KIPS Trans. Softw. and Data Eng.
Vol.9, No.2 pp.39~44 pISSN: 2287-5905

https://doi.org/10.3745/KTSDE.2020.9.2.39

Diffie-Hellman Based Asymmetric Key Exchange Method Using Collision of Exponential Subgroups

Jun Ho Song† · Sung-Soo Kim†† · Moon-Seog Jun†††

ABSTRACT

In this paper, we show a modified Diffie-Hellman key exchange protocol that can exchange keys by exposing only minimal information using pre-computable session key pairs. The discrete logarithm problem, which provides the safety of existing Diffie-Hellman and Diffie-Hellman based techniques, is modified to prevent exposure of primitive root. We prove the algorithm's operation by applying the actual value to the proposed scheme and compare the execution time and safety with the existing algorithm, shown that the security of the algorithm is improved more than the product of the time complexity of the two base algorithms while maintaining the computation amount at the time of key exchange. Based on the proposed algorithm, it is expected to provide a key exchange environment with improved security.

Keywords : Diffie-Hellman, Asymmetric Key Cryptography, Key Exchange

지수연산 부분군의 충돌을 이용한 Diffie-Hellman 기반의 비대칭 키 교환 방법

송 준 호† · 김 성 수†† · 전 문 석†††

요 약

본 논문에서는 사전연산이 가능한 세션 키 쌍을 이용하여, 최소의 정보만을 노출하여 키 교환이 가능한 변형된 Diffie-Hellman 키 교환 프로토콜을 보인다. 기존 Diffie-Hellman 및 Diffie-Hellman 기반 기법들의 보안성인 이산대수문제를 변형하여 생성원이 노출되지 않도록 설계함으로써 전송되는 암호문에 대한 공격으로부터 향상된 보안성을 가진다. 제안하는 기법에 실제 값을 적용하여 알고리즘의 동작을 증명하고 기반이 되는 기존 알고리즘과의 수행시간과 안전성을 비교 분석하여, 키 교환 시점 연산량을 유지하며 두 기반 알고리즘 시간복잡도의 곱 이상으로 알고리즘의 안전성이 향상되었음을 보였다. 제안하는 알고리즘을 기반으로 보안성이 향상된 키 교환 환경을 제공할 수 있을 것으로 기대된다.

키워드 : 디피-헬만, 비대칭 키 암호화, 키 교환 알고리즘

1. 서 론

본 논문에서는 보안성이 필요한 환경에서 통신 노드 간 대칭키를 수립하는데 널리 사용되는 Diffie-Hellman 키 교환 프로토콜을 기반으로 하는 개선 기법을 제안한다.

Diffie-Hellman을 기반으로 하는 기법들은 모두 이산대수 문제(Discrete Logarithm Problem)를 기반으로 안전성을 제공한다. 그러나 지수연산 자체의 시간 복잡도로 인하여 연산의 부하대비 안전성에 대해 문제가 제기되고 있으며, 2016년 기준으로 최소 112 bit 이상의 암호 강도를 가지는 알고리즘을 쓰도록 권고되고 있어 2048bit 이상의 공개키를 사용해야 한다[1]. 그러나 이 기준 또한 현재의 컴퓨팅 환경이 기준으로, 지속적인 하드웨어의 발달에 따라 기준이 되는 보안강도는 높아져야한다. 이에 안전한 암호화 강도를 유지하기 위해서는 요구되는 공개키의 크기가 급격히 증가한다.

이를 극복하기 위하여 작은 키 길이로 충분한 안정성을 제공할 수 있는 알고리즘으로 타원곡선 암호와 같은 응용 알고리즘들이 제안되어 사용되고 있다.

본 논문에서는 동일 키 길이에서 기존 이산대수 문제를 기반으로 하는 알고리즘보다 안전성이 향상된 방법을 제안한다. 안전성을 개선하는 과정에서 늘어난 연산량을 역원 쌍에 대한 연산을 미리 수행할 수 있도록 설계하여 키 교환 시 발생하는 절차의 부하를 분산시킨다. 이를 통해 이산대수 문제를 기반 하는 알고리즘 구조에서 비밀 값을 확장하여 안전성을 향상 시키며 메시지 교환시점에 효율성을 유지할 수 있다.

