제19회 한국정보처리학회 춘계학술발표대회 논문집 제10권 제1호 (2003. 5)

PDA 푸쉬 서비스를 위한 보안 메시지 전송 시스템 구현

이정균°, 이기영, 이기영
인천대학교 정보통신공학과

Implementation of Security Message Transfer System for PDA Push Service

Jung-Gyun Lee°, Ki-Young Lee, Ki Young Lee
Department of Information & Telecommunication Engineering, University of Incheon
tnfeco@empal.com

요 약

음성과 데이터가 같이 사용될 수 없는 2.5G의 통신형 PDA를 위한 PUSH 서비스를 제안하였고, 이 PUSH 서비스와 이동 단말기의 특성을 고려한 보안 프로토콜을 현재 상용화된 알고리즘을 적절히 적용하여 구현하였다. 제안한 PUSH 서비스는 유선 인터넷 망과는 오프라인 상태인 음성 대기용 클라이언트 단말기와 망 접속 방식으로 SMS(short message service)를 이용하였으며, SMS 메시지 수신 후에는 단말기 상에서 SMS의 데이터를 분석하여 RAS(remote access service)를 통해 데이터 채널을 생성하여 서버측의 데이터가 PUSH 되도록 구현하였다. 보안 프로토콜은 통신횟수를 줄이기 위하여 비표형식을 취했으며, SMS와 데이터의 2개의 채널을 가지는 2중 통신 방법을 이용하였다. 이를 위하여 보안 알고리즘 중 처리속도가 빠르고 적은 메모리 공간을 사용하는 SEED와 MD5 알고리즘을 사용하였으며, 더욱더 안전한 키 분배를 위하여 인증과 보안의 효과를 가지는 RSA알고리즘도 이용하였다.

* 본 연구는 한국과학재단 지정 인천대학교 멀티미디어 연구센터의 지원에 의한 것입니다.

1. 서론

모바일 컴퓨팅이 활성화되는 요즘 많은 PDA를 위한 인터넷 서비스들이 나타나고 있다. 그 중에서도 개인용 단말기의 특징을 가지고 있는 모바일 인터넷에서 PUSH 서비스의 비중은 높아 질 것이라고 본다. 이러한 모바일 인터넷 환경에서 본 논문은 PUSH 서비스를 구현하고 이에 안전한 통신을 위한 보안 프로토콜을 설계함과 동시에 처리 용량이 적은 모바일 단말기의 특징을 고려한 프로토콜을 설계함으로써 양질의 정보를 안전한 통신과 함께 받을 수 있도록 구현하였다.

모바일 단말기는 무선망을 사용하기 때문에 통신상의 많은 헛점을 보유하고 있다. 또한 서비스의 형태도 개인 일정관리, 주소록, 증권정보등 개인적인 데이터가 많기 때문에 이에 관한 보안은 필수 요소라 할 수 있다.

본 논문의 진행은 모바일 단말기의 PUSH 서비스 설계와 구현을 먼저 보여준 후 이와 같은 통신방식을 가질 때 그에 적합한 암호화 알고리즘을 적용하여 통신의 보안을 적용하는 설계하고 구현 함으로써 마치도록 한다.

2. PUSH 서비스 설계와 구현

2.1 PDA 단말기의 PUSH서비스의 특징

PDA 단말기는 일반 유선망처럼 항시 On-Line 상태가 아니다. 그 이유는 기본적인 음성 통신을 위한 기능을 기본으로 하기 때문이다. 항시 데이터 회선으로 열어놓을 경우 음성통신을 위한 대기 시간을 가질 수 없기 때문에 유선망과는 Off-Line 상태라 할 수 있다.

이러한 특징을 가지는 PDA 단말기에서 유선망으로의 접속을 위해 사용자의 요구에 따라 접속하는 것이 아니라 원하는 정보를 위한 이벤트가 발생하였을 때 접속하는 것이 첫 번째 문제이다. 본 논문에서는 PUSH 서비스를 위해 SMS Channel 과 Data Channel을 동시에 운용함으로써 모바일 단말기의 PUSH 서비스를 설계 및 구현하였다.

