ISBN 978-89-6211-645-8

ID 기반 암호 및 활용 기술

2010. 11

KiSTi www.kisti.re.kr
한국과학기술정보연구원
Korea Institute of Science and Technology Information

ISBN 978-89-6211-645-8

ID 기반 암호 및 활용 기술

2010. 11

기반기술개발실

박상배, 송중호, 최영리, 최동훈, 박동인

KiSTi www.kisti.re.kr 한국과학기술정보연구원 Korea Institute of Science and Technology Information

목차

표차례

그림차례

1. ID 기반 암호 개요

가. 공개키 암호화 방식

데이터를 암호화하는 방식은 일반적으로, 공개키 암호화 방식과 비밀키 암호화 방식으로 나눌 수 있다.

비밀키 암호화 방식은 데이터를 암호화하여 전송하고자 하는 두사용자가 동일한 비밀키를 공유하고 있고, 데이터를 나누어 가진 비밀키를 이용하여 암호화 및 복호화 한다.

비밀키 암호화 방식은 상호인증이 가능하고, 암호화 및 복호화 속도가 빠르고, 키 생성 알고리즘이 따로 필요하지 않는다는 장점이 있다.

그러나, 비밀키 암호화 방식의 단점은 통신하고자 하는 대상이 n개씩 늘어남에 따라서, 사용자가 관리해야 하는 비밀키가 n개씩 늘어난다는 단점이 있다. 또한, 비밀키를 변경하기 어려우며, 동일한 메시지를 다수의 사용자에게 전송하는 경우에 해당하는 비밀키로 암호화해야 한다는 단점이 있다.

이러한 단점을 극복하기 위해 공캐 암호화 방식이 탄생 되었다.

공개키 암호화 방식은 전송하는 데이터를 암호화 및 복호화에 쓰이는 키가 서로 다르다. 데이터를 전송하고자 하는 사용자는 공개키, 개인키를 생성하고 다른 사용자가 자신에게 데이터를 암호화하여 전송 할 수 있도록 공개키를 외부에 공개한다.

공개키 암호화 방식에서 송신자는 수신자가 이미 공개한 공개키를 가지고 데이터를 암호화 한다. 수신자는 암호화된 데이터를 수신 받으면 자신의 개인키로 복호화를 진행한다. 공개키로 암호화한 데이터는 해당하는 개인키로만 복호화가 가능하고, 사용자가 개인키로 암호화한 데이터는 공개키로만 복호화가 가능하다.

이러한 특징으로 인해서 데이터의 암호화뿐만 아니라, 서명 기능도 같이 수행 할 수 있다는 장점이 있다.

그러나, 공개키 암호화 기법에서 암호화 및 복호화에 대한 연산이 복잡하여 성능이 떨어진다는 단점을 지니고 있다.
이러한 문제점으로 인하여 현재, 암호화 통신에서는 데이터 암호화는 키를 비밀키 방식으로 진행하고, 암호화에 사용된 비밀키를 공개키로 암호화여 전송하는 형태로 두가지 방법을 혼용하여 사용하고 있다.

구분	공개키 암호	비밀키 암호
키의 관계	암호화키 ≠ 복호화키	암호화키 = 복호화키
암호화키	공개	비공개
복호화키	비공개	비공개
알고리즘	공개	공개
관리하는 키의 개수	2n	n(n-1)
암호화 속도	느림	빠름
인증 범위	누구나 인증	키 공유자

[표 1] 공개키 암호와 비밀키 암호 방식의 차이점

공개키 암호화 방식은 소인수 분해 문제를 이용한 RSA, Rabin 알고리즘과 이산대수 문제를 이용한 ElGamal, DSA, ECC 알고리즘등이 있다.

