https://doi.org/10.7840/kics.2023.48.6.630

블랙박스 암호모듈에 적용 가능한 CRYSTALS-Kyber의 은닉채널 공격과 IP카메라의 잠재적 보안 위협

최 영 락•, 염 용 진*, 강 주 성*

Subliminal Channels of CRYSTALS-Kyber Applicable in the Black Box Cryptographic Module and Potential Security Threats of IP Cameras

Youngrak Choi•, Yongjin Yeom*, Ju-Sung Kang*

요 약

은닉채널은 미국과 소련의 전략무기제한협정 과정에서 처음 등장한 개념으로 특정 개체가 은밀하게 정보를 취득할 가능성이 있는 암호시스템을 일컫는다. Young과 Yung은 현대 암호시스템에 적용 가능한 은닉채널의 일종인 SETUP(Secretly Embedded Trapdoor with Universal Protection) 개념을 창시하였다. SETUP은 공격자가 암호시스템을 변형하여 백도어를 삽입하는 메커니즘으로 실용적 관점에서 충분히 위협적인 것으로 알려져 있다. 본 논문에서는 최근 미국 NIST가 주관한 양자내성암호 표준화 공모전에서 최종 승자로 선정된 CRYSTALS-Kyber의 초기 모델에 SETUP이 형성될 수 있음을 보인다. 공격자의 배타적 공격능력을 향상시키기 위하여 X25519 키 교환 프로토콜을 사용하여 CRYSTALS-Kyber의 SETUP을 형성한다. 다음으로 SETUP에 의해 생성된 CRYSTALS-Kyber 알고리즘이 IP카메라와 스마트폰의 암호통신에 사용될 경우에 공격자가 사용자의 개인키를 추출해 낼 수 있음을 입증한다. 마지막으로 이러한 SETUP을 방지하기 위한 대응책을 알고리즘 설계 관점과 보안 정책적 측면으로 분류하여 제시한다.

키워드 : 은닉채널, SETUP, 양자내성암호, CRYSTALS-Kyber, IP카메라
Key Words : Subliminal channel, SETUP, Post-Quantum Cryptography, CRYSTALS-Kyber, IP cameras

ABSTRACT

Subliminal channel is the concept pointed out in the US-Soviet Strategic Arms Limitation Treaty, and refers to a cryptographic system in which a particular individual is likely to obtain information without anyone knowing. Later, Young and Yung invented the concept of SETUP(Secretly Embedded Trapdoor with Universal Protection), a type of Subliminal channel. Attacks by SETUP can be a sufficient threat in reality.

In this paper, we show that SETUP can be formed in the initial model of CRYSTALS-Kyber, the winner of the NIST's PQC(Post-Quantum Cryptography) Standardization process. We form SETUP of CRYSTALS-Kyber by using the X25519 key exchange protocol in order to improve the attacker's exclusive attack capability. Next,

※ 본 연구는 정부(과학기술정보통신부)의 재원으로 과학기술일자리진흥원-과학치안 공공연구성과 실용화 촉진 시범사업(1711174177)의 지원을 받아 수행되었습니다.
• First Author : Kookmin University Department of Financial Information Security, alpha1996@naver.com, 학생회원
* Kookmin University Department of Information Security, Cryptology and Mathematics, salt@kookmin.ac.kr, 종신회원; jskang@kookmin.ac.kr, 종신회원
논문번호 : 202302-013-A-RN Received January 26, 2023; Revised April 5, 2023, Accepted April 10, 2023

we demonstrate that an attacker can extract a user's private key where the CRYSTALS-Kyber generated by SETUP is used for cryptographic communication between IP cameras and smartphones. Finally, we consider countermeasures to prevent our SETUP in terms of algorithmic design principle and security policy.

Ⅰ. 서 론

은닉채널(subliminal channel)은 1978년 냉전시대에 미국과 소련의 군축 협상 과정에서 Simmons[1,2]가 최초로 언급하여 탄생한 개념이다. 당시 미국은 소련과의 전략무기제한협정(SALT: Strategic Arms Limitation Treaty)에 의해 미사일 격납고 1,000개 중 배치 가능한 미사일의 수를 100개로 제한해야 하는 상황에 놓여 있었다. 다른 한편으로는 소련의 선제공격에 대비하여 배치된 미사일의 실제 위치는 숨겨야 할 필요성이 있었다. 따라서 미국은 SALT를 준수하고 있다는 사실을 소련과 국제사회에 증명하기 위하여 격납고를 개방하지 않은 상태에서 협약된 미사일의 개수를 누구나 알 수 있게 조치하는 방법을 모색해야 했다. 두 가지 상충하는 조건을 만족시키기 위한 프로토콜(protocol) 개발을 위하여 미국과 소련은 우선 격납고에 미사일의 존재 여부를 감지할 수 있는 센서를 양국이 공동으로 참여하여 설치하기로 하였다.

센서의 출력값은 미사일이 존재하면 "1", 존재하지 않으면 "0"으로 나타낸다. k 번째($1 \le k \le 1{,}000$) 격납고에 설치된 센서의 출력값을 x_k라 놓으면, $\sum_{k=1}^{1,000} x_k = 100$을 만족하는 것이 SALT를 준수하는 조건이 된다. x_k가 포함된 정보는 양국이 사전에 제작한 전용 암호장치에 입력되어 변환된 값인 y_k가 출력된다. 전용 암호장치에 사용된 변환 메커니즘은 현재 통용되는 암호학적 용어로 설명하면 공개키암호 기반의 디지털서명(digital signature) 알고리즘으로 미국과 소련이 각각의 개인키로 메시지에 서명하는 이중 서명 방식이다. 즉, 센서의 출력값이 포함된 메시지에 먼저 미국의 개인키로 서명하고, 이 값을 다시 소련의 개인키로 서명하는 것이다. 이중 서명의 결과값인 $y_1, \ldots, y_{1,000}$을 알고 있는 당사자는 누구나 소련과 미국의 공개키를 이용하여 서명값을 검증할 수 있다. y_k에는 격납고의 위치 정보인 k가 없기 때문에 결과적으로 $\sum_{k=1}^{1,000} x_k = 100$이라는 사실은 입증된 상태에서 배치된 미사일의 위치는 보호할 수 있게 된다. 이 프로토콜의 전체적인 과정은 [그림 1]에 나타나 있다.

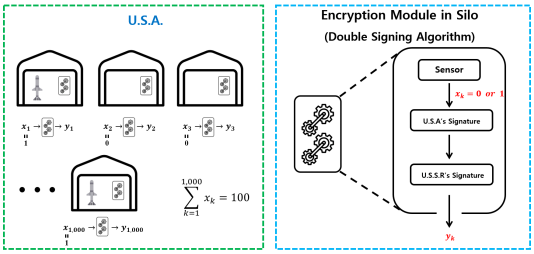


그림 1. 미국과 소련의 전용 암호장치
Fig. 1. U.S. and Soviet cryptographic module

한편, Simmons는 소련이 사용하는 서명 알고리즘에 백도어(backdoor) 등과 같은 의도적 결함이 있을 경우, 미사일의 위치가 노출될 수 있음을 지적하였다. 예를 들어, 하나의 메시지에 대한 유효 서명 값이 여러 개 있고, 소련 측이 그 서명값들을 구분할 수 있다면, 구분된 정보로부터 미국이 인지하지 못하는 상태에서 미사일의 위치 정보 추적이 가능하다는 것이다[3,4,5]. 이와 같이 프로토콜 참여자가 의식할 수 없는 상태에서 특정 개체가 원하는 정보를 얻을 가능성이 있는 시스템 상의 태생적 결함을 은닉채널(subliminal channel)이라 한다. Simmons가 언급한 은닉채널의 예시를 개념적으로 표현한 것이 [그림 2]이다.

미국과 소련이 체결한 SALT 협정 당시에 Simmons에 의해 언급된 은닉채널에 관한 문제는 양국이 세부 프로토콜 설계 과정에서 충분히 논의된 것으로 알려져 있으나, 구체적인 내용은 군사기밀이므로 명확히 알려진 문헌을 찾기는 힘들다. 그러나 Simmons에 의해 제기된 은닉채널 문제는 암호학계에서 그 이후로도 군사용 암호장치뿐만 아니라 다양한

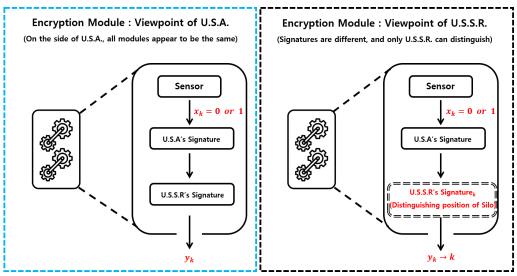


그림 2. 은닉채널이 존재하는 전용 암호장치
Fig. 2. Cryptographic module which has subliminal channel

범용의 암호시스템 설계와 구현 과정에서 반드시 고려해야 하는 중요한 주제로 인식되고 있다.

1.1 Young-Yung의 은닉채널

은닉채널의 대표적인 예시로는 SETUP(Secretly Embedded Trapdoor with Universal Protection)을 들 수 있다. SETUP은 Young-Yung[6]이 최초로 정의한 개념으로 블랙박스 환경에 있는 암호 시스템 Γ의 내부 알고리즘을 변형하여 Γ'을 생성하는 것을 뜻한다. 이 때, Γ 와 Γ'의 입력과 출력의 형태가 같아야 하고 출력은 구분 불가능해야 한다. 임의의 사용자가 악의적인 공격자의 SETUP에 의해 생성된 Γ'을 이용하여 암호통신을 할 경우, 공격자는 변형된 키 생성 알고리즘을 통해 사용자의 개인키 정보를 추출 할 수 있다. Young-Yung은 [6]에서 실제로 RSA와 같이 널리 사용되는 암호 알고리즘도 키 생성 단계에서 난수 생성 알고리즘을 의도적으로 변경할 경우 SETUP이 가능함을 보였다. [그림 3]은 사용자가 암호시스템 Γ'으로 키 생성을 했을 때 공격자가 사용자의 개인키를 추출하는 과정이다.

