通信情報保護學會論文誌
第 1 卷, 第 1 號, 1991. 12

개인정보에 기초한 키 분배방식의 분석 및 개선방안

임채훈* · 이필중*

Analysis and Improvement of ID-based Key Distribution Systems

Chae-Hoon Lim and Pil-Joong Lee

요 약

개인정보에 바탕을 둔 키 분배방식은 공개 키 분배방식에서의 공개키 인증문제나 통신량 집중현상을 자연스럽게 해결할 수 있는 매우 효율적인 방법으로 널리 연구되고 있다. 특히 사전통신 없이도 안전하게 세션키를 공유할 수 있는 비대화형의 키 분배방식은 일방통신만이 허용되는 E-mail 등에서 보안 서비스를 제공하는데 중요한 역할을 한다. 본 논문에서는 기존에 제안된 대표적인 방식들을 분석하고 안전성이나 효율성 등에 있어서 가능한 개선방안들을 제시하고자 한다. 특히 기존에 제안된 대부분의 비대화형 방식에서 공통적으로 겪고 있는 사용자간의 결탁문제를 해결한 매우 효율적인 방법으로 최근에 제안된 Maurer-Yacobi 방식을 분석하여 그 문제점을 제기하고 이를 해결할 수 있는 개선방안도 제시한다.

Abstract

An ID-based scheme provides a very efficient solution to the key distribution problem, since it can solve both the authentication problem and the communication complexity problem in a public key scheme. Especially, and ID-based noninterative key distribution system plays an crucial role in the one-way communication environment such as secure electronic mail, owing to its noninteractiveness. This paper aims at analyzing the previously proposed schemes and providing possible improvements. It also demonstraes that the Maurer-Yacobi's scheme presented in Eurocry'91 is not secure, and provides an countemeasure to overcome the security problem.

* 포항공과대학 전자전기공학과

1. 서 론

대용량의 정보를 안전하게 전송하기 위해 일반적으로 속도가 빠른 관용 암호시스템(conventional cryptosystem)을 주로 이용하게 되는데 여기에 필요한 세션키(session key)를 안전하게 분배해 주는 키 분배방식(key distribution system)에 관한 연구가 암호시스템의 효율적인 운용을 위한 선결과제가 되고 있다. 키 분배 문제를 해결하는 방안으로 관용 암호법을 이용한 중앙 집중형의 키 분배방식이 주로 사용되어 왔으나[1,2] Diiffie와 Hellman[3]이 최초로 제안한 공개 키 분배방식(public key distribution system) 이 보다 효율적인 키 관리방안으로 널리 연구되어 왔다[4-10]. 이는 공개 통신로(public channel) 상의 통신만으로 세션키 공유가 가능하도록 함으로써 세션 키 분배를 위한 비밀 통신로(secure channel)를 없앨 수 있으며 또한 시스템의 안전성이 각 사용자의 비밀 키의 비밀유지 여부에만 의존한다는 점에서 키 분배 문제를 근본적으로 해결한 최초의 키 관리 방법이었다고 할 수 있다.

그러나 이 방식에서는 공개키의 분배문제가 다시 중요한 과제로 등장하게 된다. 즉 각 사용자의 공개는 모든 다른 사용자들이 이용 가능하도록 공개해야 하며 주로 중앙의 공개키 디렉토리(public key directory)에서 관리하게 되는데, 이 경우 각 사용자들은 매 세션마다 공개키 디렉토리를 엑세스해야 하며 더우기 공개키 디렉토리로부터 받은 공개키가 정확히 자신이 원하는 상대방의 공개키인지를 확인할 수 있는 수단이 제공되어야 하므로 중앙 집중형의 키 분배방식에 필적하는 통신 복잡도(communication complexity)를 초래하며 또한 공개키의 인증문제(authentication problem)를 해결하기 위한 추가기능이 필요하게 된다. 이와같은 문제점을 해결하기 위한 방안으로 센타에서 각 사용자의 가입시에 그 사용자의 공개키를 그의 ID와 결합하여 RSA 방식으로 서명한 공개키 증명서(public key certificate)를 각 사용자에게 분배해 주고 각 사용자는 이를 서로 교환하도록 하는 방법이 제안되었다[11,12]. 이 방법은 각 사용자의 ID와 센타의 서명용 공개키를 이용하면 다른 모든 사용자들이 그 사용자의 공개키의 진위여부를 확인할 수 있게 함으로써 인증문제를 해결하였고 따라서 공개키를 중앙의 공개키 디렉토리에서 관리할 필요없이 통신을 하고자 하는 두 사용자자간에 이 증명서를 교환하고 이로부터 인증된 상대방의 공개키를 얻도록 함으로써 통신 복잡도의 증가를 피할 수 있게 하였다.

한편, 1984년 Shamir[13]는 공개키 암호시스템에서 공개키의 관리문제를 근본적으로 해결할 수 있는 방안으로 개인정보(identification information : ID)에 바탕을 둔 암호시스템(ID-based cryptosystem)의 개념을 제안하였다. 이는 특히 신분인증(indentification)이나 디지탈 서명(digital signature) 용으로 널리 연구되고 있는 것으로 누구나 알 수 있고 또한 그 사용자를 유일하게 식별해 줄 수 있는 주소, 성명, 주민등록번호 등과 같은 사용자의 개인정보를 공개키로 이용함으로써 공개키 관리문제를 자연스럽게 해결한 것이다. 그러나 개인정보에 바탕을 둔 암호시스템에서는 중앙의 키 생성센타(key generation center)가 모든 사용자들의 비밀키를 계산하여 발급하므로 모든 사용자들이 센타를 전적으로 신뢰할 수 있어야 한다는 조건을 전제로 한다. 따라서 이는 일반대중을 상대로 하는 공중통신망과 같은 범용의 통신망보다는 기업체나 연구소 등의 사설망이나 은행 등의 금융망과 같이 중앙의 본부가 신뢰할 수 있는 센타의 역할을 할 수 있는 폐쇄 사용자 그룹용(closed user group)에 이상적인 방식이라 할 수 있다. 특히 개인정보에 바탕을 둔 암호시스템은 센타가 각 사용자들의 가입시에 그의 비밀키를 계산하여 이를 센타의 공개정보와 함께 스마트카드에 내장하여 발급한다면 신분인증 프로토콜이나 디지탈 서명방식, 그리고 키 분배방식 등을 하나의 카드로 통합 구현함으로써 각종 보안서비스를 동시에 제공할 수 있으므로 미래 정보화사회에서 통합카드의 기능을 수행할 수 있다는 점에서 주요 연구대상이 되고 있다.

개인정보를 공개키로 이용하여 공개키 디렉토리를 없앰으로써 공개키 분배방식에서의 통신 복잡도나 인증문제를 해결할 수 있는 개인정보에 바탕을 둔 키 분배방식에 대한 연구도 활발히 진행되어 왔다. Okamoto-Tanaka[14], Gunther[15], Bauspieß-Knobloch[16] 등은 RSA나 EIGamal 서명을 이용하여 개인정보에 바탕을 둔 키 분배방식을 제안하였다. 이 방식들은 센타에서 분배받은 각 사용자의 ID에 바탕을 둔 비밀키를 사용하여 Diffie-Hellman 형의 키 분배방식을 구성한 것으로 어떤 공격에 대해서도 안전한 것으로 알려져 있으나 사전통신을 통하여 세션키를 계산한다는 점에서 개인정보에 바탕을 둔 암호시스템의 원래 개념과는 약간 벗어난 것이라 할 수 있다.

개인정보에 바탕을 둔 암호시스템의 원래 개념을 가장 잘 살린 것으로 통신을 하고자 하는 두 사용자간에 사전통신이 필요없이 공개정보 및 자신의 비밀키만으로 세션키를 계산할 수 있게 해주는 비대화형의 키 분배방식(noninteractive key distribution system)을 들 수 있다. 이는 특히 E-mail과 같은 일방향의 통신망을 이용하는 응용에서는 매우 중요한 역할을 하게 되며, 또한 다른 일반적인 응용에서도 대화형의 키 분배방식에 비해 통신비용을 줄일 수 있으므로 매우 효율적인 방법이라 할 수 있다. 그러나 지금까지 제안된 대부분의 방식들에서는 일정수의 사용자들이 결탁하면 센타의 비밀정보나 다른 사용자들의 비밀키를 계산하는 것이 가능하다는 문제점(conspiracy problem)을 안고 있다.

본 논문에서는 Gunther의 대화형 키 분배방식[5]에서 효율성을 높일 수 있는 개선방안을 제시하고 Tsujii-Itoh 방식[17,18] 및 Tanaka 방식[19,20] 등에서 존재하는 사용자간의 결탁문제를 해결할 수 있는 새로운 비대화형의 키 분배방식으로 최근에 제안된 Maurer-Yacobi 방식[21]을 분석하며 그 문제점을 제기하고 이를 해결할 수 있는 방안도 제시하기로 한다.

2. Gunther의 대화형 방식 및 개선방안

Gunther[15]가 제안한 대화형의 키 분배방식은 센타에서 각 사용자의 ID와 결합된 그 사용자의 비밀키를 EIGamal의 디지탈 서명을 이용하여 계산, 분배한다. 우선 센타는 유한체 GF(p), p는 소수, 와 이 유한체상의 원시원소(primitive element) g, 그리고 각 사용자 i의 기술자(descriptor)인 D_i를 p의 비트길이로 변환하는 일대일 함수 f를 선택하여 모든 사용자에게 공개한다. 또한 센타의 비밀키 $X \in Z_{p-1}$에 대응하는 공개키 $Y \equiv g^x \bmod p$를 계산하여 이 역시 모든 사용자에게 공개한다.

각 사용자 i가 시스템에 가입하면 센터는 그의 비밀키 X와 사용자 i의 ID인 $ID_i = f(D_i)$를 이용하여 다음과 같이 사용자 i의 비밀키(R_i, S_i)를 계산하여 비밀리에 전달한다.