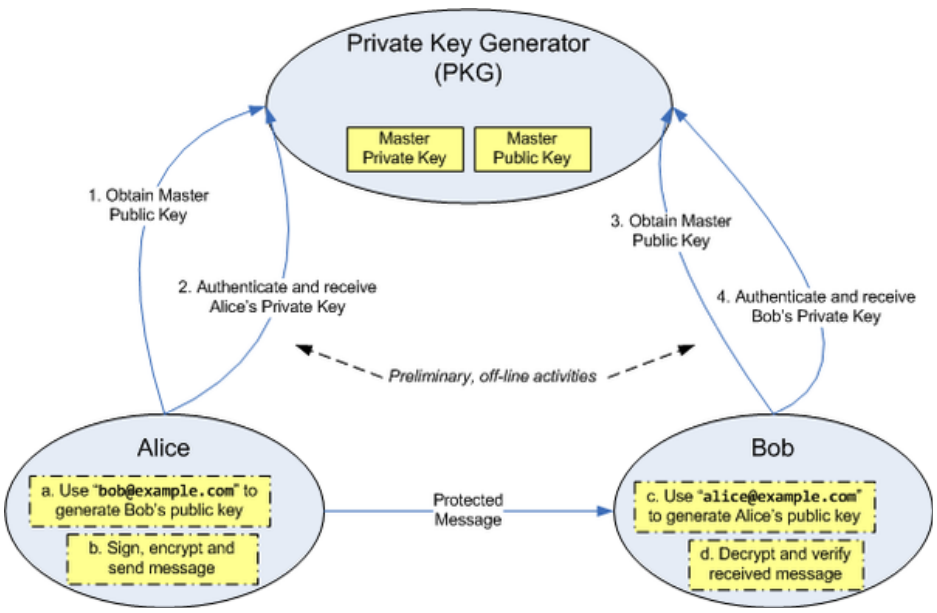
공개키 기반 암호화 알고리즘은 앞서 설명한 바와 같이 암복호와 연산속도가 느리고, 많은 컴퓨팅 연산이 필요하다는 단점이 있지만, 관리해야 하는 키의 개수가 적고, 데이터 암호화 및 서명을 동시에 수행 할 수 있다는 장점과 어느 누구든지 다른 상대와 별도의 키 교환 없이 암호화를 수행 할 수 있다는 장점으로 인해서 인터넷뱅킹, 전자문서등과 같은 많은 분야에서 활용되고 있다.

나. ID 기반 암호(ID Based Encryption)

ID 기반 암호(ID Based Encryption or Identity Based Encryption)는 식별이 가능한 ID를 가지고 인증 및 암호를 수행하는 일련의 기술을 말한다. ID 기반 암호 기술은 기존에 공개키를 활용하여 암호화를 수행하는 기법에서 키 생성과 관리 및 키 검증에 관한 처리 시간 및 효율이 떨어지는 단점을 보안하고자 1984년 Adi Shamir[1] 에 의해 제안되었다.

ID 기반 암호는 유일 무의한 ID가 공개키처럼 사용한다. ID는 텍스트 형태로써 이름, 도메인 이름 또는 물리적인 IP 주소, 이메일 주소와 같은 것이 사용 될 수 있다.

그리고, PKI(Public Key Infrastructure) 기반 공개키 시스템과 달리 공개키와 사용자 정보가 들어가 있는 인증서가 필요 없다. 따라서, 메시지를 송신하고자 하는 송신자는 수신자 또는 제3의 서버에 문의하지 않고 수신자의 ID를 알고 있으면, 수신자의 ID를 공캐키로 사용하여 데이터를 암호화하여 전송 할 수 있다.



[그림 1] ID 기반 암호화 개념도

[그림 1]은 ID 기반 암호 기법을 이용하여 Alice가 Bob에게 암호화된 메시지를 보내

는 과정에 대한 개념도이다.

각 과정을 설명하면 다음과 같다.

① Alice는 PKG의 마스터 공개키를 얻는다.

② Alice는 인증을 위해 자신의 ID를 PKG의 마스터 공개키로 암호화하여 전송한다.

③ PKG는 Alice가 전송한 정보를 마스터 개인키로 복호화하고 정상적인 사용자이면, Alice의 개인키를 생성하고 전송한다.

④ Alice는 Bob에게 메시지를 전송하기 위해 Bob의 이메일 주소를 이용하여 Bob의 공개키를 만들거나 또는 Bob의 이메일 주소를 공개키로 하여 메시지를 암호화하여 전송한다.(전송하는 메시지를 자신의 개인키로 서명한다.)

⑤ Bob은 자신의 공개키로 암호화된 메시지를 Alice로부터 받으면 이를 복호화하기 위해, PKG의 마스터 공개키를 얻는다.

⑥ Bob는 자신의 개인키를 받기 위해 PKG에게 마스터 공개키로 자신의 ID를 암호화하여 전송한다.

⑦ PKG는 Bob이 전송한 정보를 마스터 개인키로 복호화하고 정상적인 사용자이면, Bob의 개인키를 생성하고 전송한다.

