy (DIE)
 - 오픈 소스 파일 유형 식별 도구
 - Windows, Linux MacOS에서 사용가능
 - GUI 및 CLI 동시 지원
 - 주요 기능
 - 파일 형식 자동 식별 (PE, ELF, .NET 등..)
 - 패킹/암호화 식별
 - 컴파일러 및 라이브러리 식별
 - 내부 문자열 분석



비양자내성암호 탐지

- DIE 활용 API 탐지 shell script

```
> Users > suhrikim > Downloads > $ test_qv.sh
1
2  #!/bin/bash
3
4  PATTERNS="RSA,ECDH,ECC"
5  SEARCH_P=$(echo "$PATTERNS" | sed 's/,/ /g')
6
7
8  for file in *.*; do
9      if [ -f "$file" ] && [ "$file" != *.txt ] && [ "$file" != *.sh ]; then
10         echo "Detecting vulnerable algorithms in $file ..."
11
12         ret=$(/usr/bin/diec --i "$file" 2>/dev/null)
13
14         {
15             echo "===== File Information ====="
16             echo "$ret"
17             echo "==== List of Vulnerable Algorithms ====="
18             MATCHED_STRINGS=$(strings "$file" | grep -E "$SEARCH_P")
19             MATCH_COUNT=$(echo "$MATCHED_STRINGS" | wc -l)
20             echo "# Number of APIs detected: $MATCH_COUNT"
21             echo "$MATCHED_STRINGS"
22         } > "${file}_output.txt"
23
24         echo "$file analysis done ... output saved to ${file}_output.txt"
25
26     fi
27 done
```

비양자내성암호 탐지

- DIE 활용 API 탐지 shell script

```
> Users > suhrikim > Downloads > $ test_qv.sh
```

```
1
2  #!/bin/bash
3
4  PATTERNS="RSA,ECDH,ECC"
5  SEARCH_P=$(echo "$PATTERNS" | sed 's/,/\\/g')
6
7
8  for file in *; do
9      if [ -f "$file" ] && [ "$file" != *.txt ]
10      echo "Detecting vulnerable algorithms in $file"
11
12      ret=$(/usr/bin/diec --i "$file" 2>/dev/null)
13
14      {
15          echo "===== File Information ====="
16          echo "$ret"
17          echo "==== List of Vulnerable Algorithms ====="
18          MATCHED_STRINGS=$(strings "$file" | grep -E $SEARCH_P)
19          MATCH_COUNT=$(echo "$MATCHED_STRINGS" | wc -l)
20          echo "# Number of APIs detected: $MATCH_COUNT"
21          echo "$MATCHED_STRINGS"
22      } > "${file}_output.txt"
23
24      echo "$file analysis done... output saved to ${file}_output.txt"
25
26  fi
27
28 done
```

===== File Information =====

==== List of Vulnerable Algorithms ====

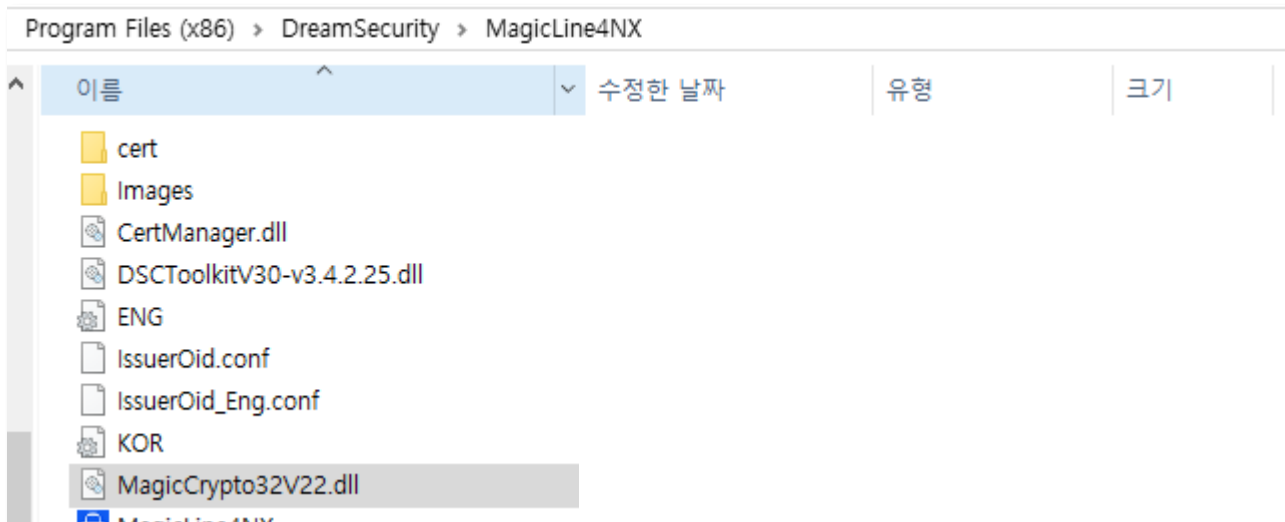
Number of APIs detected: 201

RSA_bits
RSA_meth_set_priv_enc
RSA_flags
RSA_size
RSA_set0_key
RSA_meth_set_finish
RSA_meth_set_pub_dec
RSA_meth_set0_app_data
RSA_free
RSA_new
RSA_set_flags
RSA_meth_free
RSA_meth_set_pub_enc
RSA_set_method
RSA_get0_key
RSA_meth_set_priv_dec
RSA_get_method
RSA_meth_set_init
EVP_PKEY_get0_RSA
RSA_meth_new
SSL_CTX_use_RSAPrivateKey

