NIST의 키 관리 표준

NIST Key Management Standard

신상욱(S.U. Shin) 정보보호기반연구팀 선임연구원
류희수(H.S. Ryu) 정보보호기반연구팀 선임연구원, 팀장
정교일(K.I. Jeong) 정보보호기반연구부 책임연구원, 부장

전자상거래와 같은 암호 응용 서비스들은 데이터 무결성과 기밀성과 같은 암호 서비스를 제공할 수 있는 잘 설정된 암호 기법을 필요로 한다. 그렇지만 그러한 기법들의 구현은 미리 공유 비밀키의 설정을 요구한다. 키 관리 시스템의 크기 또는 시스템을 사용하는 개체의 수가 증가함에 따라 키 설정을 위한 필요성은 키 관리 문제를 야기한다. 이를 해결하기 위해 미국의 NIST에서는 키 관리 표준을 개발하고 있다. 본 고에서는 현재 NIST에서 진행중인 키 관리 표준에 대한 동향과 고려되고 있는 기법들을 분석한다.

I. 서 론

전자상거래는 데이터 무결성과 기밀성과 같은 서비스를 제공할 수 있는 잘 설정된 암호 기법을 필요로 한다. Triple-DES, AES와 같은 대칭 암호 기법들은 이들 서비스 제공을 위한 좋은 선택들이다. 대칭키 기술을 사용한 시스템은 효율적이고 그들의 안전성 요구사항들이 잘 연구되어 있다. 더욱이 이들 기법들은 시스템들간의 상호 운영성을 용이하게 하기 위해 표준화 되어져 있거나 앞으로 표준화될 예정이다. 그렇지만 그러한 기법들의 구현은 미리 공유 비밀키의 설정을 요구한다. 키 관리 시스템의 크기 또는 시스템을 사용하는 개체의 수가 증가함에 따라 키 설정을 위한 필요성은 키 관리 문제를 야기한다. 이를 해결하기 위해 많은 기술들이 자발적인 산업 표준으로 정의되어 졌지만, 의문시되는 안전성 속성을 가진 많은 기술들이 보급되었고 몇몇 기술들은 정부의 필요성을 만족시키기에 충분한 안전성을 제공하지 않고 정부 기관들간의 상호 운영성을 촉진하지 않는다는 문제를 초래했다.

이러한 문제점들을 해결하기 위해 미국의 NIST (National Institute of Standards and Technology)에서는 키 관리 표준을 개발하고 있다[1]. 본 고에서는 현재 NIST에서 진행중인 키 관리 표준에 대한 동향과 고려되고 있는 기법들을 분석한다.

이를 위해 I장 서론에 이어 II장에서는 NIST의 키 관리 표준화 동향을 기술하고, III장에서는 키 관리 표준에서 고려되는 기술들을 고찰한다. 그리고 IV장에서 결론을 맺는다.

II. 키 관리 표준화 동향

키 관리는 키 설정과 키를 생성하고 설정하여 그 이후에 키를 다루기 위한 규칙과 프로토콜을 포함한다. 따라서 NIST는 키 관리 표준을 위해 두 가지 작업, 키 관리 기법 정의와 키 관리 가이드라인으로 나누어 이를 동시에 진행한다[2].

1. 키 관리 기법 정의

암호 키는 키 일치(key agreement) 또는 키 전

송(key transport) 기법을 사용하여 개체들간에 전자적으로 설정될 것이다. 키 일치 과정에서 키는 전송되지 않고 키 계산을 할 수 있게 하는 정보가 교환된다. 키 일치 기법은 비대칭(공개키) 기술을 사용한다. 키 전송 과정에서는 암호화된 키가 전송된다. 키 전송 기법은 대칭 또는 비대칭 기술들을 이용한다.

NIST는 키 관리 기법 정의 문서[3]를 현재 개발 중에 있다. FIPS(Federal Information Processing Standard) 또는 NIST Recommendation들은 납득할만한 키 설정 기법을 정의하기 위해 개발될 것이다. 표준 또는 권고안은 ANSI X9.42[4]로부터 D-H(Diffie-Hellman)과 MQV 키 일치(key agreement) 기법과 ANSI X9.44[5]로부터 RSA 키 일치와 키 전송, ANSI X9.63[6]으로부터 Elliptic Curve 키 일치와 키 전송기법을 선택할 것이다. 이 3가지 ANSI 문서는 현재 draft 단계에 있고 가까운 미래에 ANSI에 의해 표준으로 채택될 것으로 예상된다. NIST는 ANSI draft에 명시된 기법들 중의 부분 집합을 선택할 예정이며, key wrapping 기술(대칭 키가 또다른 대칭 키를 사용하여 암호화된다. 즉 AES (Advanced Encryption Standard) key는 AES key에 의해 암호화된다)에 대한 명세를 기법 정의 문서에 포함시킬 것이다. 문서의 계속적인 개정 과정에서 다른 기법들이 포함될 수도 있다. 기법의 선택은 기법에 의해 제공되는 안전성과 다양한 프로토콜(1-pass, 2-pass, 3-pass 프로토콜)에서 유용성에 기반할 것이다. 기법 정의 문서에 대한 개발 일정은 다음과 같다.