† 준 회 원 : 숭실대학교 컴퓨터학과 박사
†† 비 회 원 : 한국정보화진흥원 ICT융합본부 책임연구원
††† 종신회원 : 숭실대학교 컴퓨터학과 정교수
Manuscript Received : July 12, 2019
Accepted : July 26, 2019
* Corresponding Author : Jun Ho Song(jhsong@soongsil.ac.kr)

Table 1. Security Strength of Public Key Cryptographic Algorithms

Security Strength	Factorization problem	Discrete Logarithm Problem		Elliptic Curve
		Public Key	Private Key	
80 bit	1024 bit	1024 bit	160 bit	160 bit
112 bit	2048 bit	2048 bit	224 bit	224 bit
128 bit	3072 bit	3072 bit	256 bit	256 bit
192 bit	7680 bit	7680 bit	384 bit	384 bit
256 bit	15360 bit	15360 bit	512 bit	512 bit

2. 관련 연구

Diffie-Hellman 키 교환 알고리즘이 제안된 이래로 Diffie-Hellman을 기반으로 하는 다양한 방법들이 연구되었다. 안전성을 보장할 수 있는 이산대수 문제기반, 소인수분해 문제기반의 기본 알고리즘과 연구된 변형 기법을 살펴본다.

2.1 Diffie-Hellman 문제

Diffie-Hellman 알고리즘은 갈루아 체와 생성원을 이용하여 이산대수 문제를 구현하였다[2]. 기반을 두는 모든 가정은 유한순환군을 기반으로 하며, 그룹의 위수는 인수분해가 어려운 큰 소수를 약수로 가지도록 한다. 군을 생성하는 생성원이 필요하며, 군의 원소 중 하나가 주어졌을 때, 그 군의 원소를 생성하는 생성원의 승수를 찾는 것이 이산대수 문제이다.

$$\begin{aligned} &Share\,(g,p),\ p\,is\,prime\\ &A\rightarrow B:\ X\equiv g^{x}\bmod p\\ &B\rightarrow A:\ Y\equiv g^{y}\bmod p\\ &A:\ K\equiv Y^{x}\bmod p\\ &B:\ K\equiv X^{y}\bmod p\\ &SymmetricKey=g^{x\times y} \end{aligned} \tag{1}$$

2.2 사전 키 쌍 생성 기법

Nyang은 역원의 존재를 이용하여 지수연산 시간이 합의를 지연시키지 않도록 사전 키 쌍을 구성하는 방법을 제안하고 있다[3]. Diffie-Hellman 키 교환 알고리즘을 기반으로 하였으며, 전송하는 생성원의 승수에 대해 역원 쌍을 미리 계산하여 저장해 두도록 설계하였다. 사전에 생성된 역원 쌍은 '역변환 지수 문제'에 기반하여 안전하게 세션키에 연산된 값을 복호화하여 합의할 수 있도록 하였다[4].

$$\begin{aligned} &Share(g,p),\ p\,is\,prime\\ &A\rightarrow B:\ X\equiv g^{x}\bmod p\\ &B\rightarrow A:\ Y\equiv X^{y}\bmod p\\ &A:\ g^{y}\equiv Y^{\frac{1}{x}}\bmod p\\ &SymmetricKey=g^{y} \end{aligned} \tag{2}$$

2.3 Elgamal 비대칭 키 교환 기법

Elgamal은 Diffie-Hellman 키 교환 알고리즘으로 공개키 암호 방식을 제안하였다[5]. 송신 노드는 생성원에 개인키로 지수연산을 하여 공개키로 전달하고, 수신 노드는 받은 공개키를 기반으로 생성한 개인키로 지수연산을 한 뒤 전달할 메시지를 곱연산 함으로써 안전한 공개키를 만들어 전달한다. 최초 송신 노드는 전달받은 공개키 중 최초 지수연산되지 않은 암호문을 개인키로 지수연산하고, 메시지가 곱해진 암호문에 역연산을 하여 전달받은 메시지를 확인할 수 있다.

$$\begin{aligned} &PublicKey(p,g,X),\ p\,is\,prime\\ &A\rightarrow B:\ X\equiv g^{x}\bmod p\\ &B:\ C_1\equiv g^{k}\bmod p,\ C_2\equiv X^{k}m\bmod p\\ &B\rightarrow A:\ S\equiv C_1^{x}\bmod p,\ m\equiv C_2S^{-1}\bmod p\\ &SymmetricKey=m \end{aligned} \tag{3}$$

