情 報 保 護 學 會 誌
第 14 卷 第 6 號, 2004. 12

익명 통신로에 관한 최근 연구 동향

이 현 숙*, 변 진 욱*, 박 현 아*, 이 동 훈*, 임 종 인*

요 약

익명 통신로는 사용자가 자신의 정보를 임의의 통신로에 보내었을 경우에 제 3자가 사용자와 사용자가 보낸 메시지의 대응관계를 알 수 없도록 하는 기술이다. 이러한 익명 통신로의 기술은 전자선거, 전자화폐, 전자우편 등의 프라이버시의 기능을 제공해야 하는 분야에서 널리 활용된다. 본 논문에서는 익명 통신로에 관한 최신 연구 흐름을 Mix-net과 DC(Dining Cryptographers)-net 중심으로 살펴본다. 또한 향후 연구 방향에 대해서 소개한다.

Ⅰ. 서 론

사회 활동 전반에 걸쳐 정보통신망 및 컴퓨터를 이용한 개인정보의 수집 및 이용이 일상화됨에 따라 개인정보 침해 유형이 정보통신서비스 이외에 의료, 금융, 보험, 노동, 지적 재산권, 통신사업등 대부분 영역에 걸쳐 확대되고 있다. 이러한 개인정보 침해 기술 (PIT : privacy infringement technology)은 정보통신 기술의 발달과 함께 합법적인 범위 내에서 점점 지능화 고도화 되어 가고 있다.

PIT는 사용자의 프라이버시 침해와 직결되므로 이를 방어하는 프라이버시 보호 기술(PET : privacy-enhancing technologoy)을 개발하는 것은 대단히 중요하다. 일반적으로 PET는 네트워크 기반, 웹 기반, 그리고 에이전트 기반 기술들로 나누어진다. 이러한 기술들 중에서 가장 중요하게 다루는 부분이 사용자의 익명성 서비스 제공이다. 뿐만 아니라, 익명성은 전자 선거, 전자화폐, 전자우편 등의 암호 프로토콜 분야에서도 반드시 만족되어져야 하는 중요한 서비스이다. 그러므로 이를 구현한 익명 통신로에 대한 연구는 반드시 필요하다.

익명 통신로는 사용자가 자신의 정보를 임의의 통신로에 보내었을 경우에 제 3자가 사용자와 사용자가 보낸 메시지의 대응관계를 알 수 없도록 하는 기술을 의미한다. 기본적인 익명통신로는 송신자 익명성만 제공한다. 물론 양방향 통신 환경에서 송수신자의 익명성을 동시에 제공할 수도 있다[4].

이러한 익명 통신로에 대한 최초의 연구는 D. Chaum이 제안한 *Mix*형의 익명 통신로 [4] 이며 이는 송신자가 *Mix-net*을 구성해 전송하고자 하는 메시지들을 *Mix*들의 공개키들로 암호화하고, 그러한 암호문들은 *Mix* 서버들을 거치면서 복호화 되고, 재배열되어 수신자에게 익명으로 전송된다. 여기서 적어도 하나의 *Mix* 서버가 정직하다면 전체 *Mix* 네트워크에서의 송신자의 익명성은 보장된다. Chaum이 제안한 방식은 송신자가 메시지를 전송하기 위해서 구성된 모든 *Mix*들의 공개키 들이 필요하고, 송신자의 계산량 또한 크다는 단점이 있다. 그 이후로 많은 익명 통신로에 대한 기술과 공격이 제안되어왔다.

익명 통신로에 대한 또 다른 연구방향으로는 *DC-net* [3] 이 있다. *DC-net*은 non-interacitve 성질을 만족하는 익명성 보장 프로토콜이다. 즉, 사용자들은 키 교환이 끝난 후, 오직 한 번의 동보전송으로 원하는 메시지를 안전하게 전달할 수 있다. 하지만, *DC-net*을 사용하는 사용자가 *DC-net*에 입력 값을 동시에 직접 제공해야 하므로 *Mix-net*과는 다르게 중간에 *Mix* 서버를 둘 필요가 없다. 이러한 특성은 단점으로 작용한다. 왜냐하면, 여러 개의 *Mix*서버는 라우터상에서 구현하기가 용이하지만, 중간 대리인을 두지 않는 *DC-net*은 현실적인 TCP/IP 상황에서 적용되기 힘든 특성을 가지기 때문이다. 또한 정당한 참여자가 얼마든지 자신의 실체를 숨기며 전달되는 메시지를 막거나 변조 하여 전체적인 *DC-*

* 고려대학교 정보보호대학원({math33, byunstar, kokokzi}@cist.korea.ac.kr, {donghlee, jilim}@korea.ac.kr)

net 기능을 파괴할 수 있다. 이러한 것들을 막기 위해 복수의 라운드가 추가로 필요하게 된다. 이에 필요한 비용이 만만치 않으므로 현실적으로 DC-*net*에 대한 연구가 소홀히 이루어 졌다.

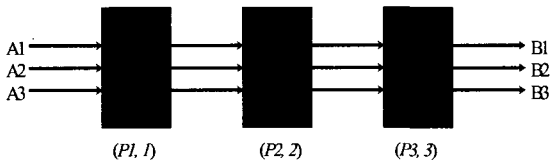
본문에서는 *Mix-net*과 *DC-net*의 기본적인 내용과 최근 연구동향을 소개하고, 앞으로 연구되어져야 할 부분에 관해서 소개하고자 한다.