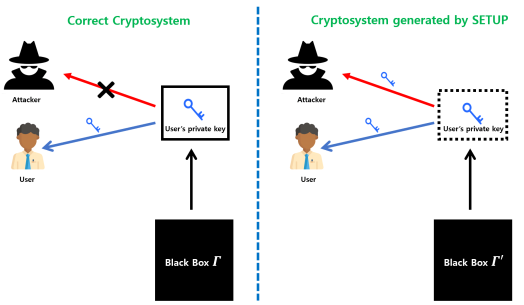


그림 3. SETUP에 의해 생성된 Γ'을 이용한 공격자의 키 추출 과정
Fig. 3. Attacker's key extraction process by using Γ' generated by SETUP

1.2 Dual_EC_DRBG의 은닉채널

미국 NIST(National Institute of Standards and Technology)의 Dual_EC_DRBG[7] 표준화 과정은 SETUP의 위험성을 시사하는 대표적인 사례이다. Dual_EC_DRBG는 2004년 NIST 워크숍에서 최초로 공개된 타원곡선 기반 난수 생성 알고리즘이다. 난수 생성을 위해 타원곡선 위의 두 점 P와 Q가 사용된다. P는 난수발생기의 내부 상태(internal state)를 갱신하는 데 쓰이는 점이고, Q는 난수발생기의 내부 상태로부터 난수를 출력하기 위해 사용되는 점이다. 여기에서 주목할 것은 내부 상태를 갱신하는 데 사용되는 함수와 난수를 출력하는 함수의 형태가 동일하다는 점이다. 즉, 타원곡선의 상수배 연산 후 x좌표만 이용하는 형태를 내부상태 갱신함수와 난수열 출력함수에 모두 사용한다. Dual_EC_DRBG는 이러한 단순한 구조와 더불어 당시의 다른 난수발생기 표준에 비해 1,000배 이상 느리고, 출력 난수의 분포도 균일하지 않아 많은 비판에 직면했음에도 불구하고 NIST가 표준으로 채택하여 많은 의문을 남겼다. 그러던 중 2007년 Microsoft의 Shumow-Ferguson[7]은 두 점 P와 Q가 하나의 난수로부터 생성될 경우, Dual_EC_DRBG의 모든 출력비트가 예측 가능하다는 사실을 지적함으로써 백도어가 삽입될 여지가 충분함을 보여주었다. 2013년도에는 Dual_EC_DRBG의 표준화 과정에 NSA가 개입했음을 암시하는 자료가 Snowden[8]에 의해 발견이 되면서 NSA의 고의적 백도어 알고리즘 삽입 의혹은 증폭되었고, 결국 Dual_EC_DRBG는 2014년부로 NIST 표준에서 사라지게 되었다.

1.3 본 논문의 연구 결과

Simmons가 SALT 과정에서 언급한 은닉채널과 Dual_EC_DRBG 표준화 과정에서 노출된 은닉채널 문제는 현실에서 충분한 위협이 될 수 있음을 실증적으로 보여준 사례이다. 따라서 은닉채널의 존재 가능성은 암호시스템 설계와 구현을 포함한 암호제품이 설치되는 모든 과정에서 반드시 고려해야 하는 매우 중요한 사안이 되었다. 최근 암호학계에서 가장 큰 관심의 대상인 양자내성암호(PQC: Post-Quantum Crypto)의 설계 및 표준화 과정에서도 은닉채널에 관한 연구는 중요한 주제로 다루어지고 있다[9,10].

본 논문에서는 먼저 Young-Yung[6]의 SETUP 개념에 대하여 심층적인 조사 분석을 실시하였다. SETUP을 만족시키기 위한 5가지 조건들의 의미와 당위성을 규명하기 위하여 우리는 적절한 예시들을 제시한다. 다음으로 최근 5년 여에 걸쳐 NIST에서 추진했던 미국의 양자내성암호 표준화 공모전에서 최종 승자로 선정된 알고리즘인 CRYSTALS-Kyber의 초기 모델 내에서 SETUP 과정을 통한 은닉채널 형성이 가능함을 학술적으로 입증한다. 여기에서 안전성 분석 대상으로 사용한 CRYSTALS-Kyber 알고리즘은 초기 논문인 [11]에 있는 'Algorithm 1 : Kyber-CPA-Keygen'이다. 우리는 RSA를 이용한 기본적인 SETUP을 먼저 서술하고, 공격자의 배타적 공격 능력을 향상시키기 위한 방법으로 Curve25519[12]를 이용한 ECDH 키 교환 메커니즘인 X25519를 사용한

SETUP을 제시한다. 또한 제시한 SETUP이 CRYSTALS-Kyber의 내부 알고리즘을 수정하지 않은 채 함께 사용하는 난수발생기의 변형만으로 가능함을 보인다.

SETUP에 의해 생성된 CRYSTALS-Kyber는 Dual_EC_DRBG의 사례와 마찬가지로 현실에서 큰 위협이 될 수 있다. 이를 실증적으로 보여주기 위하여 우리는 IP카메라 암호 통신 프로토콜에 SETUP에 의해 생성된 CRYSTALS-Kyber가 사용될 경우의 시나리오를 제시한다. 본 시나리오 상에서는 특정 개체가 모든 시스템 사용자의 IP카메라 화면을 감시할 수 있다. 여기에 사용되는 IP카메라 모델은 [13]의 IP카메라 제품들을 기준으로 한다.

우리는 SETUP 기반 은닉채널을 예방하기 위한 대응책에 대해서도 살펴본다. 보안 정책적 측면과 알고리즘 설계 관점으로 대별하여 대응책을 논한다. 암호 알고리즘 설계 단계의 대응책으로는 SETUP이 적용된 난수발생기와 정상적인 CRYSTALS-Kyber 알고리즘이 결합할 때 생성될 수 있는 은닉채널을 방지하기 위해 사용된 기법을 중심으로 살펴볼 것이다. 실제로 NIST PQC 표준으로 최종 선정된 CRYSTALS-Kyber 알고리즘 버전은 본 논문에서 제안한 은닉채널 공격에 대한 내성을 지닌다.

1.4 논문의 구성

본 논문의 구성은 다음과 같다. 2 절에서는 논문의 이해를 위한 배경지식인 RSA와 CRYSTALS-Kyber 알고리즘과 X25519 키 교환 프로토콜과 Dual_EC_DRBG 동작과정을 서술한다. 3 절에서는 Young-Yung[6]의 SETUP 개념을 5가지 조건과 함께 서술하고 RSA의 SETUP과 1.2에서 언급한 Dual_EC_DRBG의 SETUP 예시를 통해 SETUP의 조건 5가지가 암호 시스템에 적용되는 과정을 서술한다. 4절에서는 SETUP을 만족하지 않는 백도어 예시를 통해 SETUP의 조건 5가지가 어떤 의미를 가지는지 논한다. 5절에서는 본 논문의 주요 결과 중 하나인 X25519에 기반한 CRYSTALS-Kyber의 SETUP 메커니즘을 서술하고 이 메커니즘이 SETUP의 5가지 조건을 만족함을 입증한다. 6절에서는 5절에서 제시한 SETUP이 적용된 CRYSTALS-Kyber를 IP카메라의 공개키 암호로 사용할 경우 발생할 수 있는 구체적인 공격 시나리오를 제시한다. 마지막으로 7절에서는 6절에서 언급한 SETUP에 의한 공격을 방어하기 위한 대응책 두 가지를 알고리즘 설계 관점과 보안 정책적 측면에서 제시한다.

Ⅱ. 배경지식

본 절에서는 논문에서 사용하는 알고리즘을 서술한다. 주요하게 사용하는 알고리즘은 RSA와 CRYSTALS-Kyber의 키 생성 알고리즘, X25519 키 교환 프로토콜이다.

2.1 *RSA* 알고리즘

오늘날 가장 널리 알려진 공개키 암호 RSA[14]는 큰 정수의 소인수분해가 어렵다는 사실에 기반한 암호체계이다. 메시지와 암호문 공간이 n 비트인 RSA의 키 생성 알고리즘은 다음과 같다.

2.1.1 *RSA* 키 생성 알고리즘

Algorithm 1. *RSA* : *key generation*	
Input	n
Output	$pk := (N, e),\ sk := (N, p, q, d)$
1	*Choose two random distinct* $\frac{n}{2}-$*bit primes* p and q
2	$N := pq$
3	*Choose* $e \in \{2,3,\ldots\Phi(N)-1\}$ *such that* $\gcd(e,\Phi(N)) = 1$
4	*Calculate* $d \in \{2,3,\ldots\Phi(N)-1\}$ *such that* $ed \equiv 1 (\mathrm{mod}\Phi(N))$
5	***return*** $pk := (N, e),\ sk := (N, p, q, d)$

2.1.2 *RSA* 암호화 알고리즘

Algorithm 1을 사용하여 개인키와 공개키 생성을 했다면 다음과 같이 암호문을 생성할 수 있다. RSA의 평문과 암호문 공간은 모두 $[0, N)$ 이다.

Algorithm 2. *RSA* : *Encryption*	
Input	$M \in [0, N),\ pk = (N, e)$
Output	C
1	$C := M^e \ (\mathrm{mod} N)$
2	***return*** C

2.1.3 *RSA* 복호화 알고리즘

암호문을 받은 수신자는 다음 알고리즘을 통해 평문을 획득할 수 있다. RSA 복호화 알고리즘의 구조는 암호화 알고리즘과 기본적으로 동일하지만 받은 암호문의 지수에 공개키 e가 아닌 개인키 d가 입력되는 차이가 있다.

Algorithm 3. *RSA* : *Decryption*	
Input	$C \in [0, N),\ sk = (N, p, q, d)$
Output	M
1	$M := C^d \pmod N$
2	***return*** M

2.2 CRYSTALS-Kyber 키 생성 알고리즘

CRYSTALS-Kyber[11]는 Module-LWE 문제의 난해함을 기반으로 두고 있는 양자내성암호이다. 본 논문에서 서술할 CRYSTALS-Kyber의 백도어는 Kyber의 키 생성 알고리즘에 쓰이는 난수 생성 알고리즘을 수정하여 사용자의 개인키를 추출하는 방식이다. 따라서 2.2절에서는 CRYSTALS-Kyber의 알고리즘 중 키 생성 알고리즘을 중점적으로 서술한다. CRYSTALS-Kyber의 키 생성 알고리즘을 서술하기 위해서는 다음 정의들이 필요하다.

2.2.1 다항식 환(ring)

알고리즘에 사용되는 다항식 환(ring) 2개를 다음과 같이 정의한다.

$n = 2^{n'-1},\ n = 256,\ n' = 9,\ q = 7681$일 때,
$R := Z[X]/(X^n+1),\ R_q := Z_q[X]/(X^n+1).$

2.2.2 균등분포(uniform distribution)

집합 S에서 원소 s를 균등분포에 따라 선택하는 것을 $s \xleftarrow{\$} S$로 표기한다. S가 집합이 아니라 분포일 경우 위 표기법은 s를 분포 S에 따라 선택하는 것으로 간주한다.

2.2.3 분포 변형 출력 함수

분포 변형 출력 함수(extendable output function)는 균등 분포를 따르는 비트열을 입력으로 하여 출력 값을 적절히 변형함으로써 원하는 분포를 얻을 수 있는 함수를 뜻한다. 여기에서는 이러한 함수를 Sam이라 표기한다. 임의의 분포 S에 대하여 Sam의 입력 x가 균등분포를 따를 때, Sam의 출력 $y(=Sam(x))$가 분포 S를 따르도록 설계할 수 있다. 이 경우 $y \sim S := Sam(x)$로 표기한다.

2.2.4 중심 변형 이항분포

분포 변형 함수 Sam을 적절히 선택함으로써 균등분포를 따르는 서로 독립인 n-비트 확률변수들로부터 중심이 변형된 이항분포(centered binomial distribution)를 얻을 수 있다. 즉,

$$(a_i, b_i) \xleftarrow{\$} \{0,1\}^n \times \{0,1\}^n,\ i = 1,2,\ \dots,\ \eta,$$

$$Sam(\{(a_1,b_1),\ \dots,\ (a_\eta,b_\eta)\}) = \sum_{i=1}^{\eta}(a_i - b_i).$$

위에서 Sam의 출력 값은 중심이 0, 범위가 $[-\eta, \eta]$이고, 좌우 대칭인 중심 변형 이항분포를 따르게 된다. 이 분포를 β_η로 표기한다. 또한, 각 성분이 다항식 환의 한 원소인 벡터 $\vec{v} \in R^k$에 대하여, $\vec{v} \xleftarrow{\$} \beta_\eta^k$는 $\vec{v}$를 구성하는 k개 다항식의 모든 계수들이 중심 변형 이항분포인 β_η를 따른다는 것을 의미한다.