Openvpn 탐지

비양자내성암호 탐지

- API 를 활용한 탐지
 - 실험 : DreamSecurity MagicLine4NX
 - 공공·금융 사이트에서 공동인증서 기반 로그인/전자서명을 수행하는 클라이언트 모듈

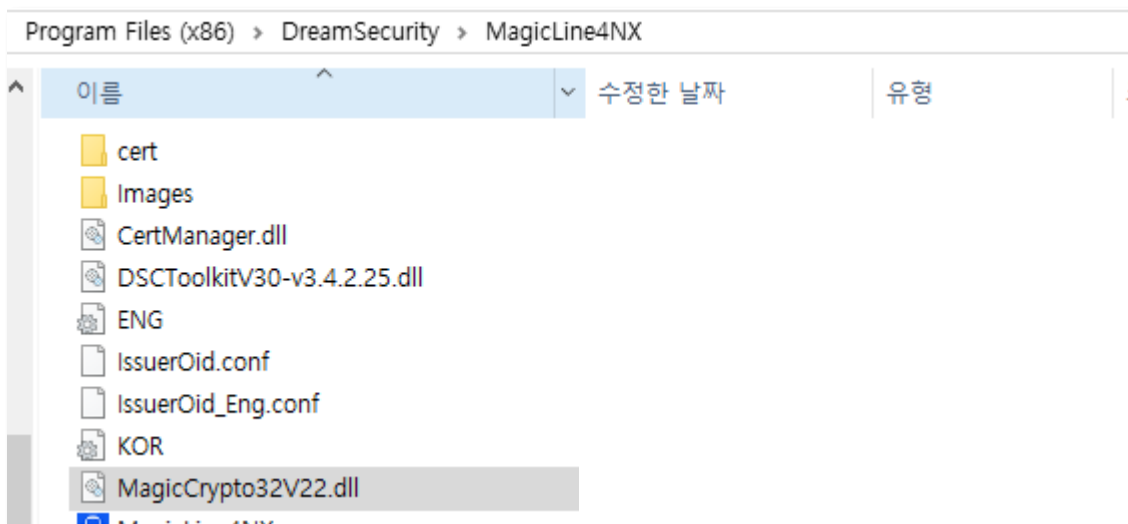


비양자내성암호 탐지

- API 를 활용한 탐지

- 실험 : DreamSecurity MagicLine4NX

- 공공·금융 사이트에서 공동인증서 기반 로그인/전자서명을
물



```
[+] ASCII hits (top 43)
BCryptCloseAlgorithmProvider
BCryptOpenAlgorithmProvider
CryptAcquireContextA
SHA256-ECDSA-B283
SHA256-ECDSA-K283
SHA256-ECDSA-P224
SHA256-ECDSA-P256
SHA256-KBDF-CTR
BCryptGenRandom
SHA256-RSA-OAEP
CryptGenRandom
SHA256-RSA-PSS
SHA256-KCDSA1
SHA256-PBKDF2
ARIA128-CCM
ARIA128-GCM
ARIA192-CCM
ARIA192-GCM
ARIA256-CCM
ARIA256-GCM
SHA256-DRBG
SHA256-HMAC
SHA384-HMAC
SHA512-HMAC
LEA128-CCM
LEA128-GCM
LEA192-CCM
LEA192-GCM
LEA256-CCM
LEA256-GCM
HIGHT-CBC
HIGHT-CTR
HIGHT-ECB
SEED-GMAC
SEED-CBC
SEED-CCM
SEED-CTR
SEED-ECB
SEED-GCM
SHA-224
SHA-256
SHA-384
SHA-512
```

PS C:\Users\subin\Documents> .\구제 #2025

LLM을 활용한 비양자내성암호 알고리즘 탐지 기술

- LLM (Large Language Model)
 - 대규모 언어 모델
 - 대량의 텍스트 데이터를 학습한 언어 모델
 - 주어진 문장을 이해하고 새로운 결과 생성