⑧ Bob는 Alice가 자신의 공개키로 암호화하여 전송한 메시지를 자신의 개인키로 복호화하여 메시지 내용을 확인한다.(전송된 메시지가 Alice가 보낸 것이 맞는지를 확인하기 위해 Alice의 공개키로 서명을 검증한다.)

PKG(Private Key Generator)는 모든 사용자의 공개키, 개인키를 생성하고 ID를 발급받은 사용자 또는 객체를 인증하며, 해당 ID에 맞는 개인키를 관리하는 주체이다.

이와 같은 역할을 수행하는 PKG는 자신에게 속한 사용자들이 주고 받는 모든 데이

터를 복호화 할 수 있고 서명을 위조 및 변조 할 수 있기 때문에, 자신이 관리하는 ID를 발급 받은 모든 사용자(사람, 호스트)가 신뢰 할 수 있어야 하며, 어떠한 공격에라도 안전해야 한다.

1984년 Adi Shamir[1] 에 의해 제안된 방법에서 서명에 관한 사항은 언급하였으나, 데이터를 암호화 하는 방법은 언급하지 않았다. 이후, 암호화를 수행하는 다양한 방법이 언급 되었으나, 효율성이 떨어지는 것과 같은 문제점이 있었다.

이러한 문제를 해결하기 위해 Dan Boneh과 Matthew K. Franklin이 타원곡석(Elliptic curve)을 이용한 "Boneh/Franklin scheme"[2]를 발표 하였다.

"Boneh/Franklin scheme"의 기반 암호방식은 다음의 4가지 알고리즘으로 구성된다.

○ Setup(k) : 보안 파라미터 k를 입력하여 그 값에 대응하는 공개 파라미터 params와 마스터키 s를 출력하는 알고리즘

- [q, G_1, G_2, e]←G(k), P←G_1, s←Z^*_q, P_{pub}=sP를 생성한다.
- H_1:{0,1}*→G^*_1이고, H_2:G_2→{0,1}^{n}이다.
- params = <q,G_1, G_2, e, n, P, P_{pub}, H_1, H_2>, 마스터 키는 s이다.

○ KeyGen(ID, params, s) : 마스터 키 s와 수신자의 ID를 입력하여 그 ID에 대응하는 비밀키 d_{ID}를 출력하는 알고리즘

- 비밀키 d_{ID} = sQ_{ID}를 생성한다. 여기서 Q_{ID}=H_1(ID)($\in G^*_1$)이다.

○ Enc(params, ID, m) : 공개 파라미터 params와 수신자의 ID와 평문 m을 입력하여 그 평문에 대응하는 암호문 c를 출력하는 알고리즘

- Q_{ID}=H_1(ID)와 r←Z^*_q를 랜덤하게 선택하고 c=<rP, m+H_2(g^r_{ID})>를 계산한다.

  여기서, g_{ID}=e(Q_{ID},P_{pub})이다.

○ Dec(params, c=<U,V>, d_{ID}) : 비밀키 d_{ID}와 암호문 c를 입력하여 암호문에 대응하는 평문 m을 출력하는 알고리즘, $m=V+H_2(e(d_{ID},U))$이다.

이와 같은 방법으로 동작하는 "Boneh/Franklin scheme"이 제안됨에 따라서 ID 기반 암호화에서 암호화를 수행 할 수 있는 안전한 방법이 마련되어 현재 다양한 분야에서 ID 기반 암호화 방법을 활용하기 위한 연구가 진행 중에 있다.

2. ID 기반 암호 활용 기술

ID 기반 암호 기법은 타원곡선 암호화 방식을 사용한다. 이로 인해서 공개키/개인키를 생성하기 위해 사용되는 연산이 비교적 단순하고, 키의 사이즈가 RSA 기반에서 키 사이즈보다 작으면서 비슷한 수준의 안전성을 제공하기 때문에, 저용량 및 저전력의 환경에서도 활용이 가능하다는 장점이 있다.