II. Mix-Net에 관한 연구동향

1. Chaum의 *Mix-net*

익명성을 제공하는 통신 채널의 전형적인 실행 방법으로 Chaum이 최초로 제안한 *Mix-net*이있다[4].

Chaum이 제안한 *Mix-net*은 송신자가 메시지를 전송하기 위해서 구성된 모든 *Mix*들의 공개키 들이 필요하고, 송신자의 계산량 또한 크게 되는 단점이 존재한다. 다음은 Chaum이 제안한 *Mix-net*이다.



〔그림 1〕 Mix net

우선 k개의 *Mix*서버를 가정하다. n명의 송신자를 A_1, ..., A_n이라고 할 때, 송신자 A_i가 자신과 자신의 메시지 m_i와의 관계를 숨기고 메시지를 전송하고자 한다. 수신자 B_i의 공개키를 E_{B_i}라 하고 *Mix* 서버 S_i의 공개키 및 개인키를 (Pi, i)이라고 한다. 여기서 서버 S_i의 역할은 각 송신자의 암호문을 복호화 하여, 난수 성분을 제거한 후 그 결과들의 순서를 랜덤하게 바꾸어 출력하는 것이다.

설명의 편의를 위해, *Mix* 서버가 3개 있고 사용자 A1이 B1에게 메시지 M_i를 전달한다고 가정하자.

〔1단계〕 각 송신자 A_i는 k개의 난수 R_1, R_2, R_3를 발생하여, 다음과 같은 암호문을 계산한 후 처음의 *Mix* 서버에게 전달한다.

$$E_{P1}(R_1, Adr_2, E_2(R_2, Adr_3, E_3(R_3, Adr_{B1}, E_{Bi}(M_i))))$$

〔2단계〕 최초의 *Mix* 서버 S_1는 수신한 암호문들을 복호화하고, 그 내용 중에서 난수 R_1을 제거한다. 난수 R_1의 의미는 확률적 공개키 사용을 의미한다. 복호화 한 내용 중 다음에 보내야할 *Mix* 서버의 주소를 얻게 되고 그 주소로 나머지 암호화된 부분을 전달한다.

$$E_2(R_2, Adr_3, E_3(R_3, Adr_{B1}, E_{Bi}(M_i)))$$

〔3단계〕 나머지 *Mix*서버들은 차례로 **단계 2**와 같은 동작을 반복 수행한다.

〔4단계〕 마지막으로, S_3는 $\{E_{Bi}(M_i)\}$를 사용자 Bi에게 전달하고 Bi는 자신의 개인키로 복호화 하여 메시지를 얻게 된다.

2. *Mix-net*의 연구 흐름

Mix-net 연구는 크게 두 가지로 나뉜다. 하나는 *Mix* 서버의 메시지 포워딩 전략에 관련된 내용이고[5] 또 하나는 *Mix-net*의 기능적 사항의 추가에 관한 내용이다[5]. 2000년대 까지는 *Mix* 서버의 메시지 포워딩 전략을 나름대로 제시하고 이에 대한 메시지 연관성 및 익명성을 계산하는 연구들이 많이 이루어졌다. 최근에는 *Mix-net*의 다양한 기능 추가들에 대한 연구가 이루어지고 있다.

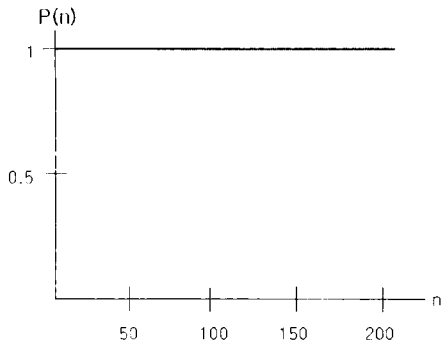
2.1 메시지 포워딩 전략에 관한 연구

Mix 서버의 주된 목적은 들어오고 나가는 메시지들의 연관성을 숨기는 것이다. 따라서 *Mix*는 어떤 수의 메시지들을 모으는 프락시의 역할을 하여 그 메시지들을 섞어서 포워딩하는데 그 섞는 방법은 여러 가지가 있다. 이러한 섞는 전략을 *Batching strategies*라 한다. 이는 섞여질 메시지를 수집하고 그것들을 다음 *Mix*로 포워딩하는 알고리즘을 말하며, 크게 다음 4가지 종류가 있다 [5,16]

o 표기법
- n : *Mix*내의 총 메시지 개수
- N_p : *Mix*가 항상 보유하고 있는 메시지 개수
- N : 새로 도착한 메시지 개수 즉, n-N_p
- $P(n)$: *Mix*내 총 메시지 개수가 n일 때, 메시지를 내보 낼 확률로 1-N_p/n 와 같다.

o Timed Mix

*Mix*는 주기적으로 내보낼 시기(flushing time)에



〔그림 2〕 Timed *Mix*

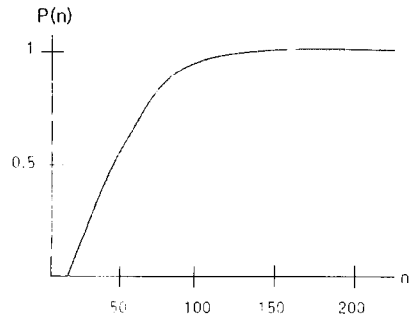
담고 있던 모든 메시지를 다 내보낸다.