- LLM 의 종류



LLM을 활용한 비양자내성암호 알고리즘 탐지 기술

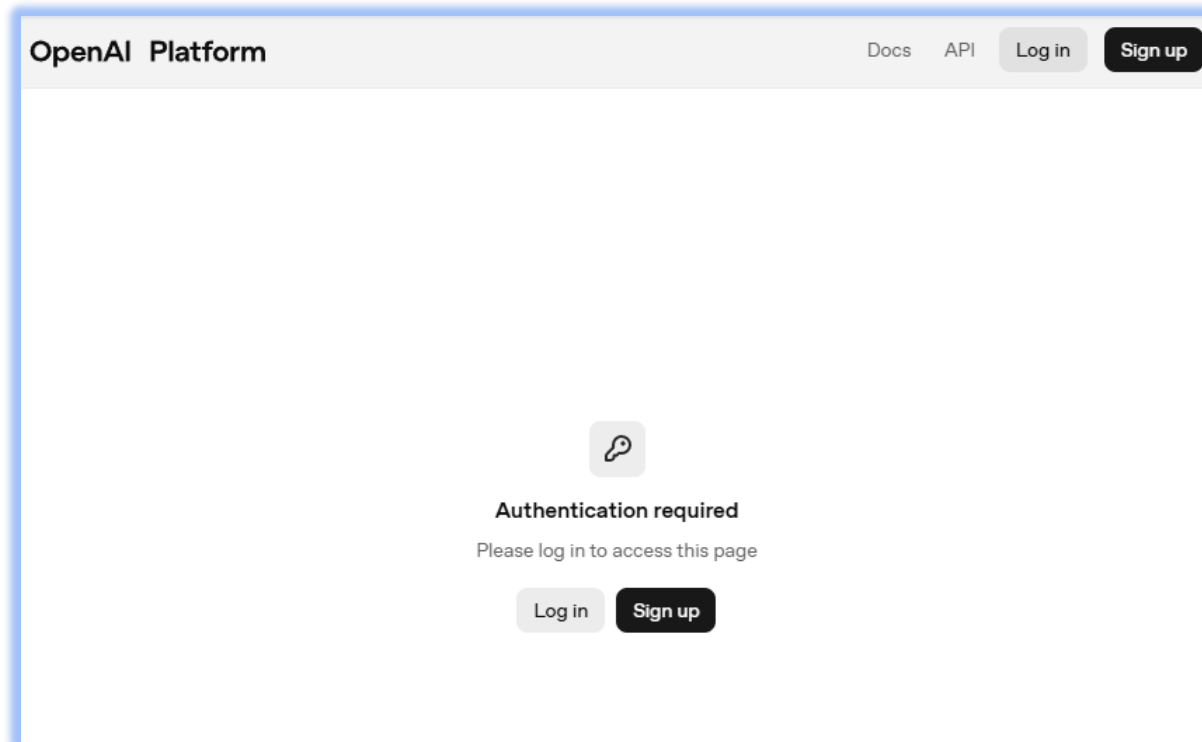
- OpenAI의 API
 - 토큰 : 언어 모델에서 텍스트의 일정 단위
 - 긴 글을 입력하거나 출력할 경우 더 많은 토큰을 사용하는데 이에 따라 비용이 증가

MODEL	INPUT	CACHED INPUT	OUTPUT
gpt-5	\$1.25	\$0.125	\$10.00
gpt-5-mini	\$0.25	\$0.025	\$2.00
gpt-5-nano	\$0.05	\$0.005	\$0.40
gpt-5-chat-latest	\$1.25	\$0.125	\$10.00
gpt-4.1	\$2.00	\$0.50	\$8.00
gpt-4.1-mini	\$0.40	\$0.10	\$1.60
gpt-4.1-nano	\$0.10	\$0.025	\$0.40
gpt-4o	\$2.50	\$1.25	\$10.00
gpt-4o-2024-05-13	\$5.00	-	\$15.00
gpt-4o-mini	\$0.15	\$0.075	\$0.60
gpt-realtime	\$4.00	\$0.40	\$16.00

1M token당 가격

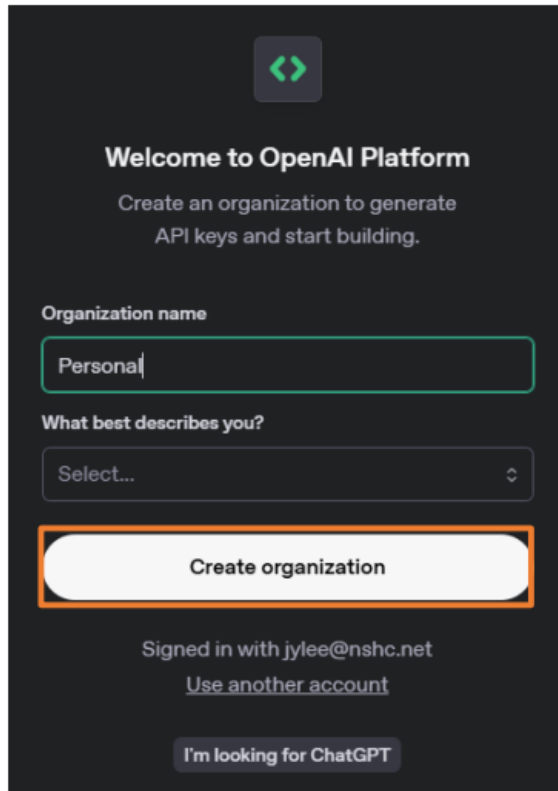
LLM을 활용한 비양자내성암호 알고리즘 탐지 기술


- OpenAI의 키 발급받기
 - 사이트: <https://platform.openai.com/api-keys>
 - Log in 후, Start building 클릭



LLM을 활용한 비양자내성암호 알고리즘 탐지 기술

- OpenAI의 키 발급받기
 - 사이트: <https://platform.openai.com/api-keys>
 - Log in 후, Start building 클릭





Welcome to OpenAI Platform

Create an organization to generate API keys and start building.