그리고, 타원곡선 암호 시스템의 안전도는 키 길이의 증가에 따라 거의 지수함수적으로 증가하므로 준 지수함수적인 증가를 갖는 RSA, ElGamal, Diffie-Hellman 등과 같은 기존의 공개키 암호시스템에 비해 장기적으로 기술의 발전에 따른 키 길이의 증가 비율면에서도 대단한 장점을 가지고 있다 (예를 들어 RSA 암호 시스템이 512 비트 정도의 타원곡선 암호 시스템과 유사한 안전도를 제공하기 위해서는 대략 15,000비트 정도의 합성수를 사용하여야 한다.)[3]

타원곡선 암호는 이러한 장점들로 인해 스마트 카드나 무선통신 단말기등과 같이 메모리와 처리능력이 제한된 응용분야에서 특히 효율적으로 사용 될 수 있다. 이에 각종 국제 표준들에서 타원곡선 암호에 대한 표준화가 활발히 진행되고 있고 또 한 실제로 다양한 보안 응용들에서도 타원곡선 암호를 속속 지원하고 있다.

가. 무선 네트워크 분야 활용 기술

USN 네트워크의 자원 제약성 때문에 초기 센서 네트워크 보안은 대칭키를 기반으로 연구되어 오 가 현재 공개키 시스템을 적용하려는 시도가 활발히 진행되고 있다. 공개키 시스템으로 폭넓게 사용되는 암호 알고리즘인, 인수분해의 어려움에 기반한 RSA 알고리즘이나 ElGamal은 서버나 개인 PC와 같은 플랫폼의 자원을 요구하므로 USN 환경에서는 적용하기에 어려움이 있다.

RSA와 동일한 암호 안전성을 제공하면서도 작은 키를 사용하는 공개키 알고리즘을 살펴본다. 미국 매사추세츠의 WPI에서는 경량 센서 노드에 탑재 가능한 저전력 공개키 암호로 Rabin, Ntru을 구현하고 성능 특성을 분석하였다. 상기 공개키 암호 알고리듬은 RSA와 동일한 보안 안전성을 제공하면서도, ECC 연산의 다소 낮은 저전력 구현 특성을 보완할 수 있다.

Rabin 기법은 RSA의 특별한 하나의 형태로써, 인수분해 문제의 어려움에 기반한 공개키 암호 시스템으로 1979년 Rabin이 제안하였으며, NtruEncrypt는 SVP의 어려움에 기반한 공개키 암호 시스템으로 1996년 Hoffstein, Pipher와 Silverman이 제안하였다. [그림 2]는 Rabin 기법과 Ntru 성능 특성을 보여주고 있다.

	Rabin	Ntru(k=1)	Ntru(k=84)
Equivalent security	60bits	57bits	57bits
Area[eqv. gates] - combinational - storage elements	16,726 8,875 7,851	2,850 523 2,327	16,200 7,000 9,200
Delay(avg. # cycles)	1,440	29,225	433
Avg.power@500kHz - static(%) - dynamic(%) - peak power	148.18μW 117.5μW (79.3%) 30.68μW (20.7%) 169.8μW	19.13μW 15.10μW (78.9%) 4.03μW (21.1%) 20.22μW	118.7μW 103.06μW (86.8%) 15.64μW (13.2%) n/a
Energy - per bit encrypted	426.76nJ 833.5pJ (512bits)	1,118.15nJ 4,235.41pJ (264bits)	102.79nJ 389.4pJ (264bits)
Throughput	177.8 kbits/s	4.52 kbits/s	304.85 kbits/s

[그림 2] Rabin 기법과 Ntru 성능 특성

적절한 알고리즘과 구현 파라미터를 선정하고 저전력 기술을 적용할 경우, 력 소비를 20μW 이하로 암호 연산을 수행할 수 있다. 이는 배터리 전지를 사용하는 센서노드 환경에서 충분히 사용할 수 있음을 보여준다.

노스캐롤라이나 주립 대학에서 타원 곡선 암호 알고리즘을 TinyOS 상에서 구현하여 키를 안전하게 분배하고 있으며, 실제 사용을 위해 타원 곡선 기반 암호화 프로토콜인 ECIES와 키 분배 프로토콜인 ECDH, 서명 기법인 ECDSA 프로토콜을 구현 하였다[4],[5]. 해당 기술은 MICAz와 Telosb, TmoteSky에서 사용할 수 있으며, SECG에서 추천하는 128bit와 160bit, 192bit 타원곡선을 사용하고 있다. 성능을 보면 전자서명에 3.17초, 검증에 4.04초가 소요된다. 이는 비록 대칭키 암호 알고리듬 기반의 키 분배 기법보다는 다소 긴 시간이지만, 실제 응용에 사용되는 경우에도 충분히 실제 사용할 수 있는 시간이다[4].