SMS(Short Massage Service)는 모바일 단말기가 가지는 일반적인 특성일 뿐 아니라 SMS Data도 모바일 단말기에서 처리가능하기 때문에 이는 적절한 연결방식이 될 수 있다.

2.2 PUSH 서비스의 설계

본 논문에서는 PUSH서비스의 구현을 아래와 같이 하였다. 일반 음성통신의 특성을 가지는 모바일 단말기에 부가적인 기능인 SMS와 Data Connection(RAS:Remote Access Server)를 통한 두 개의 채널을 적절히 혼합하여 설계 하였다.

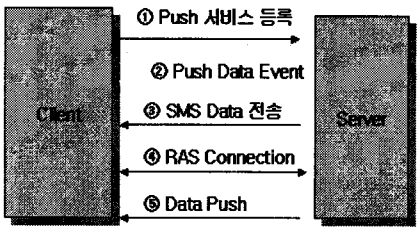

그림 1. Push 서비스 설계

그림 1에서 ①은 원하는 서버의 Push 서비스를 위한 등록 단계이다. ①은 원하는 Push 서비스의 형태와 자신 단말기 번호와 같은 개인 정보를 등록하는 단계이다. ②는 PUSH 데이터 생성 단계이다. 이는 정해진 시간이 아니기 때문에 이벤트라는 명칭으로 했다. ③은 Off-Line 단말기를 호출하는 단계이다. 이때 서버쪽의 어떠한 메시지가 도착했는 지와 서버의 주소를 알려주고 이를 받기 위한 준비를 하는 과정이다. ④는 SMS로 온 데이터를 기준으로 접속을 시도하여 Data Channel을 만든다. ⑤의 단계에서 사용자에게 원하는 데이터를 PUSH해 준다.

①의 단계는 서비스를 받기위해 한번만 시행 할 뿐이고 실제 PUSH 서비스의 중심이 되는 통신은 ③④⑤의 단계이다. 3번의 통신횟수를 거쳐야 하는 이유는 Off-Line의 PDA 단말기를 On-Line으로 만들어야 하기 때문에 ③④의 과정이 필요하다.

2.3 PUSH 서비스의 구현

본 논문에서 PUSH 서비스의 구현은 PDA 단말기를 Windows CE 3.0 OS를 가지는 iPAQ으로 구현하였으며 Push Server와 SMS Server, Content Server은 윈도우 2000의 OS를 가지는 PIII-1G PC로 구현하였다. 그 외에 SMS를 위한 셀룰러 단말기도 사용하였다. 실제 SMS 데이터와 Data Connection도 실제 RAS서비스를 통해 구현하였다.

그림2의 단계 ①은 Push 서비스를 위한 등록 단계이다. 자신의 PDA 단말기의 번호($ID_{[A]}$)를 서버 쪽에 등록한다. 이 단계는 최초 서비스를 위한 등록단계이므로 전체 서비스에서 1회 실행된다. 또한 유선 인터넷 서비스를 통해서 등록하거나 또는 PDA 단말기에서 직접 등록할 수 있다. 통신 방법은 Data Channel을 이용한다.

단계 ②는 등록된 단말기의 번호($ID_{[A]}$)를 DB에 삽입한다.

단계 ③은 PUSH 데이터 생성단계로 수시로 발생하는 PUSH 데이터($Push\_Data_{[A]}$)를 원하는 서비스 사용자의 단말기 번호($ID_{[A]}$)와 함께 Content DB에 저장한다.