2.4 RSA 비대칭 키 교환 기법

RSA 알고리즘은 환을 이용하여 암호화와 복호화를, 곱셈군을 이용하여 키 생성을 하여 소인수분해 문제를 구현하였다[6]. 두 개의 소수를 고르고, 두 소수를 곱하여 암호화와 복호화를 위한 환으로 활용한다. 환에서 역원관계를 만들기 위해 승수의 성질을 이용하여 각 소수의 오일러 토션트 함수 값을 곱한 군에서 역원을 이루는 두 개의 값을 선택하여 공개값과 비밀값으로 선정한다. 두 값은 암호화와 복호화를 위한 환에서 선택한 하나의 생성원에 지수연산을 하였을 시 항등원을 생성한다. 공개값으로 지수연산하여 암호화한 값을 받아 비밀값으로 지수연산하여 복호화 하여 전달받은 메시지를 확인할 수 있다.

$$\begin{aligned} &Share(n,e)\\ &select\,(p,q)\,is\,prime\\ &n=p\times q,\ \phi(n)=(p-1)\times(q-1)\\ &e\times d\bmod\phi(n)\equiv 1\\ &A\rightarrow B:\ C\equiv M^{e}\bmod n\\ &B:\ C^{d}\bmod n\equiv M \end{aligned} \tag{4}$$

위의 알고리즘들의 안전성은 이산대수의 문제와 소인수분해의 문제 두 가지로 생각할 수 있으며, 이를 메시지와 전송되는 암호화된 메시지에 대한 수식으로 정리하였을 경우, 이산대수의 문제는 승수를 비밀로 하고, 소인수분해의 문제는 밑수를 비밀로 하는 순환공격과 동일한 수준의 복잡도 안전성을 제공한다.

$$\begin{aligned} &when\ discrete\ \log arithm\ problem\\ &y=g^{x}\bmod p,\ x\,is\,\mathrm{secret}\\ &when'\ factorization\ problem\\ &y=g^{x}\bmod p,\ g\,is\,\mathrm{secret} \end{aligned} \tag{5}$$

본 논문에서는 기존 비대칭 키 키 교환 알고리즘의 안전성을 향상시키기 위해 순환부분군의 역원을 이용하여 밑수와 승수를 모두 비밀로 하는 생성원이 노출되지 않는 비대칭 키 교환 방법을 제안한다.

3. 지수연산 부분군의 충돌을 이용한 비대칭 키 교환 방법

노드A와 노드B는 하나의 소수를 공유하여 노드B가 생성한 세션키에 대한 합의를 수행한다. 본 장에서 제안하는 키

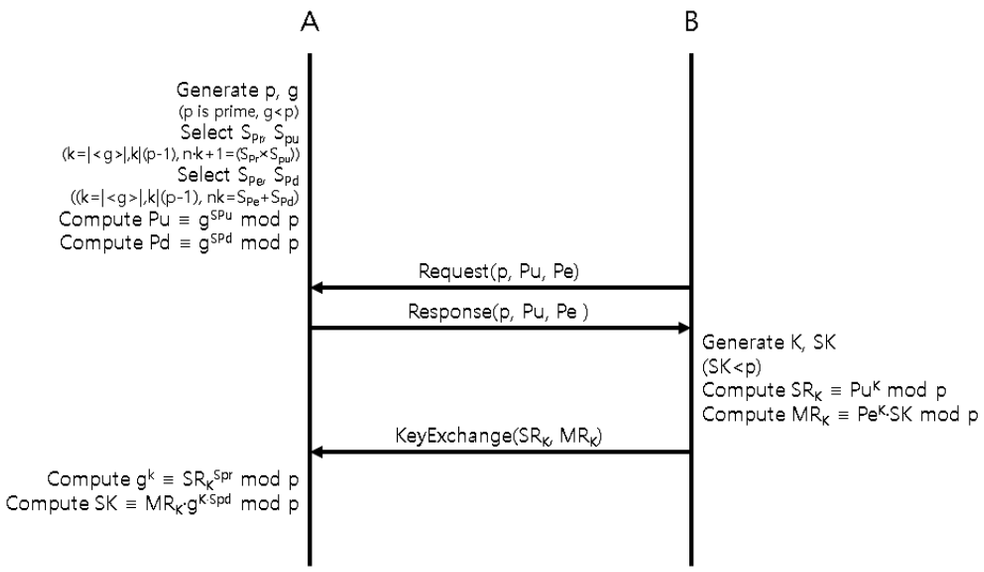

Fig. 1. Diffie-Hellman Based Asymmetric Key Exchange Method using Collision of Exponential Subgroups Procedure

교환 절차는 Fig. 1과 같다.