Organization name

What best describes you?

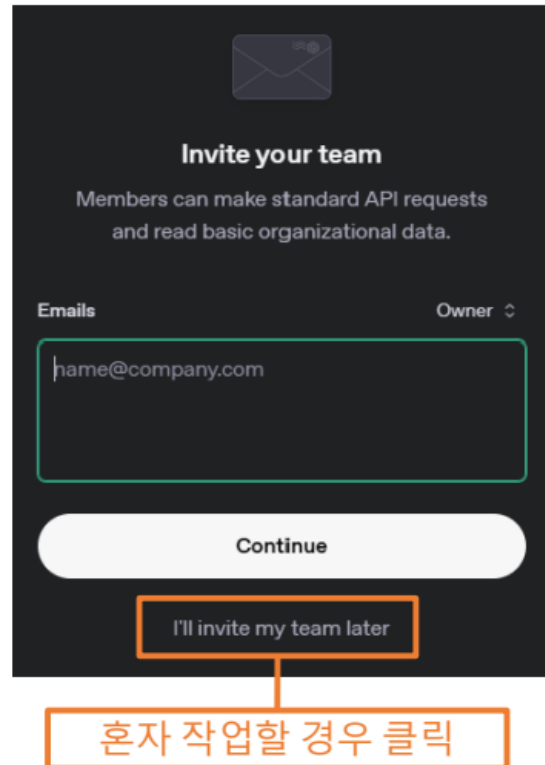
Select...


Create organization

Signed in with jylee@nshc.net

[Use another account](#)

[I'm looking for ChatGPT](#)





Invite your team

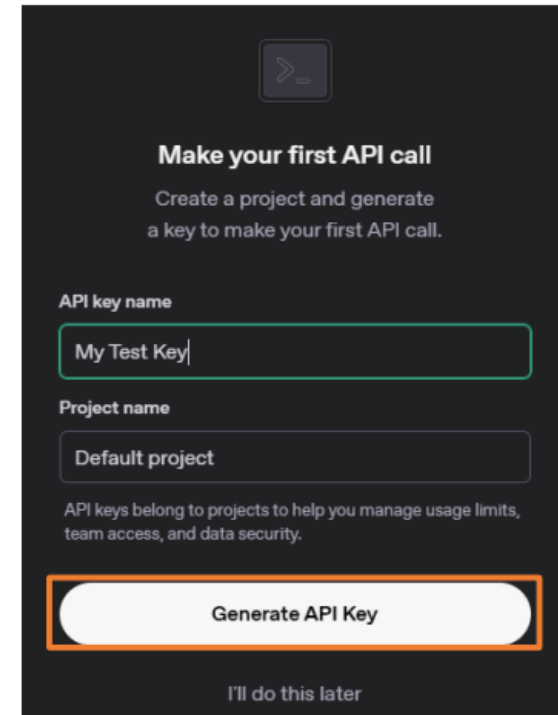
Members can make standard API requests and read basic organizational data.


Emails Owner

Continue

I'll invite my team later

혼자 작업할 경우 클릭





Make your first API call

Create a project and generate a key to make your first API call.

API key name

Project name

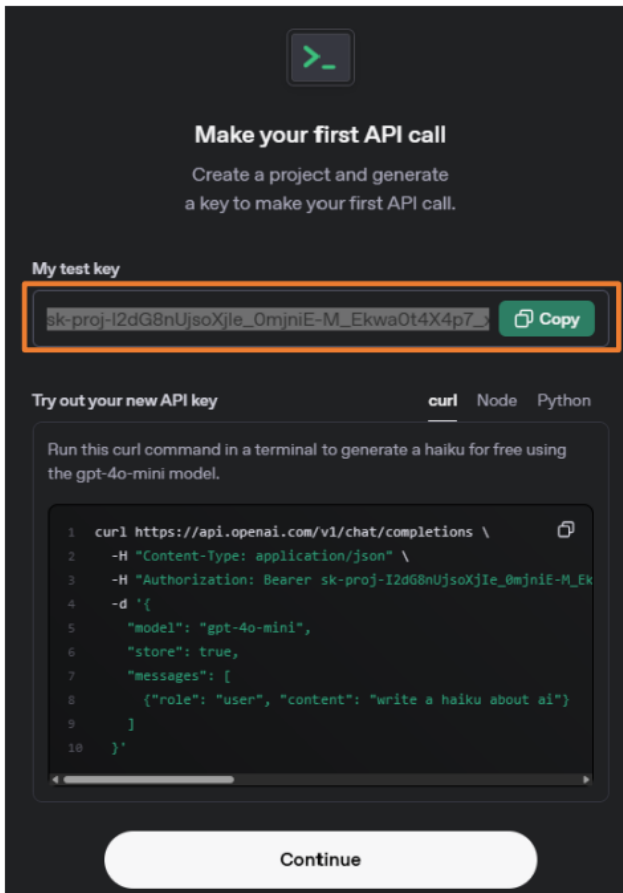
API keys belong to projects to help you manage usage limits, team access, and data security.