XTR 공개키 암호 알고리듬은 Crypto2000에서 처음 소개되었고, 안전성의 관점에서 볼 때, 부분군에서의 DLP 문제에 기반을 두고 있다. 그러나 XTR은 부분군의 원소를 표현하고 계산하는 데 표준적인 방법을 사용하지 않으며, 전통적인 방법보다 대폭적인 통신상/계산상의 이점을 갖는다. 1024bit RSA의 안전성과 동일한 XTR은 ECC

에 기반을 둔 암호 시스템과 속도 및 안전성 면에서 비슷하다. XTR의 공개키는 ECC보다 두 배 정도 크지만 RSA와 ECC의 파라미터 초기화 시간보다 무시할 만큼 작은 시간이 소요된다. 따라서 XTR은 센서 네트워크 환경에서 RSA와 ECC의 좋은 대안이 될 수 있다[6]

나. 대용량 데이터보호 분야 활용

인터넷의 발전과 대용량 데이터의 급속한 유통으로 데이터 서비스 사용자가 급격히 증가하고 있다. 이러한 인터넷 서비스 데이터의 지속적인 증가로 인해 시스템의 구축비용과 확장성이 인터넷 서비스 업체의 경쟁력 확보에 중요한 요소가 되고 있다. 또한, 인터넷 서비스 데이터량의 지속적인 증가로 대량의 원시 데이터로부터 정보를 가공 처리하는 과정, 체계화된 정보의 저장 관리 및 유용한 정보를 추출하기 위한 분석 등에 분산 컴퓨팅 기술을 적용하는 움직임이 활발히 진행되고 있다.

구글, 야후 등 글로벌 인터넷 서비스 업체들은 인터넷 서비스 플랫폼의 중요성을 인식하고 자체 연구 개발을 수행, 저가 상용 노드를 기반으로 한 대규모 클러스터 기반의 분산컴퓨팅 플랫폼 기술을 개발 활용하고 있다.[7][8] 대용량 데이터 처리 및 저장 관리가 필요한 대표적인 어플리케이션으로는 인터넷 서비스 분야 외에 예를 들면, 비즈니스 인텔리전스 등 다른 응용 영역으로 확대하여 클라우드 서비스로 제공하려는 비즈니스 모델이 제시되고 있다. 이와 같이 분산 컴퓨팅 환경에서 다양한 데이터 서비스가 가능해지면서 대용량 데이터의 분산관리가 주요 이슈로 떠오르고 있다.

한편, 대용량 데이터의 다양한 이용 형태로부터 악의적인 공격자나 내부 사용자에 의한 보안 취약성 및 프라이버시 침해가 발생할 수 있다. 향후, 전자정부나 민간기업 등에서 취급하는 정보의 양이 점차 대규모화됨에 따라 개인정보 등의 프라이버시 관련 정보를 안전하게 저장해야 한다. 따라서 데이터의 안전한 저장·관리 문제 해결이 시급한 실정으로서 보안상의 위험성을 고려하여 데이터를 암호화한다. 그러나 암호화 방식은 키 분배 및 관리 운용상의 제약이 존재할 수 있다.

이러한 제약사항을 극복하기 위해 최근 프록시 재암호화(Proxy Re-encryption) 기법을 이용하여 암호화된 데이터를 복호할 때 복호권한을 위임하여 다른 사용자가 복호할 수 있도록 한다. 이와 같은 장점을 가지는 프록시 재암호화(Proxy Re-encryption)기법을 ID 기반 암호화 기법과 결합하여 이용하고 있다.

앞서 설명한 ID 기반 Proxy Re-encryption 기법의 알고리즘[9]은 다음과 같다.

Proxy Re-encryption 기능을 가지는 ID 기반 암호이다. ID기반 암호는 Setup, KeyGen, Enc, Dec의 4개의 알고리즘에 re-encryption key generating key 를 출력하는 RGKG(Re-encryption key Generating Key Generation), re-encryption key의 생성을 하는 RG(Re-encryption key Generation)와 암호문을 변환하는 Re-encrypt로부터 완성되는 7개의 알고리즘으로 구성된다.

또한, re-encryption key의 생성을 송신자가 생성하여 송신자가 암호문 복호자를 결정할 수가 있다.

○ Setup(k) : 보:안파라미터 k를 입력하여 공개파라미터 params와 master key s를 생성한다.