o Timed pool *Mix*

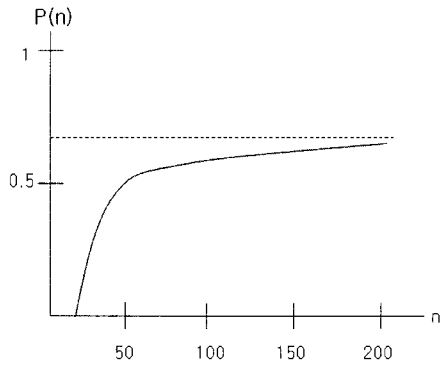
*Mix*는 항상 일정한 수의 메시지 N_p개를 풀에 보관하고 있다가 주기적으로 내보낸다. n을 플러싱 할 시기에 가지고 있는 메시지의 총 개수라 하면, t초마다 $n-N_p$개의 메시지를 내보낸다. 즉, 마지막으로 메시지를 내보낸 이후에 *Mix*에 도착한 메시지 개수 N만큼을 현재 *Mix* 내에 있는 메시지들 중에서 랜덤하게 선택해서 내보내는 것이다. 만약 내보낼 시기에 단지 N_p개의 메시지가 있다면 *Mix*는 어떠한 메시지도 내보내지 않고 그 이상의 수가 모여져야지만 $n-N_p$개의 메시지를 내보낸다. 다음 그림은 $N_p=20$일 때를 나타낸다.

o Timed dynamic pool Mix (Cottrell Mix)

이 *Mix*는 들어오는 메시지의 수가 임계치 값 N_p보다 큰 상태에서 일정 시간만료가 되면 메시지를 내보내는 구조이다. 그러므로 전체적으로 메시지를 내보내는 시간이 동적일 수 있다. 여기서는 *Mix*내의 메시지의 수가 $n=N+N_p$이라고 한다면, 내보내는 메시지의 수는 N이 아니라 fraction 함수 f에 의해서 좌우된다. f는 n에서



〔그림 3〕 Timed pool *Mix*



〔그림 4〕 Timed dynamic pool *Mix*

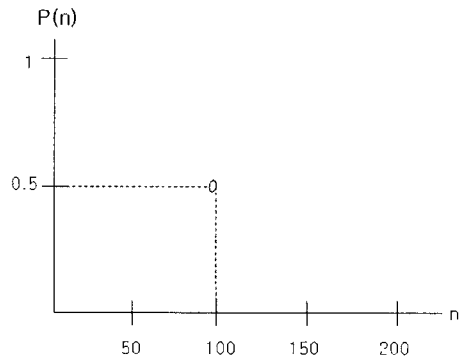
임계치 값을 뺀 값들에 대해서 *Mix*에서 나가는 메시지의 비율을 의미하는 함수이다. 예를 들어, $f(n-N_p)=0.7$로 설정할 수 있다. 다음 그림은 N_p는 20이고, $P(n)=f(1-N_p/n)$이다.

o Threshold pool Mix

위 세 개의 *Mix*는 얼마나 자주 메시지를 내보내는지를 나타내는 시간주기(T)에 관한 함수이다. 만약 $T=0$으로 놓는다면, 임계점을 제외하고 메시지를 내보낼 확률 $P(n)=0$이다. 즉, 임계치를 $N+N_p$라 한다면, *Mix*내에 저장된 메시지가 임계치에 도달하기만 하면 N만큼의 메시지를 랜덤하게 선택해서 내보낸다. 다음 그림에서 나타난 그래프는 임계치가 100 이고, 풀의 크기 N_p가 50인 경우이다.

o 익명성(anonimity)의 측정

위의 Batch strategy를 사용하는 *Mix*의 익명성을 측정하는 방법은 다양하다. [15]에서 *Mix* 내에 있는 엔트로피 개념을 이용하여 다음과 같이 정의하였다.



〔그림 5〕 Threshold pool *Mix*

$$E_{total} = E_{mix} + \sum_{0<x<=n} p_x E_x$$

E_{mix}는 현재 *Mix* 자체가 포함하고 있는 메세지의 엔트로피를 의미하며, p_x는 x번째의 *Mix*에 하나의 주어진 메시지가 존재할 확률이다. 즉, *Mix*에 존재하는 하나의 메시지의 익명성은 *Mix* 자체가 가지는 엔트로피 값에다 *Mix*내의 모든 메시지들의 익명성을 더한 값으로 정의하였다. (이에 대한 자세한 설명은 [15]를 참조한다.)

*Mix*내의 메시지에 대한 익명성 측정은 새로운 *Mix* 서버가 설계 되었을 때 중요한 평가도구로써 활용될 수 있다. 또한 USN과 같은 차세대 IT 환경에서는 이러한 익명성 평가방법이 달리 적용될 수도 있다. 이에 대한 논의와 연구가 필요하다.