2.2.5 Compress 와 Decompress 함수

$d < \lceil \log_2(q) \rceil$를 만족하는 자연수 d에 대하여 Compress와 Decompress 함수를 다음과 같이 정의한다.

(1) $x \in Z_q$에 대하여

$$Compress_q(x,d) := \left\lfloor \frac{2^d x}{q} \right\rceil \bmod 2^d.$$

(2) $x \in Z_{2^d}$에 대하여

$$Decompress_q(x,d) := \left\lfloor \frac{qx}{2^d} \right\rceil.$$

Compress 함수는 Z_q에서 Z_{2^d}로의 전사함수이고, Decompress 함수는 Z_{2^d}에서 Z_q로의 단사함수이다. Compress와 Decompress 함수를 쓰는 주요한 이유는 키, 암호문의 크기를 줄이기 위해서다.

2.2.6 CRYSTALS-Kyber 키 생성 알고리즘

CRYSTALS-Kyber는 키 생성 파라미터로 k, η, d_t를 사용한다. k는 키 생성에 필요한 행렬 $\vec{A}$와 벡터 $\vec{s}$와 $\vec{e}$의 원소의 개수를 결정짓는 파라미터이고, η는 다항식 환의 원소를 성분으로 갖는 벡터 $\vec{s}$와 $\vec{e}$의 계수의 범위를 결정짓는 파라미터이다. 파라미터 d_t는 벡터 $\vec{A}\vec{s}+\vec{e}$의 각 성분의 계수를 줄이는 Compress 함수를 결정짓는다.

주목할 사실은 CRYSTALS-Kyber는 256비트 난수 ρ와 σ로부터 공개키, 개인키를 생성한다는 점이다. ρ와 σ는 공개키 생성에 관여하고, 개인키 생성에는 σ만 관여한다. 이 사실은 후술할 CRYSTALS-Kyber의 백도어 형성에 중요한 요인으

로 작용한다. 다음은 CRYSTALS-Kyber의 키 생성 알고리즘이다.

Algorithm 4. CRYSTALS−Kyber : key generation	
Input	k, η, d_t
Output	$pk := (\vec{t}, \rho),\ sk := \vec{s}$
1	$\rho,\sigma \xleftarrow{\$} \{0,1\}^{256}$
2	$\vec{A} \sim R_q^{k\times k} := Sam(\rho)$
3	$(\vec{s},\vec{e}) \sim \beta_\eta^k \times \beta_\eta^k := Sam(\sigma)$
4	$\vec{t} := Compress_q(\vec{A}\vec{s}+\vec{e}, d_t)$
5	***return*** $pk := (\vec{t}, \rho),\ sk := \vec{s}$

2.3 X25519

X25519는 Curve25519를 사용하는 ECDH 키교환 메커니즘으로 2006년 Bernstein[12]에 의해 발표되었다. 일반적인 타원곡선 군(Group)의 상수배 연산은 타원곡선 위의 점 (x,y)를 모두 사용하지만 Curve25519는 타원곡선의 x좌표만 사용하여 상수배 연산을 한다.

Curve25519는 유한체 $GF(2^{255}-19)$를 사용하는 타원곡선 $y^2=x^3+486662x^2+x$ 위에서 정의한 군이다. X25519는 Curve25519의 Cofactor가 8인 부분군 H를 사용하며 H의 위수(Order)는 $2^{252}+27742317777372353535851937790883648493$이고 이 값은 소수로 알려져 있다. Curve25519는 타원곡선 위의 점과 정수의 상수배 연산 시 x좌표만 사용한다. 따라서 본 논문에서는 정수 n과 타원 곡선 위의 두 점 (x,y)와 (w,z)가 관계식 $n \cdot (x,y)=(w,z)$를 만족할 때 $n \cdot x = w$으로 표기한다. <표 1>는 X25519의 주요 특징을 정리한 것이다.

표 1. X25519의 주요 특징
Table 1. Features of X25519

Elliptic curve	Curve25519
Subgroup	$H=<(9,y)>$
Base point	$x=9$
Cofactor	8
Public key	256-bit(Last bit is always zero)
Private key	256-bit(First three bits and last bit are fixed to zero, and the second to last bit is fixed to one.)
Security level	Almost 128-bit

2.4 Dual EC DRBG 동작과정

Dual EC DRBG는 타원곡선 기반 PRNG 알고리즘으로 2004년 NIST 워크숍에서 최초로 공개되었다[7]. Dual EC DRBG의 동작과정을 서술하기 위하여 우리는 먼저 난수발생기의 기본적인 개념과 구성요소를 서술한다.

2.4.1 난수발생기

난수발생기란 암호시스템에서 암호화 시 사용되는 키, 논스(Nonce) 등을 요청할 때, 필요한 난수 비트열을 출력하는 알고리즘이다. 난수발생기는 TRNG(True Random Number Generator)와 PRNG(Pseudo Random Number Generator)로 구성되어 있다. TRNG는 비(非)결정론적 알고리즘으로 어떤 잡음원으로부터 생성된 난수를 후처리 하여 비트열을 출력한다. 반면, PRNG는 결정론적 알고리즘으로 TRNG의 출력 비트열을 입력으로 받아 최종 난수 비트열을 출력한다.

PRNG는 내부 상태 $s_i (i \geq 0)$, 내부 상태 갱신 함수 f, 난수 비트열 출력 함수 g로 구성되어 있다. f는 s_i를 s_{i+1}로 갱신시키고 g는 s_i로부터 난수 비트열 r_i를 출력한다. 암호시스템이 난수 비트열을 요청하면 PRNG는 요청한 난수 비트열을 출력할 때까지 f로 s_i를 갱신하면서 g로 난수 비트열을 순차적으로 출력한다. [그림 4]는 PRNG가 f, g를 이용하여 난수 비트열을 순차적으로 출력하는 과정을 나타낸 것이다.

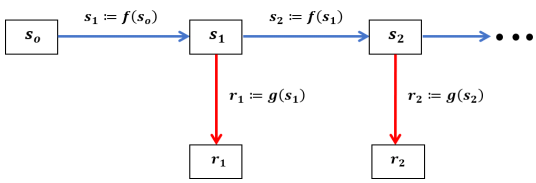


그림 4. PRNG 동작과정
Fig. 4. Generation process of pseudo-random bits

2.4.2 Dual EC DRBG 동작과정

Dual_EC_DRBG는 사용할 타원 곡선과 곡선 위의 두 점 P와 Q를 파라미터로 사용한다. 타원곡선은 P-256, P-384, P-521 중 하나를 선택한다. 타원곡선 위의 두 점 P와 Q는 기본적으로 상수로 주어져 있으며 필요에 따라 사용자가 직접 설정할 수 있다[15]. Dual_EC_DRBG의 내부 상태 갱신 함수 f와 난수 비

트열 출력 함수 g는 다음과 같다.

(1) $f(s_i) := \pi(s_i \bullet P) = s_{i+1}$.
(2) $g(s_i) := Disgard(\pi(s_i \bullet Q), 16) = r_i$.

여기서 π는 타원곡선 위의 점 (x, y)를 입력으로 받아 x를 출력하는 함수고 $Disgard$는 입력의 마지막 16 비트를 제외한 비트를 순서대로 출력하는 함수다. [그림 5]는 Dual_EC_DRBG의 동작 과정을 나타낸 것이다.

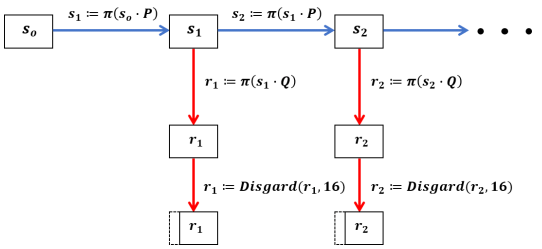


그림 5. Dual_EC_DRBG 동작과정
Fig. 5. Generation process of pseudo-random bits by Dual_EC_DRBG

Ⅲ. SETUP 개념과 예시

본 절에서는 SETUP의 정의와 예시를 서술한다. SETUP은 블랙박스 환경에 있는 암호 시스템 Γ의 내부 알고리즘을 변형하여 Γ'을 생성하는 것을 뜻한다. 여기서 블랙박스 환경이란 암호시스템을 사용하는 사용자가 암호시스템의 입출력만으로 내부 알고리즘의 정상 동작 여부를 확인할 수 있는 환경을 뜻한다.

암호 시스템 Γ에 백도어를 삽입하고자 하는 공격자는 백도어가 다음과 같은 조건을 만족하길 기대할 것이다.

(1) 사용자가 백도어가 삽입된 암호시스템 Γ'으로 공개키와 같은 공개된 파라미터를 출력하면 공격자는 출력을 이용하여 사용자의 개인키와 같은 비밀 파라미터를 얻을 수 있다.
(2) Γ'의 구체적인 알고리즘이 공개되더라도 백도어를 이용한 공격은 공격자만 가능하다.
(3) Γ'에 백도어가 삽입될 수 있다는 사실이 밝혀져도 공격자가 사용자의 비밀 파라미터를 추출한 증거는 남지 않는다.

SETUP은 공격자의 관점에서 위 조건을 만족시키기 위해 고안된 백도어 설계 메커니즘이다. 이를 달성하기 위하여 공격자는 공개키 암호의 일 방향성을 이용한다. 즉, 공격자는 공개키 암호 Θ로 공개키 μ와 개인키 ν를 생성한 뒤 μ를 암호시스템 Γ에 삽입하여 Γ'을 생성한다. Γ'은 μ로 사용자의 개인키와 같은 비밀 파라미터를 암호화하여 사용자의 공개 파라미터를 생성한다. 공격자는 사용자의 공개 파라미터를 ν로 복호화하여 사용자의 비밀 파라미터를 획득한다. 삽입한 백도어는 공격자의 개인키가 있어야 사용 가능하기 때문에 개인키 소유자인 공격자만 공격이 가능하다.

SETUP의 구체적인 정의는 다음과 같다.

3.1 SETUP의 정의

암호 시스템 Γ의 알고리즘 변형인 SETUP Γ'은 다음을 만족한다.

1. Γ'의 입력형태와 출력형태는 Γ와 같아야 한다.
2. Γ'은 공격자의 공개키를 이용한 공개키 암호화 함수 Θ를 포함하고 있어야 한다.
3. Γ'은 공격자의 개인키를 이용한 복호화 함수가 포함되지 않아야 한다.
4. Γ'의 출력엔 사용자의 개인키 정보가 있어야 하며, 공격자만 배타적으로 이를 추출해낼 수 있어야 한다.
5. Γ와 Γ'의 출력은 다항시간 내에 구별 불가능해야 한다.