제안 기법에서 표기하는 파라미터의 설명은 Table 2와 같다.

노드A는 곱셈에 대해 역원를 가지는 잉여류 집합을 위한 소수 p를 정한다. 그리고 Z^*_p에 대한 순환부분군을 생성할 생성원으로 p보다 작은 생성원 g를 선택한다. 생성원 g의 멱승으로 생성되는 순환부분군의 위수는 (p-1)의 약수들로 도출된다.

노드A는 키 교환 절차가 이루어지기 전, 노드B의 암호키를 전달받기 위해 멱승의 역원 쌍을 사전에 생성한다. 지수의 곱연산에 대한 순환부분군의 첫 번째 값은 지수의 곱에 대한 항등원으로 각 순환부분군의 첫 번째 지수를 분리하여 생성원 g에 대한 지수연산 값과 그에 대한 역원을 도출할 수 있다. 순환부분군의 위수가 각 순환부분군의 크기이므로, (1,1) 쌍을 가지는 첫 번째 순환부분군을 제외한 순환부분군의 첫 번째 값은 순환부분군의 위수를 n번 반복한 값에 1을 더하여 구할 수 있다. n번째 반복된 순환부분군의 첫 번째 값의 지수에 대해 곱의 관계를 가지는 두개의 수를 정하여 개인 비밀키 S_{Pr}와 공개 비밀키 S_{Pu}로 지정한다.

$$k=|<g>|, k|(p-1), nk+1=S_{Pu}\times S_{Pr} \tag{6}$$

곱연산에 대한 역연산이 준비되면, 메시지나 세션키(SK)를 전달받기 위한 지수의 합연산에 대한 역원을 구한다. 지수의 합연산에 대한 역원 쌍의 합은 지수의 곱연산에 대한 순환부분군의 마지막 값의 지수이다. 지수의 곱연산의 역원 쌍과 마찬가지로, n번째 반복된 순환부분군의 마지막 값의 지수에 대해 합의 관계를 가지는 두개의 수를 정하여 암호 비밀키 S_{Pe}와 복호 비밀키 S_{Pd}로 정한다.

$$k=|<g>|, k|(p-1), nk=S_{Pe}+S_{Pd} \tag{7}$$

순환부분군을 생성한 생성원 g와 공개 비밀키 S_{Pu}를 지수연산하여 공개키 Pu를 생성하고, 암호 비밀키 S_{Pe}를 지수연산하여 공개키 Pe를 생성한다. A는 연산의 바탕이 되는 유한 순환군 Z^*_p를 식별하는 소수 p와 공개키 Pu, Pe를 안전한 통신을 위한 공개키로 저장한다.

$$\begin{aligned} &Pu\equiv g^{S_{Pu}}\bmod p\\ &Pe\equiv g^{S_{Pe}}\bmod p\\ &A\rightarrow B:\ PublicKey(p,Pu,Pe)\end{aligned} \tag{8}$$

노드B에서 키 교환에 대한 요청이 오면, 노드A는 준비된 공개키 (p, Pu, Pe)를 전달한다. 노드B는 안전한 전송을 위한 암호화키 K와 전달하고자 하는 메시지인 세션키 SK를 생성한다. SK는 Z^*_p에서 해석되어야 하므로 p보다 작아야 한다. 전달받은 p로 Z^*_p를 확인하고, Pu를 K로 지수연산하여 SR_K를 생성한다. SK는 Pe를 K로 지수연산한 값에 곱하여 안전하지 않은 경로로 전송되더라도 안전한 값 MR_K를 생성한다. 그리고 생성된 SR_K와 MR_K를 노드A로 전달한다.