2.2 *Mix-net* 기능에 관한 연구

1) Universal *Mix-net*

송신자가 직접 해당 *Mix* 들의 공개키를 이용하여 각각 암호화 한 다음 메시지를 전송하는 기존의 *Mix-net* 방식은 송신자가 모든 *Mix*들의 공개키를 미리 알아서 각각 암호화해야 한다는 단점이 존재한다. 이와 달리 최근에는 송신자가 *Mix*를 미리 구성하여 각각의 공개키로 암호화 할 필요 없이 안전한 공개 게시판을 이용하여 언제든지 전송하고자 하는 메시지를 안전하게 전송할 수 있는 Universal *Mix-net*이 등장 하였다[8]. 즉, *Mix*들의 공개키가 별도로 요구되지 않는다.

송신자가 먼저 보내고자 하는 메시지 m 과 1을 각각 암호화 하여 게시판에 고정된 길이로 게시한다. 그 이후 일정한 시간이 지나면 각 *Mix*들은 랜덤한 값만을 이용하여 게시판 전체 메시지들을 재 암호화(re-encryption)하고, 재배열(re-order)하여 게시판을 갱신하게 된다. 이러한 믹싱의 과정이 여러 *Mix*서버들에 의해서 끝나면 수신자는 먼저 1의 암호문 부분을 자신의 개인키를 이용해 복호화 해서 1이 나오면 해당 메시지가 자신에게 보내어진 메시지라 확신하고 메시지 m을 복호화 하게 된다.

구체적인 설명은 다음과 같다.

[1단계] 키 생성 알고리즘 : 수신자의 공개키, 개인키 생성

$$KG \rightarrow (PK, SK) = (y = g^a, a)$$

[2단계] 암호화 알고리즘

$$UE \rightarrow [(my^{r_1}, g^{r_1}); (y^{r_2}, g^{r_2})] \quad for\ (r_1, r_2) \in_U Z_q^2$$

[3단계] 재 암호화 알고리즘

$$URE \rightarrow (my^{r_1 + r_2 r_3}, g^{r_1 + r_2 r_3}); (y^{r_2 r_3}, g^{r_2 r_3})$$
$$for\ (r_3, r_4) \in_U Z_q^2$$

[4단계] 복호화 알고리즘

$$UD \rightarrow If \frac{y^{r_2 r_3}}{(g^{r_2 r_3})^x} = 1 \ , \ \frac{my^{r_1 + r_2 r_3}}{(g^{r_1 + r_2 r_3})^x} = m$$
$$If \frac{y^{r_2 r_3}}{(g^{r_2 r_3})^x} \neq 1 \ , \ the\ decryption\ fails$$

암호화 알고리즘은 송신자가 처음으로 암호화하는 것을 의미하며 재 암호화는 각 *Mix*들이 차례로 재암호화 하는 것을 의미한다. 마지막에 수신자는 비로소 복호화 알고리즘을 통해 메시지를 전달 받는다.

2) Incomparable Public Key 프로토콜[19]

Waters는 수신자가 하나의 아이디로 하나의 공개키를 사용할 때 생길 수 있는 문제점을 지적하였는데, 익명의 수신자에게 보낼 메시지를 암호화할 공개키를 여러 명의 송신자가 공유하였을 때, 이들은 자신들이 실제 같은 수신자에게 메시지를 전송하고 있다는 것을 유추해 낼 수 있다는 것이다. 따라서, 송신자가 동일한 수신자와 통신하고 있다 라는 사실을 숨기기 위하여 수신자는 여러 개의 익명의 아이디와 여러 개의 다른 공개키가 필요하다. 이러한 문제점에 대한 답안으로 Waters가 공개키를 비교 할 수 없도록 하는 Incomparable Public Key 프로토콜을 제안하였다. 제안된 프로토콜에서는 비 대칭키 암호 시스템으로 데이터를 암호화하는데 사용될 수 있는 고유한(unique) 공개키는 다수인 반면, 복호화에 사용되는 비밀 키는 하나뿐이다. 이 때, 같은 비밀 키에 대응되는 두 공개키는 *equivalent*하다고 말하고, 그렇지 않다면, *non-equivalent*하다고 말한다.

이 프로토콜에서는 수신자는 단순히 멀티캐스트 주소와 Incomparable Public Key키의 쌍으로 익명의 아이디를 만든다. 수신자는 *equivalent* 하지만, 고정된 공개키를 사용한다. 구체적인 설명은 다음과 같다. 다음에서는 오직 두 개의 *equivalent* 한 키만 가정하였다.

[1단계] 키 생성 알고리즘 : 수신자의 공개키와 개인키 생성

생성원 : g, h, PK : 공개키, SK : 개인키

$$KG \rightarrow (PK, SK) = (y = g^a, a)$$
$$= (y = h^a, a)$$

〔2단계〕 암호화 알고리즘
전달되는 메시지 형태 : $((d,e),h,M)$

$$(d,e) = (g^r, g^{ar}K)$$
$$h = H(r)$$
$$M = E_K(r,(g,g^a),m)$$

단, K는 메시지 암호화 용 키이고 r은 랜덤 값이며, m은 전달하는 메시지이다.

〔3단계〕 복호화 알고리즘
1. 먼저 수신자는 비밀 키를 알고 있으므로 $K=e/d^a$를 알 수 있다. 이 키를 이용해서 M부분을 복호화 해서 r,g,g^a,m 을 얻는다.
2. $h=H(r)$을 검증하고 공개키가 equivalent 한지 검사한다.
3. $g^r=d$를 검사한다.
4. 1~3 과정이 올바르면 복호화된 메시지를 출력한다.