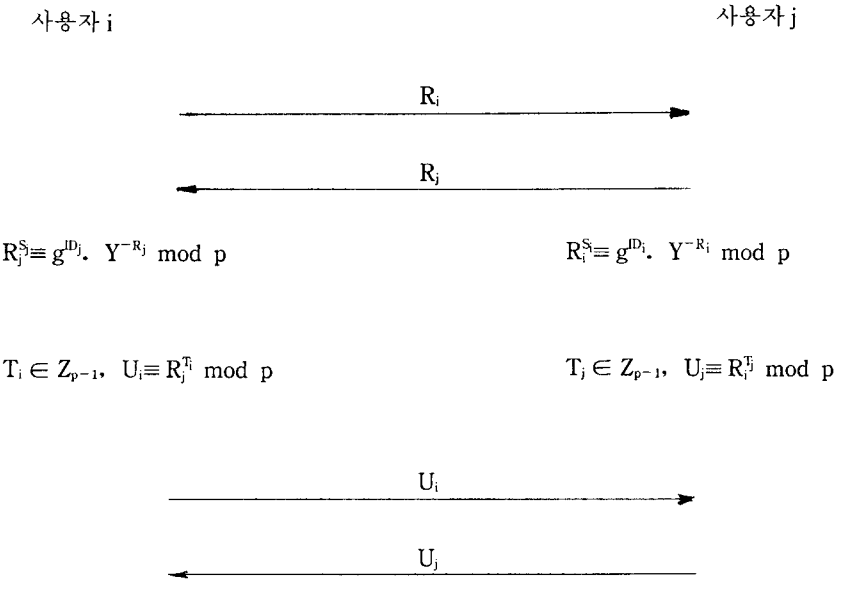
$$ID_i \equiv XR_i + K_iS_i \bmod p-1,\ R_i \equiv g^{K_i} \bmod p,\ K_i \in Z_{p-1},\ GCD(K_i,\ p-1)=1.$$

여기서 K_i는 센타가 랜덤하게 선택한 값으로 모든 사용자에게 비밀을 유지해야 할 것이다. K_i마저 알려지면 센타의 비밀키 X가 노출될 것이기 때문이다. 더우기 사용자간의 결탁에 의한 센타의 비밀키 노출을 막기 위해서는 각 사용자의 비밀키를 계산할 때 서로 다른 K_i를 사용해야 할 것이다. 한편 K_i를 $GCD(K_i,\ p-1)=1$이 되도록 선택하였으므로, R_i 역시 g와 마찬가지로 원시원소가 된다는 것을 알 수 있다.

위 서명의 인증은 (R_i, S_i)를 이용하여 $g^{ID_i} \equiv Y^{R_i} \cdot R_i^{S_i} \bmod p$ 인지를 검사하는 것으로 이는 $R_i^{S_i} \equiv g^{ID_i} \cdot Y^{-R_i}$가 되고 여기서 R_i를 공개하더라도 사용자 i의 비밀키 S_i가 노출될 염려는 없다는 것을 알 수 있다. 따라서 $R_i^{S_i}$는 센타의 공개정보와 사용자 i의 ID_i 및 R_i만으로 계산될 수 있고 위에서 언급했듯이 R_i는 원시원소이므로 이 $R_i^{S_i}$를 Diffie-

Hellman형 키 분배방식에서의 공개키로 사용할 수 있음을 알 수 있다. 그러나 여기서 R_i는 센타의 비밀키 노출을 막기 위해 각 사용자마다 다른 값을 사용하였으므로 이를 서로 교환해야 할 것이다.

다음에 Gunther에 의해 제안된 대화형 키 분배방식을 간략히 기술한다. 센타의 공개키는 모든 사용자들에게 알려져 있고 또한 각 사용자 i의 ID인 ID_i는 미리 정해진 포오멧의 기술자 D_i 및 공개된 함수 f(.)로부터 누구나 계산할 수 있다고 가정한다.



$$K_i \equiv U_j^{S_i} \cdot (R_j^{S_j})^{T_i} \equiv R_i^{S_i T_j} \cdot R_j^{S_j T_i} \bmod p = K_j \equiv U_i^{S_j} \cdot (R_i^{S_j})^{T_j} \bmod p$$

위의 키 분배방식에서 상대방의 공개키 $R_j^{S_j}$를 계산하는 것은 곧 센타의 서명을 인증하는 과정이므로 인증된 공개키(authenticated public key)를 얻을 수 있으며 또한 각 사용자는 자신의 비밀키를 이용하면 센타가 각 사용자의 비밀키를 생성하는 방법과 마찬가지로 메세지에 대한 EIGamal 서명을 생성시킬 수 있다는 장점이 있다. 즉 공개된 hash 함수 h로 메세지 M을 압축하여 h(M)을 계산하고 GCD(T, p−1)=1인 T를 랜덤하게 선택하여 $V \equiv R_i^T \bmod p$를 계산한 다음 $h(M) \equiv S_i V + TW \bmod p-1$을 만족하는 W를 구하면 ($R_i$, V, W)가 메세지 M에 대한 서명이 된다. 이 서명의 인증은 $R_i^{h(M)} \equiv (R_i^{S_i})^V \cdot V^W \equiv (g^{ID_i} \cdot Y^{-R_i})^V \cdot V^W \bmod p$를 만족하는지 여부를 검사하면 될 것이다.

한편 위의 키 분배방식에서는 EIGamal 서명의 성질상 상대방의 공개키 계산시 서로 다른 밑을 사용하게 되므로 순차적으로 두번씩의 전송을 해야하고 또한 세션키 계산에 필요한 계산량이 상당히 많다는 것을 알 수 있다. 여기서는 Schnorr의 서명[22]을 사용함으로써 공통의 밑을 사용할 수 있고 계산효율을 높일 수 있는 방안을 제시하기로 한다. 먼저 센타는 Schnorr의 서명에서와 같이 g의 위수(order)로 $g^q \equiv 1 \bmod p$가 되는 p−1의 약수인 소수 q를 추가로 공개하여 지수를 계산하는 법(modulus)으로 사용하며 센타의 비밀키 X도 Z_q상에서 선택하고 해당 공개키 $Y \equiv g^x \bmod p$를 모든 사용자에게 공개한다. 법 p는 512비트 정도의 소수로 잡고 q는 p−1의 약수로서 약 160비트 정도의 소수가 되게

한다. 이산대수 계산시 기본원소 g의 위수 q를 알고 있을때 매우 효율적인 알고리즘은 Pollard의 Monte Carlo 법[23]으로 그 시간복잡도는 $O(\sqrt{q})$ 정도로 주어지므로 q는 최소한 140 비트 이상이 되도록 선택해야 할 것이다. 이제 사용자의 비밀키 생성을 Schnorr의 서명을 이용하여 q를 법으로 하여 다음과 같이 계산한다면 공통의 밑 g를 사용할 수 있을 것이다. 여기서 함수 h는 공개된 해쉬함수로 임의의 메세지를 t 비트길이의 압축된 값으로 변환하는 함수이다(t는 약 80비트 정도가 되게 한다). 즉

$$XH_i + K_i \equiv S_i \bmod q,\quad H_i = h(R_i,\ ID_i),\quad R_i \equiv g^{K_i} \bmod p,\quad K_i \in Z_q.$$

센타는 위와같이 계산한 $(H_i,\ S_i)$를 각 사용자 i에게 비밀리에 전해준다. 그러면 각 사용자는 i는 $R_i \equiv g^{S_i} Y^{-H_i} \bmod p$를 계산하여 $H_i = h(R_i,\ ID_i)$가 성립하는지를 검사함으로써 자신의 비밀키를 인증할 수 있다. 이때 각 사용자 i의 공개키는 $P_i \equiv g^{S_i} \equiv Y^{H_i}.\ R_i \bmod p$와 같이 R_i가 주어지면 기본원소 g를 공통의 밑으로 하여 쉽게 계산할 수 있다. 이제 두 사용자 i, j는 Z_q 상에서 비밀 랜덤수 ρ_i, ρ_j를 선택한 후 예비통신 단계에서 $\{R_i,\ T_i \equiv g^{\rho_i}\}$, $\{R_j,\ T_j \equiv g^{\rho_j}\}$를 서로 교환하면 다음과 같이 MTI 방식[24]의 세션키를 계산할 수 있다. 즉

$$K_i \equiv (Y^{H_j}.\ R_j)^{\rho_i}.\ T_j^{S_i} \equiv g^{S_j\rho_i}.\ g^{\rho_j S_i} \equiv (Y^{H_i}.\ R_i)^{\rho_j}.\ T_i^{S_j} \equiv K_j \bmod p.$$

위의 변형된 방식에서 각 사용자 i의 세션키 계산은 다음과 같이 수행하면 q의 비트길이를 n이라 할때 평균적으로 약 (3.375n+5)번의 모듈라 곱셈이면 세션키 계산이 가능하다. 여기서 H_j는 상대방으로부터 받은 R_j와 상대방의 ID_j를 이용하여 공개된 해쉬함수 h로 $H_j = h(R_j,\ ID_j)$와 같이 계산한다.

① $T_i \equiv g^{\rho_i} \bmod p$: 평균 1.5n번의 모듈라 곱셈

② $E_1 \equiv H_j.\ \rho_i \bmod q$, $E_2 \equiv \rho_i$, $E_3 = S_i$: 1번의 모듈라 곱셈

③ $K_S \equiv Y^{E_1}.\ R_j^{E_2}.\ T_j^{E_3} \equiv g^{s_j\rho_i}.\ g^{\rho_j\rho_i} \bmod p$: 평균 (1.875n+4)번의 모듈라 곱셈.

위의 세션키 계산에서 과정 ③은 다음과 같은 순서로 계산한다. 우선 YR_j, YT_j, R_jT_j, $YR_jT_j \pmod p$는 미리 계산하여 두고 필요할때 사용하도록 하며 지수 E_i는 이진수로 $E_i = (e_{i1},\ e_{i2},\ \cdots,\ e_{il})$, $e_{ij} \in \{0, 1\}$로 표시된다고 가정한다.

i) i=n, z=1로 둔다.

ii) 다음의 과정을 i=1이 될때까지 반복한다.

$\{z \equiv z^2.\ Y^{e_{1i}}.\ R_j^{e_{2i}}.\ T_j^{e_{3i}} \bmod p,\ i = i-1\}$

iii) $K_s = z$가 구하고자 하는 계산결과이다.

위의 계산과정 ii)에서 e_{1i}, e_{2i}, e_{3i}가 모두 0인 경우를 제외하면 매 비트마다 두번씩의 모듈라 곱셈이 필요하므로 평균적으로 1.875n 번의 모듈라 곱셈이면 원하는 결과를 얻을 수 있다. 법 q의 크기가 n=160비트라고 하면 약 545번의 모듈라 곱셈이 소요되므로 한번의 512비트 모듈라 멱승 (modular exponentiation) 연산보다 더 적은 계산량으로 세션키 공유가 가능함을 알 수 있다. 또한 전송정보 T_i는 원래의 방식에서와는 달리 $T_i \equiv g^{\rho_i} \bmod p$와 같이 기본원소 g를 밑으로 이용하므로 각 사용자마다 고유한 비밀의 사전처리 알고리즘(preprocessing algorithm)을 이용한다면 휴지기간(idle time) 동안에 계산할 수 있을 것이므로 1.5n번의 모듈라 곱셈을 줄이는 것도 가능할 것이다. 이 경우 약 305번의 모듈라 곱셈이면 세션키 계산이 가능하므로 매우 효율적임을 알 수 있다.

한편 위와같이 각 사용자의 비밀키를 생성하면 각 사용자들도 자신의 비밀키를 이용하여 원하는 메세지에 대한 Schnorr의 서명을 생성시킬 수 있으므로 원래의 방식에서보다 훨씬 효율적인 서명의 생성 및 인증이 가능할 것이다.

위의 개인정보에 기초한 대화형의 키 분배방식은 ElGamal의 서명에 의해 인증된 상대방의 공개키를

얻을 수 있다는 점을 제외하면 Diffie-Hellman형의 공개 키 분배방식과 거의 같음을 알 수 있다. 또 다른 대표적인 대화형 키 분배방식인 Okamoto의 방식[14]에서는 합성수 n을 이용하여 RSA 서명에 바탕을 두어 각 사용자의 비밀키를 계산한다. 그러나 공개 키 분배방식에서의 공개키의 인증은 RSA 서명을 이용한 Kohnfelder[11]의 공개키 증명서나 Girault[12] 등의 self-certified public key 등에 의해 매우 효율적으로 제공될 수 있으며 더우기 이들 방식에서는 센타가 각 사용자의 비밀키를 계산하는 것이 불가능하므로 사용자의 측면에서 보면 개인정보에 기초한 키 분배방식 보다는 훨씬 바람직한 방법이라 할 수 있다.