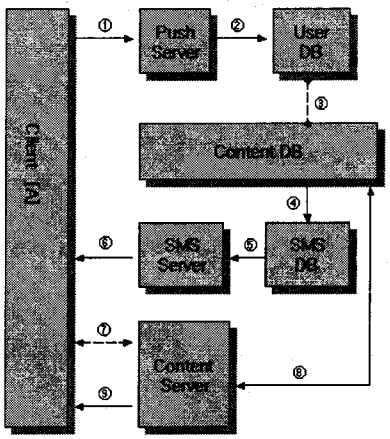

그림 2. Push 서비스 구현

단계 ④에서 Content DB의 삽입과 동시에 같은 Transaction으로 Trigger 명령을 통해 SMS DB에 PUSH 데이터의 번호($ID\_Push\_Data_{[A]}$)와 단말기 번호($ID_{[A]}$)를 삽입한다.

단계 ⑤는 이 데이터를 SMS Server에 전송한다.

단계 ⑥은 SMS Server가 SMS Channel을 통해 단말기 번호($ID_{[A]}$)를 참조하여 Content Server의 주소($ID_{[content\_server]}$)와 PUSH 데이터의 번호($ID\_Push\_Data_{[A]}$)를 함께 전송한다.

단계 ⑦은 단계 ⑥에서 받은 SMS Data를 처리한 후 Content Server의 주소($ID_{[content\_server]}$)를 참조하여 RAS Connection을 한다. 이로써 Data Channel이 생성된다.

단계 ⑧을 생성된 Data Channel을 통해 Content DB에서 단말기 번호($ID_{[A]}$)에 해당하는 PUSH 데이터($Push\_Data_{[A]}$)를 Content Server에서 준비한다.

단계 ⑨를 통해 사용자 단말기($ID_{[A]}$)에 PUSH 데이터를 전송함으로써 PUSH 데이터를 전송하고 Data Channel의 연결을 끊는다.

3. 보안 프로토콜의 설계 및 구현

3.1 PDA PUSH 서비스를 위한 보안 개요

PDA 단말기에서 많은 데이터 흐름을 가지게 될 PUSH 서비스는 PUSH 서비스의 특성상 개인적인 성향의 정보를 많이 가지게 될 것이다. 그러나 모바일 단말기 PUSH 서비스는 다음과 같은 문제를 가지고 있다.[4]

첫째, PDA 단말기가 가지는 저 용량성과 저 처리 능력이다. 이는 안전한 보안 방식을 사용하기에는 너무도 많은 작업시간과 작업공간을 필요로 한다.

둘째, 공중망 통신을 이용하기 때문에 도청이 쉽다. 이는 공중망 특성으로 인한 어디서나 도청이 가능하다는 관점뿐아니라 푸시 서비스가 가지는 개인적인 정보의 남용으로 이어 질 수 있다.

셋째, 위장에 의한 접근이 가능하다. 제 3자가 SMS 메시지를 받은 것처럼 위장하여 Content Server에 접근하여 PUSH 정보를 받아갈 수가 있다.

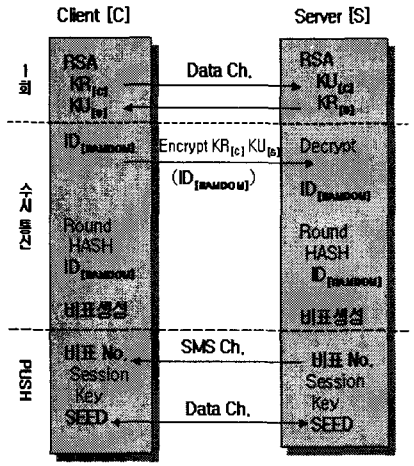

그림3. 모바일 PUSH를 위한 보안 개요

그림3은 본 논문의 서비스를 위한 보안의 개요이다. 단계는 다음과 같다.