※ 기법 정의 문서에 대한 개발 일정

June 2001	· RSA 기법의 개요에 대한 문서 · 가능하다면 D-H 기법을 포함 · public comment 받음
July 2001	· 키 wrapping 기법 제안 · public comment 받음
October 2001	· NIST의 선택을 논의하고 기법 정의 문서의 다음 단계를 계획하기 위해 워크샵이 개최
Thereafter	· 선택된 RSA 기법, 키 wrapping 기법, 가능하다면 D-H 기법에 대한 초기 NIST 권고안 개발 · public comment 받음 · ANSI X9.42에 정의된 D-H와 MQV 기법에 대해 작업 계속 진행 · D-H와 MQV에 관한 draft 문서 개발 · 개발 진행 과정과 마지막 단계를 논의하기 위해 두번째 워크샵(Nov. 1-2, 2001) 개최 · D-H와 MQV 기법을 포함하기 위해 NIST 권고안 갱신 · public comment 받음 · ANSI X9.63에 정의된 Elliptic Curve 기법에 관해 연구 계속 진행 · 현재까지의 진행 상황을 논의하기 위해 세번째 워크샵 개최 · Elliptic Curve 기법을 포함하기 위해 NIST 권고안을 수정하거나 모든 권고된 기법을 포함하는 FIPS draft 개발 · public comment 받음

2. 키 관리 가이드라인

암호 키를 사용하는 과정의 안전성과 신뢰성은 그 키에 행해진 보호에 직접적으로 의존한다. 이를 위해 NIST는 키 관리 가이드라인 문서[7]를 개발 중에 있다. 이는 NIST 권고안 또는 NIST Special Publication이 암호 키의 라이프 사이클 관리(키의 생성, 설정, 저장, cryptoperiod, 복구, 파괴)에 대한 정부의 가이드라인을 제공하기 위해 개발될 것이다. 비밀키와 공개키 쌍 중에서 개인키는 노출로부터 보호되어야 한다. 모든 키들은 수정(대치와 허가되지 않은 삭제)되는 것으로부터 보호되어야 한다. 키를 수신하거나 설정하는 개체는 누가 키를 전송했는지, 누구와 키를 설정하는지, 키의 목적은 무엇인지에 관해 확신을 가져야 한다.

가이드라인은 다음을 포함할 것이다.

- cipher suites의 협정(키 설정을 위해 어떤 알고리듬을 사용할 것인지, 암호화를 위한 것인지, 키 크기는 얼마인지 등)
- 키 전송 기법에 사용되는 키의 생성과 분배
- PKI 관련 이슈들
- 생성으로부터 파괴까지 키의 처리
- 암호 모듈로 키 입력
- security association의 관리
- 키의 cryptoperiod
- 프로토콜 이슈
- 상호 운영성
- 구현 지침

- 유효성(validation)과 테스트
- 보증(assurance)(도메인 파라미터와 공개키 유효성, 정확한 구현)
- 권고되는 파라미터, 인코딩 제약 사항들, 지수 크기, 키 크기
- 책무(accountability)
- 키 복구/archiving과 백업

가이드라인 문서 개발 일정은 다음과 같다.

※ 가이드라인 문서 개발 일정

Summer 2001	public review를 위한 문서/개요
October 2001	진행 과정을 논의하기 위해 워크샵 개최
Thereafter	권고안/Special Pub. 개발 계속 두번째 워크샵 이전에 갱신된 draft 문서 제공 두번째 워크샵(Nov. 1-2, 2001)에서 진행과정 논의 NIST 권고안/Special Pub.의 최종 draft를 개발 public comment 받음

III. 키 설정 기법

많은 IT(Information Technology) 시스템들은 그것들이 처리하는 데이터의 기밀성과 무결성을 보호하기 위해 잘 설정된 암호 기법 적용을 필요로 한다. FIPS 197에 정의된 AES, FIPS 46-3에 채택된 Triple-DES, FIPS 198에 정의된 HMAC과 같은 알고리듬들이 이들 서비스를 제공하기 위해 선택될 수 있다. 이들 알고리듬들은 시스템간의 상호 운영성을 용이하게 하기 위해 표준화 되어졌다. 그렇지만 이들 알고리듬들의 사용은 사전에 공유된 keying material의 설정을 요구한다. 신뢰되는 전달자가 keying material을 수동으로 전달할 수 있다. 그렇지만 시스템을 사용하는 개체의 수가 증가하면 keying material 분배에 관련되는 작업도 지수적으로 증가한다. 이를 해결하기 위해 NIST는 키 관리 표준을 개발하고 있다. 현재 NIST에서 개발중인 키 관리 기법 정의 문서는 ANSI에 의해 개발된 표준인 ANSI X9.42와 ANSI X9.63으로부터 키 설정 기법 개발을 위한 접근법을 제공한다. 최종적인 기법 정의 문서는 ANSI X9.44에서의 키 전송 기법을 포함할 것이다. 또한 키 wrapping 기술, 마스터 키로부터 유도되는 키들에 관한 논의들도 포함될 계획이다.