조건 1번의 입력 형태와 출력 형태란 입출력 변수의 개수, 이름, 비트열 길이와 같이 가시적으로 확인할 수 있는 입력과 출력의 특성을 의미한다. 조건 2번과 3번은 조건 4번을 만족시키기 위한 선행 조건이다. 조건 2번과 3번으로 인해 Γ'의 알고리즘이 공개된다 하더라도 Θ로 생성한 개인키를 가지고 있는 공격자만 공격을 할 수 있다. 조건 5번에서 언급하는 출력의 구별 불가능성이란 Γ와 Γ'의 출력 비트열의 확률분포가 유사하여 빠른 시간 안에 두 분포를 구분할 수 없다는 의미이다.

3.2 *RSA* SETUP

SETUP의 대표적인 예시로 RSA의 변형 알고리즘을 들 수 있다[6]. RSA 키 생성 알고리즘인 Algorithm 1이 다음과 같이 수정되었다고 가정해보자.

Algorithm 5. RSA SETUP	
Input	n
Output	$pk := (N, e)$, $sk := (N, p, q, d)$
1	*Choose two random distinct* $\frac{n}{2}-$*bit primes* p and q
2	$N := pq$
3	$e := p^E \bmod(M)$
4	**if** $\gcd(e, \Phi(N)) \neq 1$: ***goto*** *line* 1
5	*Calculate* $d \in \{2,3,...\Phi(N)-1\}$ *such that* $ed \equiv 1 \pmod{\Phi(N)}$
6	***return*** $pk := (N, e)$, $sk := (N, p, q, d)$

여기서 Algorithm 5 내부에 있는 M과 E는 공격자가 사전에 생성한 RSA 공개키다. $M < N$을 만족하고 M의 비트 수는 N과 같은 것을 사용한다. 공격자는 M과 E를 이용하여 백도어를 형성하는데 그 과정은 Algorithm 5의 line 3에 나타나 있다. Algorithm 5의 line 3은 line 1에서 생성한 p를 공격자의 공개키로 암호화 하여 e를 생성한다. 따라서 사용자의 공개키는 공격자 입장에서 복호화 가능한 암호문으로 취급되고 공격자는 자신의 개인키 D로 e를 복호화하여 p를 획득할 수 있다. 위 백도어가 구체적으로 SETUP의 조건 5가지를 만족하는지 확인 해보면 다음과 같다.

1. Algorithm 1과 Algorithm 5의 입력은 모두 n이고 출력은 모두 $pk = (N, e)$, $sk = (N, p, q, d)$으로 같다.
2. Algorithm 5는 공격자의 공개키를 이용한 공개키 암호화 함수를 line 3에 포함하고 있다. 즉, Algorithm 5는 사용자의 개인키 p를 공격자의 RSA 공개키 E로 암호화 하여$(p^E \bmod(M))$ 사용자의 공개키 e를 생성한다.
3. Algorithm 5의 내부에는 공격자의 공개키 E에 대응되는 개인키 D가 존재 하지 않는다.
4. 공격자는 사용자의 개인키를 다음과 같이 추출할 수 있다.
   (1) 공격자는 사용자의 공개키 e를 관측한 뒤 연산 $e^D \pmod{M}$을 수행한다.
   (2) $e^D \pmod{M} = p^{ED} \pmod{M} = p$. 따라서 p를 획득한다.
   (3) p로 N을 나누어 q를 획득한다.
   (4) p, q로 $\Phi(N)$을 계산한다 $(\Phi(N) = (p-1)(q-1))$.
   (5) 공격자는 사용자의 개인키 d를 유클리드 알고리즘을 사용하여 계산한다.

   위 공격을 수행하기 위해서는 공격자가 소유한 RSA 개인키 D가 필요하다. 따라서 위 공격은 D를 소유한 공격자만 배타적으로 수행할 수 있다.
5. p는 무작위로 생성된 소수다. 따라서 $e = p^E \pmod{M}$는 범위 $[0, M)$ 내에서 균등분포에 가깝다. N과 M의 비트 수는 동일하므로 Algorithm 1과 Algorithm 5에서 생성한 e는 다항시간 내에 구분할 수 없다.

3.3 Dual_EC_DRBG의 SETUP

Dual_EC_DRBG는 서론에서 언급했듯이 SETUP이 가능한 알고리즘이다. SETUP이 가능한 이유는 타원곡선 위의 두 점 P와 Q가 독립적으로 생성되지 않을 여지가 있기 때문이다. 만약 공격자가 암호시스템 설계 과정에서 Dual_EC_DRBG를 PRNG로 사용하고 타원곡선 위의 두 점 P와 Q가 스칼라 값 α에 대하여 관계식 $P = \alpha \cdot Q$를 만족하도록 설정되었을 경우 공격자는 난수 비트열의 일부 관측하는 것만으로도 추후 생성될 난수 비트열을 전부 획득할 수 있다. 타원곡선 위의 두 점 P와 Q가 독립적으로 선택된 Dual_EC_DRBG를 Γ, 타원곡선 위의 두 점 P와 Q가 관계식 $P = \alpha \cdot Q$를 만족하도록 설정된 Dual_EC_DRBG를 Γ'라 놓으면 Γ'은 SETUP의 조건 5가지를 다음과 같이 만족한다.

1. Γ'은 Dual_EC_DRBG의 규격에 맞는 두 점을 선택했다. 따라서 출력형태는 Γ와 같다.
2. Γ'은 $\alpha \cdot Q$를 알고리즘에 포함하고 있다. 만약 공격자가 ECDH 키 생성 과정을 통해 개인키 α와 공개키 $\alpha \cdot Q$를 생성했다면 Γ'은 공격자의 공개키 $\alpha \cdot Q$를 이용한 암호화 함수를 포함한 것으로 여길 수 있다.
3. Γ'은 알고리즘 내부에 α를 포함하지 않는다. 따라서 Γ'은 공격자의 개인키를 이용한 복호화 함수를 포함하지 않는다.
4. 공격자는 사용자의 비밀 파라미터 s_{i+1}를 다음과 같이 추출 할 수 있다.
   (1) 공격자는 사용자의 난수 비트열의 일부를 관측하여 r_i를 획득한다.
   (2) 공격자는 2^{16}의 전수조사를 통해 최종 r_i 출력 이전에 버려진 16 비트를 복구하여 $\pi(s_i \cdot Q)$를 획득한다.

(3) 공격자는 타원곡선 위의 점 $s_i \cdot Q$의 x좌표 $\pi(s_i \cdot Q)$를 이용하여 $s_i \cdot Q$의 y좌표를 복구한다. 즉, 점 $s_i \cdot Q$을 획득한다.

(4) 공격자는 연산 $\alpha \cdot (s_i \cdot Q)$를 수행한다.

(5) $\alpha \cdot (s_i \cdot Q) = s_i \cdot (\alpha \cdot Q) = s_i \cdot P$. 따라서 공격자는 $s_i \cdot P$를 획득하고 연산 π를 이용하여 $s_{i+1}(=\pi(s_i \cdot P))$을 획득한다.

(6) 내부 상태 s_{i+1}를 획득한 공격자는 r_i 이후에 생성되는 모든 난수 비트열 $r_j\,(j \geq i+1)$을 획득할 수 있다.

위에서 서술한 공격을 수행하기 위해서는 공격자의 ECDH 개인키 α가 필요하다. 따라서 위 공격은 개인키 α를 소유한 공격자만 배타적으로 수행할 수 있다.

5. Γ와 Γ'의 출력이 구분 불가능하다는 사실은 DDH(Decisional Diffie–Hellman) 가정으로부터 도출된다[16]. DDH 가정은 위수가 q인 순환군 G의 생성원 g와 균등분포를 따르는 $a,b,c \in Z_q$ 대하여, (g^a, g^b, g^{ab})과 (g^a, g^b, g^c)는 확률적으로 구분 불가능하다는 것을 의미한다. 이때, (g^a, g^b, g^{ab})를 DDH triplet이라 부른다. 만일 순환 군을 타원곡선 군으로 두고 타원곡선 군의 생성원 C에 대하여 Q가 $Q = \beta \cdot C$로 표현된다면 $(\alpha \cdot C, Q, P)$는 DDH triplet을 이룬다. 따라서 두 점 P와 Q는 타원곡선 위에서 무작위로 뽑힌 두 점과 확률적으로 구분 불가능하다.

Dual_EC_DRBG는 백도어가 내부에 고의적으로 삽입되어도 이에 대한 확증을 찾아내는 것은 불가능함을 실증적으로 보여준 알고리즘이다. Dual_EC_DRBG는 두 점 P와 Q를 사용자 설정으로 변경할 수 있지만 [15]에서는 문서에서 제시한 두 점을 사용하는 것을 권고했다. 만약 [15]에서 권고한 두 점이 어떤 상수 α에 대하여 관계식 $P = \alpha \cdot Q$를 만족한다면 Dual_EC_DRBG의 알고리즘 설계자는 α를 이용하여 사용자의 난수 비트열을 예측할 수 있게 된다. 이러한 문제로 인하여 [15]에 제시된 Dual_EC_DRBG의 두 점 P와 Q는 그 출처에 대하여 많은 의문이 있었다. 그러나 NIST는 어떤 근거로 두 점을 선택했는지에 대한 공식적인 답을 내놓지 않았고 결과적으로 NIST가 고의적으로 백도어를 삽입했다는 확증은 찾을 수 없었다.

Ⅳ. SETUP을 만족하지 않는 백도어

SETUP의 조건 5가지는 3절의 도입부에 서술한 백도어가 갖춰야 할 조건 3가지를 충족하기 위하여 고안되었다. 암호시스템의 입출력이 표준과 다를 경우 사용자는 암호시스템에 백도어가 삽입되었음을 유추할 수 있기 때문에 SETUP의 조건 1번은 자명한 조건이다. 그러나 나머지 조건들의 당위성은 3절에서 서술한 RSA의 SETUP과 Dual_EC_DRBG의 SETUP을 통해서 파악하기는 어려울 수 있다. 따라서 본 절에서는 SETUP의 조건 2, 3, 4, 5번을 만족하지 않는 백도어 알고리즘 예시를 살펴보고 이 예시들을 통해 SETUP의 각 조건들의 당위성을 규명한다.

4.1 SETUP의 조건 2, 3, 4번을 만족하지 않는 백도어

다음 알고리즘은 AES를 이용하여 CRYSTALS-Kyber의 키 생성 알고리즘을 변형한 것이다. 여기서 $AES_Enc_{256}(m,k)$는 256 비트 평문 m을 키 k로 암호화 하는 연산을 의미한다. 반대로 256 비트 암호문 c를 k로 복호화하는 연산은 $AES_Dec_{256}(c,k)$로 표기한다.