3) Reusable Anonymous Return Channel[9]

Chaum이 제안한 기본 *Mix-net*에서는 송신자의 익명성은 보장할 수 있으나 수신자가 익명으로 송신자에게 응답하는 것은 불가능했다. 이러한 기본 *Mix-net*의 확장된 디자인으로 한 번의 응답을 허용하는 스킴이 있으나 실질적으로 제약 사항이 많았고, 다른 확장된 디자인으로 여러 번의 응답을 허용하는 스킴이 존재하나 이는 트래픽 분석에 취약했다. 왜냐하면 특정수의 응답을 보냄으로써 누가 같은 수의 메시지를 모으는지 관찰함으로써 공격자는 메시지의 근원지를 추적할 수 있기 때문이다.

이 논문에서는 익명의 메시지의 송신자가 재사용 가능하며 익명의 리턴 채널을 구축하는 프로토콜을 제안한다. 이런 채널은 익명의 메시지를 받은 수신자가 한번 이상의 익명의 응답을 하는 것을 가능하게 한다. 다른 메시지들에 대해 응답하는 수신자는 두개의 리턴 채널이 같은 것인지 확인할 수 없고, 따라서 같은 사람에게 응답하고 있는지 알 수 없는 것이다. 그리고 다수의 수신자들은 같은 리턴 채널을 통해 여러 개의 응답을 보낼 수 있다. 전체적인 프로토콜 설명은 방대하므로 생략한다.([9] 참조)

4) Length Preserving을 만족하는 *Mix-net* 프로토콜

Mix 서버에 입력되는 메시지와 출력되는 메시지들의 길이가 틀리다면, 공격자는 쉽게 메시지의 연결성을 파악하게 된다. 그러므로 입출력 메시지의 길이를 고정되게 유지하는 작업은 반드시 필요하다.

가장 쉬운 방법으로, 각각의 공개키로 암호화된 메시지가 *Mix*들을 거치면서 일정한 길이만큼 줄어들게 되는데 각 *Mix*에서 일정 길이만큼을 임의로 패딩 시켜주는 것을 생각해 볼 수 있다.

하지만 이러한 방법은 공격자가 패딩된 부분의 일정 부분을 마킹 함으로써 메시지의 연결성을 쉽게 깰 수 있다.

가장 최근의 연구결과로는 [12]가 있다. 이 논문에서는 CCA(chosen ciphertext attack)에 안전한 공개키 암호알고리즘과 스트림암호 및 MAC을 혼합한 hybrid 한 length preserving *Mix-net*을 설계하였다. 또한 비연결성에 대한 안전성을 랜덤오라클 모델에서 증명하였다.

3. *Mix-net*의 향후 연구 과제

지금까지의 *Mix-net* 연구는 *Mix* 서버들이 각기 다른 *batch strategy*를 사용했을 때 발생하는 가능한 공격들과 이에 대한 익명성을 측정하는 논문들이 주를 이루었다. 이와 더불어, 다양한 *Mix-net* 기능들을 추가하고 이에 대한 익명성 및 비연결성을 증명하는 방향으로 연구가 이루어졌다.

하지만 지금까지 개발된 *Mix-net* 기술들을 차세대 USN 환경에서 그대로 쓰기에는 무리가 있다. 예를 들어 공개키를 사용하는 *Mix-net* 구조는 엄청난 비용을 초래한다. 또한 USN의 센서 특성에 맞는 *Mix-net* 설계도 쉬운 일이 아니다. 유비쿼터서 센서 노드에서 라우팅 할 때 어떻게 입출력의 관계를 최소의 비용으로 숨기느냐 하는 문제는 아주 좋은 연구주제가 될 수 있을 것이다. 차세대 IT 환경에 맞는 익명 통신로 개발이 시급하다.

Ⅲ. DC.-Net에 관한 연구동향

1. *DC-net*의 연구 흐름[3]

*DC-net*은 n명의 통신 참여자가 동시에 자신의 암호문을 전송하는 방식으로 사용자간의 통신이 아닌 단 한 번의 브로드캐스트만으로 이루어진다. 이 때, 한 사람이 익명으로 메시지를 보내기 위해서는 모든 참가자가 동시에 자신의 메시지를 보내야 한다. 이러한 특성 때문에 악의적인 사용자가 프로토콜 수행 도중에 갑자기 멈추거나, 옳지 않은 메시지를 보내는 방법으로 프로토콜 수행을 방해할 수 있다. 또, 사전 키 공유와 각 참가자의 메시지

전송량이 많은 비효율성 문제로 인하여 많은 연구가 진행되지 않았다. 반면, *DC-net*은 메시지를 일정하게 보내어 트래픽 분석을 방지할 수 있다. 또, 공격자가 모든 전송 라인을 제어한다 할지라도, 그는 메시지의 송신자에 관한 어떠한 정보도 얻을 수 없으므로 송신자 익명성을 제공하는 특성을 갖는다[3].