1. 기호와 약어

H	승인된 해시 함수
$Text_1$, $Text_2$	키 확인 과정동안 사용되어지고 keying material을 설정하는 개체간에 전달되는 선택적인 비트 스트링
U	키 설정 과정의 한 개체 또는 그 개체의 신분을 나타내는 비트 스트링
V	키 설정 과정의 다른 개체 또는 그 개체의 신분을 나타내는 비트 스트링
X||Y	두 스트링 X와 Y의 연결
ANSI X9.42의 기호들	
p, q, g	도메인 파라미터
mod p	정수 값에서 modulo p reduction
r_U, r_V	개체 U와 개체 V의 ephemeral 개인키
t_U, t_V	개체 U와 개체 V의 ephemeral 공개키
x_U, x_V	개체 U와 개체 V의 static 개인키
y_U, y_V	개체 U와 개체 V의 static 공개키
Z	키 유도 함수를 사용하여 keying material을 유도하기 위해 사용되는 공유 비밀
Z_e	D-H primitive를 사용하여 계산된 ephemeral 공유 비밀
Z_s	D-H primitive를 사용하여 계산된 static 공유 비밀
ANSI X9.63의 기호들	
[X]	스트링 X의 포함이 선택 사항이라는 것을 지시
a, b	타원 곡선식을 정의하는 필드 원소
avf(P)	타원 곡선 포인트 P의 associate value
$d_{e,U}$, $d_{e,V}$	개체 U와 개체 V의 ephemeral 개인키
$d_{s,U}$, $d_{s,V}$	개체 U와 개체 V의 static 개인키
FR	사용된 basis의 indication
G	타원 곡선에서 distinguished 포인트
h	포인트 G의 order에 의해 나누어지는 타원 곡선의 order
n	포인트 G의 order
φ	Infinity라 불리는 타원 곡선의 특수한 포인트. 이 포인트는 타원 곡선의 덧셈 항등원이다.
q	필드 크기
$Q_{e,U}$, $Q_{e,V}$	개체 U와 개체 V의 ephemeral 공개키
$Q_{s,U}$, $Q_{s,V}$	개체 U와 개체 V의 static 공개키
SEED	타원 곡선이 랜덤하게 생성된다면 존재하는 선택적인 비트 스트링
x_P	포인트 P의 x 좌표
y_P	포인트 P의 y 좌표
Z	키 유도 함수를 사용하여 keying material을 유도하기 위해 사용되는 공유 비밀
Z_e	D-H primitive를 사용하여 계산된 ephemeral 공유 비밀
Z_s	D-H primitive를 사용하여 계산된 static 공유 비밀

2. 키 설정 알고리듬 분류

암호 keying material은 키 일치 또는 키 전송 기법을 사용하여 개체들간에 전자적으로 설정된다. 키 일치 과정 동안 설정되는 keying material은 전송되지 않고 keying material 계산을 위해 필요한 정보가 개체간에 교환된다. 키 일치 기법은 비대칭키(공개키) 기술을 사용한다. 키 전송 과정 동안 암호화된 keying material이 keying material을 생성한 개체로부터 다른 개체로 전송된다. 키 전송 기법은 대칭 또는 공개키 기술을 이용한다.

ANSI X9.42와 X9.63의 기법들은 이산 대수(discrete logarithm) 문제의 어려움에 기반한다. ANSI X9.42는 유한체 위에서 계산되고 ANSI X9.63은 타원 곡선을 사용하여 계산된다.

3. 암호학적 요소들

가. 도메인 파라미터

ANSI X9.42와 X9.63에 명시된 것과 같이 이산 대수 기반 암호들은 공개키와 개인키 쌍이 도메인 파라미터에 관하여 생성될 것을 요구한다. 이 도메인 파라미터는 파라미터들이 적절히 생성되었다는 것을 보장하도록 검증되어야 한다. 도메인 파라미터가 공개 정보이지만 키 쌍을 사용하는 개체들에 대해 주어진 키 쌍과 도메인 파라미터간의 정확한 대응이 유지되도록 관리되어야 한다. 도메인 파라미터는 일정 시간동안 고정되어질 수 있고 다수의 키 설정 기법에 사용되어질 수 있다.

ANSI X9.42와 X9.63에서 몇몇 기법들은 분리된 도메인 파라미터들이 같은 기법에서 static과 ephemeral 키에 사용되는 것을 허용한다. 그렇지만 NIST의 문서에서는 하나의 도메인 파라미터 집합만을 사용한다. 즉 같은 도메인 파라미터가 주어진 기법에서 static과 ephemeral 키에 대해 사용된다. 도메인 파라미터의 생성과 검증에 관한 것은 ANSI 9.42와 ANSI 9.63 문서를 따른다.

나. 개인키와 공개키

1) 개인키와 공개키 생성

Static과 ephemeral 키 쌍은 같은 primitive를 사용하여 생성된다. ANSI X9.42에 명시된 기법에 대해 개인키 x와 공개키 y는 도메인 파라미터(p, q, g)를 사용하여 생성된다. ANSI X9.63에 명시된 기법에 대해 개인키 d와 공개키 Q는 도메인 파라미터(q, FR, a, b, [SEED], G, n, h)를 사용하여 생성된다.

Static 공개키/개인키가 키 설정 과정에서 참여자에 의해 요구되면 static 키는 키 설정 과정에서 참여 전에 미리 생성되어야 한다. 각 개인키는 통계적으로 유일해야 하고 예측 불가능해야 하고 승인된 난수 발생기를 사용하여 생성되어야 한다. 같은 개인키가 하나 이상의 도메인 파라미터 집합에 사용되지 말아야 한다.