Generate API Key

[I'll do this later](#)

LLM을 활용한 비양자내성암호 알고리즘 탐지 기술

- OpenAI의 키 발급받기
 - 사이트: <https://platform.openai.com/api-keys>
 - Log in 후, Start building 클릭



Make your first API call

Create a project and generate a key to make your first API call.

My test key

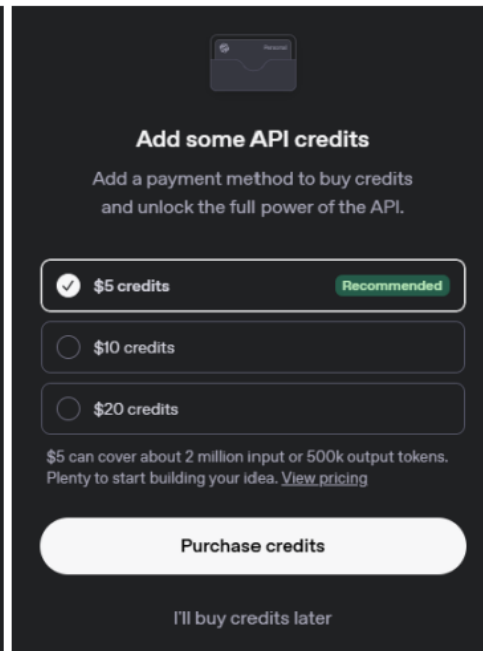
sk-proj-I2dG8nUjsoXjIe_0mjniE-M_EkwaOt4X4p7... [Copy](#)

Try out your new API key **curl** Node Python

Run this curl command in a terminal to generate a haiku for free using the gpt-4o-mini model.

```
1 curl https://api.openai.com/v1/chat/completions \
2 -H "Content-Type: application/json" \
3 -H "Authorization: Bearer sk-proj-I2dG8nUjsoXjIe_0mjniE-M_EkwaOt4X4p7..." \
4 -d '{
5   "model": "gpt-4o-mini",
6   "store": true,
7   "messages": [
8     {"role": "user", "content": "write a haiku about ai"}
9   ]
10 }'
```

[Continue](#)



Add some API credits

Add a payment method to buy credits and unlock the full power of the API.

☒ \$5 credits [Recommended](#)

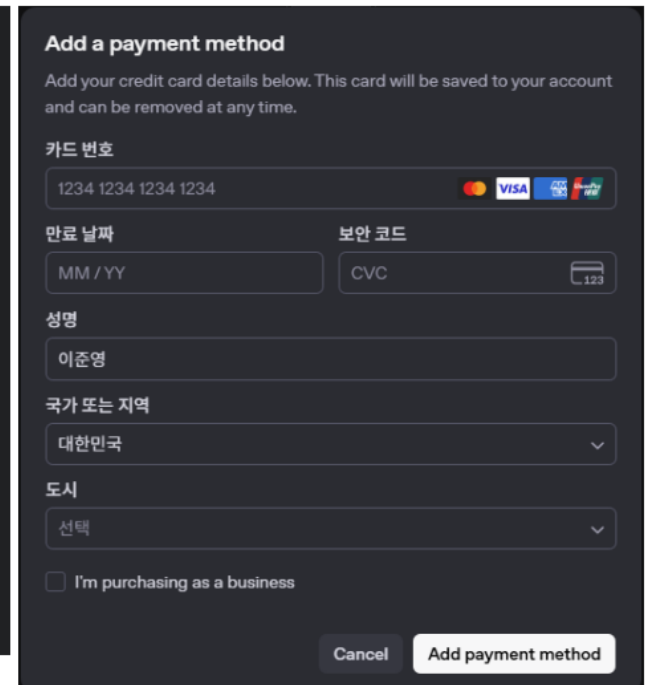
☐ \$10 credits

☐ \$20 credits

\$5 can cover about 2 million input or 500k output tokens. Plenty to start building your idea. [View pricing](#)

[Purchase credits](#)

[I'll buy credits later](#)



Add a payment method

Add your credit card details below. This card will be saved to your account and can be removed at any time.