클라이언트는 RSA 알고리즘을 이용하여 개인키($KR_{[C]}$)와 공개키($KU_{[C]}$)를 생성한다. 생성된 공개키($KU_{[C]}$)를 서버에 등록하고 서버로부터 서버의 공개키($KU_{[S]}$)를 얻어온다. 임의의 수($ID_{[RANDOM]}$)를 생성하여 서버쪽으로 보낸다. 이때 자신의 개인키($KR_{[C]}$)와 서버의 공개키($KU_{[S]}$)로 암호화하여 보낸다. 서버와 클라이언트는 정의된 Round Hash 함수를 통해 비표를 생성한다. 이후 PUSH 메시지가 발생 할 때마다 서버는 SMS Channel을 통해 비표번호를 기존 정보와 같이 전송하고 Data Channel이 생성되면 SEED 암호화를 통하여 비표 번호에 근거한 세션키로 통신한다.[2]

3.2 PDA PUSH 서비스를 위한 보안 프로토콜 설계 및 구현

3.2.1 초기 등록 단계

본 단계는 PUSH 서비스를 받기 위한 서비스 등록 단계로 다음과 같이 진행된다.

RSA알고리즘에 근거하여 클라이언트는 두 개의 서로 다른 큰 소수 p,q를 구하고, n=pq과 Ø=(p-1)(q-1)를 계산하여 구한다. gcd(e, Ø)=1인 정수 e(1<e<Ø)를 임의로 선택하고 확장된 유크리드 알고리즘을 이용하여 ed≡1(mod Ø)인 유일한 정수 d(1<d<Ø)를 계산한다. 이에 클라이언트는 공개키($KU_{[C]}$) Client[n, e]와 개인키($KR_{[C]}$) Client[n, d]가 생성된다. 역시 서버쪽도 같은 방식으로 공개키($KU_{[S]}$)와 개인키($KU_{[S]}$)를 준비한다. 첫 등록단계에서 클라이언트는 서버쪽에 공개키($KU_{[C]}$) Client[n, e]를 전송하고 서버쪽에서는 클라언트에게 공개키($KU_{[S]}$) Server[n, e]를 전송한다.[1][3][5]

3.2.2 비표 생성단계

본 단계는 안전한 세션키를 생성하기 위한 비표 생성단계로 64개의 비표를 만든다.

비표를 생성하기 위한 첫 ID값은 클라이언트로부터 임의로 만들어진 수를 받는다. 하지만 이 단계의 정보가 중요한 만큼 RSA의 암호화와 복호화를 이용한다.

클라이언트는 임의의 수($ID_{[RANDOM]}$)를 생성한다. 생성한 임의의 수는 클라이언트의 개인키($KR_{[C]}$) Client[n, d]로 암호화 한다.

$KR_{[C]}(ID_{[RANDOM]}) = ID_{[RANDOM]}^{\ d} \bmod n \equiv ID_{[KR\_C]}$

그 다음 서버의 공개키($KU_{[S]}$) Server[n, e]로 암호화 한다.

$KU_{[S]}(ID_{[KR\_C]}) = ID_{[KR\_C]}^{\ e} \bmod n \equiv ID_{[KR\_C \,\|\, KU\_S]}$

이와 같은 과정으로 도착한 메시지를 서버쪽에서는 다시 서버의 개인키로 복호화하고 다시 클라이언트의 공개키로 복호화 함으로써 원 임의의 수($ID_{[RANDOM]}$)를 알수 있다. 클라이언트의 임의의 수($ID_{[RANDOM]}$)을 이용하여 서버와 클라이언트 각각은 표2와 같이 정의된 Round Hash함수를 통해 비표를 생성한다.[5]

```
b=ID[RANDOM];
for (i=0; i<64; I++){
   b=rand(b)
   a=md5(b);
   비표파일 += a; // i Line
}
```