2) 공개키 검증

안전한 키 설정은 공개키/개인키 쌍의 검증에 의존한다. NIST의 기법 정의 문서는 키의 수신자에 의해 수행되어지는 공개키 검증을 요구한다. Static 공개키는 수신자 또는 수신자가 신뢰하는 개체에 의해 검증되어야 한다. Static 공개키 검증은 다음 세 가지 방법 중의 한 가지로 달성된다.

① explicit 공개키 검증 수행. 즉 공개키가 특정한 수학적 성질을 가지는지를 검사한다. 이 방법은 ANSI X9.42의 7.4절과 ANSI X9.63의 5.2.2절에서 방법 1이다.

② 다른 개체(즉 CA)가 방법 1을 사용하여 공개키를 검증했다는 확신을 수신한다. 이것은 ANSI X9.42의 7.4절에서 방법 2와 ANSI X9.63의 5.2.2절에서 방법 3이다.

③ 다른 개체(즉 CA)가 신뢰되는 루틴을 사용하여 공개키를 검증했다는 확신을 수신한다. 이것은 ANSI X9.42의 7.4절과 ANSI X9.63의 5.2.2절에서 방법 4이다.

각 ephemeral 공개키는 공유 비밀을 유도하기

위해 사용되기 전에 수신자에 의해 검증되어야 한다. Ephemeral 공개키는 위의 방법 1 또는 2를 사용하여 검증된다.

두 가지 ANSI 표준은 공개키 검증을 위한 추가적인 선택 사항을 포함한다. 이 선택 사항은 implicit 공개키 검증으로 이것은 신뢰되는 루틴을 사용하여 공개키를 생성함으로써 수행된다. 기법 정의 문서는 이 방법을 포함하지 않는다.

3) 키 쌍 관리

공개키/개인키 쌍은 대응하는 도메인 파라미터에 정확히 관련되어야 한다. 개인키는 허가되지 않은 노출, 수정, 대치로부터 보호되어야 한다. 따라서 개인키는 허가되지 않은 접근으로부터 보호되어야 한다.

각 static 개인키의 cryptoperiod는 명확히 정의되어야 한다. Static 개인키는 cryptoperiod가 만료되면 파괴되어야 한다. Ephemeral 키는 가능한 한 사용 시점에 생성되어야 한다. Ephemeral 개인키는 공유 비밀이 계산되면 즉시 파괴되어야 한다.

Static 공개키의 수신자는 공개키, 도메인 파라미터, 키를 소유한 개체간의 바인딩에 대한 보장을 획득해야 한다. 이 보증은 신뢰되는 개체에 의해 서명된 공개키 인증서를 검증함으로써 제공된다.

Static 공개키는 신뢰되는 방법으로 획득되어야 한다. 즉 CA에 의해 서명된 인증서를 통해 획득되거나 또는 공개키 소유자가 수신 개체에 의해 신뢰되고 수신된 데이터의 출처가 인증될 수 있다면 공개키 소유자로부터 직접 획득되어야 한다.

Static 키 쌍은 하나 이상의 키 설정에 사용될 수 있다. 그렇지만 다른 공개키/개인키 쌍은 다른 목적에 대해 사용되어야 한다.

다. 키 유도 함수

키 유도 함수는 공유 비밀로부터 keying material을 유도하기 위해 다음처럼 사용된다. NIST의 기법 정의 문서에서는 다음의 키 유도 함수를 정의하고 있다.

Function call: kdf(Z, OtherInput) 여기서 OtherInput은 U, V, keydatalen, hashlen, [SharedInfo]이다.

- Input description:

① 비트 스트링 Z는 공유 비밀이다.

② 비트 스트링 U와 V는 참여 개체의 신분을 나타낸다. U는 키 설정 프로토콜을 개시하는 개체이고 V는 응답자이다.

③ keydatalen은 생성될 keying material의 길이이다. 이 값은 $hashlen \times (2^{32}-1)$보다 작아야 한다.

④ hashlen은 keying material을 유도하기 위해 사용되는 해시 함수의 비트 길이이다.

⑤ 선택 사항인 SharedInfo는 개체 U와 V에 의해 공유된 데이터로 구성된다.

- Process:

① 32-비트 big-endian 비트 스트링 counter를 0x00000001로 설정한다.

② For i =1 to i =[$keydatalen/hashlen$], do the following:

$Hash_i = H(Z \| counter \| U \| V \| [SharedInfo])$

counter 증가

i 증가

③ $Hhash_j$ 는 $keydatalen/hashlen$ 이 정수이면 $Hash_i$ 를 나타낸다. 그렇지 않으면 $Hash_j$ 의 왼쪽 ($keydatalen-(hashlen \times (j-1))$) 비트를 나타낸다.

④ $DerivedKeyingMaterial = Hash_1 \| Hash_2 \| \ldots \| Hash_{j-1} \| Hash_j$

- Output: 비트 스트링 *DerivedKeyingMaterial* 의 *keydatalen* 비트

유도된 keying material은 하나 이상의 키 또는 다른 암호 keying material로 parse될 수 있다.

$Hashlen \times (2^{32}-1)$ 비트보다 크거나 같은 비트 길이의 스트링에 대해 키 유도 함수를 호출하는 시도는 “invalid”를 출력하고 중단해야 한다.