예를들어 Girault의 방식을 살펴보자. 먼저 센타는 RSA 서명용 공개키 n, e를 공개하고 해당 비밀키 d($ed=1 \bmod \lambda(n)$)를 계산하여 비밀리에 간직한다. 각 사용자 i는 자신의 비밀키 S_i를 선택하여 비밀로 하고 이에 대응하는 공개키 $g^{S_i} \bmod p$을 계산하여 그의 ID와 함께 센타에 제공하면 센타에서는 $(g^{S_i}-ID_i)^d \bmod n$의 형태로 RSA 서명한 P_i를 사용자 i에게 주어 이를 공개키로 이용하는 것이다. 이제 통신을 하고자 하는 두 사용자 i, j는 자신들의 공개키와 랜덤하게 선택한 비밀수 ρ_i ρ_j를 이용하여 전송정보 $\{P_i, T_i \equiv g^{\rho_i}\}$, $\{P_j, T_j \equiv g^{\rho_j}\}$를 서로 교환한다. 이때 각 사용자는 상대방으로부터 받은 공개키로부터 키 분배에 사용할 원래의 공개키인 g^{S_i}, g^{S_j}를 센타의 공개키를 이용하면 쉽게 계산할 수 있다. 즉 $P_i^e+ID_i \equiv g^{S_i} \bmod n$과 같이 계산되므로 공개키 e를 2나 3정도의 작은 수로 선택한다면 몇번의 모듈라 곱셈(modular multiplication)이면 인증된 상대방의 공개키를 계산할 수 있을 것이다. 이와같이 상대방의 공개키를 얻은 후에는 위의 방식에서와 마찬가지 방법으로 그들간의 세션키를 계산할 수 있다.

이처럼 각 사용자의 비밀키가 노출되지 않는다는 동일한 전제하에서는 공개 키 분배방식에서의 공개키 인증문제는 쉽게 해결되므로 센타에 대한 신뢰성의 정도에 있어서 공개키 분배방식이 개인정보에 기초한 키 분배방식 보다는 보다 바람직한 키 분배방식이라 할 수 있다. 그러나 개인정보에 기초한 키 분배방식의 가장 큰 장점은 사전통신이 필요없이 센타의 공개정보와 상대방의 ID만을 이용하여 세션키를 계산할 수 있는 비대화형 방식을 설계할 수 있다는 점이다.

3. 비대화형 키 분배방식의 분석 및 개선방안

일반적으로 센타의 비밀정보와 사용자의 ID를 이용하여 각 사용자의 비밀키를 계산하는 비대화형 방식에서는 센타의 비밀정보의 일부가 각 사용자의 비밀키로 분산되는 결과를 초래하므로 일정 수의 사용자들이 결탁하면 센타의 비밀정보의 일부 혹은 전부를 계산해 낼 수 있다는 문제점이 제기되어 왔고 이를 극복할 수 있는 비밀키 생성방법을 개발하려는 노력도 꾸준히 진행되고 있다. 사용자간의 결탁문제는 비밀공유방식(secret sharing scheme)을 응용한 Blom의 SKGS(Symmetric Key Genration System)[25]에서 그 전형적인 예를 볼 수 있으며 이외에도 Tsujii-Itoh 방식이나 Tanaka의 방식을 포함하여 Matsumoto-Imai의 KPS(Key Predistribution System)[26] 등 지금까지 제안된 대부분의 방식들에서 가장 큰 문제점으로 지적되고 있다. 이러한 문제점을 극복할려는 노력의 일환으로 Tsujii와 Chao[27]는 Crypto'91에서 매우 복잡한 과정을 통하여 세션키를 생성하는 새로운 방법을 발표하였으나 이에 필요한 엄청난 계산량(공통의 법으로 사용되는 합성수 N(n비트)의 2배 길이의 지수에 대한 2n번의 모듈라 멱승 연산)을 고려하면 실용성보다는 사용자자간의 결탁문제가 비대화형 방식의 본질적인 문제점이 아니라는 사실을 보일려는 시도였다고 생각된다. 한편 Maurer와 Yacobi[21]는 Eurocrypt'91에서 비대화형 키 분배방식을 설계하는 다른 접근법으로 센타의 비밀정보를 이용하는 대신 사용자의 ID의 모듈라 제곱에 대한 이산대수를 사용자의 비밀키로 계산하는 새로운 방식을 제안하였다.

이 장에서는 우선 3.1절에서 Tsujii-Itoh[18] 및 Tanaka[20]의 비대화형 키 분배방식에서 사용자간의 결탁에 의한 안전성 파괴문제를 살펴본다. 그리고 3.2절과 3.3절에서는 이와같은 결함이 존재하지 않는 매우 효율적인 방식으로 Maurer와 Yacobi에 의해 제안된 비대화형 방식을 분석하여 그 문제점을 제기하고 이를 해결할 수 있는 개선방안도 제시하기로 한다.

3.1 Tsujii-Itoh 및 Tanaka의 비대화형 키 분배방식

Tsujii-Itoh 방식은 유한체(finite field)나 정수링(integer ring) 상에서의 이산대수 문제(discrete logarithm problem)의 어려움에 그 바탕을 둔다. 여기서는 두 큰 소수 p, q의 곱으로 구성되는 n비트 길이의 합성수 m(=pq)를 법으로 하는 정수링상의 방식을 기술하고 그 문제점인 사용자간의 결탁(conspiracy)에 의한 시스템의 안전성 파괴문제를 간략히 살펴보기로 한다.

우선 각 사용자 i의 공개키는 그 사용자를 유일하게 기술하는 정해진 포오맷(format)의 기술자(descriptor) $\mathbf{D}_i$를 공개된 일대일 함수 f(예 : RSA)로 변환시킨 n차원 이진 벡타(n-dimensional binary vector)로 구성된 $\mathbf{ID}_i$로 표기한다. 즉

$$\mathbf{ID}_i = f(\mathbf{D}_i) = (e_{i1}, e_{i2}, \cdots, e_{in}),\ e_{ik} \in \{0, 1\}\ (1 \le k \le n).$$

센타는 두 개의 큰 소수 p, q를 발생시켜 m=pq, λ(m)=LCM(p−1, q−1)를 계산하고 곱셈군(multiplicative group) $(Z/mZ)^*$의 generator g를 발생시킨다(여기서 λ(m)은 m에 대한 Carmichael 함수 값으로 m이 두 소수 p, q의 곱으로 구성될 때는 p−1과 q−1의 최소공배수(least common multiple)로 주어진다). 이 중 m, g는 공개하고 λ(m)은 비밀로 한다. 센타의 비밀키 **X** 및 공개키 **Y**는 다음과 같이 $(Z/mZ)^*$ 상의 이산대수 문제에 바탕을 두어 계산한다. 여기서 $Z_{\lambda(m)}$은 λ(m)으로 나누었을때의 나머지를 나타낸다.

$$\mathbf{X} = (x_1, x_2, \cdots, x_n),\ x_k \in Z_{\lambda(m)}\ (1 \le k \le n).$$
$$\mathbf{Y} = (y_1, y_2, \cdots, y_n),\ y_k \equiv g^{x_k} \bmod m\ (1 \le k \le n).$$

이제 센타는 각 사용자 i의 비밀키 S_i를 센타의 비밀키 **X**와 그 사용자의 공개키인 $\mathbf{ID}_i$를 이용하며 다음과 같이 계산한 다음 센타의 공개키와 함께 각 사용자에게 전달한다.

$$S_i = \mathbf{X} \cdot \mathbf{ID}_i = \Sigma_{1 \le k \le n}\ x_k e_{ik} \bmod \lambda(m).$$

여기서 모든 사용자들의 비밀키는 서로 달라야 할 것이므로 센타의 비밀키 **X**는 임의의 n차원 이진벡타 **I**, **J**에 대하여 다음의 조건을 만족하도록 선택한다.

$$\mathbf{X} \cdot \mathbf{I} \ne \mathbf{X} \cdot \mathbf{J} \bmod \lambda(m),\ \mathbf{I} \ne \mathbf{J}.$$

이와같은 벡타 **X**를 발생시키는 가장 쉬운 방법은 다음과 같이 Merkle-Hellman[28]의 Knapsack 암호에서 공개키를 발생시키는 과정을 이용하는 것이다. 먼저 센타는 초증가 수열(superincreasing sequence) $\{v_k\}$ $(1 \le k \le n)$, $\Sigma_{1 \le k \le n}\ v_k < \lambda(m)$, 를 발생 시키고 λ(m)과 서로소(relatively prime)인 w를 선택하여 $x_k \equiv v_k \cdot w \bmod \lambda(m)$인 $\{x_k\}$ $(1 \le k \le n)$를 계산하는 것이다. $\{v_k\}$의 초증가 성질로부터 위의 조건을 만족하는 것은 당연하며 $\{x_k\}$는 센타의 비밀키로 비밀이 유지될 것이므로 Knapsack 암호의 안전성과는 무관하게 된다.

이제 센타로부터 그의 비밀키 S_i와 센타의 공개정보 m, g, f, **Y**를 분배받은 사용자 i는 임의의 다른 사용자 j의 공개키 $P_j \equiv g^{s_j}$를 다음과 같이 사용자 j의 ID와 자신의 비밀키 및 센타의 공개정보로부터 쉽게 계산할 수 있다.

$$P_j \equiv \prod_{1\le k\le n} y_k^{e_{jk}} \equiv \prod_{1\le k\le n} g^{x_k e_{jk}} \equiv g^{\sum_{1\le k\le n} x_k e_{jk}} \equiv g^{S_j} \bmod m.$$

따라서 두 사용자간의 세션키 K_s는 Diffie-Hellman 방식으로 $K_s \equiv P_j^{S_i} \equiv g^{S_i S_j} \bmod m$와 같이 계산할 수 있을 것이다.

위의 방식은 문헌[18]에 설명되었듯이 (n+1)명의 사용자가 서로 결탁하여 비밀키를 공유하면 센타의 비밀인 λ(m)을 계산할 수 있고 따라서 m의 두 소인수를 알면 센타의 비밀정보 **X**와 동등한 정보를 계산할 수 있다는 것이 Coppersmith에 의해 지적되었다. 여기서는 Shamir에 의해 개발된 보다 일반적인 공격법으로 (n+2)명의 사용자가 결탁하여 센타의 비밀정보 **X**를 직접 계산할 수 있는 방법을 설명하기로 한다. 이는 각 사용자의 비밀키를 센타의 비밀정보 **X**와 해당 사용자의 ID의 벡타곱으로 계산한 선형성에 기인한다.