○ Extract(params, s, ID) : 공개 파라미터 params, master key s, 사용자 개별의 ID를 입력으로 ID에 대응하는 사용자 비밀키 dID를 생성한다, 이는 PKG나 관리자 등 신뢰 할 수 있는 기관이 행한다.

○ RGKG(params) : 공개 파라미터 params를 입력하여 re-encryption key와 암호문의 생성에 사용하는 re-encryption key generating key a를 생성한다.

○ RG(IDA, IDB, a) : 각사용자의 IDA, IDB, 랜덤 re-encryption key generating key a를 입력하여 re-encryption key $\pi_{A\rightarrow B}$를 계산한다.

○ Encrypt(params, ID, a, m) : 공개 파라미터 params, 사용자 개별의 ID, re-encryption key generating key a, 평문 m을 입력하여 암호문을 C를 생성한다. (즉, 송신자가 RGKG에서 출력된 re-encryption key generating key a로부터ID를 공개키로 평문 m을 암호화하여암호문 C를 생성)

○ Re-encrypt(C, $\pi_{A\rightarrow B}$) : Proxy가 암호문 C와 re-encryption key $\pi_{A\rightarrow B}$를 입력하여암호문 C를 C′로 변환한다. 여기서, Proxy는 암호문 변환만을 하는 기관이다.

○ Decrypt(params, dID, C′) : 공개 파라미터 params, 사용자의 비밀키 dID, 암호문 C′를 입력으로서 평문 m으로 복원한다.

여기서 re-encryption key generation key는 re-encryption key나 암호문을 작성하는데 필요한 키이다. Re-encryption key generation key는 re-encryption key의 생성과 암호문의 생성에는 필요하지만, 복호에는 필요가없다. RGKG는 종래의 Proxy Re-encryption에 없는 알고리즘이며 re-encryption key generation key를 생성한다. 또한, Encrypt는 암호문을 작성 할 경우에 ID 기반 암호와 달리 공개키 ID외 에 re-encryption key generation key를 필요로 한다.

다. P2P 분야 활용[10]

2005년 9월 NGN2005에서는 2003년을 기점으로 하여 인터넷에서 가장 많은 트래픽을 차지하는 서비스가 P2P 서비스라는 통계를 발표하였고, 같은 해 Network World지에서도 인터넷 트래픽의 60~89% 정도의 트래픽을 P2P 응용이 차지하고 있다고 하였다. 실제로 현재 수많은 사용자가 P2P서비스를 이용하고 있으며 다양한 P2P 응용이 급속 출현되고 있는 현실이다. 과거에는 P2P 응용하면 instant messaging, file sharing, distributed computing 정도를 떠올렸으나, 지금은 VoIP, video streaming, multicast, web caching, game, virtual office 등 무척이나 다양한 분야의 응용에 적용되고 있는 실정이다. 문제는 저작권 문제나 보안 문제와 같은 P2P 서비스의 부작용을 최소화하면서 안전하고 효율적인 방법으로 P2P 서비스를 사용하는 환경을 조성하는 것에 있다.

P2P는 자체적인 성격의 분산컴퓨팅 기술을 기반으로 하고 있으므로 분산 환경에서의 특수한 보안 취약성에 대한 분석의 선행이 필수적이다. P2P 네트워크 특유의 보안 문제점에 대한 이해가 요구되며 더불어 대응책을 고려함에 있어 기존의 보안 메커니즘의 적절한 변이를 숙고해야 할 필요성이 있다.

P2P의 대표적인 보안 취약성은 다음과 같다.

○ Whitewashing

P2P 네트워크는 기본적으로 peer의 자유로운 참여로 조성되는 네트워크 구조다. 이러한 특징은 각 peer의 익명성(anonymity)을 보장하는 장점이 되는 반면에 free-rider[10]인 peer들이 손쉽게 저렴한 비용으로 사용 실체에 대한 검증 없이 새로운 ID를 가지고 P2P 네트워크에 참여 할 수 있는 기회를 제공하는 문제점을 안고 있다. 일례로, 단순히 온라인상에서 주민등록번호로 인증을 하고 ID 발급을 하면, 본인의 동의 없이 도용/오용된 주민등록번호인 경우에 수많은 ID의 생성을 야기 시킬 뿐만 아니라 악의적 이용자의 자유로운 인터넷 출입을 허용하게 되는 것이 현 실정이다.