라. MAC

메시지 인증 코드(Message Authentication Code: MAC)는 대칭키와 데이터의 함수이다. 키 설정 기법에서 개체는 수신 또는 유도된 데이터에 MacTag를 계산하는 것이 요구된다. MacTag는 데이터가 정확히 수신 또는 유도되었다는 것을 확인하기 위해 다른 개체에게 전송된다.

MAC 함수는 키 확인을 제공하기 위해 사용된다. 또한 키 설정 기법의 구현을 검증하기 위해 사용된다. Tag 계산과 검사는 ANSI X9.42의 7.8절과 ANSI X9.63의 5.7절에 정의되어 있다.

마. Associate Value Function(Elliptic Curve Only)

Associate value function은 타원 곡선에 관련된 정수를 계산하기 위해 MQV 키 일치 기법 계열에 의해 ANSI X9.63에서 사용된다. avf(P)를 도메인 파라미터(q, FR, a, b, [SEED], G, n, h)를 사용하는 ANSI X9.63의 5.6.1절에 정의된 포인트 P($P \neq \varphi$)의 associate value function으로 정의한다.

바. 암호 해시 함수와 난수 발생기

해시 함수가 요구되면(즉 키 유도 함수 또는 MAC 계산을 위해 HMAC이 사용될 때), 승인된 해시 함수가 사용되어야 한다. 또한 표준이 랜덤하게 생성된 값의 사용을 요구하면, 그 값들은 승인된 난수 발생기를 사용하여 생성되어야 한다.

사. 키 확인

키 확인(key confirmation)은 개체가 같은 키를 유도했다는 보증을 제공하기 위해 사용된다. 키 설정 과정에서 개체들은 표준에 의해 제공된 기법들 중 하나를 사용하여 공유 비밀 Z를 먼저 설정해야 한다. 그 후에 각 개체는 6.3절을 사용하여 keying material을 유도하여 다음처럼 MacKey와 MacData로 분리해야 한다.

MacKey || *MacData* = *DerivedKeyingMaterial*

키 설정 과정에서 응답자 V는 $MacData_V$ = 0x02||V||U||*EphemPubKeyV*||*EphemPubKeyU*||[$Text_1$]으로 설정하고 $MacTag_V$를 계산하여 개시자 U에게 전송한다. 키 설정 과정에서 개시자 U는 $MacData_U$ = 0x03||U||V||*EphemPubKeyU*||*EphemPubKeyV*||[$Text_2$]으로 설정하고 $MacTag_U$를 계산하여 응답자 V에게 전송한다. 두 개체가 MacData를 구성하기 위한 모든 정보를 안다면 MacData는 전송될 필요가 없다.

아. 공유 비밀 계산

공유 비밀 계산을 위한 프리미티브는 ANSI 표준에 정의되어 있다. 각 키 설정 기법은 정확히 하나의 프리미티브 사용이 요구된다. 기법 정의 문서에 포함되는 기법에 의해 사용되어야 하는 5개의 ANSI 프리미티브들은 다음과 같다.

① ANSI X9.42 7.5.1절의 Diffie-Hellman 프리미티브. 이 프리미티브는 dhHybrid1, dhEphem, dhHybridOneFlow, dhOneFlow, dhStatic 기법에 의해 사용되어야 한다. 이들 기법들은 유한체 연산과 Diffie-Hellman 알고리듬에 기반한다.

② ANSI X9.63 5.4.2절의 수정된 Diffie-Hellman 프리미티브. 이 프리미티브는 Full Unified Model, Ephemeral Unified Model, 1-Pass Unified Model, 1-Pass Diffie-Hellman, Static Unified Model 기법들에 의해 사용되어야 한다. 이들 기법들은 타원 곡선 연산과 Diffie-Hellman 알고리듬에 기반한다.

③ ANSI X9.42 7.5.2.1절의 MQV2 프리미티브. 이 프리미티브는 MQV2 인터랙티브 기법에 의해 사용되어야 한다. 이 기법은 유한체 연산과 MQV 알고리듬에 기반한다.

④ ANSI X9.42 7.5.2.2절의 MQV1 프리미티브. 이 프리미티브는 MQV1 store-and-forward 기법에 의해 사용되어야 한다. 이 기법은 유한체 연산과 MQV 알고리듬에 기반한다.

⑤ ANSI X9.63 5.5절의 MQV 프리미티브. 이 프

리미티브는 Full MQV, 1-Pass MQV 기법에 의해 사용되어야 한다. 이 기법들은 타원 곡선 연산과 MQV 알고리듬에 기반한다.

공유 비밀은 공유된 keying material로 직접 사용되지 말아야 한다. 공유된 keying material은 공유 비밀에 키 유도 함수를 적용하여 계산되어야 한다.

4. 키 일치 기법들

NIST에서 고려하고 있는 키 일치 기법들은 3가지 범주로 분류된다. 범주의 분류는 키 일치 과정에서 두 개체에 의해 사용되는 ephemeral 키의 개수에 기반한다(<표 1> 참조). 범주 C(i)에서 개체 U와 V는 총 i개의 ephemeral 키 쌍을 가진다. 첫번째 범주 C(2)는 두 개체에 의해 ephemeral 키 쌍의 생성을 요구하는 기법으로 이루어진다(두 개체 참여 기법. 즉 interactive 또는 2-way 기법). 두번째 범주 C(1)은 단지 한 개체에 의해 ephemeral 키 쌍의 생성을 요구하는 기법으로 구성된다(한 개체 참여. 즉 store-and-forward 또는 1-way 기법). 세번째 범주 C(0)는 ephemeral 키를 사용하지 않는 기법으로 이루어진다(static 기법. 즉 passive 기법).