우선 (n+1)명의 사용자 i($1\le i\le n+1$)가 결탁하면 다음과 같은 선형 합동식계(system of linear congruences)가 주어지며 여기서 첫째 메트릭스 **D**의 n개의 칼럼벡타들이 정수링 상에서 선형독립(linearly independent)이라고 가정한다.

$$\begin{bmatrix} \mathbf{ID}_1 \\ \mathbf{ID}_2 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{ID}_{n+1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ \cdot \\ \cdot \\ \cdot \\ S_{n+1} \end{bmatrix} - \begin{bmatrix} c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ c_{n+1} \end{bmatrix} \qquad \lambda(m) = \mathbf{D}.\mathbf{X}.$$

위의 식은 다음과 같이 변형될 수 있다.

$$\begin{bmatrix} \mathbf{ID}_1 & S_1 \\ \mathbf{ID}_2 & S_2 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \mathbf{ID}_{n+1} & S_{n+1} \end{bmatrix} \begin{bmatrix} -x_1 \\ -x_2 \\ \cdot \\ \cdot \\ -x_n \\ 1 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ c_{n+1} \end{bmatrix} \qquad \lambda(m) = \mathbf{D}'.\mathbf{X}'.$$

메트릭스 **D**의 n칼럼벡타들이 정수링상에서 선형독립이라는 가정으로부터 메트릭스 **D**′의 (n+1) 칼럼들 역시 선형독립일 확률은 매우 높을 것이므로 메트릭스 **D**′는 정수링 상에서 nonsingular(따라서 판별식 $\det(\mathbf{D}') \neq 0$)할 것이다. 한편 위의 식은 **D**′.**X**′$\equiv$**0** mod λ(m)이 되고 만일 메트릭스 **D**′가 Z/λ(m) 상에서 nonsingular 한다면 **X**′$=$**0** mod λ(m)이 된다. 그러나 **X**′$=[$**X**, 1$]^T$는 결코 0 벡타가 될 수 없으므로 $\det(\mathbf{D}') \equiv 0 \bmod \lambda(m)$이 될 것이다. 또한 $\det(\mathbf{D}') \neq 0$이므로 $\det(\mathbf{D}')$는 λ(m)의 정수배가 된다. 더우기 (n+2)명의 사용자중 다른 (n+1)명의 사용자들간의 결탁에 의해서도 위와 같은 결과를 얻는다면, 즉 마찬가지 방법으로 메트릭스 **D**″에 대해 $\det(\mathbf{D}'') \equiv 0 \bmod \lambda(m)$를 얻는다면, 이 두

판별식의 최대공약수 GCD(det(**D**′), det(**D**″))는 λ(m)의 작은 배수가 될 것이므로 λ(m)을 쉽게 계산할 수 있을 것이다.

위와 같은 공격방법에 의해 m의 두 소인수 p, q를 구하면 첫번째 메트릭스식에서 n명의 사용자로 구성된 $\mathbf{D}\cdot\mathbf{X}\equiv\mathbf{S} \bmod \lambda(m)$, $\mathbf{S}=[S_1, S_2, \cdots, S_n]^T$로부터 센타의 비밀정보인 **X**나 혹은 이와 동등한 벡타 **X**′를 쉽게 구할 수 있다. 여기서 det(**D**)가 반드시 λ(m)(항상 짝수)과 서로소가 되지는 않을 것이나 일반적으로 m의 두 소인수 p, q는 p±1, q±1이 큰 소수를 포함하도록 선택되므로 d=GCD(det(**D**), λ(m))는 작은 숫자가 될 것이다. 편의상 두 소수를 $p=2^{c_1}p'+1$, $q=2^{c_2}q'+1$(p′, q′는 소수, c_1, c_2는 작은 정수로 $c_1>c_2$)로 두면 d는 최대로 2^{c_1}이 될 것이므로 λ(m)′=λ(m)/d=p′q′를 법으로 하여 계산된 **X**′는 원래의 **X**의 λ(m)′의 배수의 차이가 날 것이다. 따라서 이 **X**′를 이용하여 구한 임의의 사용자 k의 비밀키 S_k'는 원래의 비밀키 S_k와 λ(m)′의 배수 차이가 날 것이므로 최대 C_1번의 시도이면 원래의 비밀키를 구할 수 있다. Tsujii 등은 안전성 파괴에 필요한 결탁자의 수를 배가시키는 방법을 제시했으나 이 경우 센타의 비밀정보 및 공개정보 역시 배가되므로 실용성있는 개선책으로 보기는 어려울 것이다. 그러나 개인정보에 기초한 암호시스템은 센타에서 모든 사용자의 비밀키를 생성하는 특성상 기업체의 사설망이나 금융망 등과 같은 폐쇄 사용자 그룹에서의 각 지점들간의 비밀통신이 주요 적용대상이 될 것이므로 사용자 수가 안전성 파괴에 필요한 결탁자 수 이하의 소규모인 경우는 매우 효율적으로 사용될 수 있을 것이다.

한편 H. Tanaka는 Crypto'87에서 비대화형 방식으로는 최초의 시도라 할 수 있으나 계산량 및 메모리 등에 있어서 비실용적이며 마찬가지로 사용자간의 결탁문제가 존재하는 ID에 기초한 키 분배방식을 제안하였다[19]. 후에 Tanaka는 사용자의 계산량을 줄일 수 있도록 이를 약간 변형시켰으며 이는 문헌[20]에 그 안전성이 분석되어 있다. 여기서는 이 변형된 방식 및 그 안전성 분석을 간단히 소개하기로 한다. 이 방식의 시스템 파라미터는 다음과 같다.

◆ 센타의 공개정보 : m(=pq), g.

◆ 센타의 비밀정보 : $\lambda(m)=\mathrm{LCM}(p-1, q-1)$, $\mathbf{X}=\{x_i\}$ $(1\le i\le n)$, $x_i\in Z_{\lambda(n)}$.

◆ 각 사용자 i의 공개키 : $\mathbf{ID}_i=(e_{i1}, e_{i2}, \cdots, e_{in})$, $e_{ij}\in\{0, 1\}$.

◆ 각 사용자 i의 비밀키 : $\mathbf{G}_i=\{g_{ik}\}$ $(1\le k\le n)$.

(센타에 의해 계산 : $g_{ik}\equiv g^{S_i x_k} \bmod m$, $S_i\equiv\mathbf{X}\cdot\mathbf{ID}_i\equiv\sum_{1\le k\le n} x_k e_{ik} \bmod \lambda(m)$).

이제 두 사용자 i, j의 세션키는 다음과 같이 계산될 수 있다.

$$K_{ij}\equiv\prod_{1\le k\le n} g_{ik}^{e_{jk}}\equiv\prod_{1\le k\le n} g^{S_i x_k e_{jk}}\equiv g^{S_i\sum_{1\le k\le n} x_k e_{jk}}\equiv g^{S_i S_j} \bmod m.$$

이 방식에서는 사용자들의 결탁에 의해 센타의 비밀정보인 **X**를 계산하는 것은 불가능하나 다음과 같이 다른 사용자의 비밀키를 계산하는 것이 가능하다. 어떤 사용자 t의 ID인 $\mathbf{ID}_t$가 다른 r명의 사용자들의 ID의 선형결합으로 표시될 수 있다고 가정하자. 즉 $\mathbf{ID}_t=\sum_{1\le k\le n} c_i\mathbf{ID}_i=c_1\mathbf{ID}_1+c_2\mathbf{ID}_2+\cdots+c_r\mathbf{ID}_r$, c_i는 정수, 이면 $S_t\equiv c_1S_1+c_2S_2+\cdots+c_rS_r \bmod \lambda(m)$이 성립할 것이고 따라서 $g_{ti}\equiv g^{S_t x_i}\equiv(g^{S_1 x_i})^{c_1}\cdot(g^{S_2 x_i})^{c_2}\cdots g^{S_r x_i})^{c_r} \bmod m$, i=1, 2, …, n과 같이 계산되므로 이 r명의 사용자들이 결탁하여 그들의 비밀키를 공유한다면 사용자 t의 비밀키를 쉽게 계산할 수 있다. 이와같은 문제점이 제기되는 것은 Tsujii-Itoh 방식에서와 마찬가지로 각 사용자들의 비밀키 계산에 사용된 S_i가 센타의 비밀정보의 선형결합으로 구성되기 때문이며 이와 같은 문제는 사용자의 비밀키 생성에서 센타의 비밀정보와 사용자 ID의 선형함동식 관계가 사용되는 한 항상 존재하게 된다.

3.2 Maurer-Yacobi의 비대화형 방식

Maurer와 Yacobi는 Eurocrypt'91에서 한번의 모듈라 멱승만으로 세션키 공유가 가능하다는 점에서 매우 효율적인 새로운 비대화형 키 분배방식을 제안하였다[21]. 이절과 다음 절에서는 이들이 제안한 방식을 분석하여 문제점을 제기하고 이를 극복할 수 있는 개선방안을 제시하고자 한다.

Maurer-Yacobi 방식은 적절히 선택된 큰 합성수 m을 소인수분해하는 것은 계산상 불가능하다는 사실과 어느 정도 크기의 소수 p에 대해서는 이산대수를 계산하는 것이 가능하다는 사실을 바탕으로 한다. 현재의 알고리즘 및 기술 수준으로는 소인수분해가 불가능하도록 합성수 m을 선택하되 이 합성수의 소인수들을 알고 있는 센타는 각 소인수들에 대한 이산대수 계산이 가능하도록 소인수들의 크기를 적절히 선택하면 센타는 각 사용자의 ID를 mod m으로 제곱한 결과에 대한 이산대수를 계산할 수 있을 것이므로 이를 사용자의 비밀키로 삼자는 것이다. 즉 센타는 각 사용자 i의 비밀키로 $ID_i^2 \equiv g^{S_i} \bmod m$을 만족하는 S_i를 계산하여 해당 사용자에게 전해준다. 그러면 시스템에 가입한 어떤 사용자 i도 상대방 j와의 세션키를 $K_S \equiv (ID_j^2)^{S_j} \equiv \bmod m$과 같이 한번의 모듈라 멱승이면 계산할 수 있으므로 기존의 어떤 방식보다도 효율적이라 할 수 있다. 여기서 각 사용자의 ID를 모듈라 제곱한 결과에 대해 이산대수를 계산한 것은 합성수 m을 법으로 사용하는 경우 모든 가능한 수에 대해 기본원소 g를 밑으로 하는 이산대수가 반드시 존재하지는 않기 때문이다. 반면 합성수 m과 서로소인 임의의 수의 모듈라 제곱에 대해서는 항상 g를 밑으로 하는 이산대수가 존재한다는 것을 보일 수 있다. 이는 후에 설명하기로 하고 우선 이 방식의 기반이 되는 소인수분해와 이산대수 문제를 푸는 알고리즘들에 대해 현재의 기술수준으로 계산 가능한 숫자의 한계를 간략히 살펴보기로 한다.