Table 2. Exponential Subgroups Asymmetric Key Exchange Parameter

Parameter	Description
p	prime
g	number in 1~p-1
S_{Pr}	private multiplication private key of A
S_{Pu}	private multiplication public key of A
S_{Pe}	private addition public key of A
S_{Pd}	private addition private key of A
Pu	exchange multiplication public key
Pe	exchange addition public key
K	generated secretkey by B
SK	generated session key by B
SR_K	result of B's multiplication secret value
MR_K	result of B's addition secret value

$$SR_K \equiv Pu^K \bmod p$$
$$MR_K \equiv Pe^K \times SK \bmod p, (SK < p)$$
$$B \rightarrow A : KeyExchange(SR_K, MR_K) \quad (9)$$

노드B에서 전달받은 SR_K에 S_{Pe}지수연산의 곱 역원인 S_{Pr}을 지수연산하여 g^K값을 계산한다. 도출된 g^K에 지수연산의 합의 역원인 S_{Pd}을 지수연산하여 MR_K와 곱하면 g^K의 S_{Pe}승과 g^K의 S_{Pd}승이 역원 연산으로 상쇄되고 메시지인 세션키 SK가 도출된다.

$$g^K \equiv SR_K^{S_{Pr}} \bmod p$$
$$SK \equiv MR_K \times g^{K \times S_{Pu}} \bmod p \quad (10)$$

교환된 SK는 그 자체로 메시지로 쓰일 수 있으며, 대칭키 암호화에 사용하여 안전한 채널을 구성하는데 활용될 수 있다.

4. 파라미터 검증

본 장에서는 직접 확인 가능한 작은 소수를 이용하여 제안하는 알고리즘의 절차와 적용을 보인다.

Table 3. Example of Algorithm Application in Z^*_{487} - Application of Parameter Value

Parameter	Value
p	487
g	463
S_{Pr}	4
S_{Pu}	61
S_{Pe}	324
S_{Pd}	162
K	392
SK	324

소수 p는 487로 지정하여 유한순환군 Z^*_{487}에서 알고리즘을 적용한다. 순환부분군을 생성할 생성원 g로 463을 선택하였다. 생성원 463으로 생성되는 순환부분군의 위수는 243으로 지수연산의 곱셉의 역원을 가지는 쌍의 곱은 243n+1을 해로 가진다. 검증 과정에서는 1차 순환부분군의 값을 기준으로 개인 곱비밀키 S_{Pr}은 4, 교환 곱비밀키 S_{Pu}는 61로 선택한다.

$$k = 243$$
$$243n + 1 = 4 \times 61, (n = 1) \quad (11)$$

지수의 합연산에 대한 역원 쌍의 합은 g로 생성되는 순환부분군의 마지막 값 1의 지수이다. 검증 과정에서는 지수의 합연산의 역원 쌍에 대해 2차 순환부분군의 항등원의 지수를 기준으로 암호 비밀키 S_{Pe}는 324, 복호 비밀키 S_{Pd}는 162로 정한다. 예시는 지수의 곱연산에 대한 역원 쌍은 1차 순환부분군에서, 지수의 합연산에 대한 역원 쌍은 2차 순환부분군을 기준으로 보였으나 연산은 유한순환군 Z^*_{487}을 기반으로 하므로, n값에 따라 여러 키 쌍을 생성할 수 있다.

$$k = 243$$
$$243n = 324 + 162, (n = 2) \quad (12)$$

순환부분군을 생성한 생성원 463과 공개 비밀키 61을 지수연산하여 공개키 Pu를 생성하고, 암호 비밀키 324를 지수연산하여 공개키 Pe를 생성한다. A는 연산의 바탕이 되는 유한순환군 Z^*_p를 식별하는 소수 487과 공개키 Pu, Pe를 안전한 통신을 위한 공개키로 저장한다.

$$132 \equiv 463^{61} \bmod 487$$
$$254 \equiv 463^{324} \bmod 487$$
$$A \rightarrow B : PublicKey(487, 132, 254) \quad (13)$$

노드B에서 통신에 대한 요청이 오면, 노드A는 준비된 공개키 (487, 132, 254)를 전달한다. 노드B는 안전한 전송을 위한 암호화키 392와 전달하고자 하는 메시지, 세션키 324를 생성한다. 전달받은 소수 487로 Z^*_{487}상에서 132를 392로 지수연산하여 SR_K를 생성하고, 254를 392로 지수연산한 값에 세션키 324를 곱하여 MR_K를 생성한다. 그리고 생성된 SR_K값 127과 MR_K값 170을 노드A에게 전달한다.