각 범주는 개체에 의해 static 키의 사용에 의해 분류되는 하나 이상의 범주들로 구성된다. 하위 범주 C(i,j)에서 개체 U와 V는 총 i개의 ephemeral 키 쌍과 j개의 static 키 쌍을 가진다(<표 2> 참조).

기법들은 유한체 연산(FF)을 사용하는지 타원 곡선 연산(EC)을 사용하는지에 따라 다시 분류된다. 기법들은 Diffie-Hellman 또는 MQV 프리미티브를 사용한다. 따라서 예로 C(2,2,DH,FF)는 유한체 연산을 이용한 Diffie-Hellman 프리미티브를 사용하는 두 개의 ephemeral 키와 두 개의 static 키를 가진 기법으로 dhHybrid1 기법을 완전히 분류한다(<표 3> 참조).

<표 1> 키 일치 기법 범주

범주	주석
C(2): 두 개체 참여 (interactive, 2-way)	각 개체가 ephemeral 키 쌍을 생성한다.
C(1): 한 개체 참여 (store-and-forward, 1-way)	개시자만이 ephemeral 키 쌍을 생성한다.
C(2): static(passive)	ephemeral 키를 사용하지 않는다.

<표 2> 키 일치 기법 하위 범주

범주	주석
C(2): 두 개체 참여 (interactive, 2-way)	C(2,2): 각 개체가 하나의 ephemeral 키 쌍을 생성하고 하나의 static 키 쌍을 가진다.
	C(2,0): 각 개체가 하나의 ephemeral 키 쌍을 생성하고 static 키 쌍을 가지지 않는다.
C(1): 한 개체 참여 (store-and-forward, 1-way)	C(1,2): 개시자는 하나의 ephemeral 키 쌍을 생성하고 하나의 static 키 쌍을 가진다. 응답자는 static 키 쌍만을 가진다.
	C(1,1): 개시자는 하나의 ephemeral 키 쌍을 생성하고 static 키 쌍을 가지지 않는다. 응답자는 static 키 쌍만을 가진다.
C(2): static(passive)	C(0,2): 각 개체는 static 키 쌍만을 가진다.

<표 3> 키 일치 기법들

범주	하위 범주	프리미티브	연산	기법	완전한 분류
C(2)	C(2,2)	DH	FF	dhHybrid1	C(2,2,DH,FF)
C(2)	C(2,2)	DH	EC	Full Unified Model	C(2,2,DH,EC)
C(2)	C(2,2)	MQV	FF	MQV2	C(2,2,MQV,FF)
C(2)	C(2,2)	MQV	EC	Full MQV	C(2,2,MQV,EC)
C(2)	C(2,0)	DH	FF	dhEphem	C(2,0,DH,FF)
C(2)	C(2,0)	DH	EC	Ephemeral Unified Model	C(2,0,DH,FF)
C(1)	C(1,2)	DH	FF	dhHybrid OneFlow	C(1,2,DH,FF)
C(1)	C(1,2)	DH	EC	1-Pass Unified Model	C(1,2,DH,EC)
C(1)	C(1,2)	MQV	FF	MQV1	C(1,2,MQV,FF)
C(1)	C(1,2)	MQV	EC	1-Pass MQV	C(1,2,MQV,EC)
C(1)	C(1,1)	DH	FF	dhOne Flow	C(1,1,DH,FF)
C(1)	C(1,1)	DH	EC	1-Pass Diffie-Hellman	C(1,1,DH,EC)
C(0)	C(0,2)	DH	FF	dhStatic	C(0,2,DH,FF)
C(0)	C(0,2)	DH	EC	Static Unified Model	C(0,2,DH,EC)

키 일치 과정에서 각 개체는 같은 도메인 파라미터를 사용해야 한다. 이들 파라미터들은 키 일치 과정의 개시 이전에 설정되어야 한다.

일반적인 플로 다이어그램이 기법의 각 범주에 대해 제공된다. 플로 다이어그램에서의 점선 화살표는 개체 자신들 또는 CA와 같은 third party에 의해 분배되어지는 static 공개키의 분배를 표시한다. Static 공개키는 각 개체의 공개키가 개체의 신분과 도메인 파라미터 집합에 연결되도록 신뢰되는 방법(즉 CA에 의해 서명된 공개키 인증서를 통해)으로 분배되어야 한다. 연결 과정은 static 공개키의 검증을 포함해야 한다. Static 공개키의 획득은 키 일치 과정 이전 또는 키 일치 과정 동안 중에서 사용 이전에 임의의 시간에 발생할 수 있다.

플로 다이어그램에서 실선 화살표는 키 일치 과정 동안 발생하는 ephemeral 공개키의 분배를 표시한다. Ephemeral 키 쌍은 가능한 한 사용 시점에 생성되어야 한다. 그리고 계산이 완료되면 즉시 파괴되어야 한다.