4.1.1 CRYSTALS-Kyber Backdoor1

Algorithm 6. CRYSTALS−Kyber Backdoor1	
Input	k, η, d_t
Output	$pk := (\vec{t}, \rho)$, $sk := \vec{s}$
1	$\sigma \xleftarrow{\$} \{0,1\}^{256}$
2	$\rho := AES\_Enc_{256}(\sigma, K)$
3	$\vec{A} \sim R_q^{k \times k} := Sam(\rho)$
4	$(\vec{s}, \vec{e}) \sim \beta_\eta^k \times \beta_\eta^k := Sam(\sigma)$
5	$\vec{t} := Compress_q(\vec{A}\vec{s} + \vec{e}, d_t)$
6	***return*** $pk := (\vec{t}, \rho)$, $sk := \vec{s}$

Algorithm 6은 CRYSTALS-Kyber의 개인키가 난수 σ에서 파생되는 사실을 이용한 백도어이다. Algorithm 5는 사용자의 RSA 개인키 p를 공격자의 RSA 공개키 E와 M으로 암호화하여 사용자의 RSA 공개키 e를 생성했다. 이와 비슷하게

Algorithm 6은 σ를 공격자의 AES 대칭키 K로 암호화하여 ρ를 생성한다. 사용자가 Algorithm 6을 사용하여 공개키, 개인키 쌍을 생성할 경우 공격자는 사용자의 ρ값을 확인한 뒤 $AES\_Dec_{256}(\rho, K) = \sigma$ 연산을 통해 σ를 획득한다. σ를 획득한 공격자는 $Sam(\sigma)$ 연산을 사용하여 사용자의 개인키를 획득할 수 있다. Algorithm 6과 Algorithm 5의 백도어 생성 원리는 같다. 하지만 Algorithm 6은 공개키를 사용하여 σ를 암호화 하지 않고 대칭키(AES)를 사용하여 암호화 한다는 점에서 Algorithm 5와 차이가 있다. 따라서 Algorithm 6은 SETUP의 2번 조건을 만족하지 않는다.

Algorithm 6은 SETUP의 2번 조건뿐만 아니라 조건 3, 4번도 만족하지 않는다. 3번 조건의 경우 알고리즘에 직접적으로 공격자의 대칭키 K가 드러나 있기 때문에 만족하지 않는다. 4번 조건의 경우 알고리즘의 구조를 파악한 객체는 K를 이용하여 임의의 사용자의 개인키를 추출 할 수 있기 때문에 만족하지 않는다.

4.2 SETUP의 조건 5번을 만족하지 않는 백도어

다음 알고리즘은 고정된 소수를 사용하는 RSA의 키 생성 알고리즘이다.

4.2.1 Modified ***RSA*** [6]

Algorithm 7. Modified RSA

Input	n
Output	$pk := (N, e)$, $sk := (N, p, q, d)$
1	*Choose random* $\frac{n}{2}-$*bit prime* p and *fixed prime* q
2	$N := pq$
3	*Choose* $e \in \{2,3,...\Phi(N)-1\}$ *such that* $\gcd(e, \Phi(N)) = 1$
4	*Calculate* $d \in \{2,3,...\Phi(N)-1\}$ *such that* $ed \equiv 1 (\mathrm{mod} \Phi(N))$
5	***return*** $pk := (N, e)$, $sk := (N, p, q, d)$

Algorithm 7의 q는 고정 값이다. 따라서 사용자가 Algorithm 7을 사용하여 키 생성을 했을 경우 공격자는 사용자의 공개값 N을 확인한 뒤 소인수분해를 이용하여 사용자의 개인키를 추출할 수 있다. 그러나 Algorithm 7은 고정된 q를 사용하기 때문에 사용자는 키 생성을 수 차례 해보는 것으로 암호 알고리즘이 변형된 것을 파악 할 수 있다. 이러한 이유로 Algorithm 7은 SETUP의 조건 5번을 만족 하지 않는다.

Algorithm 7은 SETUP의 조건 5번을 만족하지 않지만 알고리즘 변형과정이 매우 단순하기 때문에 이를 통해 조건 5번의 당위성을 파악하긴 힘들다. 따라서 알고리즘 변형과정이 단순하지 않은 예시를 통해 5번 조건의 당위성을 설명하고자 한다.

4.2.2 CRYSTALS-Kyber Backdoor2

다음 알고리즘은 SETUP의 5번 조건을 만족하지 않는 CRYSTALS-Kyber의 키 생성 알고리즘을 변형한 백도어이다.

Algorithm 8. CRYSTALS−Kyber Backdoor2

Input	k, η, d_t
Output	$pk := (\vec{t}, \rho)$, $sk := \vec{s}$
1	$\rho \xleftarrow{\$} \{0,1\}^{256}$
2	$\vec{A} \sim R_q^{k \times k} := Sam(\rho)$
3	$Counter := \rho$
4	$c := 0$
5	$\vec{s}, \vec{e} := \vec{0} (\in R_q^k)$
6	**for** $i = 1$ ***to*** k :
7	**for** $j = 1$ ***to*** n :
8	$c = AES\_Enc_{256}(Counter, K) \bmod^{\pm} (2\eta+1)$
9	$\vec{s}[i] = \vec{s}[i] + cx^{j-1}$
10	$Counter = Counter + 1 (\mathrm{mod} 2^{256})$
11	**for** $i = 1$ ***to*** k :
12	**for** $j = 1$ ***to*** n :
13	$c = AES\_Enc_{256}(Counter, K) \bmod^{\pm} (2\eta+1)$
14	$\vec{e}[i] = \vec{e}[i] + cx^{j-1}$
15	$Counter = Counter + 1 (\mathrm{mod} 2^{256})$
16	$\vec{t} := Compress_q(\vec{A}\vec{s} + \vec{e}, d_t)$
17	***return*** $pk := (\vec{t}, \rho)$, $sk := \vec{s}$

여기서 $\bmod^{\pm}(2\eta+1)$ 연산은 비트열 B를 입력으로 받아 $[-\eta, \eta]$ 범위 내의 정수를 출력하는 함수로 정확한 정의는 다음과 같다.

정의. 비트열 B의 자연수 표현을 b라 하자. 이때 조건 $-\eta \le b' \le \eta$와 $b = b' + \lambda(2\eta+1)$를 동시에 만족하는 정수 쌍 λ와 b'는 유일하다. 이러한 정수 쌍 λ와 b'에 대하여 $B \bmod^{\pm}(2\eta+1)$를 다음과 같

이 정의한다.

$$B \bmod^{\pm}(2\eta+1) := b'.$$

즉, 연산 $\bmod^{\pm}(2\eta+1)$ 은 중심을 0으로 옮긴 모듈러 연산이다.

Algorithm 8은 난수 σ를 사용하지 않고 line 1에서 생성한 ρ와 공격자의 AES 대칭키 K를 이용하여 $\vec{s}$와 $\vec{e}$를 생성한다. $\vec{s}$와 $\vec{e}$는 k차 벡터로 n차 다항식을 성분으로 갖는다. Algorithm 8의 line 6부터 line 15는 총 $2kn$개의 다항식 계수를 for loop를 사용하여 정의하는 과정이다. 다항식의 계수들은 line 8과 line 13에서 확인할 수 있듯이 $c = AES_Enc_{256}(Counter, K) \bmod^{\pm}(2\eta+1)$로 정의한다. 이때, $Counter$ 값은 다항식의 계수마다 다르기 때문에 결과적으로 다항식의 계수들은 출력의 범위가 $[-\eta, \eta]$ 내의 정수인 균등분포를 따르게 된다.

공격자가 AES키 K를 통해 사용자의 개인키를 획득하는 과정은 다음과 같다.

(1) 사용자가 Algorithm 8을 사용하여 공개키, 개인키 쌍을 생성하면 공격자는 사용자의 공개키 ρ를 관측한다.
(2) 공격자는 (1)에서 획득한 ρ와 자신이 소유한 AES 대칭키 K를 이용하여 Algorithm 8의 line 3부터 line 10을 수행하여 사용자의 개인키 $\vec{s}$를 획득한다.

Algorithm 4로 생성한 $\vec{s}$를 구성하는 다항식의 계수들은 분포 β_η를 따른다. 그러나 Algorithm 8로 생성한 $\vec{s}$를 구성하는 다항식의 계수들은 균등분포를 따른다. 따라서 Algorithm 8로 키 생성을 반복적으로 할 경우, 사용자는 $\vec{s}$의 분포가 β_η^k를 따르지 않음을 확인할 수 있다. 따라서 Algorithm 8은 SETUP의 조건 5번을 만족하지 않는다.

V. CRYSTALS-Kyber SETUP

본 절에서는 논문의 주요 결과인 SETUP을 통해 생성한 CRYSTALS-Kyber의 키 생성 알고리즘을 서술한다. CRYSTALS-Kyber는 난수 ρ와 σ로부터 공개키, 개인키를 생성하기 때문에 Algorithm 5와 비슷한 방식으로 SETUP을 형성할 수 있다.

5.1 *RSA*를 이용한 CRYSTALS-Kyber SETUP

CRYSTALS-Kyber의 가장 간단한 SETUP은 Algorithm 6을 변형하여 얻을 수 있다. Algorithm 6은 난수 σ를 AES로 암호화하여 난수 ρ를 생성했다. 그러나 AES는 대칭키 암호이기 때문에 공격자의 배타적인 공격능력을 보장 시킬 수 없었고 Algorithm 6은 SETUP의 조건 2, 3, 4번을 만족하지 않았다. 따라서 AES 대신 공개키 암호 RSA를 사용하여 ρ를 생성하는 방식을 고려 해 볼 수 있다. 다음 알고리즘은 RSA를 이용한 CRYSTALS-Kyber의 SETUP이다.

5.1.1 CRYSTALS-Kyber SETUP Ver1

Algorithm 9. CRYSTALS−Kyber SETUP Ver1

Input	k, η, d_t
Output	$pk := (\vec{t}, \rho),\ sk := \vec{s}$
1	$\sigma \xleftarrow{\$} \{0,1\}^{256}$
2	$\sigma := \sigma (\bmod M)$
3	$\rho := \sigma^E (\bmod M)$
4	$\vec{A} \sim R_q^{k\times k} := Sam(\rho)$
5	$(\vec{s}, \vec{e}) \sim \beta_\eta^k \times \beta_\eta^k := Sam(\sigma)$
6	$\vec{t} := Compress_q(\vec{A}\vec{s}+\vec{e}, d_t)$
7	***return*** $pk := (\vec{t}, \rho),\ sk := \vec{s}$

여기서 Algorithm 9 내부에 있는 E와 M은 공격자가 사전에 생성한 RSA 공개키다. ρ가 256 비트 난수이기 때문에 M은 128비트 소수 2개의 곱으로 생성한다. Algorithm 9가 SETUP을 만족하는지 확인해 보면 다음과 같다.

1. Algorithm 4와 Algorithm 9의 입력은 k, η, d_t이고 출력은 $pk := (\vec{t}, \rho),\ sk := \vec{s}$으로 서로 같다.
2. Algorithm 9는 line 3에 공격자의 공개키 E와 M을 이용한 RSA 공개키 암호화 함수를 포함하고 있다.
3. Algorithm 9는 공격자의 RSA 개인키 D를 포함하지 않는다.
4. 공격자는 다음 과정을 통해 사용자의 개인키를 획득할 수 있다.
   (1) 공격자는 사용자의 공개키 ρ를 관측한 뒤

연산 $\rho^D(\mathrm{mod}\,M)$을 수행한다.

(2) $\rho^D(\mathrm{mod}\,M)=\sigma^{ED}(\mathrm{mod}\,M)=\sigma$. 따라서 공격자는 난수 σ를 획득한다.

(3) 공격자는 $Sam(\sigma)$ 연산을 수행한다. $Sam(\sigma)=(\vec{s},\vec{e})$. 따라서 공격자는 사용자의 개인키 $\vec{s}$를 획득한다.