$$127 \equiv 132^{392} \bmod 487$$
$$170 \equiv 254^{392} \times 324 \bmod 487$$
$$B \rightarrow A : KeyExchange(127, 170) \quad (14)$$

노드A는 노드B로부터 127과 170을 전달받는다. 암호화키를 풀기위해 노드B에서 전달받은 127에 지수연산의 곱 역원인 개인 곱비밀키 4를 지수연산하여 g^K를 도출한다. 도출된 g^K값 442에 지수연산의 합의 역원인 복호 합비밀키 162를 지수연산하여 170과 곱하면 합비밀키의 역원 쌍 $g^K \cdot S_{Pe}$와 $g^K \cdot S_{Pd}$가 상쇄되고 메시지인 세션키 SK값 324가 도출된다.

$$442 \equiv 127^4 \bmod 487$$
$$324 \equiv 170 \times 442^{162} \bmod 487 \quad (15)$$

제안한 절차에 따라 생성원인 g값이 안전하지 않은 경로상에 노출되지 않으며, 안전하지 않은 경로 상으로 전송되는 g의 연산 값은 모두 지수연산의 결과 값으로 최소 이산대수 문제를 기반으로 하는 안전성이 보장된다.

5. 효율성 및 보안성 평가

본 장에서는 제안한 알고리즘을 기존 알고리즘과 비교하여 연산의 효율성과 안전성에 대해서 평가한다.

제안 알고리즘의 효율성을 평가하기 위해서는 비교 대상 알고리즘과 동일한 절차를 보아야 한다. 그러나 제안 알고리즘의 사전 키 생성 절차는 타 알고리즘에 존재하지 않으며, 제안하는 알고리즘은 키 교환단계에 연산량을 줄이기 위하여 사전 생성이 가능하도록 설계하였다. 이에 키 교환을 위한 사전 절차를 미리 수행한 것으로 가정하여, 키 교환 요청이 발생한 시점에서 키 교환 시 소요되는 연산부하에 대해 분석하였다. 키 교환은 사전 검증한 데이터를 기반으로 하여 Diffie-

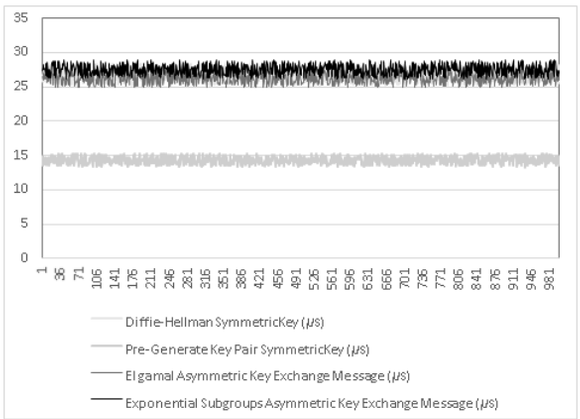

Fig. 2. Diffie-Hellman, Pre-Generation Key Pair, Elgamal and Proposed Algorithm Comparison of Time Required for Calculation of the Same Key

Hellman, Nyang의 사전 역원 생성, Elgamal과 제안 알고리즘을 대상으로 동일수준의 키 연산을 기준으로 각 1000000회 연산하여, 각 10000번의 수행시간마다 표본화하여 100건의 수행시간을 대푯값으로 정리하였다. 제안한 알고리즘은 기존 알고리즘과 수행시간의 증가율을 비교해 볼 경우 Diffie-Hellman 대비 102%, 사전 역원 생성 대비 193%, Elgamal 대비 107%의 수행시간을 보였다. 이는 제안 알고리즘의 효율성이 기존의 알고리즘과 큰 차이가 없음을 보여준다.

다음으로 제안 알고리즘의 공격에 대한 안전성을 평가하기에 앞서, 제안 알고리즘의 연산이 이루어지는 군이 안전한지에 대한 평가를 하였다. 이를 위해 지수연산의 역원이 존재하는 순환부분군을 생성하는 생성원이 군의 원소 중 얼마나 되는지 알아야한다. 만약 역원이 존재하는 순환부분군을 생성하는 생성원의 개수가 적을 경우, 제안 알고리즘은 설계에서 의도된 보안성의 향상을 보일 수 없다. 이에 작은 소수를 대상으로 생성원으로 만들어지는 부분군 중 순환부분군이 차지하는 비율과 개수를 분석하였다.