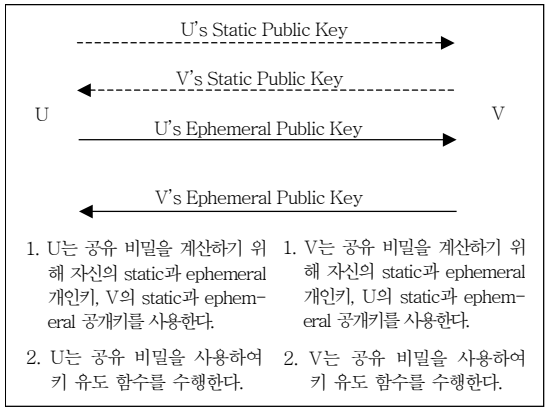
가. 두 개체 참여(interactive, 2-way), C(2)

이 범주에서 각 개체는 ephemeral 키 쌍을 생성하여 상대방에게 ephemeral 공개키를 전송한다. 두 개체는 공유 비밀을 유도하기 위해 유사한 계산을 수행한다. 그렇지만 키 유도 계산과 키 확인 계산은 개시자와 응답자에 대해 다르다.

이 범주는 static 키의 사용에 의해 결정되는 두 개의 하위 범주로 구성된다. 첫번째 하위 범주에서 각 개체는 static과 ephemeral 키 모두를 가진다. 두번째 하위 범주에서 각 개체는 ephemeral 키만을 가진다.

1) 각 개체는 static 키 쌍을 가지고 ephemeral 키 쌍을 생성한다. C(2,2)

이 기법에 대해 각 개체(U와 V)는 static 키 쌍을 가지고 키 일치 과정 동안 ephemeral 키 쌍을 생성한다. 모든 키 쌍은 같은 도메인 파라미터를 사용하여 생성되어야 한다. 개체 U와 V는 키 설정 과정

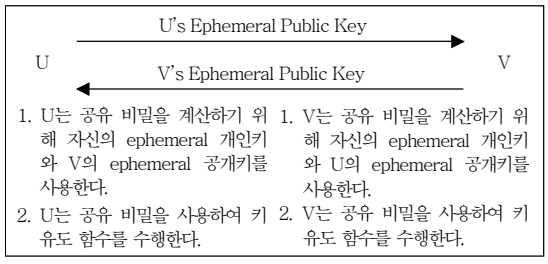


(그림 1) 각 개체가 static과 ephemeral 키 쌍을 모두 사용할 때 일반적인 프로토콜

이전에 생성된 서로의 static 키를 얻는다. 두 개체는 ephemeral 공개키/개인키 쌍을 생성하여 ephemeral 공개키를 교환한다. Static과 ephemeral 키를 사용하여 두 개체는 공유 비밀을 생성한다. 공유된 keying material은 공유 비밀로부터 유도된다((그림 1) 참조). 여기에는 dhHybrid1, C(2,2,DH,FF), Full Unified Model, C(2,2,DH,EC), MQV2, C(2,2,MQV, FF), Full MQV, C(2,2,MQV,EC) 기법이 속한다.

2) 각 개체는 ephemeral 키 쌍을 생성하고, static 키 쌍을 사용하지 않는다. C(2,0)

이 범주에 대해 Diffie-Hellman 기법들만이 명시된다. 각 개체(U와 V)는 같은 도메인 파라미터를 가지고 ephemeral 키 쌍을 생성한다. 두 개체는 ephemeral 공개키 쌍을 교환한 후 공유 비밀을 생성한다. Keying material은 공유 비밀로부터 유도된다



(그림 2) 각 개체가 ephemeral 키 쌍을 생성하고 static 키를 사용하지 않을 때 일반적인 프로토콜

((그림 2) 참조). 여기에는 dhEphem, C(2,0,DH,FF), Ephemeral Unified Model, C(2,0,DH,EC) 기법이 속한다.

나. 한 개체 참여(client/server, store-and-forward), C(1)

이 범주에서 키 일치에 참여하는 개체들은 키 일치 과정을 개시했는지에 의존하여 공유 비밀을 결정하기 위해 다른 계산을 수행한다. 개체 U는 개시자이고 개체 V는 응답자라고 하자. 개시자만이 ephemeral 키 쌍을 생성한다.

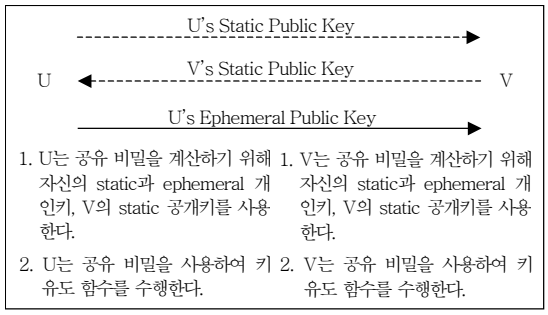
이 범주는 개체들에 의해 static 키 쌍의 소유에 의해 결정되는 두 가지 하위 범주로 구성된다. 첫번째 하위 범주에서 개시자와 응답자는 모두 static 키 쌍을 가지고 개시자는 ephemeral 키 쌍을 생성한다. 두번째 하위 범주에서는 개시자가 ephemeral 키 쌍을 생성하지만 static 키 쌍을 가지지 않는다. 응답자는 static 키 쌍만을 가진다.