5. σ는 256비트 난수다. 따라서 $\rho=\sigma^E(\mathrm{mod}\,M)$는 범위 $[0,M)$ 내에서 균등분포에 가깝다. M의 비트 수가 256이므로 Algorithm 9에서 생성한 ρ는 256비트 난수와 확률적으로 구분 불가능하고 ρ와 σ는 서로 독립인 256비트 난수로 취급할 수 있다. 따라서 Algorithm 4와 Algorithm 9의 출력은 다항시간 내에 구분 불가능하다.

Algorithm 9는 위와 같이 SETUP의 5가지 조건을 전부 만족한다. 그러나 RSA는 키 길이 대비 낮은 보안강도를 가지기 때문에 다음과 같은 이유로 공격자의 배타적 키 추출 능력이 완전히 보장받는다고 보기는 어렵다.

n비트를 쓰는 RSA의 보안강도는 약 $\frac{1.923\times\sqrt[3]{n\times\ln(2)}\times\sqrt[3]{[\ln(n\times\ln(2))]^2}-4.69}{\ln(2)}$으로 알려져 있다[17]. 이 식에 따른 $n=256$을 사용하는 RSA의 보안강도는 약 40비트다. 40비트는 일반 가정용 PC로도 한 시간 안에 수행 할 수 있는 연산능력이다. 즉, Algorithm 9를 파악하고 있는 객체는 공격자의 RSA 개인키 D를 모르더라도 한 시간 안에 40비트의 전수조사를 통해 난수 σ를 획득할 수 있고 이를 이용하여 사용자의 개인키 $\vec{s}$를 획득할 수 있다. 따라서 공격자의 배타적 키 추출 능력이 완벽히 보장받을 수 있다고 하기는 어렵다.

5.2 X25519를 이용한 CRYSTALS-Kyber SETUP

Algorithm 9는 백도어 형성 과정에서 공개키 암호 RSA를 사용했다. 그러나 RSA는 보안강도 1 비트 당 요구되는 암호문 공간의 비트 수가 큰 공개키 암호다. 따라서 256 비트 난수 ρ를 생성하기 위해선 낮은 보안강도를 가지는 RSA를 사용할 수 밖에 없었고 결과적으로 Algorithm 9의 구조를 파악한 객체는 공격자가 아니더라도 전수조사를 통해 사용자의 개인키를 획득할 수 있었다. 따라서 공격자의 입장에서 백도어의 성능을 높이기 위해서는 보안강도 1 비트 당 요구되는 암호문 공간의 비트 수가 작은 공개키 암호가 필요하다. 이러한 조건을 만족시키는 암호로는 대표적으로 타원곡선의 x좌표만 사용하여 상수 배 연산을 하는 X25519 키 교환 프로토콜을 들 수 있다. 따라서 Algorithm 9의 기본적인 구조를 가져가되 사용하는 공개키 암호를 RSA에서 X25519로 대체할 경우 5.1.1에서 언급한 공개키 암호의 낮은 보안강도 문제를 해결할 수 있다. 다음 알고리즘은 X25519를 이용한 CRYSTALS-Kyber의 SETUP이다.

5.2.1 CRYSTALS-Kyber SETUP Ver2

Algorithm 10. CRYSTALS−Kyber SETUP Ver2

Input	k, η, d_t
Output	$pk:=(\vec{t},\rho)$, $sk:=\vec{s}$
1	$x\xleftarrow{\$}\{0,1\}^{256}$
2	$x:=x\wedge\overbrace{000111...1110}^{256-bit}$
3	$x:=x\vee\overbrace{000000...0010}^{256-bit}$
4	$\rho:=x\bullet 9$
5	$\rho[255]\xleftarrow{\$}\{0,1\}$
6	$\sigma:=x\bullet(A)$
7	$\sigma[255]\xleftarrow{\$}\{0,1\}$
8	$\vec{A}\sim R_q^{k\times k}:=Sam(\rho)$
9	$(\vec{s},\vec{e})\sim\beta_\eta^k\times\beta_\eta^k:=Sam(\sigma)$
10	$\vec{t}:=Compress_q(\vec{A}\vec{s}+\vec{e},d_t)$
11	***return*** $pk:=(\vec{t},\rho)$, $sk:=\vec{s}$

여기서 Algorithm 10 내부에 있는 A는 공격자가 X25519 키 생성 프로토콜을 통해 생성한 공개키로 $A=g\bullet 9$이다. 위 알고리즘의 SETUP 형성 방식은 Dual_EC_DRBG의 SETUP 형성 방식과 유사하다. Dual_EC_DRBG를 배포한 공격자는 사용자가 생성한 난수 비트열 r_i를 확인하는 것으로 내부 상태 s_{i+1}을 추출하는 공격이 가능했다. 이와 비슷하게 Algorithm 10을 배포한 공격자는 사용자의 공개키 ρ를 확인하는 것으로 난수 σ를 추출하는 공격이 가능하다. 구체적으로 Algorithm 10의 line 4는 Dual_EC_DRBG의 난수 비트열 출력 함수에 대응되고 line 6은 Dual_EC_DRBG의 내부 상태 갱신함수에 대응된다.

이외에도 Algorithm 10의 각 line의 특징을 살펴보면 다음과 같다.

line 2, 3 : x의 첫 3비트, 마지막 2비트를 고정시키는 과정이다. 이 과정을 통해 x는 X25519의 개인키 공간에 속하게 된다.

line 5, 7 : Curve25519는 유한체 $F_{2^{255}-19}$를 사용하기 때문에 타원곡선 위의 점과 스칼라의 상수배 연산 결과 값의 마지막 비트는 항상 0이다. 따라서 마지막 비트의 무작위성을 만족시키기 위하여 line 5와 line 7을 도입한다. 결과적으로 난수 ρ와 σ는 임의의 256 비트 난수와 구분할 수 없게 된다.

Algorithm 10이 SETUP의 조건들을 만족하는지 확인해보면 다음과 같다.

1. Algorithm 4와 Algorithm 10의 입력은 k, η, d_t이고 출력은 $pk := (\vec{t}, \rho)$, $sk := \vec{s}$으로 서로 같다.
2. Algorithm 10의 line 6은 공격자의 공개키 A를 이용한 X25519 키 교환 과정으로 볼 수 있다.
3. Algorithm 10의 내부에는 공격자의 공개키 A에 대응되는 개인키 g가 없다.
4. 공격자는 사용자의 개인키를 다음과 같이 획득할 수 있다.
   (1) 공격자는 사용자의 사용자의 공개키 ρ를 관측한다.
   (2) ρ의 마지막 비트를 0으로 변환한다. 즉, 변환된 값은 $\rho = x \cdot 9$이다.
   (3) $\sigma_1 := g \cdot \rho$ 연산을 수행한다. 따라서 $\sigma_1 = g \cdot (x \cdot 9) = x \cdot (g \cdot 9) = x \cdot A$.
   (4) σ_1의 마지막 비트를 1로 바꿔 σ_2를 생성한다.
   (5) σ_1으로 연산 $Sam(\sigma_1) = (\vec{s_1}, \vec{e_1})$을 하여 사용자의 개인키 후보 $\vec{s}_1$을 획득한다.
   (6) σ_2으로 연산$Sam(\sigma_2) = (\vec{s_2}, \vec{e_2})$을 하여 사용자의 개인키 후보 $\vec{s}_2$를 획득한다.
   (7) $\vec{s}_1$과 $\vec{s}_2$ 중 하나는 사용자의 개인키가 된다.

   위 공격을 수행하기 위해선 공격자의 X25519 개인키 g값이 필요하다. 개인키 g값이 없을 경우 전수조사를 통해 g값을 찾아야 하는데 X25519는 보안강도가 약 128비트이기 때문에 전수 조사를 통해 g값을 찾는 것은 사실상 불가능하다. 따라서 Algorithm 10은 Algorithm 9와 다르게 공격자만 배타적으로 공격이 가능한 것으로 볼 수 있다.
5. $(g \cdot 9,\ x \cdot 9,\ gx \cdot 9)$는 X25519 개인키 공간에 속하는 g와 x에 대하여 DDH triplet을 이룬다. 또한, $g \cdot 9$와 $x \cdot 9$와 $gx \cdot 9$의 마지막 비트 값은 0이고 ρ와 σ는 마지막 비트를 무작위로 선택한다. DDH 가정에 따라서 ρ와 σ는 독립적으로 생성된 256비트 난수 두 개와 확률적으로 구분 불가능하고 Algorithm 4와 Algorithm 10의 출력은 다항시간 내에 구분할 수 없다.

Algorithm 10은 백도어 형성에 쓰이는 공개키 암호로 X25519를 선택함으로써 Algorithm 9의 낮은 보안강도 문제를 해결하였다. 그러나 X25519의 보안강도는 약 128비트로 양자컴퓨터가 상용화되었을 시 Algorithm 10을 파악하고 있는 객체는 Algorithm 9와 마찬가지로 공격자의 X25519 개인키 g를 모르더라도 전수조사를 통해 난수 σ를 획득할 수 있다. 따라서 Algorithm 10과 같은 형태의 SETUP은 양자내성암호와 기존암호가 공존하는 양자내성암호로의 전환기에 유효한 공격법으로 볼 수 있다.

5.3 난수발생기의 변형을 통한 CRYSTALS-Kyber의 SETUP

Algorithm 9, 10은 CRYSTALS-Kyber의 키 생성 알고리즘을 변형한 SETUP이다. 그러나 CRYSTALS-Kyber의 키 생성 알고리즘은 변형하지 않고 CRYSTALS-Kyber와 함께 사용하는 난수발생기를 변형하여 동일한 백도어를 제공하는 SETUP을

Algorithm 11. *Malicious random number generator*

Input	$(-)$
Output	(ρ, σ)
1	$x \xleftarrow{\$} \{0,1\}^{256}$
2	$x := x \wedge \overbrace{000111...1110}^{256-bit}$
3	$x := x \vee \overbrace{000000...0010}^{256-bit}$
4	$\rho := x \cdot 9$
5	$\rho[255] \xleftarrow{\$} \{0,1\}$
6	$\sigma := x \cdot (A)$
7	$\sigma[255] \xleftarrow{\$} \{0,1\}$
8	***return*** (ρ, σ)

구성할 수 있다. 예를 들어 Algorithm 4에서 사용하는 난수 ρ와 σ가 Algorithm 11의 출력일 경우 공격자는 5.2.1에 서술한 공격을 동일하게 수행할 수 있다.

Ⅵ. CRYSTALS-Kyber SETUP Ver2를 이용한 공격 시나리오

본 절에서는 5.2.1에서 서술한 CRYSTALS-Kyber SETUP Ver2가 암호시스템에 삽입되었을 경우 발생할 수 있는 공격 시나리오를 서술한다. SETUP에 의해 생성된 암호시스템은 보안제품에 탑재될 경우 심각한 위험을 초래할 수 있다. 본 논문에서는 특별히 스마트폰과 IP카메라의 암호통신에 CRYSTALS-Kyber SETUP Ver2 알고리즘을 사용할 경우 발생할 수 있는 공격 시나리오에 대해 다룬다. 논문에서 기준으로 삼은 IP카메라 모델은 [13]의 IP카메라 제품들이다.