1.1 Chaum의 *DC-net*

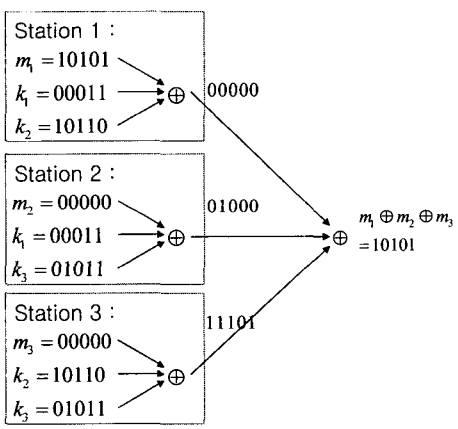
Chaum에 의해 제안된 스킴은 binary superposed sending에 기초한다. 각 사용자 스테이션은 적어도 하나의 다른 스테이션과 임의의 길이의 비트 스트링(binary random bit stream)을 공유한다. 이 임의의 비트 스트링은 비밀 키의 역할을 하게 된다.

1) Binary Superposed Sending

각 한 번의 송신 단계에서 모든 사용자 스테이션은 공유한 모든 키 비트들과 메시지 비트를 이진 덧셈 계산한다(superposed). 메시지 전송을 원하지 않는 스테이션은 그들 키 비트들의 합을 산출하여 0을 보낸다. 전송받은 값을 모두 ⊕한다. 모든 사용자 스테이션에 분산되어진 결과는 전송된 모든 메시지 비트들의 ⊕한 결과이다. 만약 정확히 한 참여자만이 메시지를 전송했다면, 메시지는 각 참여자에게 전체 ⊕의 결과로서 성공적으로 전달된다. 만약에 충돌이 감지되었다면 랜덤한 횟수의 라운드 후에 메시지를 재전송하는 방법으로 해결하게 된다.

아래에 그 그림을 나타내었다.

공격자의 경우에 각 스테이션에 의해 전송된 메시지 비트들에 관해 알 수 있는 유일한 것은 그들의 parity sum이다. 왜냐하면 그 패드의 parity만이 알려진 때, 메시지 비트들은 one-time-pad에 의해서 은닉되고, 오직 그들의 패리티만이 노출된다. 완벽한 수신자 익명성은 안전한 브로드캐스트에 의해 보증되어 질 수 있고, 메시지의 비밀성은 메시지 암호화에 의래서 이루어진다.



〔그림 6〕 *DC-net*의 구조

2) Slotted ALOHA

Superposed sending은 충돌이 있는 추가적인 multi-access channel을 인지할 수 있으며, 충돌은 익명을 보존하는 multi-access protocol에 의해 해결되어져야 한다.

이런 multi-access protocol은 문자들의 고정된 수 c를 메시지로 연결시켜 각 메시지들을 c가 연속되는 라운드들(slots이라 불린다)에 전송시킨다.

효과적이진 않지만 간단한 프로토콜로 slotted ALOHA가 있다.

사용자 A_i가 메시지 전송을 원한다면, 다음 슬롯에서 전송한다. 만약 충돌이 발행하면, A_i는 충돌을 감지하고, 랜덤한 횟수의 슬롯 후에 다시 메시지를 재전송 하게 된다.

충돌을 피하기 위해서 좀 더 효율적인 프로토콜은 예약 프로시져를 사용한다. 실제 메시지를 위한 많은 슬롯들이 예약 프레임에 의해 선행되어지고, 이것은 특별한 참여자를 위해 그것들을 보존하기 위해 사용된다.

3) 키 교환

Superposed sending은 거대한 양의 랜덤 하게 선택된 키들을 교환해야 한다. 이 작업은 쉬운 일이 아니다. 비밀 키를 공유하기로 한 참여자들 각 쌍의 한 멤버는 사전에 미리 반복되지 않은 랜덤한 키 스트림을 생성해서 별개의 구분되어진 광디스크에 그 키 스트림의 똑같은 복사본 두개를 만든다. 그런 후 그 디스크를 통신 파트너에게 제공한다. 비용이 적게 드는 대체 접근은 암호학적인 유사 랜덤 넘버 생성기에 의해 생성된 키 스트림을 사용한다. 물론 동일한 키를 만들기 위해 두 개체가 잘 알려진 키 교환 기법들을 사용할 수도 있다.

4) Fail-stop 브로드캐스트[18]

특별한 참여자 A_i의 전송되는 메세지가 능동 공격자에 의해 어떻게 추적 가능한지 〔18〕에 잘 나타나 있다. 가령, 능동 공격자와 정당한 참여자 A_i가 통신한다고 가정하고, A_i는 메시지를 전송하면 거기에 답할 것이다. 공격자가 메시지를 연속적으로 오직 한명의 참여자에게 보내고, 의미 없는 메시지들을 다른 모든 사람들에게 전송한다고 하자. 능동 공격자는 정당한 사용자 A_i가 메시지를 받았는지 받지 않았는지를 체크하여 신원 확인할 수 있다. 이러한 능동 공격이 가능하다면, *DC-net*은 수신

자 추적불가능성을 보증하지 않으며, 따라서 송신자 추적 불가능성도 보증할 수 없다. [18]에서, 이러한 문제점에 대한 해결책으로 *DC-+Net*을 제시하고 있다. 이것은 능동 공격에도 불구하고 무조건적인 추적불가능성을 제공한다는 것을 증명하고 있다. *DC-+net*은 *DC-net*을 사용하나, 안정한 브로드캐스트를 fail-stop 브로드캐스트로 대체했다. fail-stop 브로드캐스트는 정직한 두 명의 참여자가 서로 다른 입력 문자를 받았을 때 메시지 전송을 멈추게 함으로써 능동 공격을 막게 된다.