1) 개시자가 static 키 쌍을 가지고 ephemeral 키 쌍을 생성한다. 응답자는 static 키 쌍을 가진다. C(1,2)

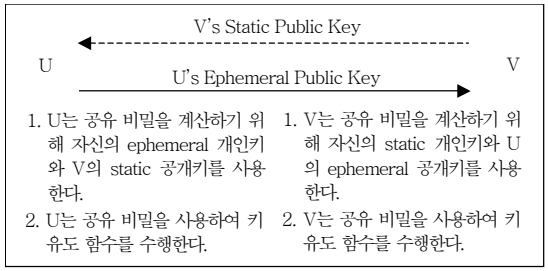
이 기법에서 개체 U는 static과 ephemeral 공개키/개인키 쌍 모두를 사용하고, 개체 V는 static 개인키/공개키 쌍만을 사용한다. 개체 U와 개체 V는 상대방의 static 공개키를 신뢰되는 방법으로 획득한다. 개체 U는 또한 개체 V에게 자신의 ephemeral 공개키를 전송한다. 공유 비밀은 static과 ephemeral 키를 사용하여 개체들에 의해 생성된다. 공유 keying material은 공유 비밀을 사용하여 유도된다((그림 3) 참조). 여기에는 dhHybridOneFlow, C(1,2,DH,FF), 1-Pass Unified Model, C(1,2,DH,EC), MQV1, C(1,2,MQV,FF), 1-Pass MQV, C(1,2,MQV,EC) 기법이 포함된다.

2) 개시자가 ephemeral 키 쌍만을 생성한다. 응답자는 static 키 쌍만을 가진다. C(1,1)

이 기법에서 개체 U는 ephemeral 키 쌍을 생성



(그림 3) 개시자는 static과 ephemeral 키 쌍을 모두 가지고 응답자는 static 키 쌍만을 가질 때 일반적인 프로토콜

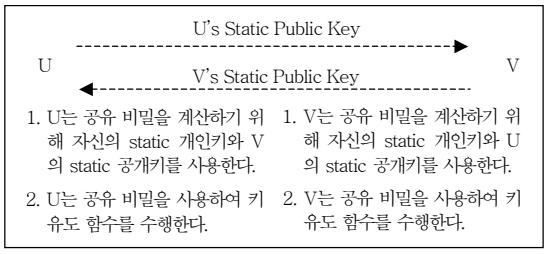


(그림 4) 개시자는 ephemeral 키 쌍만을 모두 가지고 응답자는 static 키 쌍만을 가질 때 일반적인 프로토콜

하지만 static 키 쌍을 가지지 않는다. 개체 V는 static 개인키/공개키 쌍만을 사용한다. 개체 U는 개체 V의 static 공개키를 신뢰되는 방법으로 획득하고 개체 V에게 자신의 ephemeral 공개키를 전송한다. 개체들은 자신의 개인키와 상대방의 공개키를 이용하여 공유 비밀을 계산한다. 공유 keying material은 공유 비밀을 사용하여 유도된다((그림 4) 참조). dhOneFlow, C(1,1,DH,FF)와 1-Pass Diffie-Hellman, C(1,1,DH,EC) 기법이 여기에 속한다.

다. Static key only, C(0)

이 범주에서 각 개체들은 같은 도메인 파라미터를 사용하여 생성된 static 키 쌍만을 가진다. 각 개체는 상대방의 static 공개키를 획득하여 자신의 static 개인키와 상대방의 static 공개키를 이용하여 공유 비밀을 계산한다. Keying material은 키 유도



(그림 5) 각 개체는 static 키 쌍만을 가진다

함수와 공유 비밀을 사용하여 유도된다((그림 5) 참조). 여기에는 dhStatic, C(0,2,DH,FF)와 Static Unified Model, C(0,2,DH,EC) 기법이 속한다.

IV. 결 론

전자상거래와 같은 암호 응용 서비스들은 데이터 무결성과 기밀성과 같은 암호 서비스를 제공할 수 있는 잘 설정된 암호 기법을 필요로 한다. 그렇지만 그러한 기법들의 구현은 미리 공유 비밀키의 설정을 요구한다. 키 관리 시스템의 크기 또는 시스템을 사용하는 개체의 수가 증가함에 따라 키 설정을 위한 필요성은 키 관리 문제를 야기한다. 이를 해결하기 위해 미국의 NIST에서는 키 관리 표준을 개발하고 있다. 본 고에서는 현재 NIST에서 진행중인 키 관리 표준에 대한 동향과 고려되고 있는 기법들을 분석하였다.

국내에서도 전자상거래와 같은 암호 응용 서비스의 사용과 정부 기관에서 민감한 데이터 보호를 위해 암호 기술 사용의 필요성이 계속적으로 증대되고 있다. 따라서 국내에서도 국내 실정에 맞는 키 관리 표준의 개발이 시급히 추진되어야 할 것으로 사료된다.

참 고 문 헌

[1] NIST Key Management Standard Web Site, http://csrc.nist.gov/encryption/kms/

[2] White Paper of NIST Key Management Standard, http://csrc.nist.gov/encryption/kms/white-paper.pdf

[3] Key Establishment Schemes Document, http://csrc.nist.gov/encryption/kms/schemes_workshop_doc.pdf

[4] ANSI X9.42 -2001, "Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography."

[5] ANSI X9.44(Draft), "Public Key Cryptography for the Financial Services Industry: Agreement and Key Transport Using Factoring-Based Cryptography."

[6] ANSI X9.63(Draft), "Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Key Cryptography."

[7] Key Management Guideline, http://csrc.nist.gov/encryption/kms/guideline-1.pdf