6.1 Wifi 기반 IP카메라

Wifi 기반 IP카메라는 스마트폰 전용 어플리케이션이 존재한다. IP카메라 사용자는 어플리케이션에 접속하여 서버 회원가입을 하고 로그인을 하면 IP카메라를 서버에 등록할 수 있다. 서버에 등록된 IP카메라는 어플리케이션으로 서버 로그인을 한 스마트폰을 매개로 Wifi를 연결 할 수 있다. Wifi 기반 IP카메라의 구체적인 작동 메커니즘은 다음과 같다.

(1) IP카메라를 전원에 연결한다.
(2) 스마트폰을 Wifi에 연결한다.
(3) 스마트폰으로 IP카메라 전용 어플리케이션에 접속하여 서버에 로그인한다.
(4) 어플리케이션을 이용하여 IP카메라를 탐색한 후 서버에 IP카메라를 등록한다.
(5) 어플리케이션을 이용하여 IP카메라를 Wifi에 연결시킨다. 여기서 IP카메라에 연결된 Wifi는 스마트폰에 연결된 Wifi이다.
(6) 사용자는 스마트폰 어플리케이션에 접속하여 서버 로그인을 하면 등록되어 있는 IP카메라의 화면을 보거나 원격으로 조종할 수 있다.

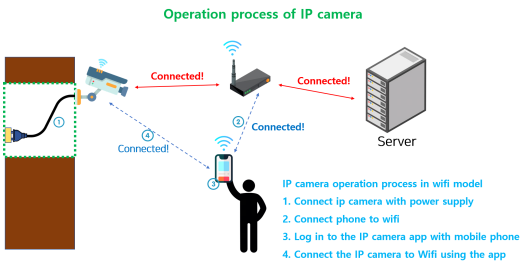


그림 6. IP카메라 동작 과정
Fig. 6. Operation process of IP camera

6.2 스마트폰과 IP카메라 사이의 암호 통신 과정

스마트폰 어플리케이션으로 서버에 로그인을 하면 서버에 등록되어 있는 IP카메라는 즉시 스마트폰과 대칭키 교환을 한다. 대칭키 교환이 끝나면 IP카메라는 영상을 공유 대칭키로 암호화한 뒤 스마트폰으로 송신한다. 공유 대칭키 교환은 공개키암호 PKE에 의해 이루어진다. IP카메라와 스마트폰 사이의 영상 송수신 과정은 다음과 같다.

6.2.1 PKE를 이용한 서버의 공개키, 개인키 생성

(1) 공개키, 개인키 생성 : 사용자가 스마트폰 어플리케이션으로 서버 회원가입을 하여 ID와 PW를 생성하는 순간 서버는 ID와 PW를 사용하여 ID에 대응되는 공개키 pk와 개인키 sk를 생성한다.
(2) 서버에는 사용자의 pk와 $H(PW)$만 저장된다 (H : 해시함수).

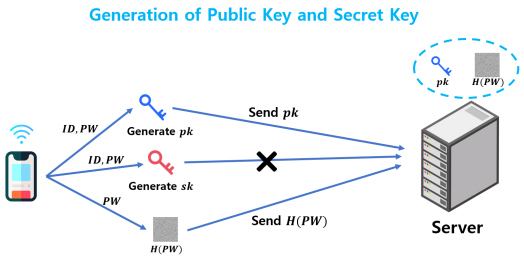


그림 7. 서버의 키 생성 과정
Fig. 7. Key generation by server

6.2.2 공유 대칭키 교환 및 암복호화 과정

(1) 스마트폰 어플리케이션으로 서버 로그인 시 서버는 로그인한 ID에 대응되는 공개키 pk를 IP카메라로 전송한다.
(2) IP카메라는 암호시스템 내부에 존재하는 난수 발생기로 스마트폰과의 통신에 필요한 대칭키 K를 생성한다.
(3) IP카메라는 영상 이미지를 K로 암호화하여 암호화된 영상 C를 생성한다.
(4) IP카메라는 pk로 K를 암호화하여 K_{pk}를 생성한다.

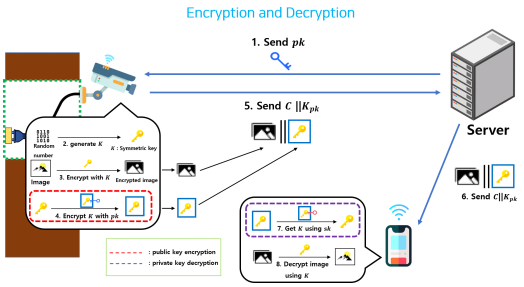


그림 8. Wifi 기반 IP카메라와 스마트폰의 암호통신 과정
Fig. 8. Encryption and decryption process between Wifi-based IP camera and smartphone

(5) IP카메라는 $C \| K_{pk}$를 서버로 전송한다.
(6) 서버는 IP카메라가 송신한 $C \| K_{pk}$를 스마트폰으로 전송한다.
(7) 스마트폰은 개인키 sk로 K_{pk}를 복호화하여 K를 획득한다.
(8) 스마트폰은 획득한 K로 C를 복호화 한 뒤 영상 이미지를 화면에 띄운다.

6.3 CRYSTALS-Kyber SETUP Ver2을 이용한 공격 시나리오

악의적인 공격자가 CRYSTALS-Kyber SETUP Ver2를 포함하고 있는 암호시스템 Γ'를 유포했다고 가정하자. 만약 스마트폰과 IP카메라 사이의 암호 통신이 Γ'을 통해 이루어질 경우 공격자는 서버에 등록된 모든 사용자의 ID에 대응되는 sk를 추출 할 수 있다. 사용자의 sk를 추출한 공격자는 영상 송신 과정에서 노출된 K_{pk}를 복호화하여 사용자와 IP카메라 사이의 공유 대칭키 K를 얻을 수 있다. 이후 공격자는 획득한 K로 암호화된 IP카메라 영상 C를 복호화하여 영상을 확인할 수 있다. CRYSTALS-Kyber SETUP Ver2를 이용한 IP카메라 공격 시나리오는 다음과 같다.

6.3.1 악의적인 공격자의 CRYSTALS-Kyber SETUP Ver2를 이용한 공개키,개인키 생성

6.2.1절에 따라 CRYSTALS-Kyber SETUP Ver 2가 암호시스템 내부의 공개키 암호로 사용된다면 개인키와 공개키 생성과정에서 ID와 PW가 사용되어야 한다. CRYSTALS-Kyber SETUP Ver 2의 개인키와 공개키는 256비트 난수 x로부터 생성된다. 따라서 가장 간단한 방식으로 $ID \| \| PW \| \| y$을 256 비트로 해싱한 값을 난수 x로 놓는 방식을 고려해볼 수 있다. 여기서 y는 솔트(salt) 값이고 256 비트 난수를 사용한다.

A가 공격자의 X25519 공개키일 때 IP카메라용 CRYSTALS-Kyber SETUP Ver2 알고리즘은 다음과 같다.

Algorithm 12. Malicious Server's key generator	
Input	k, η, d_t, ID, PW
Output	$pk := (\vec{t}, \rho)$, $sk := \vec{s}$
1	$y \xleftarrow{\$} \{0,1\}^{256}$
2	$x := H(ID \| \| PW \| \| y)$ $(H : Hash\,function)$
3	$x := x \wedge \overbrace{000111\ldots1110}^{256-bit}$
4	$x := x \vee \overbrace{000000\ldots0010}^{256-bit}$
5	$\rho := x \bullet 9$
6	$\rho[255] \xleftarrow{\$} \{0,1\}$
7	$\sigma := x \bullet (A)$
8	$\sigma[255] \xleftarrow{\$} \{0,1\}$
9	$\vec{A} \sim R_q^{k \times k} := Sam(\rho)$
10	$(\vec{s}, \vec{e}) \sim \beta_\eta^k \times \beta_\eta^k := Sam(\sigma)$
11	$\vec{t} := Compress_q(\vec{A}\vec{s} + \vec{e}, d_t)$
12	***return*** $pk := (\vec{t}, \rho)$, $sk := \vec{s}$

6.3.2 공격자의 공격과정

사용자가 스마트폰 어플리케이션으로 서버 회원가입을 한 순간 사용자의 개인키와 공개키는 Algorithm 12에 의해 생성된다. 사용자 U가 서버 회원가입을 하여 공개키 pk_U, 개인키 sk_U를 생성했다고 하자. 공격자는 다음 과정을 거쳐 U의 IP카메라 화면을 감시 할 수 있다.

(1) 공격자는 U의 IP카메라가 송신하는 $C \| \| K_{pk_U}$를 관측한다.
(2) 공격자는 $pk_U = (\vec{t_U}, \rho_U)$를 관측한 뒤 ρ_U를 추출한다.
(3) 공격자는 5.2.1에 서술한 CRYSTALS-Kyber SETUP Ver2의 공격 과정에 따라 sk_U를 추출한다.
(4) 공격자는 획득한 sk_U로 K_{pk_U}를 복호화하여 U의 스마트폰과 IP카메라 사이의 공유 대칭키

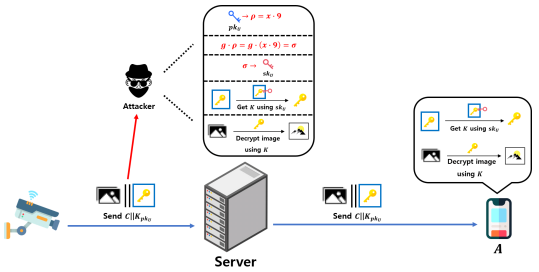


그림 9. 은닉채널 공격에 의한 영상 추출 과정
Fig. 9. Extacting private image by using subliminal channel

K를 획득한다.

(5) 공격자는 획득한 K로 암호화된 영상 C를 복호화하여 U의 IP카메라 영상 이미지를 획득한다.

Ⅶ. SETUP 기반 은닉채널 대응책

본 절에서는 SETUP 기반 은닉채널을 예방하기 위한 대응책에 대하여 논한다. 대응책은 암호 알고리즘 설계 관점과 보안 정책적 측면으로 나누어서 살펴볼 수 있다.

7.1 암호 알고리즘 설계 관점의 대응책

암호 시스템의 SETUP이 구조적으로 불가능하도록 설계하여 특정 환경에서 SETUP을 방지할 수 있는 대책이 존재한다. 여기에서 특정 환경이란 사용자가 난수발생기를 제외한 CRYSTALS-Kyber 알고리즘의 내부코드를 확인할 수 있는 환경을 뜻한다. 이러한 환경은 2절에서 가정한 블랙박스 환경과는 부합하지 않지만, 현장에서 암호 시스템의 적합성 평가는 난수발생기보다는 암호 알고리즘 평가에 더 중점을 두기 때문에 충분히 의미 있는 환경으로 볼 수 있다. 실제로 암호시스템 내의 암호 알고리즘을 부분적으로 암호모듈 검증 시 난수발생기로 외부 하드웨어 모듈을 사용하거나 외부 모듈의 난수를 사용해도 검증을 통과할 수 있다.