소인수분해와 이산대수 문제를 푸는 알고리즘들의 실행시간(asymptotic running time)을 추정하는 함수로 $L(x)=\exp(\sqrt{\log x \log\log x})$가 공통적으로 사용된다. 현재까지 알려진 알고리즘으로 유한체 GF(p) 상에서 이산대수를 계산하는 가장 빠른 알고리즘은 Coppersmith, Odlyzko, Schroeppel 등이 개발한 것으로 $L(p)^{1+o(1)}$ 정도의 시간복잡도를 갖는다[29]. 이 방법에 의하면 소-중형 컴퓨터로도 65-70자리 정도의 소수에 대한 이산대수를 계산하는 것이 가능하며 대규모의 병렬처리를 이용하는 경우 약 100자리 정도의 소수에 대한 이산대수 계산도 가능한 것으로 보고 있다.

한편 소인수분해 알고리즘으로 가장 빠른 것은 Pollard에 의해 개발된 NFS(Number Field Seive) 알고리즘이며 $\exp((\log m)^{1/3}(\log\log m)^{2/3})$ 정도로 추정되는 시간복잡도를 가지나 아직까지는 $r^s \pm e$ 형태의 특수한 숫자에 그 응용이 제한되고 있다[30]. 이를 일반화시키는 작업이 진행되고 있으나 그 실용성 여부는 아직까지 불투명하다. 보다 일반적인 숫자를 소인수분해하는 가장 효율적인 알고리즘으로는 QS(Quadratic Sieve)의 변형인 MPQS(Multiple Polynomial Quadratic Sieve)[31]로 $L(m)^{1+o(1)}$의 시간복잡도를 갖는 것으로 추측된다. 이 방법에 의해 대규모의 병렬처리를 통해 소인수분해한(발표된 바로는) 가장 큰 숫자는 111자리수이며 자리수 3개의 증가에 실행시간은 약 2배 정도로 늘어나는 것으로 평가되므로 이 방법에 의해서도 소인수분해 가능한 숫자는 당분간 120 자리수 이하로 볼 수 있을 것이다[32]. 한편 실행시간이 합성수 m의 소인수 p의 크기에 의존하는 알고리즘으로 Lenstra의 ECM(Elliptic Curve Method)[33]이 있으며 대략 $L(p)^{\sqrt{2+o(1)}}$ 정도의 시간복잡도를 갖는 것으로 추정되며 이 방법으로 찾은 가장 큰 소인수는 Silverman이 찾은 38자리수로 알려져 있다. 대규모 병렬처리시에는 대략 40-45자리 소인수까지 찾을 수 있는 것으로 보고 있다.

이상에서 간략히 살펴 보았듯이 55-70자리 정도의 소인수 3-4개 정도를 곱하여 합성수 m를 구한다면 현재까지 알려진 알고리즘으로는 이를 소인수분해하는 것이 완전히 불가능함을 알 수 있

다. 일반화된 NFS가 실용화된다고 하더라도 ECM 보다 항상 많은 실행시간이 요구되도록 소인수의 수를 증가시킬 수 있을 것이므로 ECM을 기준으로 소인수의 크기를 결정하면 될 것이다. 또한 합성수 m에 대해 이산대수를 계산하는 가장 빠른 방법은 먼저 m을 소인수분해한 후 각 소인수에 대해 이산대수를 계산하는 방법으로 알려져 있으므로 공격자로서는 이 m을 먼저 소인수분해하려 할 것이다. 이때 공격자가 이 방식을 깨기 위해 필요한 계산량과 센타가 각 사용자의 비밀키를 계산하는데 필요한 계산량의 차이는 대략 $L(p)^{\sqrt{2-1+o(1)}}=L(p)^{0.414+o(1)}$ 정도로 현재의 알고리즘으로는 하드웨어 등의 기술수준의 발전이 공격자보다는 시스템 설계자에게 더 유리하게 작용함을 알 수 있다.

Maurer와 Yacobi가 제시하였듯이 합성수 m를 선택하는 다른 방법으로 Pohlig-Hellman의 이산대수 계산법[34]과 Pollard의 p−1 소인수분해법[35]을 그 바탕으로 삼을 수도 있다. 즉 m을 약 100자리 정도의 두 소수 p_1, p_2의 곱으로 구성하되 p_1-1과 p_2-1을 적당히 작은 소수들로만 구성되게 하는 것이다. 이와같은 m에 대해 가장 효율적인 소인수분해법과 이산대수 계산법으로 Pollard의 p−1 방법은 실행시간이 p_i-1의 가장 큰 소수에 의존하는 반면 Pohlig-Hellman 방법은 p_i-1의 가장 큰 소수의 제곱근에 의존하게 된다. 따라서 센타의 계산량을 k배 증가시키면 공격자의 계산량은 k^2배로 증가하게 될 것이므로 기술발전에 따른 계산량의 증가는 앞에서 선택한 m 보다 훨씬 더 센타에 유리하게 작용할 것이다. 현재의 기술수준으로는 p_i-1을 대략 13−15자리 정도의 소수들로 구성하는 것이 적절한 것으로 보고 있다. 10^{15}까지의 소수의 갯수는 약 3×10^{13}개 정도이므로 위와같이 선택한 약 200(664비트) 자리 정도의 m을 Pollard의 p−1 방법으로 소인수분해하기 위해서는 200자리수에 대한 모듈라 멱승을 약 10^{12}번을 수행해야 할 것이다. 이는 초당 100번의 모듈라 멱승연산을 수행할 수 있는 특수칩을 사용한다고 가정하더라도 약 317년의 시간이 소요될 것이다. 현재의 기술수준에서 RSA 전용칩으로도 한번의 512비트 모듈라 멱승 연산에 약 30msec 정도의 시간이 걸린다는 것을 고려하면 이와같은 m을 소인수분해하는 것은 완전히 불가능함을 알 수 있다[36].

다른 방식들과 비교할때 Maurer-Yacobi 방식은 각 사용자의 비밀키를 계산하는데 필요한 센타의 계산량이 상당히 많다는 것을 단점으로 지적할 수 있다. 그러나 센타는 각 사용자가 가입시 한번씩만 이를 계산하면 되고 또한 GF(p) 상의 이산대수를 계산하는 (Coppersmith 등의) 알고리즘이 데이타베이스를 구축하는 사전계산단계(precomputation step)에서 많은 시간이 소요되는 반면 각각의 이산대수를 구하는데는 $L(p)^{1/2+o(1)}$ 정도로 훨씬 빠르게 계산되므로 적적한 크기의 p에 대해서는 충분히 실현 가능할 것이다. 필요하면 센타는 고성능의 컴퓨터를 사용하거나 병렬처리를 함으로써 보다 빠른 서비스를 제공할 수도 있을 것이다.

이제 합성수 m을 법으로 하는 정수링에서 m과 서로소인 모든 수의 모듈라 제곱은 기본원소 g를 밑으로 하는 이산대수를 갖는다는 것을 보이기로 한다. 여기서 g는 m의 인수인 각 소수에 대해 원시원소가 되도록 선택한다. GF(p) 상에서 원시원소는 $\phi(\phi(p))=\phi(p-1)$개가 존재할 것이므로 이와같은 기본원소 g를 찾는 것은 가능할 것이다. 편의상 m의 소인수 $p_i(i=1, 2, \cdots, r)$들은 $(p_i-1)/2=p_i'$가 각 i에 대해 서로 다른 소수가 되도록 선택한다. 그러면 $m=p_1\cdot p_2\cdot \cdots p_r$에 대해 $\lambda(m)=2\cdot p_1'\cdot p_2'\cdot \cdots p_r'$가 되고 g의 위수는 이 $\lambda(m)$과 같게 될 것이다. 우선 $y\equiv g^x \bmod m$를 만족하는 x를 구하는 경우를 생각하면 이 x는 다음과 같이 각 소수 p_i에 대한 r개의 이산대수를 Chinese Remainder Theorem(CRT)에 의해 결합한 결과가 될 것이다.

$$y\equiv y_1\equiv g^{x_1} \bmod p_1 \rightarrow x\equiv x_1 \bmod(p_1-1=2p_1').$$
$$y\equiv y_2\equiv g^{x_2} \bmod p_2 \rightarrow x\equiv x_2 \bmod(p_2-1=2p_2').$$
$$\vdots \qquad\qquad \vdots$$
$$y\equiv y_r\equiv g^{x_r} \bmod p_r \rightarrow x\equiv x_r \bmod(p_r-1=2p_r').$$

위의 합동식에서 각 p_i-1은 서로소가 아니므로 (2를 공통으로 갖는다) 반드시 해가 존재하지는 않으며 x_i들이 모두 짝수이거나 모두 홀수일때만 해를 갖게 된다. 이는 다음과 같은 두개의 합동식이 해를 갖기 위한 조건을 생각하면 쉽게 알 수 있다. 즉 $x\equiv x_1 \bmod p_1-1$, $x\equiv x_2 \bmod p_2-1$이 해를 갖기 위해서는 첫 번째식에서 $x=x_1+(p_1-1)t$, t은 정수,로 주어지고 이를 두번째식에 대입하면 $(p_1-1)t\equiv x_2-x_1 \bmod p_2-1$이 성립한다. 이 합동식이 해를 갖기 위해서는 $GCD(p_1-1, p_2-1)=2=GCD(x_2-x_1, p_2-1)$을 만족해야 하며 따라서 x_1, x_2가 모두 짝이거나 모두 홀수일때만 해를 갖는다는 것을 알 수 있다. 이때의 해는 $(p_1-1)(p_2-1)/2=2p_1'p_2'$를 법으로 하여 $x=x_1+(p_1-1)t$, $t\equiv(x_2-x_1)p_1'^{-1} \bmod p_2'$로 주어진다. 여기서 $p_1'^{-1}$는 법 p_2'에 대한 p_1'의 역원을 나타낸다.

즉 $p_1'^{-1}\ p_1'\equiv 1 \bmod p_2'$.

한편 $y^2\equiv g^x \bmod m$의 경우 각 소수 p_i에 대해 $y^2\equiv g^{x_i} \bmod p_i$의 해는 $x_i\equiv 2u_i \bmod (p_i-1)$로 주어진다. 여기서 u_i는 p_i를 법으로 하는 y의 이산대수, 즉 $y\equiv g^{u_i} \bmod p_i$를 만족하는 값이다. 따라서 $y^2\equiv g^x \bmod m$의 해 x는 다음 합동식들의 해가 되며 이 해가 존재 한다는 것은 쉽게 알 수 있다.

$y\equiv y_1\equiv g^{u_1} \bmod p_1 \rightarrow x\equiv 2u_1 \bmod (p_1-1=2p_1')$.
$y\equiv y_2\equiv g^{u_2} \bmod p_2 \rightarrow x\equiv 2u_2 \bmod (p_2-1=2p_2')$.
. .
. .
$y\equiv y_r\equiv g^{u_r} \bmod p_r \rightarrow x\equiv 2u_r \bmod (p_r-1=2p_r')$.

위의 합동식의 해는 CRT에 의해 $x\equiv\Sigma_{1\le i\le n}2u_i M_i N_i \bmod \lambda(m)$, $M_i=\lambda(m)/(2p_i')$, $N_i\equiv M_i^{-1} \bmod p_i'$와 같이 주어질 것이다. 따라서 각 사용자 i의 비밀키 S_i는 먼저 m의 각 소인수 p_i에 대해 $ID_i\equiv g^{u_i} \bmod p_i$를 풀어서 u_i를 구하고 $x\equiv 2u_i \bmod (p_i-1)$을 CRT에 의해 계산한 결과가 된다.