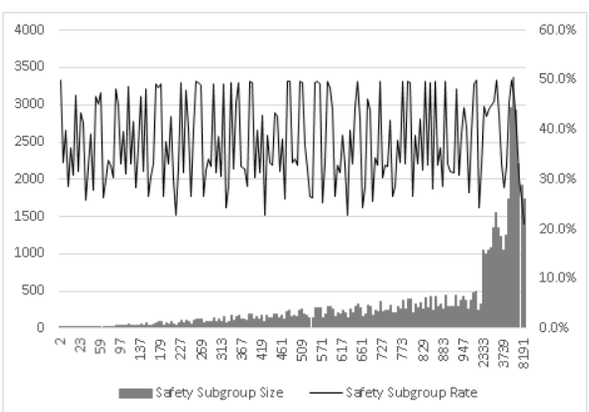

Fig. 3. The Size Ratio of the Circulating Subgroup

대상이 되는 소수는 1000보다 작은 소수 168개와 8191보다 작은 소수 17개를 선정하여 지수 곱의 위수를 구하여 정리하였다. 순환부분군을 이루는 생성원은 각 소수의 잉여류별로 최소 21.1%에서 최대 50%의 분포를 보였다. 평균 순환부분군은 37.4%의 비율로 발생하였으나, 실제 순환부분군의 개수는 순환군의 크기가 클수록 커지므로 순환부분군의 생성 비율이 21.1%로 비율이 가장 작은 소수 8171의 원소 중 순환부분군을 생성하는 생성원이 1728개로 가장 큰 것을 확인할 수 있다.

생성된 순환부분군의 크기를 확인하였으나, 아직 순환부분군의 크기가 실제 공격 시 어느 정도의 안전성을 향상시킬 수 있는지는 알기 어렵다. 이에 순환부분군의 개수를 확인한 8191보다 작거나 같은 소수 195개를 대상으로 공격 시 안전성을 분석하였다. 분석 과정에서 비교할 알고리즘으로 이산대수의 복잡성에 기반을 둔 알고리즘과 소인수분해의 복잡성에 기반을 둔 알고리즘을 가정하고, 각 알고리즘에서 전송되는 메시지를 탈취하여 탈취한 메시지 값을 생성하는 키를 찾아내는데 소요되는 시간을 측정하였다. 단, 소인부분해의 복잡성에 기반을 둔 알고리즘의 경우 동일한 시간 복잡도를 가지는 것으로 알려진 순환공격을 수행하였다.

Table 4. Comparison of Decryption Time for Attacks Against Discrete Algebra Problem, Factorization Problem, and Proposed Algorithm

mod	discrete logarithm problem (ms)	Cyclic Attack (Prime Factorization) (ms)	Proposal cryptography algorithm (ms)
2333	170	388	159765464
2339	168	378	159627868
2393	196	489	206852181
2399	193	405	182073233
2939	365	676	369809303
3119	384	766	362236953
3137	378	983	511901845
3733	628	1331	1003186218
3739	596	1237	1004980665
3793	678	1626	848701027
3797	675	1337	883505419
5939	2201	4804	5126812750
7193	3676	8023	21562858176
7331	3850	8301	10736682097
7333	3844	8146	22050046987
7393	3840	9803	23028630984
8191	5245	12045	35560360072

같은 컴퓨팅 환경에서 각 군의 생성원들로 생성원별 동일 메시지 값을 생성하는데 까지 걸린 시간을 측정하였으며, 공격하는 밑수 g는 각 소수의 절반보다 큰 임의의 완전부분군을 대상으로 하였다. 이산대수의 복잡성에 기반을 둔 알고리즘은 Equation (5)를 기준으로 밑수 g, 모듈러 p에서 지수연산 값 x를 1부터 증가시켜 공격하였고, 소인수분해의 복잡성에 기반을 둔 알고리즘은 Equation (4)의 암호화된 메시지 S에 대한 공격으로, x로 반복 지수연산하여 메시지를 찾아낼

때까지 걸린 연산시간을 측정하였다. 단, 제안하는 알고리즘에서는 공격 시 전송되는 메시지와 일치하는 g^x 후보군에 대한 검증을 수행하는 절차도 필요하나, 이 과정은 타 알고리즘에 대한 공격 시 존재하지 않는 과정으로 비교하기 어려우므로 단순히 메시지 값을 생성하는 밑수와 승수 쌍을 구하는데 걸리는 시간만을 측정하여 비교하였다. 따라서 제안 알고리즘에 대한 공격 시 공격자는 그림에 표기된 시간 외에 g^x 후보군에 대한 검증을 수행하여야 하나 본 비교절차에서는 제외하였다.