5.3절에서 서술한 바와 같이 원래의 CRYSTALS-Kyber 키 생성 알고리즘을 수정하지 않은 상태에서 함께 사용하는 난수발생기만을 변형하는 형태의 SETUP이 존재한다. 따라서 사용자가 CRYSTALS-Kyber의 알고리즘 코드를 확인할 수 있는 환경에 놓여 있다 할지라도 공격자는 여전히 사용자의 개인키를 추출하는 공격을 시도 할 수 있다. 하지만 CRYSTALS-Kyber의 키 생성 알고리즘 표준을 Algorithm 13과 같이 수정한다면, 5.3절에서 제시한

Algorithm 13. CRYSTALS−Kyber(2021)
: key generation

Input	$k,\ \eta,\ d_t$
Output	$pk := (\vec{t}, \rho),\ sk := \vec{s}$
1	$d \xleftarrow{\$} \{0,1\}^{256}$
2	$(\rho, \sigma) := G(d)$
3	$\vec{A} \sim R_q^{k\times k} := Sam(\rho)$
4	$(\vec{s}, \vec{e}) \sim \beta_\eta^k \times \beta_\eta^k := Sam(\sigma)$
5	$\vec{t} := Compress_q(\vec{A}\vec{s} + \vec{e}, d_t)$
6	***return*** $pk := (\vec{t}, \rho),\ sk := \vec{s}$

난수발생기를 변형하는 형태의 SETUP은 불가능하다.

Algorithm 4와 Algorithm 13의 차이점은 키 생성에 쓰이는 난수의 개수다. Algorithm 4에서는 두 개의 난수 ρ와 σ를 난수발생기에 요청하고, Algorithm 13에서는 한 개의 256비트 난수 d를 난수발생기에 요청한다. 5절에서 서술한 공격의 핵심은 난수 ρ와 σ를 독립적으로 생성하지 않고, ρ를 먼저 생성한 뒤 공격자의 공개키로 ρ를 암호화하여 σ를 생성하는 것이다. 그러나 Algorithm 13은 하나의 난수 d로부터 명시된 함수 G를 통해 ρ와 σ를 생성하기 때문에 사용자가 알고리즘 코드를 확인할 수 있는 환경에서는 5.3절에서 언급한 형태의 SETUP은 불가능하다. 실제로 CRYSTALS-Kyber의 키 생성 알고리즘은 초기 모델로 Algorithm 4를 사용했으나, NIST PQC 표준화 공모전에서 최종적으로 선정된 CRYSTALS-Kyber의 키 생성 알고리즘은 Algorithm 13의 형태를 갖추고 있다[18].

7.2 보안 정책적 측면의 대응책

SETUP을 방지하는 가장 확실하면서도 실용적인 방법은 보안 정책적 측면에서 암호모듈 검증제도를 운영하는 것이다. 우리나라의 KCMVP(Korea Cryptographic Module Validation Program), 일본의 JCMVP(Japan Cryptographic Module Validation Program), 미국의 CMVP(Cryptographic Module Validation Program) 등은 각국이 운영 중인 암호모듈 검증제도의 대표적인 예이다. 암호모듈 검증제도란 암호모듈의 안전성과 구현 적합성을 검증하는 제도로 제품에 탑재된 암호모듈 내부의 소프트웨어 및 하드웨어의 세부 동작과정이 암호모듈 검증기준을 따르는지 평가하는 제도다. 검증 대상이 되는 제품은 암호모듈에 포함된 암호 알고리즘의 코드와 하드웨어 입출

력의 흐름과 같은 모듈 내부의 동작과정을 모두 공개한 상태에서 적합성을 검증 받는다. 따라서 CRYSTALS-Kyber SETUP Ver2와 같이 독립된 난수 ρ와 σ를 쓰지 않고, 난수 x로부터 ρ와 σ를 생성하는 형태의 알고리즘 변형은 쉽게 탐지되어 검증을 통과할 수 없게 된다.

Ⅷ. 결 론

본 논문의 연구 결과는 크게 두 가지다. 첫 번째 연구결과로 Young과 Yung의 SETUP 개념을 양자내성 암호 CRYSTALS-Kyber의 초기 모델에 적용하여 알고리즘 내부에 은닉채널이 존재함을 보였다. 우리는 선행 연구인 Young-Yung의 SETUP 개념과 *RSA*를 이용한 SETUP을 분석하였고, SETUP에 의한 공격이 실증적으로 가능함을 Dual_EC_DRBG의 SETUP을 통해 살펴보았다. 또한, 알고리즘에 삽입된 백도어가 SETUP이 되기 위한 5가지 조건의 당위성을 다양한 알고리즘 예시를 제시하여 규명하였다. 그리고 CRYSTALS-Kyber의 키 생성 알고리즘에 적용 가능한 두 가지 형태의 SETUP을 제시하였다. 첫 번째 SETUP은 난수 σ를 공격자의 *RSA* 공개키로 암호화하여 난수 ρ를 생성하는 방식이다. 그러나 이 방식을 사용할 경우 사용하는 *RSA*의 보안강도가 낮기 때문에 공격자의 배타적 공격능력을 보장할 수 없다. 이러한 문제점을 해결하기 위하여 우리는 사용 비트 수 대비 보안강도가 높은 X25519 프로토콜을 이용한 SETUP을 구성하였다.

다음으로 우리는 SETUP에 의해 생성된 알고리즘이 암호모듈 내부에 존재할 경우 발생할 수 있는 실증적인 공격 시나리오와 이에 대한 대응책을 제안하였다. 구체적으로, 스마트폰과 IP카메라의 통신과정 중 발생할 수 있는 공격 시나리오를 제시했다. SETUP에 대한 대응책의 경우 크게 두 가지 관점에서 고려하였다. 보안 정책적 측면의 대응책으로 암호모듈 내부의 모든 동작과정을 검증하는 암호모듈 검증제도와 암호 알고리즘 설계 측면의 대응책을 제안하였다. 우리는 CRYSTALS-Kyber의 키 생성 알고리즘에 사용되는 난수를 하나만 사용할 경우 SETUP이 형성되지 않음을 입증하였다. 또한, NIST PQC 표준화 공모전에 최종 선정된 CRYSTALS-Kyber의 키 생성 알고리즘도 이러한 형태를 갖추고 있음을 확인하였다.

References

[1] G. J. Simmons, "Subliminal channels: Past and present," *Eur. Trans. Telecommun.*, vol. 5, no. 4, pp. 459-474, Jul./Aug. 1994. (https://doi.org/10.1002/ett.4460050408)

[2] G. J. Simmons, "The history of subliminal channels," *IEEE J. Sel. Areas in Commun.*, vol. 16, no. 4, pp. 452-462, 1998. (https://doi.org/10.1109/49.668969)

[3] G. J. Simmons, "The prisoners' problem and the subliminal channel," in *Advances in Cryptology,* pp. 51-67, Boston, MA, 1984. (https://doi.org/10.1007/978-1-4684-4730-9_5)

[4] G. J. Simmons, "The subliminal channel and digital signatures," in *Wkshp. Theory and Appl. Cryptographic Techniques*, pp. 364-378, Berlin, Heidelberg, 1985. (https://doi.org/10.1007/3-540-39757-4_25)

[5] G. J. Simmons, "A secure subliminal channel (?)," *Conf. Theory and Appl. Cryptographic Techniques,* pp. 33-41, Berlin, Heidelberg, 1986. (https://doi.org/10.1007/3-540-39799-X_5)

[6] A. Young and M. Yung, "The dark side of "black-box" cryptography or: Should we trust capstone?," *Annu. Int. Cryptology Conf.,* vol. 96, pp. 89-103, Berlin, Heidelberg, 1996. (https://doi.org/10.1007/3-540-68697-5_8)

[7] D. J. Bernstein, T. Lange, and R. Niederhagen, "Dual EC: A standardized back door," *The New Codebreakers,* pp. 256-281, Berlin, Heidelberg, 2016. (https://doi.org/10.1007/978-3-662-49301-4_17)

[8] https://en.wikipedia.org/wiki/Dual_EC_DRBG

[9] https://csrc.nist.gov/Projects/post-quantum-cryptography

[10] https://kpqc.or.kr

[11] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM," 2018 *IEEE EuroS&P,* pp. 353-367, 2018. (https://doi.org/10.1109/EuroSP.2018.00032)

[12] D. J. Bernstein, "Curve25519: New Diffie-

Hellman speed records," *Int. Wkshp. Public Key Cryptography,* pp. 207-228, Berlin, Heidelberg, 2006. (https://doi.org/10.1007/11745853_14)

[13] https://www.hikvision.com/en/products/IP-Products.

[14] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM,* vol. 21, no. 2, pp. 120-126, 1978. (https://doi.org/10.1145/359340.359342)

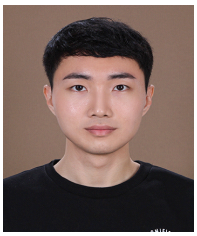
[15] E. B. Barker and J. M. Kelsey, "*Recommendation for random number generation using deterministic random bit generators (revised),*" Washington, DC, USA: US Department of Commerce, Technology Administration, National Institute of Standards and Technology, Computer Security Division, Information Technology Laboratory, 2007, from https://www.nist.gov/publications/recommendation-random-number-generation-using-deterministic-random-bit-generators-1

[16] https://en.wikipedia.org/wiki/Decisional_Diffie%E2%80%93Hellman_assumption.

[17] NIST, CSE, "*Implementation guidance for FIPS PUB 140-2 and the cryptographic module validation program,*" 2021, from https://csrc.nist.gov/csrc/media/projects/cryptographic-module-validation-program/documents/fips140-2/fips1402ig.pdf

[18] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "*CRYSTALS-Kyber algorithm specifications and supporting documentation,*" 2021, from https://pq-crystals.org/kyber/data/kyber-specification-round3-20210131.pdf

최 영 락 (Youngrak Choi)



2022년 2월 : 국민대학교 수학과 학사

2022년 3월~현재 : 국민대학교 일반대학원 금융정보보안학과 석사과정

<관심분야> 암호이론, 난수성 분석, 은닉채널

염 용 진 (Yongjin Yeom)

1992년 2월 : 서울대학교 수학과 학사

1994년 2월 : 서울대학교 수학과 석사

1999년 2월 : 서울대학교 수학과 박사

2000년 4월~2012년 2월 : ETRI 부설 연구소 책임연구원/팀장

2006년 12월~2007년 12월 : Columbia 대학교 방문연구원

2012년~현재 : 국민대학교 수학과 부교수

2013년~현재 : 국민대학교 BK21+ 미래 금융정보보안 인력양성사업단 교수

<관심분야> 암호구현 및 분석, 보안시스템 평가

강 주 성 (Ju-Sung Kang)

1989년 2월 : 고려대학교 수학과 학사

1991년 2월 : 고려대학교 수학과 석사

1996년 2월 : 고려대학교 수학과 박사

1997년~2004년 : 한국전자통신연구원 선임연구장/팀장

2001년~2002년, 2010년 : 벨기에 루벤대학 COSIC 방문 연구원

2004년~현재 : 국민대학교 수학과 교수

2013년~현재 : 국민대학교 BK21+ 미래 금융정보보안 인력양성사업단 교수

<관심분야> 암호이론, 정보보안 프로토콜, 안전성 분석 및 평가