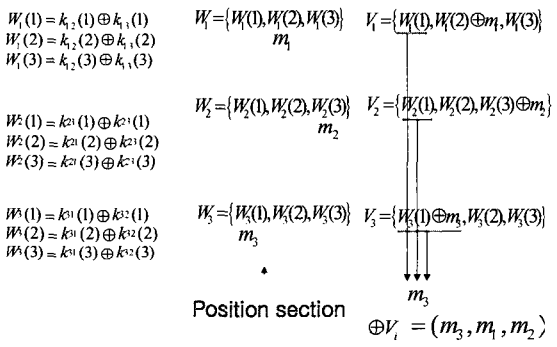
5) *DC-net*의 재고찰[10]

*DC-net*은 단 한명의 악의적인 사용자에 의해서도 옳지 않은 메시지를 보내거나, 메시지를 보내지 않는 방법으로 공격당하기 쉽다. 즉, 한명의 악의적인 공격자에 의해 다른 사람들이 메시지를 복호화 할 수 없게 된다. *DC-net*에서 송신자는 메시지를 전달하기 위해서 자신의 메시지를 키 패드에 추가하여 벡터(키와 메시지의 ⊕ 값)를 각각 구성하게 되는데 이때, 메시지를 첨부할 벡터상의 위치인 c_i를 선택해야 한다. 다음의 예는 3명의 송신자가 메시지를 전달하는 과정이다[10].

이 때, 각 사용자마다 메시지를 첨부할 위치인 c_i의 위치는 메시지가 제대로 첨부되었는지를 확인하기 위해서 모두 달라야 하고, 이러한 위치선택의 문제도 쉽지는 않다.

이러한 문제를 해결하기 위해서 메시지의 위치를 미리 선정하여 송신자에게 미리 알려주는 예약 프로시져를 사용하기도 한다. 이 경우, 악의적인 사용자는 *DC-net*을 공격하기 위해서 자신이 예약하지 않은 위치에 메시지를 보내는 방법으로 공격이 가능하다. 그러나, 그러한 공격의 경우는 예약 프로시저에 의해서 사전 단계에 찾아낼 수 있다.

그러나 이러한 방법 때문에 정당한 송신자의 익명성까지 침해받게 되는 문제가 발생하게 된다는 것을 Waidner와 Pfitzman은 지적하였다[18]. 그들은 이러한 점을 보완하여 예약 프로시져 기술을 확장하게 된다. 그러나 그러한 방법도 주어진 위치 c_i에서의 공격자 파악만이 가능할 뿐, 메시지를 복원하는 기능은 제공하지 못하였다. 즉, 메시지를 복원하기 위해서는 메시지를 재전송하는 방법이 필요하며 이를 위해 추가적인 라운드가 첨가되어야 한다.



[그림 7] 전송자가 3명인 *DC-net*의 예

아주 최근 Golle와 Juels[10]에 의해 불법적인 참여자의 추적과 효율적인 메시지 복원이라는 두 문제가 해결되었다. 제안된 시스템에서는 오직 한 번의 브로트캐스트만으로 높은 확률로 메시지를 복구할 수 있다. 물론 공격자도 추적할 수 있다. 또한, 공격된 사실을 쉽게 판별할 수 있도록 처음에 브로드캐스트 된 보조 데이타의 정당성 검증을 사용한다.

특히 이전의 방법들에서는 메시지를 복구하기 위해서는 많은 계산량과 전송량을 요한다. 그러나 [10]에서는 단 한번의 추가된 브로드캐스트만으로 완전 메시지 복구(full fault recovery)가 가능하다.

2. *DC-net* 향후 연구 과제

지금까지의 *DC-net* 연구는 정당한 공격자가 네트워크를 방해했을 때의 문제점을 해결하는데 초점을 두었다. 대다수의 연구결과들은 많은 전송량과 계산량을 요했으며, 이를 효율적으로 해결한 결과도 나왔다[10]. 하지만 [10]에서는 불법적인 공격자를 찾는데 참여자 수만큼의 연산량을 필요로 한다. 이는 연산량이 사용자수에 비례하므로 전체적인 확장성(scalability)을 보장받지 못한다. 이를 상수번의 비용으로 줄이는 것은 좋은 연구주제가 될 수 있다.

V. 결 론

지금까지 익명통신의 연구 흐름을 *Mix-net*과 *DC-net* 중심으로 살펴보았다. 하지만 이러한 모든 결과들을 USN에 그대로 적용하기에는 많은 무리가 있다. 왜냐하면 제안된 모든 프로토콜들은 공개키 기반 구조를 사용하며, ad-hoc 성격의 네트워크를 가정하지 하기 때문이다.

USN 상황에서 전자투표 및 경매를 하거나 전자화폐를 사용한다면 USN 환경에 맞는 익명 통신로 개발은 반드시 필요하다. 그러므로 이에 대한 연구가 시급하다.

참 고 문 헌

[1] L. von Ahn, A. Bortz and N. J. Hopper, "k-anonymous message transmission". In Proc. of ACM CCS'03, PP122-130. ACM Press, 2003.

[2] M. Bellare, A. Boldreva, A. Desai and D. Pointcheval. "Key-privacy in public-key encryption". In Proc. of ASIACRYPT '01, pp.566-582. LNCS 2248.