3.3 Maurer-Yacobi 방식의 문제점 및 개선방안

위에서 소개한 Maurer-Yacobi 방식은 mod m으로 재곱수에 대해서는 항상 기본원소 g를 밑으로 하는 이산대수가 존재한다는 사실에 바탕을 두어 각 사용자의 비밀키를 계산하게 되는데 이는 곧 다음과 같은 문제점을 야기시킨다. 즉 ID_i에 대해서는 이산대수가 존재하지 않으나 $ID_i^2 \bmod m$에 대해서는 이산대수가 존재한다는 사실은 곧 이 이산대수를 이용하면 m의 소인수 중의 일부를 계산할 수 있다는 것이다. 위의 비밀키 생성과정에서도 알 수 있듯이 각 사용자의 비밀키는 항상 짝수가 되며 이는 곧 많은 소인수분해 알고리즘들의 바탕이 되는 $X^2\equiv Y^2 \bmod m$ 형태의 합동식을 주게 되고, 또한 ID_i에 대해서는 이산대수가 존재하지 않으나 $ID_i^2 \bmod m$에 대해서는 이산대수가 존재한다는 사실은 곧 $X\neq\pm Y$가 성립함을 의미하게 되므로 이와같은 ID_i를 갖는 사용자는 그의 비밀키를 이용하면 쉽게 m의 소인수 중의 일부를 계산할 수 있다.

위의 사실을 계산이 간단한 작은 숫자를 예로들어 설명해 보기로 한다. 합성수 m을 $m=p_1\cdot p_2\cdot p_3=227\cdot 347\cdot 467=36785123$으로 잡으면 $p_1-1=226=2\cdot 113$, $p_2-1=346=2\cdot 173$, $p_3-1=466=2\cdot 233$이므로 $\lambda(m)=2\cdot 113\cdot 173\cdot 233=9109834$가 된다. 그리고 세 소인수 모두에 대해 $GF(p_i)$ 상에서 최소의 원시원소는 2이므로 $(Z/mZ)^*$의 기본원소로 $g=2$를 잡는다. 이제 사용자 A의 ID를 $ID_A=LCH=4C43\,48_{16}$ (16진수 ASCII 코드값)$=4213496$이라고 가정하면 이 사용자의 비밀키 $S_A=x$는 다음과 같이 계산된다.

$ID_A=4213496\equiv 149\equiv 2^{x_1} \bmod 227 \rightarrow x_1=221 \rightarrow x\equiv 221 \bmod 226$.
$ID_A=4213496\equiv 222\equiv 2^{x_2} \bmod 347 \rightarrow x_2=312 \rightarrow x\equiv 312 \bmod 346$.
$ID_A=4213496\equiv 222\equiv 2^{x_3} \bmod 467 \rightarrow x_3=104 \rightarrow x\equiv 104 \bmod 466$.

위의 x_i값들이 모두 짝수이거나 홀수가 아니므로 $g=2$를 밑으로 하는 ID_A의 이산대수는 존재하지 않는다는 것을 알 수 있다. 그러나 $ID_A'=421349 6^2 \equiv 18198772 \equiv 2^x \bmod 36785123$을 만족하는 x는 존재할 것이며 이는 다음의 합동식을 CRT로 풀어서 구할 수 있다.

$$x \equiv 2.221 \bmod 226 (=2.113)$$
$$x \equiv 2.312 \bmod 346 (=2.173)$$
$$x \equiv 2.104 \bmod 466 (=2.233)$$

따라서 $x \equiv 2.221.173.233.60+2.312.113.233.21+2.104.113.173.81 \equiv 3393154 \bmod 9109834$가 된다. 이와같이 구한 x는 $ID_A^2 \equiv 4213496^2 \equiv 18198772 \equiv 2^x \bmod 36785123$를 만족함을 확인할 수 있다. 한편 위 식은 $ID_A^2 \equiv 4213496^2 \equiv 2^{3393154} \equiv (2^{1696577})^2 \bmod 36785123$이 되고 여기서 $2^{1696577} \equiv 6805836 \bmod 36785123$이므로 결국 $4213496^2 \equiv 6805836^2 \bmod 36785123$이 성립한다. 따라서 GCD(6805836−4213496, 36785123)=GCD(2592340, 36785123)=227로 m의 소인수 중의 하나를 구할 수 있게 된다.

다른 사용자 B의 경우를 하나 더 예로 들어 보자. 사용자 B의 ID를 $ID_B=LPJ=4C504A_{16}=5001290$이라고 하면 다찬가지 방법으로 $x \equiv 2.62.173.233.60+2.268.113.233.21+2.425.113.173.81 \equiv 181292 \bmod 9109834$와 같이 계산되고 $ID_B^2 \equiv 5001290^2 \equiv 11222421 \equiv 2^{1812192} \equiv (2^{906096})^2 \equiv 31467674^2 \bmod 36785123$로부터 GCD(31467674−5001290, 36785123)=78769(=227.347)을 얻는다.

다음에는 각 소수에 대한 ID의 이산대수가 모두 짝수이거나 모두 홀수인 경우를 살펴보자. 사용자 C의 ID를 $ID_C=CHO=43484F_{16}=4409423$이라고 하면 각 소수 p_i에 대한 이산대수는 다음과 같이 모두 짝수로 주어진다.

$$ID_C=4409423 \equiv 175 \equiv 2^{x_1} \bmod 227 \rightarrow x_1=176 \rightarrow x \equiv 176 \bmod 226.$$
$$ID_C=4409423 \equiv 94 \equiv 2^{x_2} \bmod 347 \rightarrow x_2=190 \rightarrow x \equiv 190 \bmod 346.$$
$$ID_C=4409423 \equiv 9 \equiv 2^{x_3} \bmod 467 \rightarrow x_3=434 \rightarrow x \equiv 434 \bmod 466.$$

따라서 이때는 ID_C 자체에 대한 이산대수, 즉 $ID_C \equiv 4409423 \equiv 2^x \bmod 36785123$을 만족하는 x가 존재하며 $x/2 \equiv 88.173.233.60+95.113.233.21+217.113.173.81 \equiv 3166687 \bmod 4554917 (=113.173.233)$로부터 $x \equiv 6333374 \bmod 9109834$과 같이 계산된다. 이 경우 $ID_C^2 \equiv 4409423^2 \equiv 36720541 \equiv 2^x \bmod 36785123$을 만족하는 x를 앞의 두 예에서처럼 계산한다면 $x \equiv 3556914 \equiv 2.6333374 \bmod 9109834$가 될 것이고 앞의 두 예에서처럼 m에 대한 소인수의 일부를 얻는 것은 불가능하다. 즉 이 경우는 사용자 C는 비밀키로 $S_C=3556914$을 받게 되며 이로부터 $2^{S_C/2} \equiv 3237500 \equiv -ID_C \bmod 36785123$와 같이 계산하더라도 ID_C^2에 대한 두 해 $\pm ID_C$ 중의 하나를 얻을 뿐이다.

마찬가지로 각 소수 p_i에 대한 이산대수가 모두 홀수인 사용자 D를 예로들어보면 $ID_D=KDJ=4B444A_{16} \equiv 4932682$일때 $ID_D^2 \equiv 4932682^2 \equiv 16030389 \equiv 2^x \bmod 36785123$을 만족하는 x는 $x \equiv 5462732 \bmod 9109834$로 주어지며 $2^{S_D/2} \equiv 2^{2731366} \equiv 31852441 \equiv -ID_D \bmod 36785123$이 됨을 알 수 있다.

위의 예들에서 볼 수 있듯이 각 소인수에 대한 ID의 이산대수가 모두 짝수이거나 홀수가 아닌 경우에는 언제나 m의 인수 중의 일부를 계산할 수 있을 것이므로 이 방식을 그대로 사용할 수는 없을 것이다. 각 사용자들이 세션키를 계산하는데 기본원소 g는 필요없으므로 g를 비밀로 할 수도 있으나 이 경우 임의의 두 사용자 i, j가 결탁하였을때 그들의 비밀키가 $GCD(S_i, S_j)=2$를 만족한다면 언제나 유클리드 공격법(Euclidean attack)에 의해 g^2을 계산할 수 있으므로 효과가 없을 것이다. 즉 GCD

$(S_i, S_j)=2$이면 확장 유클리드 알고리즘(extended Euclidean algorithm)에 의하면 $S_iu+S_jv=2$를 만족하는 u, v를 구할 수 있고 따라서 $(ID_i^2)^u \cdot (ID_j^2)^v \equiv g^2 \bmod m$와 같이 g^2을 구할 수 있기 때문이다.

이와같은 문제점을 간단히 해결할 수 있는 방법으로 센타는 위와같이 계산한 각 사용자 i의 비밀키 $X_i(ID_i^2 \equiv g^{X_i} \bmod m)$를 그대로 각 사용자에게 전달할 것이 아니라 센타 자신만의 비밀 랜덤수 $S(GCD(S, \lambda(m))=1)$를 선택하여 $S_i \equiv S.X_i \bmod \lambda(m)$을 각 사용자의 비밀키로 전해 주는 것이다. 그러면 이를 받은 각 사용자는 i는 S나 X_i를 절대로 알 수 없으며 이는 다수의 사용자가 결탁하는 경우도 마찬가지이다. 따라서 위의 예에서와 같이 자신의 비밀키로부터 $X^2 \equiv Y^2 \bmod m (X \neq \pm Y)$ 형태의 합동식을 얻는 것은 불가능하므로 위에서 제기된 문제점을 완전히 극복할 수 있다. 그러나 각 사용자의 비밀키를 이와같이 계산하더라도 두 사용자 i, j는 원래의 방식에서와 마찬가지 방법으로 그들간의 세션키를 계산할 수 있다. 즉 $K_i \equiv (ID_j^2)^{S_i} \equiv g^{X_j S_i} \equiv g^{X_j X_i S} \equiv g^{X_i X_j S} \equiv (ID_i^2)^{S_j} \equiv K_j \bmod m$이 되므로 S를 전혀 모르더라도 공통의 비밀키를 계산할 수 있다.

위와같이 변형된 방식의 단점은 각 사용자의 비밀키를 그의 ID와 관련시켜 주는 관계식이 존재하지 않는다는 점이다. 즉 만일 센타가 $P \equiv g^{s^{-1}} \bmod m$을 센타의 공개키로 모든 사용자에게 공개한다면 사용자 i의 비밀키 S_i에 대해 $P^{S_i} \equiv g^{X_i} \equiv ID_i^2 \bmod m$이 성립할 것이나 이 경우는 원래의 방식에서 존재하던 문제점이 다시 제기되므로 이와같은 P를 공개할 수는 없다. 결국 변형된 방식은 두 사용자간의 세션키 분배용으로는 안전하게 사용할 수 있으나 한 사용자에 대한 비밀키로서의 역할을 한다고 보기는 어렵다.