각 알고리즘에 공격 결과는 도식화하여 비교하기엔 현저한 차이가 있어, 1000 이상의 소수 17개의 공격 결과에 대한 비교 표로 작성하였다. 이산대수 문제와 소인수분해 문제에 대한 공격은 밑수의 크기에 비례하여 약 2배의 차이를 보였으나, 제안 알고리즘은 해당 모듈러 값에 해당하는 모든 순환부분군에서 일치하는 메시지 값을 구하여 단순히 확률적으로 비교하기 힘든 시간차이를 보였다.

위의 평가를 통해 제안 알고리즘은 활용단계에서 연산량에 큰 증가량 없이, 충분한 후보 키 그룹을 가지며, 향상된 시간 복잡도를 제공할 수 있음을 확인하였다.

6. 결 론

본 논문에서는 지수의 곱연산에 의한 순환부분군을 이용하여 생성원까지 비밀값으로 하는 Diffie-Hellman 키 교환 방법이 가능함을 보였다. 또한 제안하는 알고리즘의 키 교환 시 필요한 연산량과, 후보 키 그룹의 크기, 공격에 대한 연산 부하량을 측정하여 효율성과 안전성을 검증하였다. 기존의 알고리즘과 비교하여 같은 크기의 키 쌍에서 크게 향상된 안전성을 보이므로, 안전성이 향상된 키 교환 환경을 제공할 수 있을 것으로 기대된다.

References

[1] NIST, “Recommendation for Key Management”, NIST Special Publication 800-57 Part 1, Revision 4, 2016.

[2] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE Transactions on Information Theory*, Vol.22, Issue 6, pp.644-654, Nov. 1976.

[3] Dae Hun Nyang and Kyung Hee Lee, “Information Security: One Variant of Diffie-Hellman Key Exchange Protocol,” *The KIPS Transactions: Part C*, Vol.14, No.6, pp.9-17, Oct. 2010.

[4] A.-R. Sadeghi and M. Steiner, “Assumptions related to discrete logarithms: Why subtleties make a real difference,” *Advances in Cryptology - EUROCRYPT 2001 - International Conference on the Theory and Application of Cryptographic Techniques*, Proceedings, 2045: 244-261, 2001.

[5] T. Elgamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Transactions on Information Theory IEEE Trans. Inform. Theory Information Theory*, IEEE Transactions on. Vol.31, Issue 4, pp.469-472, Jul. 1985.

[6] R. L. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” *Communications of the ACM*, Vol.21, Issue 2, pp.120-126, Feb. 1978.

송 준 호

https://orcid.org/0000-0002-0853-4803
e-mail : jhsong@soongsil.ac.kr
2011년 서일대학교 인터넷정보(전문학사)
2011년 평생교육진흥원 컴퓨터공학(학사)
2014년 숭실대학교 컴퓨터학과(석사)
2020년 숭실대학교 컴퓨터학과(박사)
관심분야 : Information Security, Network Security, Cryptography

김 성 수

https://orcid.org/0000-0003-4025-733X
e-mail : cryptoauth@nia.or.kr
2016년 숭실대학교 컴퓨터학과(박사)
2014년 ~ 2016년 안양대학교 교양대학 겸임교수
2017년 ~ 현 재 한국정보화진흥원 ICT융합본부 책임연구원
관심분야 : Information Security, Authentication Theory, Cryptographic Algorithm

전 문 석

https://orcid.org/0000-0002-6932-2257
e-mail : mjun@ssu.ac.kr
1981년 숭실대학교 전자계산학과(학사)
1986년 University of Maryland Computer Science(석사)
1989년 University of Maryland Computer Science(박사)
1991년 ~ 현 재 숭실대학교 컴퓨터학과 정교수
관심분야 : Information Security, Network Security, Cryptography