[3] D. Chaum, The dining cryptographers problem : unconditional sender and recipient untraceability. In Journal of Cryptography, 1(1), pp.65-75, 1988.

[4] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms". Comm. *ACM*, Vol. 24, No. 2, pp. 84-88, Feb. 1981.

[5] C. Diaz and A. Serjantov, "Generalising Mixes", In the Proceedings of the privacy Enhancing Technologies workshop(PET 2003), March 2003.

[6] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin. "Secure Distribution Key Generation for Discrete-Log Based Cryptosystems". In Proc. of Eurocrypt'99, pp. 295-310. Springer-Verlag, 1999. LNCS 1592.

[7] S.Goldwasser and S. Micali. "Probabilistic encryption". Jaurnal Computer and System Sciences, Vol.28, pp.270-299, 1984.

[8] P. Golle, M. Jakobsson, A. Juels, and P. Syverson. "Universal Re-Encryption for Mixnets". In Proc. of the 2004 RSA Conference, Cryptographer's track, San Francisco, USA, February 2004.

[9] P. Golle, M. Jakobsson. "Reusable Anonymous Return Channels". In Proc. of WPES '03, October 30, 2003, Washington, DC, USA.

[10] P. Golle, A. Juels, "Dining Cryptograpers Revisited". In Proc. of Eurocrypt'04, LNCS 3027, pp 456-473, 2004.

[11] K. Kurosawa. "Multi-recipient Public-Key Encryption with Shortened Ciphertext". PKC 2002, LNCS 2274, pp. 48-63, 2002.

[12] B. Moller, "Provably Secure Public-Key Encryption for Length-Preserving Chaumian Mixes". CT-RSA 2003, April 2003.

[13] W. Ogata, K. Kurosawa, K. Sako and K. Takatan. "Fault tolerant anonymous channel". In Proc. of ICICS'97, pp. 440-444, 1997. LNCS 1334.

[14] T. Pedersen. "A Threshold cryptosystem without a trusted party." In Proc. of Eurocrypt'91, pp.522-526, 1991.

[15] A. Serjantov, R. Dingledine, and P. Syverson, "From a Trickle to a Flood: Active Attacks on Several Mix Types", *In the Proceedings of* Information Hiding Workshop, October 2002.

[16] A. Serjantov, R. E. Newman, On the Anonymity of Timed Pool Mixes, In the Workshop on Privacy and Anonymity Issues in Networked and Distributed Systems, Athens, Greece, May 2003, pages 427-434.

[17] M. Waidner, "Unconditional sender and recitient untraceability in spite of active attacks", In Proc. of Eurocrypt'89, pp. 302-319. LNCS 434.

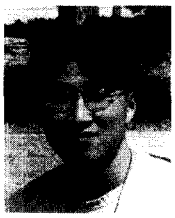
[18] M. Waidner and B. Pfitzmann, "The Dining cryptographyers in the disc : Unconditional sender and recitient untraceability with computaionally secure serviceability", In Proc. of Eurocrypt'90.

[19] B. Waters, E. Felten, and A. Sahai, "Receiver Anonymity via Incomparable Public Key", In the Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS 2003), October 2003, pages 112-121.

〈著 者 紹 介〉

이 현 숙 (Rhee Hyun-sook)
학생회원

1998년 2월 : 단국대학교 수학과 졸업
2000년 2월 : 단국대학교 수학과 수학
2001년 3월~현재 : 고려대학교 정보보호대학원 박사수료
〈관심분야〉 암호 프로토콜, 익명성 연구, DB 보안



변 진 욱 (Byun Jin-wook)
학생회원

2001년 2월 : 고려대학교 전산학과 졸업
2003년 2월 : 고려대학교 정보보호대학원 석사
2003년 3월~현재 : 고려대학교 정보보호대학원 박사과정
〈관심분야〉 암호프로토콜, 키 교환, 익명성 연구, DB 보안

박 현 아 (Park Hyun-a)
학생회원

2003년 2월 : 고려대학교 수학과 졸업
2003년 3월~현재 : 고려대학교 정보보호대학원 석사과정
〈관심분야〉 암호프로토콜, 익명성 연구, DB 프라이버시

이 동 훈 (Lee Dong-hoon)
정회원

1983년 8월 : 고려대학교 경제학사
1987년 12월 : Oklahoma University 전산학 석사
1992년 5월 : Oklahoma University 전산학 박사
1992년 8월 : 단국대학교 전자계산학과 전임강사
1993년 3월 ~ 1997년 2월 : 고려대학교 전산학과 조교수
1997년 3월 ~ 2001년 2월 : 고려대학교 전산학과 부교수
2001년 2월~현재 : 고려대학교 정보보호대학원 교수
〈관심분야〉 암호프로토콜, 암호이론, USN 이론, 키 교환, 익명성 연구, PET 기술

임 종 인 (Lim Jong-in)
정회원

1980년 2월 : 고려대학교 수학과 졸업
1982년 2월 : 고려대학교 수학과 석사
1986년 2월 :고려대학교 수학과 박사
1986년 9월 ~2001년 1월 :고려대학교 자연과학대학 정교수
2001년 2월 ~ 현재 : 고려대학교 정보보호대학원 원장, 고려대학교 정보보호기술연구센터 센터장
〈관심분야〉 암호 이론, 암호 정책, PET 기술