Whitewashing에 대한 구체적인 대안 모델로써, 먼저 strict model을 고려할

수 있다. 중앙신뢰기관 (central trusted authority) 또는 신뢰할 수 있는 로그인 서버(trusted login server)에 의해 강력한 할당 기법을 바탕으로 각 peer에게 고유한 ID를 부여한다. 그러나 본 기법은 중앙 서버의 개입을 통해ID가 부여되고, 관리된다는 구조적 문제점을 가지고 있다.

즉, 현실적으로는 PKI와 같은 인증을 위한 인프라가 있지만 실체 확인을 위하여 F2F 검증을 하여야 하므로 P2P와 같은 분산 환경에서는 불가능한 방법이며, P2P 서비스를 위하여 현재의 은행에서 F2F검증을 대행하여야 하며, PKI 기반의 인증서가 없는 사람은 P2P 서비스를 받을 수 없다는 문제점이 발생된다.

두번째 모델로써, 중앙신뢰기관을 통한 메커니즘의 사용이 불가능하다면, reputation model을 고려할 수 있다. 즉, P2P 네트워크에 참여하고자 하는 whitewashers를 포함한 모든 새로운 peer에게 합리적인 방법으로 penalty를 부여하는 방법을 고려할 수 있다. 그러나 해당 방어 기법은 reputation system과 같이 cost를 기반으로 peer의 악의성 여부를 판단하는 시스템상에서만 고려될 수 있다는 한계를 가지고 있다. 또한, peer간 상대적이고 동적인 cost 증감을 고려해야 하므로 P2P 네트워크로의 빈번한 참여 시도는 P2P 시스템 전체의 성능 저하를 가져올 수 있다.

○ ID Spoofing

본 공격은 악의적 peer가 바로 자기 자신의 식별정보(Identity, ID)를 속여 다른 대상 시스템을 공격하는 기법이다. 공격자는 획득한 식별정보를 target peer에 접근하는 ID로 사용하거나 두 peer 사이의 통신과정에서 responder peer인척 함으로써 비인가된 통신을 지속할 수 있다.

이와 같은 공격이 가능한 이유는 기본적으로 분산 환경을 기반으로 한 현재의 인터넷에서 ID 발급이 자유롭게 허용되어 이에 대한 추적이 어렵기 때문이다. 즉 아무런 규칙 없이 ID 발행을 허용한다면 ID가 도용되었을 때에 허가된 영역, 권한 외에서의 사용에 대한 능동적 대처가 어렵게 된다. 그러므로 ID 발급에 대한 최소한의 인증서버가 있어서 ID에 대한 명확성과 제한성을 부여하여 ID 검증 체계를 강화 해야만 한다.

이러한 공격은 ID 인증 기반의 접근 제어와 패킷 필터링 접근 제어, 취약점 서비스 사용의 제거, 암호화 프로토콜의 사용을 통해서 방어가 가능하다.

○ Reputation

상기 언급한 whitewashing 및 IP spoofing 등의 공격을 통해 획득된 신뢰성 없는 ID를 가지고 사이버 공간에서 문제점을 야기 시키는 공격자가 있다 하더라도 공격자 본인이 해당 ID를 사용하지 않았다고 거짓말을 하면 ID 사용의 실체를 검증하지 못하므로 ID의 소유자가 사이버공간에서 한 행위에 대하여 최종 부인을 할 수 있게 된다. 즉, 본 공격 유형에 대해서는 행위 부인에 대한 실제 검증이 어려운 문제점이 있다.

○ Man-in-the-Middle Attack

Peer 간에 상호인증을 통해 보안적으로 신뢰성 있는 통신 채널이 생성되더라도 중간자 공격(이하 MITM) 공격방법을 통해 통신 채널에서 전송되는 데이터들의 악의적 수집이 가능하다. 본 공격은 peer 간에 전송되는 데이터 스트림(data stream)의 불법 수정이나 거짓 데이터 스트림 생성을 통한 신분위장(masquerade), 재전송(replay), 메시지 불법수정(modification of message), 그리고 서비스 부인(denial of service) 등의 공격을 감행하는 특징이 있다.

즉, 악의적인 객체 또는 공격자가 중간에서 가로챈 중요한 데이터의 내용을 거짓된 내용으로 수정함으로써 인증된 객체가 수정된 내용을 믿게 되어 통신중인 peer에게 있어 강한 위협 요인에 노출되게 만드는 P2P 네트워크상의 대표적인 적극적 공격 유형이다.