카드 번호

1234 1234 1234 1234

만료 날짜

MM / YY

보안 코드

CVC

성명

이준영

국가 또는 지역

대한민국

도시

선택

☐ I'm purchasing as a business

[Cancel](#) [Add payment method](#)

LLM을 활용한 비양자내성암호 알고리즘 탐지 기술

- LLM을 활용한 비양자내성암호 탐지 적용 방안
 - 소스코드나 바이너리 파일의 연산 구조를 분석하여 암호 알고리즘을 판단
 - 암호 알고리즘이 들어간 함수 명, 파일 명, 변수 명 유무에 따른 탐지
 - 암호 알고리즘 표준문서나 기반 소스코드를 참고 유무에 따른 탐지

LLM을 활용한 비양자내성암호 알고리즘 탐지 기술

- 목표: 알고리즘 관련 문자열을 제거한 소스코드 대상 탐지

```
Welcome X detect_llm.py 5 X C test.c C test2.c 1
C: > Users > suhrikim > Documents > 과제 > 2025국보연 > detect_llm.py > ma

227 def main():
228     ap = argparse.ArgumentParser(description="AI-only crypt
229     ap.add_argument("--dir", type=str, default=".", help="
230     ap.add_argument("--model", type=str, default="gpt-4o-n
231     ap.add_argument("--metadata", type=str, default=None,
232     ap.add_argument("--include-hidden", action="store_true
233     ap.add_argument("--api-key", type=str, default=os.envi
234     args = ap.parse_args()
235
236     root = Path(args.dir).resolve()
237     if not root.is_dir():
238         print(f"--dir가 폴더가 아닙니다: {root}")
239         sys.exit(1)
240
241     meta = load_metadata(Path(args.metadata)) if args.meta
242     client = openai_client(args.api_key)
243
244     targets = []
245     for p in sorted(root.iterdir()):
246         if p.is_dir(): continue
247         if not args.include_hidden and p.name.startswith('.')
248         if p.suffix.lower() == ".c" or is_probably_binary(p)
249         targets.append(p)
250
```

모델/metadata/api-key 등 입력

LLM을 활용한 비양자내성암호 알고리즘 탐지 기술

- 목표: 알고리즘 관련 문자열을 제거한 소스코드 대상 탐지

```
SYSTEM_SOURCE = (  
    .... "당신은 '암호 구현 분석'에 숙련된 시니어 리서처입니다."  
    .... "아래 c/헤더 원문 블록만 근거로 사용된/구현된 암호 알고리즘을 판별하고,"  
    .... "알고리즘만 추출하세요. 관련된 주석은 극한 불확실성을 줄이기 위해 제거합니다." .....
```

C: > Users > suhrikim > Documents > 과제 > 2025국보연 > detect_llm.py > main

```
199 def analyze_one(client, model: str, path: Path, meta: Optional[Dict[str, Any]], args=None):  
215     blocks = sample_text_blocks(txt, block_chars=80_000, blocks=5,  
216     .... system_prompt = SYSTEM_SOURCE  
217     .... user_prompt = build_user_for_source(path.name, blocks, meta)  
218  
219     raw = call_chat_completion(client, model, system_prompt, user_prompt)  
220     raw = coerce_json_only(raw)  
221  
222     # 저장  
223     (outdir / f"{path.name}.analysis.json").write_text(raw, encoding="utf-8")  
224     (outdir / f"{path.name}.analysis.txt").write_text(json_to_txt(raw), encoding="utf-8")  
225     return outdir / f"{path.name}.analysis.json"  
226
```

Openai api를 활용해 분석

LLM을 활용한 비양자내성암호 알고리즘 탐지 기술

- 목표: 알고리즘 관련 문자열을 제거한 소스코드 대상 탐지

```
file Edit Selection View Go ... < ->
Welcome detect_llm.py 5 test.c x
C: > Users > suhrikim > Documents > 과제 > 2025국보연 > test.c
276
277 void XXX_enc(uint8_t *ct, uint8_t *pt)
278 {
279     for(int i=0; i<Nb; i++)
280     {
281         ct[i]=pt[i]^Rkey[0][i];
282     }
283     #ifdef DEBUG
284     PrintValue(ct);
285     #endif
286     for(int i=1; i<numr; i++)
287     {
288         for(int j=0; j<Nb; j++)
289         {
290             ct[j]=sbox[ct[j]];
291         }
292     }
293     #ifdef DEBUG
294     PrintValue(ct);
295     #endif
296     //sub1
```

test.c (AES)

```
Welcome x detect_llm.py 5 test.c test2.c 1 x
C: > Users > suhrikim > Documents > 과제 > 2025국보연 > test2.c > ...
181
182 int32_t
183 K_XXX_key_schedule(uint8_t *key,
184                   uint32_t keyLength,
185                   XXX_ALG_INFO *AlgInfo)
186 {
187     uint8_t tempKey[XXX_MAX_KEY_SIZE];
188     uint8_t t[16], w1[16], w2[16], w3[16];
189     uint8_t *w0, *rk;
190     int i, j, R, dir;
191
192     if(AlgInfo == NULL || key == NULL)
193         return -1;
194     dir = AlgInfo->dir;
195
196     if(keyLength < XXX_MIN_KEY_SIZE){
197         return -2;
198     }
199     else if((16 <= keyLength) && (keyLength < 24)){
200         AlgInfo->rounds = R = 12;
201         AlgInfo->keyLength = 16;
202     }
```

test.c (ARIA)