한편 모든 사용자들의 비밀키는 항상 짝수이므로 $S.X_i$와 $\lambda(m)$은 모두 짝수이므로 $S_i \equiv S.X_i \bmod \lambda(m)$ 역시 짝수가 된다. 다음과 같이 두 사용자간의 세션키 계산시 상대방의 ID 자체에 자신의 비밀키로 멱승을 하더라도 공통의 세션키를 얻을 수 있다. 즉

$$K_i \equiv ID_j^{S_i} \equiv (ID_j^2)^{S_i/2} \equiv g^{X_j S_i/2} \equiv g^{X_j S X_i/2} \equiv g^{X_i S_j/2} \equiv (ID_i^2)^{S_j/2} \equiv ID_i^{S_j} \equiv K_j \bmod m.$$

위에서 구한 두 사용자 A, B간의 세션키 계산결과로 이를 확인해 보자. 우선 센타의 비밀 랜덤수 S를 $S=7054364$로 선택하면 사용자 A, B의 비밀키는 각각 $S_A \equiv S.X_A \equiv 7054364.3393154 \bmod 9109834 \equiv 8207190$, $S_B \equiv S.X_B \equiv 7054364.1812192 \bmod 9109834 \equiv 4624186$이 된다. 이제 두 사용자간의 세션키는 $ID_B^{S_A} \equiv 5001290^{8207190} \equiv 13706942 \equiv 4213496^{4624186} \equiv ID_A^{S_B}$로 동일함을 확인할 수 있다. 이는 3.1절의 비밀키 계산과정에서 알 수 있듯이 g 대신에 g^2을 기본원소로 사용한다면 항상 원래의 비밀키의 1/2에 해당하는 값을 비밀키로 계산하게 된다는 사실에 기인한다.

한편 Maurer와 Yacobi는 법 m이 두 소수의 곱일 때는 사용자의 ID를 변형하여 그에 대응하는 이산대수가 항상 존재하도록 하는 다른 방법을 제시하였다. 즉 $m=p_1p_2$이고 $GCD(p_1-1, p_2-1)=2$인 경우 임의의 수 x에 대해 Jacobi symbol(x/m)가 1일 때만 m을 법으로 하는 x의 이산대수가 존재한다는 사실을 이용한다. Jacobi symbol(x/m)는 $(x/p_1).(x/p_2)$로 쓸 수 있고 $(x/m)=1$인 경우는 $(x/p_1)=(x/p_2)=1$이거나 $(x/p_1)=(x/p_2)=-1$인 경우이다. 또한 소수 p_i에 대한 Jacobi symbol(곧 Legendre symbol) $(x/p_i)=1$인 경우 (곧 x가 $GF(p_i)$에서 quadratic residue인 경우)는 $GF(p_i)$에서 x에 대한 이산대수가 짝수인 경우이며 $(x/p_i)=-1$인 경우는 그 이산대수가 홀수인 경우이다. 따라서 $GF(p_1)$과 $GF(p_2)$에서 x에 대한 이산대수가 모두 짝수이거나 모두 홀수일 때만 Jacobi symbol(x/m)는 1의 값을 갖게 된다. 앞에서 살펴 보았듯이 m의 각 소인수를 법으로 하는 이산대수가 모두 짝수이거나 모두 홀수일

때는 항상 m을 법으로 하는 이산대수가 존재하게 되므로 곧 (x/m)=1일 때는 항상 x에 대한 이산대수가 존재하게 된다. Maurer와 Yacobi는 이와 같은 사실을 이용하여 사용자 i의 공개키를 그의 ID_i 보다 크거나 같은 수로서 (x/m)=1이 되는 가장 작은 정수 x로 정의하였다. Jacobi symbol(x/m)는 m의 소인수를 모르더라도 누구나 쉽게 계산할 수 있으므로 센타에서 이와같은 x에 대한 이산대수를 각 사용자의 비밀키로 계산하여 주면 각 사용자들은 상대방의 공개키를 상대방의 ID를 이용하면 쉽게 계산할 수 있을 것이다.

이 방식에서는 앞에서 살펴본 사용자의 ID의 제곱에 대한 이산대수를 계산하던 방식에서 존재하던 문제점은 존재하지 않게 되나 상대방의 공개키를 얻기 위해서는 상대방의 ID 값에서 시작하여 Jacobi symbol이 1이 되는 수 x를 찾아야 하므로 사용자의 ID에 따라서는 여러번의 Jacobi symbol을 계산해야 할 것이다. 그러나 여기서도 단지 한번의 Jacobi symbol만을 계산하면 상대방의 공개키를 결정할 수 있도록 간단히 변형시키는 것이 가능하다. 즉 센타에서 m의 두 소인수 p_1, p_2를 선택할 때 3.2절에서 설명했듯이 p_i-1이 13-15 자리의 소수들을 포함하도록 하되 (2/m)=-1이 되도록 $p_1 \equiv \pm 1 \bmod 8$, $p_2 \equiv \pm 3 \bmod 8$을 만족하도록 한다. 여기서 소수 p에 대해 Legendre symbol $(2/p)$는 $(2/p)=(-1)^{(p^2-1)/8}$임을 상기하면 (2/m)=-1이 됨을 쉽게 알 수 있다. 그러면 사용자 i의 공개키를 결정할 때 단지 Jacobi symbol (ID_i/m)를 계산하여 그 값이 1이면 ID_i가 그의 공개키가 되며 그렇지 않을 때는 $(2ID_i/m)$가 항상 1이 될 것이므로 $2ID_i$가 사용자 i의 공개키가 될 것이다. 따라서 센타에서 각 사용자 i의 비밀키를 계산할 때 (ID_i/m)의 값에 따라 ID_i나 $2ID_i$에 대한 이산대수를 계산하여 주면 임의의 두 사용자들이 그들간의 세션키를 계산할 때에도 단지 한번의 Jacobi symbol 계산만으로 상대방의 공개키를 얻을 수 있게 될 것이므로 계산량을 줄일 수 있을 것이다. 물론 여기서 한 사용자의 ID 값에 2를 곱한 것(즉 ID를 한 비트 왼쪽으로 쉬프트시킨 값)이 다른 사용자의 ID와 일치하지 않아야 한다는 가정을 전제해야 하지만 이는 사용자의 ID의 비트 길이를 법 m의 비트길이보다 최소한 한 비트 정도 작게 제한한다면 $2ID_i$가 다른 ID_j와 mod m으로 일치할 확률은 거의 없을 것이므로 무리없이 가정할 수 있을 것이다.

마찬가지 방법으로 법 m의 소인수 p_1, p_2를 $p_1 \equiv 1 \bmod 4$, $p_2 \equiv 3 \bmod 4$가 되도록 선택한다면 Jacobi symbol$(-1/m)$은 -1이 될 것이므로 사용자 i의 공개키는 (ID_i/m)의 값에 따라 ID_i나 $-ID_i$가 될 것이다. 즉 만일 (ID_i/m)의 값이 1이면 그의 공개키는 ID_i가 될 것이고 그렇지 않으면 $(-ID_i/m)$의 값이 항상 1이므로 $-ID_i$ (즉 m-ID_i)가 공개키가 될 것이다. 이 경우는 어떤 두 사용자의 ID를 합한 값이 법 m과 같지 않아야 한다는 조건을 전제해야 하며 이런 가능성은 거의 무시할 수 있을 것이다. 이 방식에서는 $g^{s_i} \equiv ID_i \bmod m$과 같이 사용자의 비밀키와 ID 사이에 분명한 관계식이 성립할 뿐더러 ID의 제곱에 대한 이산대수를 계산하던 방식에서 발생하던 문제점도 존재하지 않으므로 Maurer와 Yacobi의 원래 주장과는 달리 이 방식이 보다 나은 방식이라 할 수 있다.

이상에서 Maurer-Yacobi 방식을 분석하여 그 문제점을 제기하고 이를 해결할 수 있는 개선방안을 제시하였다. 이 방식은 사용자의 계산량이나 메모리 사용량 등에 있어서 지금까지 제안된 어떤 방식보다도 효율적이므로 비대화형 키 분배방식으로 매우 유용하게 사용될 것으로 생각된다. 단지 센타에서 각 사용자의 시스템 가입시 그의 비밀키를 계산하여 발급하는데 상당한 시간이 걸린다는 것이 단점으로 지적될 수 있으나 사용자의 측면에서는 세션키 공유에 단지 한번의 모듈라 멱승이면 충분하므로 매우 효율적이라 할 수 있다.

3.4 랜덤한 세션키의 생성

모든 비대화형의 키 분배방식은 항상 동일한 세션키를 초래하므로 이를 직접메세지의 암호에 사

용하는 것은 바람직하지 않을 것이다. 대신 위에서 계산한 세션키를 두 사용자간의 master key M으로 하여 $M^R \equiv g^{S_i S_j R}$을 실제로 메세지의 암호에 사용되는 세션키로 사용할 수 있다. 여기서 사용된 랜덤수 R은 암호화된 메세지와 함께 전송할 수도 있으나 과거의 세션키가 노출되었을때 공격자에 의한 세션키의 재사용을 방지할 수 있도록 timestamp T를 전송하고 이 T에 의존하는 랜덤수를 사용하는 편이 보다 안전할 것이다. 물론 T에 의존하는 랜덤수를 발생시키는 방법은 모든 사용자들에게 알려져 있어야 할 것이다. 비효율적이기는 하나 $R = g^T \pmod m \oplus T$와 같이 R을 발생시킬 수도 있을 것이며 보다 효율적인 방법으로 공개된 의사난수 발생기(pseudorandom number generator)에서 T를 seed로 사용하는 방법을 들 수 있다.

한편 개선된 Maurer-Yacobi 방식에서 세개 이상의 소수들의 곱으로 구성된 합성수 m에 대해 λ(m)을 공개했을 때의 그 안전성을 살펴보자. 3.1절에서 언급했듯이 $m = p_1 p_2 \cdots p_r$과 같이 m을 55-70자리 정도의 소수 p_i들로 구성하되 $(p_i - 1)/2 = p_i'$ 역시 소수가 되도록 한다면 $\lambda(m) = 2\, p_1' p_2' \cdots p_r'$이 알려지더라도 λ(m)을 소인수분해하는 것은 여전히 불가능하며 또한 m과 λ(m)으로부터 m의 소인수를 구하는 것도 역시 불가능하다. $ID_i^{x_i} \equiv g^{X_i} \bmod m$을 만족하는 X_i를 구하기 위해서는 m의 각 소인수들을 법으로 하는 이산대수를 계산할 수 있어야 하므로 이 방식의 비밀정보는 λ(m)이 아니라 m의 소인수들이다. 또한 위에서 설명하였듯이 원래의 Maurer-Yacobi 방식의 결함을 극복하기 위해 사용자의 비밀키 생성을 $ID_i^{x_i} \equiv g^{X_i} \bmod m$을 만족하는 X_i와 센타의 비밀랜덤수 S를 이용하여 $S_i = X_i \cdot S \bmod \lambda(m)$과 같이 계산하도록 변형시켰는데 여기서도 λ(m)의 공개가 X_i나 S를 구하는데 아무런 도움을 주지 않는다는 것을 알 수 있다. 따라서 m의 이 두 소수의 곱으로 구성된 경우와는 달리 이와같은 경우에는 λ(m)을 공개하더라도 그 안전성이 전혀 위협받지 않음을 알 수 있다.