MITM에 대비하기 위해서는 다음과 같은 요구사항을 만족해야만 한다.

- 상대 객체에 대한 안전한 인증 서비스 필요함
- 인증된 객체만이 패킷을 복호화 할 수 있는 암호화 키 서비스 필요함
- 교환되는 모든 메시지들이 중간에서 수정될 수 없어야 하며 수정된 경

우 이를 탐지해야 함

- 각 객체 시스템에 개인 firewall, anti-virus 프로그램과 같은 보안 대책이 수립되어야 함
- 등록된 데이터의 위치 정보에 오류가 있음을 인식했을 때 피해가 확산되지 않도록 빠른 대처 방법이 필요함

이러한 문제점은 ID 기반 암호화 기법을 활용하여 보안 할 수 있다. 아이디를 생성하는 데 있어 사용자를 구분 지을 수 있는 홍채, 지문 등의 바이오 정보를 추가하여 그것을 공개키로 이용할 수 있다. 사용자의 바이오 정보를 해시한 값이 '74123BC45'인 경우 아이디는 'ob@abc.com|D74123BC45'와 같이 사용될 수 있고 이것이 곧바로 Bob의 공개키가 된다. 이러한 방식은 전자여권과 같이 사용자의 신분을 확인할 수 있는 시스템 구축에 매우 유용할 수 있고, P2P와 같은 네트워크 상에서도 충분히 활용이 가능하다.

현재까지 개발된 아이디 기반 암호 기술의 한 가지 문제점은 공개키에 해당하는 개인키를 서버(KGC)가 생성하여 사용자에게 전달해야 한다는 것이다.

하지만, P2P 상에서 KGC와 같이 모두가 신뢰할 수 있는 서버를 두고 신뢰할 수 있는 통신 채널을 통해 온라인으로 개인키를 전달하는 것은 어려운 문제일 수 있다. 그러나 기존의 서버가 사용자를 직접적으로 인증해 주는 역할을 수행하는 반면 KGC는 사용자간의 암호 기술을 적용하는 데 있어 필요한 파라미터만 설정해 주기 때문에 사용자 간의 상호작용에 도움을 주는 역할을 수행하는 것으로 볼 수 있다. 이런 운용상의 문제점이 존재하기는 하지만 아이디 기반 암호 기술은 P2P 상에서 사용자의 아이디와 보안 기술을 결합할 수 있는 강력한 도구가 될 수 있으며, 이에 대한 연구가 진행 중에 있다.

[참고문헌]

1. Adi Shamir, “Identity-Based Cryptosystems and Signature Schemes. Advances in Cryptology”, Proceedings of CRYPTO 84, Lecture Notes in Computer Science, 1984

2. Dan Boneh, Matthew K. Franklin, “Identity-Based Encryption from the Weil Pairing Advances in Cryptology”, Proceedings of CRYPTO 2001

3. 임채훈, 이동훈, “타원곡선 암호 알고리즘”, TTA 저널 제80호, Page 98, TTA

4. 김호원, 이석준, 오경희, “센서네트워크 보안 기술 개발동향,” 정보보호학회지, 제18권 제2호, 2008. 4.

5. TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wiress Sensor Networks, Ver 1.0, http//discovery.csc.ncsu.edu/software/TinyECC, 2007. 2. 11.

6. 이신경, 이해동, 정교일, 최두호, “안전한 USN을 위한 정보보호기술 동향”, 전자통신동향분석 제23권 제4호, ETRI, 2008. 08

7. Jeff Dean, "Handling Large Datasets at Google: Current Systems and Future Directions," Data-Intensive Computing Symposium, 2008.

8. Raghu Rmakrishan, "Sherpa: Cloud Comptuin of the Third Kind," Data-Intensive Computing Symposium, 2008.

9. 鈴木秀輔, 齊藤泰一, "Proxy Re-encryption 機能をもつIDベース暗号" Symposium on Cryptography and Information Security, 2008.

10. 권혁찬, 문용혁, 구자범, 고선기, 나재훈, 정종수, “P2P 표준화 및 기술 동향”, 전자통신동향분석 제 22권 제1호, ETRI, 2007. 02

11. Michal Feldman et al., "ree-Riding and Whitewashing in Peer-to-Peer Systems,"the 3rd Annual Workshop on Economics and Information Security (WEIS2004), 2004.

12. http://www.gnutella.com/

13. http://www.kazaa.com/