LLM을 활용한 비양자내성암호 알고리즘 탐지 기술

- TEST 1 : 알고리즘 정보 없이 탐지

```
suhrikim@ex-desk:~/qed/testai$ python3 detect_llm.py --dir . --model gpt-4o --api-key sk-pro
```

```
대 상 파 일 수 : 2
```

```
[1/2] test.c 분석 중 ...
```

```
↳ JSON 저장 : reports/test.c.analysis.json
```

```
[2/2] test2.c 분석 중 ...
```

```
↳ JSON 저장 : reports/test2.c.analysis.json
```

```
suhrikim@ex-desk: ~/qed/testai/reports$ cat test.c.analysis.txt
```

```
[file] test.c
```

```
[algorithms]
```

```
- AES (family=AES)
```

```
· blockSizeBits: 128
```

```
· keySizeBits: 128
```

```
· rounds: 10
```

```
· evidence: The code defines constants and operations typical of AES, such as a 16-byte block s  
10-round structure (numr = 10), and the use of S-boxes and inverse S-boxes for substitution. The ke  
schedule uses a round constant array (CONS) and operations like RotWord and SubWord, which are  
characteristic of AES key expansion. The use of a 128-bit key is implied by the size of the Rkey  
array and the initial key size in the main function.
```

```
[notes]
```

```
The implementation includes both encryption and decryption functions, as well as a CTR mode  
operation, which is consistent with AES usage. The presence of specific AES components like the
```

test.c (AES) → AES 로 분석

LLM을 활용한 비양자내성암호 알고리즘 탐지 기술

- TEST 1 : 알고리즘 정보 없이 탐지

```
suhrikim@ex-desk:~/qed/testai$ python3 detect_llm.py --dir . --model gpt-4o --api-key sk-pro
```

```
대 상 파 일 수 : 2
```

```
[1/2] test.c 분석 중 ...
```

```
↳ JSON 저장 : reports/test.c.analysis.json
```

```
[2/2] test2.c 분석 중 ...
```

```
↳ JSON 저장 : reports/test2.c.analysis.json
```

```
suhrikim@ex-desk:~/qed/testai/reports$ cat test2.c.analysis.txt
```

```
[file] test2.c
```

```
[algorithms]
```

```
- AES (family=AES)
  · blockSizeBits: 128
  · keySizeBits: [128, 192, 256]
  · rounds: {'128': 12, '192': 14, '256': 16}
  · evidence: The code defines a block size of 16 bytes (128 bits) and supports key sizes of 16, 24, (128, 192, and 256 bits). The S-box used in the code matches the AES S-box, and the key schedule involves multiple rounds with transformations similar to AES. The number of rounds is set to 12, 14, or 16 based on the key size, which aligns with AES-128, AES-192, and AES-256 specifications.
```

```
[notes]
```

```
The algorithm appears to be a variant or implementation of AES, given the use of an S-box similar to AES and the key schedule structure. However, the specific transformations and round constants (RCON) are not fully defined in the provided code snippet.
```

test2.c (ARIA) → AES 로 분석

LLM을 활용한 비양자내성암호 알고리즘 탐지 기술

- TEST 2 : ARIA의 경우 알고리즘 정보 입력 후 탐지

```
Welcome detect_llm.py 5 test.c test2.c 1 !
C: > Users > suhrikim > Documents > 과제 > 2025국보연 > ! alginfo.yaml
1  algorithms:
2    - name: ARIA
3      family: ARIA
4      blockSizeBits: 128
5      keySizeBits: [128, 192, 256]
6      rounds: {"128": 12, "192": 14, "256": 16}
7      roundKeyConstants:
8
9        - name: C1
10         bytes: [0x51, 0x7c, 0xc1, 0xb7, 0x27, 0x22, 0x0a, 0x9
11
12         - name: C2
13         bytes: [0x6d, 0xb1, 0x4a, 0xcc, 0x9e, 0x21, 0xc8, 0x2
14
15         - name: C3
16         bytes: [0xdb, 0x92, 0x37, 0x1d, 0x21, 0x26, 0xe9, 0x7
17
18     sboxes:
19       S1:
20         bytes: [0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc
21
22         0xb7, 0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc, 0x3
```

alginfo.yaml

LLM을 활용한 비양자내성암호 알고리즘 탐지 기술

- TEST 2 : ARIA의 경우 알고리즘 정보 입력 후 탐지

```
suhrikim@ex-desk:~/qed/testtai$ python3 detect_llm.py --dir . --model gpt-4o --api-key sk-proj-iArkobN2LDKA52j0BYy9HPgCVqftYNTwwDa
--metadata alginfo.yaml

대상 파일 수 : 2
[1/2] test.c 분석 중 ...
↳ JSON 저장 : reports/test.c.analysis.json
[2/2] test2.c 분석 중 ...
↳ JSON 저장 : reports/test2.c.analysis.json
suhrikim@ex-desk:~/qed/testtai$ cd reports/
suhrikim@ex-desk:~/qed/testtai/reports$ cat test2.c.analysis.txt
[file] test2.c

[algorithms]
- ARIA (family=ARIA)
  · blockSizeBits: 128
  · keySizeBits: [128, 192, 256]
  · rounds: {'128': 12, '192': 14, '256': 16}
  · evidence: The code defines constants for block and key sizes that match ARIA's specifications. The S-boxes and
round key constants (KRK_12, KRK_14, KRK_16) are consistent with ARIA. The key schedule and
encryption functions use operations typical of ARIA, such as the use of S-boxes and key rotation.

[notes]
The code appears to implement the ARIA block cipher, as indicated by the use of ARIA-specific
S-boxes and round key constants. The structure and operations in the key schedule and encryption
functions align with ARIA's design.suhrikim@ex-desk:~/qed/testtai/reports$
```