이 사실을 이용하면 위의 세션키의 계산도 한번의 모듈라 멱승만으로 계산할 수 있을 것이다. 즉 $KS \equiv M^R \equiv ID_j^{S_i R} \bmod m$의 계산에서 지수부분을 λ(m)을 법으로 reduction 시킨 다음 모듈라 멱승을 수행하는 것이다.

세션키를 얻는 다른 방법으로 두 사용자간의 공유키 $M \equiv ID_j^{S_i} \equiv g^{S_i S_j}$를 계산한 다음 전송정보로 timestamp T를 이용하여 $Z \equiv (M.\ T \bmod m) \oplus R$와 같이 계산한 Z를 T와 함께 전송하는 것이다. 여기서 R은 사용자 i가 랜덤하게 선택한 비밀 수이며(Z, T)는 $M^R = g^{S_i S_j R} \bmod m$을 세션키로 하여 암호화된 메세지와 함께 전송하면 될 것이다. 그러면 사용자 j 역시 공유키 M을 계산할 수 있으므로 전송정보 (Z, T)로부터 비밀랜덤수 R을 얻을 수 있고 따라서 세션키 M^R을 계산하여 수신된 암호문을 복호화할 수 있을 것이다. 그러나 이 방법에서는 세션키 계산을 위해 두번의 모듈라 멱승이 필요함을 알 수 있다.

마지막으로 두 소수의 곱으로 구성된 m에 대해 $g^{S_i} \equiv ID_i \bmod m$이 성립하는 두번째 방식에서 랜덤한 세션키를 계산하는 다른 예를 들어본다. 우선 사용자 i는 랜덤한 비밀 수 ρ_i를 선택하면 $Z_i \equiv g^{\rho_i} \bmod m$을 계산하고 이 Z_i와 timestamp T, 그리고 자신의 ID 및 상대방 j의 ID를 이용하여 공개된 일방성 해쉬함수 h로 $C = h(Z_i,\ T,\ ID_i,\ ID_j)$를 계산한다. 그리고 Schnorr의 서명에서처럼 $R_i = \rho_i + CS_i$을 구해 $T \equiv ID_j^{R_i} \bmod m$을 계산한 다음 $K_{ij} = h(T)$를 사용자 j와의 세션키로 사용하여 메세지 m을 암호화한 후 이 암호문과 함께 Z_i, T를 전송한다. 여기서 각 사용자는 λ(m)을 알지못하므로 R_i는 단순한 산술연산에 의해 계산되고 따라서 이는 법 m의 비트길이보다 대략 C의 비트길이만큼 큰 수가 될 것이다. 그러면 이를 받은 사용자 j는 마찬가지 방법으로 C를 계산한 후 $(ID_C,\ Z_i)^{S_j} \equiv g^{S_i S_j} \equiv T \bmod m$을 얻어 세션키 $K_{ij} = h(T)$를 계산할 수 있을 것이다. 여기서 세션키는 일방성 해쉬함수의 결과이므로 세션키가 알려진다 하더라도 이로부터 아무런 정보도 얻을 수 없을 것이다.

4. 결 론

본 논문에서는 개인정보에 기초한 키 분배방식으로 기존의 대표적인 방식들을 분석하고 안전성이나 효율성을 향상시킬 수 있는 개선방안들을 제시하였다. Gunther의 방식에서 Schnorr의 서명을 사용함으로써 한번의 모듈라 멱승 연산보다 더 적은 계산량으로 세션키를 계산할 수 있음을 보였고 또한 최근에 발표된 Maurer-Yacobi 방식을 분석하여 안전성에 문제가 있음을 밝혀내고 이를 해결할 수 있는 개선방안도 제시하였다. 특히 Maurer-Yacobi 방식과 같은 안전한 비대화형의 키 분배방식은 사전통신이 필요없이 센타의 공개정보와 상대방의 ID만으로 세션키 계산이 가능하므로 E-mail과 같은 일방향의 비밀통신에 특히 유용하며 다른 일반적인 응용에서도 통신비용을 줄일 수 있으므로 매우 효율적으로 사용될 수 있을 것이다.

참 고 문 헌

1. W.F.Ehrsam, S.M. Matyas, C.H. Meyer, and W.L. Tuchman, "A cryptographic key management scheme for implementing the Data Encryption Standards," IBM Systems J., 17, No. 2, 1978, pp. 106-125.

2. "Banking-key management(wholesale)," International Standard ISO 8732, International Organization for Standardization, Geneva, 1988.

3. W. Diffie and M.E. Hellman, "New directions in cryptography," IEEE Trans. Inform. Theory, IT-22, 6, 1976, pp. 644-654.

4. K.S. McCurley, "A key distribution system equivalent to factoring," J. Cryptology, Vol. 1, No. 2, 1988, pp. 95-106.

5. J.A. Buchman and H.C. Williams, "A key-exchange system based on imaginary quadratic fields," J. Cryptology 1, 1988, pp. 107-118.

6. V. Miller, "Use of elliptic curves in cryptography," Proc. Crypto'85 : Advances in Cryptology, Lecture notes in Computer Science 218, Springer-Verlag, 1986, pp. 417-426.

7. T. Yamamoto and R. Akiyama, "A data encryption device incorporating fast PKDS," Proc. IEEE Global Telecom. Conf., Nov-Dec. 1983, pp. 1085-1090.

8. T. Matsumoto, Y. Takashima, and H. Imai, "On seeking smart public key distribution systems," Trans. IECE Japan, Vol. E. 69, No. 2, 1986, pp. 99-106.

9. 이필중, 임채훈, "일반화된 Diffie-Hellman키 분배방식의 안전성 분석," 한국통신학회논문지 제16권 7호, 7/91, pp. 575-597.

10. 임채훈, 이필중, "효율적인 키 분배 및 암호시스템의 제안 : 제 I 부 제 II 부," '91 정보보호학술발표회논문집, 11/91, pp. 19-39.

11. L. Kohnfelder, "Towards a practical public key cryptosystem," B.S. thesis, MIT, Cambridge, MA, 1978.

12. M. Girault, "Self-certified public keys," Proc. Eurocrypt'91.

13. A. Shamir, "Identity-based cryptosystems and signature schemes," Proc. Crypto'84 : Advances in Cryptology, Leture notes in Computer Science 196, Springer-Verlag, 1985, pp. 47-53.

14. E. Okamoto and K. Tanaka, "Key distribution system based on identfication information," IEEE J. Selected areas in comm., Vol. 17, No. 4, 1989, pp. 481-485.

15. C.G. Gunther, "An identity-based key-exchange protocol," Proc. Eurocrypt'89.

16. F. Bauspieß and H. Knobloch, "How to keep authenticity alive in a computer network," Proc. Eurocrypt'89 : Advances in Cryptology, Lecture notes in Computer Science 434, Springer-Verlag, 1990, pp. 29-37.

17. S. Tsujii and T. Itoh, "ID-based cryptosystem using discrete logarithm problem," Electronic Letters, Vol. 23, No. 24, 1987, pp. 1318-1320.

18. S. Tsujii and T. Itoh, "An ID-based cryptosystem based on the discrete logarithm problem," IEEE J. Selected areas in com., Vol. 7, No. 4, 1989, pp. 467-473.

19. H. Tanaka, "A realization scheme for the identity-based cryptosystem," Proc. Crypto'87 : Advances in Cryptology, Lecture notes in Computer Science 293, Springer-Verlag, 1988, pp. 340-349.

20. T. Matsumoto and H. Imai, "On the security of some key sharing schemes," the 1990 Symp. Cryptography and Infom. Security, Japan, 1990, pp. 1-6(in Japanese).

21. U. M. Maurer and Y. Yacobi, "Non-interactive public key cryptography," Proc. Eurocrypt'91.

22. C. P. Schnorr, "Effcient identification and signatures for smart cards," Proc. Crypto'89 : Advances in Cryptology, Lecture notes in Computer Science 435, Springer-Verlag, 1990, pp. 239-251.

23. J. M. Pollard, "Monte Carlo methods for index computation mod p," Math. Comp. 32, 1978, pp. 918-924.

24. T. Matsumoto, Y. Takashima, and H. Imai, "On seeking smart public key distribution systems," Trans. IECE Japan, Vol. E. 69, No. 2, 1986, pp. 99-106.

25. R. Blom, "An optimal class of symmetric key generation systems," Proc. Eucrypt'84 : Advances in Cryptology, Lecture notes in Computer Science 209, Springer-Verlag, 1985, pp. 335-338.

26. T. Matsumoto and H. Imai, "On key predistribution system," Proc. Crypto'87 : Advances in Cryptology, Lecture notes in Computer Science 293, Springer-Verlag, 1988, pp. 185-193.

27. S. Tsujii and J. Chao, "A new ID-based key sharing system," Proc. Crypto'91.

28. R. C. Merkle and M. E. Hellman, "Hiding information and signatures in trap-door knapsacks," IEEE Trans. Inform. Theory, IT-24, No. 5, 1978, pp. 525-530.

29. D. Coppersmith, A. M. Odlyzko, and R. Schroeppel, "Discrete logarithms in GF)(p)," Algorithmica, Vol. 1, 1986, pp. 1-15.

30. A. K. Lenstra, H. W. Lenstra, M. S. Manasse, and J. M. Pollard, "The number field sieve," Proc. ACM Symp. on Theory of Computing (STOC), 1990, pp. 564-572.

31. R. D. Silverman, "The multiple polynomial quadratic sieve," Math. Comp., Vol. 18, 1987, pp. 329-339.

32. A. K. Lenstra and M. S. Manasse, "Factoring with two large primes," Proc. Eurocrypt'90.

33. H. W. Lenstra, "Factoring integers with elliptic curves," Annals Math., Vol. 126, 1987, pp. 649-673.

34. S. C. Pohlig and M. E. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance," IEEE Trans. Inform. Theroy, IT-24, No. 1, 1978, pp. 106-110.

35. J. M. Pollard, "Theorems on factorization and primality testing," Proc. Cambridge Philos. Soc., Vol. 76, 1974, pp. 521-528.

36. E. F. Brickell, "A Survey of hardware implementations of RSA," Proc. Crypto'89 : Advances in Cryptology, Lecture notes in Computer Science 435, Springer-Verlag, 1990, pp. 368-390.

□ 著者紹介

林 采 薰(正會員)

1983年 3月 서울大學校 電子工學科 學士
1989年 2月 韓國데이타通信(株) 技術本部 勤務
1992年 2月 浦項工科大學 電子電氣工學科 碩士
現在 : 浦項工科大學 電子電氣工學科 博士過程 在學中

李 弼 中(正會員)

1951年 12月 30日生
1974年 2月 서울大學校 電子工學科 學士
1977年 2月 서울大學校 電子工學科 碩士
1982年 6月 U.C.L.A. System Science, Engineer
1985年 6月 U.C.L.A. Electrical Engineering, Ph.D,
1980年 6月～1985年 8月 : Jet Propulsion Laboratory, Senior Engineer
1985年 8月～1990年 2月 : Bell Communications Research, M.T.S.
1990年 2月～現在 : 浦項工科大學 電子電氣工學科, 副教授