test2.c (ARIA) → ARIA 로 분석

LLM을 활용한 비양자내성암호 알고리즘 탐지 기술

- TEST 3 : DES로 함수명 변경 후 탐지 (추가 정보 X)

```
suhrikim@ex-desk:~/qed/testai$ cat reports/test.c.analysis.txt
```

```
[file] test.c
```

```
[algorithms]
```

```
- AES (family=AES)
```

```
· blockSizeBits: 128
```

```
· keySizeBits: 128
```

```
· rounds: 10
```

```
· evidence: The code includes an S-box and inverse S-box which are characteristic of AES. The use of a 16-byte block size (Nb = 16) and a 10-round structure (numr = 10) aligns with AES-128. The round key generation process and operations like SubBytes, ShiftRows, and MixColumns (sub1, sub3, inv_sub1, inv_sub3) are consistent with AES encryption and decryption processes.
```

```
[notes]
```

```
Despite the presence of 'DES' in function names and comments, the implementation details such as the S-box, block size, and round structure clearly indicate AES rather than DES. The use of a 128-bit block size and 10 rounds is specific to AES-128.suhrikim@ex-desk:~/qed/testai$ cat reports/test2.c.analysis.txt
```

```
[file] test2.c
```

```
[algorithms]
```

```
- AES (family=AES)
```

```
· blockSizeBits: 128
```

```
· keySizeBits: [128, 192, 256]
```

```
· rounds: {'128': 10, '192': 12, '256': 14}
```

```
· evidence: The presence of the S-box (S) and the key schedule constants (KRK_12, KRK_14, KRK_16) are indicative of AES. The function K_DES_key_schedule uses key lengths of 16, 24, and 32 bytes, which correspond to AES-128, AES-192, and AES-256. The rounds are set to 12, 14, and 16, which align with AES's 10, 12, and 14 rounds for 128, 192, and 256-bit keys respectively. The use of a 16-byte block size also matches AES's 128-bit block size.
```

```
[notes]
```

```
The file name and some variable names suggest DES, but the implementation details such as the S-box and key schedule are consistent with AES, not DES. The block size and key sizes further confirm this.suhrikim@ex-desk:~/qed/testai$
```

test2.c (ARIA) → AES 로 분석

LLM을 활용한 비양자내성암호 알고리즘 탐지 기술

- TEST 4 : DES로 함수명 변경 후 탐지 (추가 정보 0)

```
suhrikim@ex-desk:~/qed/testai$ cat reports/test.c.analysis.txt
[file] test.c

[algorithms]
- AES (family=AES)
  · blockSizeBits: 128
  · keySizeBits: 128
  · rounds: 10
  · evidence: The code defines constants and functions that are characteristic of the AES algorithm, such as the S-box and inverse S-box tables, the round constant array (CONS), and the key schedule process (RoundKeyGen function). The use of 128-bit blocks (Nb = 16 bytes) and 10 rounds (numr = 10) further supports this identification.

[notes]
The code appears to implement AES encryption and decryption, despite the misleading use of 'DES' in function names and comments. The presence of AES-specific components like the S-box, inverse S-box, and round constants confirms that reports/test2.c.analysis.txt
[file] test2.c

[algorithms]
- ARIA (family=ARIA)
  · blockSizeBits: 128
  · keySizeBits: [128, 192, 256]
  · rounds: {'128': 12, '192': 14, '256': 16}
  · evidence: The code defines constants KRK_12, KRK_14, and KRK_16, which are used for different key lengths and rounds, matching the ARIA specification. The S-boxes and key schedule logic also align with ARIA's structure.

[notes]
The code appears to implement the ARIA block cipher, as indicated by the use of specific S-boxes and round constants that match the ARIA specification. The use of 'DES' in variable names is misleading and likely a placeholder or error, as the algorithm characteristics do not match DES encryption.
```

두 파일 제대로 탐지

결론 및 향후 계획

- LLM을 활용한 비양자내성암호 탐지 결과

	TEST 1	TEST 2	TEST 3	TEST 4
	코드 only	코드 & 레퍼런스	다른 알고리즘으로 변경	다른 알고리즘으로 변경 & 레퍼런스
AES	O	O	O	O
ARIA	X	O	X	O

- 결과 분석
 - 국산 암호의 경우 코드 만 가지고 분석하기 어려운 경향이 있음
 - 향후 공개키 암호 + 실행파일로 확장할 예정
 - LLM의 분석 근거를 활용해 프롬프트 최적화 고려

Thank you