표 1. 비표생성 알고리즘

3.3.3 PUSH 서비스 단계

본 단계는 PUSH메시지를 전송하는 단계에서 이루어 진다. PUSH 서비스의 속도를 향상 시키기위해 준비된 비표를 사용한다. 또한 보다 안전한 통신을 위해 비표의 번호는 SMS Channel을 이용하고 PUSH 데이터는 Data Channel을 이용한다. SMS Channel로 온 데이터에는 서버쪽의 Conntent 서버의 주소와 Push Message의 번호가 들어있고 또한 보안을 위한 비표 번호가 들어있다. 이에 클라이언트는 해당 서버에 접속하여 서버로부터 온 데이터를 받는다. 받은 데이터는 SEED로 암호화 되어있지만 서버와 같이 약속된 비표 번호에 해당하는 비표파일의 Line 번호에 해당하는 비표내용이 받은 데이터의

SEED암호의 키가 된다.[2]

3.3.4 시스템 구성

가. 푸쉬 서버

본 시스템의 푸쉬 서버는 두 개의 시스템으로 구성되어있다. ContentDB, SMSDB, UserDB를 갖춘 윈도우 2000 서버 컴퓨터와 SMS 셋톱박스가 연결되어 SMS전송을 담당한다. SMS데이터는 SMS DB와 연동되어 발송하게 되며 ContentDB는 유선망 접속으로 데이터를 전송 할 수 있다.

본 논문에서 제안한 보안 프로토콜의 모듈은 모두 3가지로 RSA 키생성 모듈, Round Hash 모듈, Seed 파일단위 처리 모듈로 구성되어 있다.

나. 푸쉬 클라이언트

푸쉬 클라이언트는 SMS 데이터를 받을 수 있도록 무선 모뎀이 장착되어 있어야 하며 이 응답을 처리할 수 있도록 모뎀과의 연결 포트에 대한 처리가 되어 있다. 본 논문에서는 ipaq을 모델로 SKT의 nate 모듈, 019 i-kit 모듈이 구현되어 있다. 또한 PDA의 특성인 절전모드로 인한 시스템 sleep 타임에도 SMS 메시지를 받으면 sleep 타임에서 깨어 날 수 있도록 wake up 구현 하였다.

역시 본 논문에서 제안한 클라이언트 보안 모듈로 RSA 키생성 모듈, Round Hash 모듈, Seed 파일단위 처리 모듈로 구성되어 있다.

4. 결론 및 향후과제

본 논문은 모바일 푸시 서비스를 기준으로 하여 작성하였으며 이에 적절한 보안 요소를 위해 상용 알고리즘 중 안전한 형식을 따르는 비대칭 형식의 RSA와 데이터 통신의 고속화를 위한 SEED 방식을 취함으로써 저사양 저속 모바일 환경에서 안전하면서 빠른 속도의 성능을 발휘하는데 초점을 맞추었다.

향후, PDA 단말기를 위한 특성화된 PUSH서비스만을 기준으로 작성한 보안프로토콜로 좀 더 확장된 형식의 보안 서비스를 위해서는 상호 통신을 위한 보안 요소도 구비되어야 할 것이다.

본 논문에서 PDA 단말기를 Pocket PC로 한정했지만 다른 플렛폼에서도 원활히 동작하도록 이에 적절히 변형되어야 하는 것도 또 하나의 과제이다.

참 고 문 헌

[1] RFC 2437, B. Kaliski, J. Staddon, "RSA Cryptography Specifications Version 2.0", October 1998

[2] TTA.KO-12.0004 "128비트 블록암호알고리즘 표준", 한국정보통신기술협회, 1999년 9월

[3] 우원택, "전자상거래에서의 RSA 알고리듬의 분석과 구현", 한국정보시스템학회, 2000년도 춘계학술대회 발표논문집, 2000

[4] 김성열, 정일용, 오명옥, 배용근, "효율적인 그룹키 분배 및 갱신을 위한 보안 프로토콜의 설계", 한국정보처리학회, 정보처리학회논문지, 2002

[5] 최용락, 소우영, 이재광, 이임영 역, ",컴퓨터 통신 보안", 도서출판 그린, 2002

[6] William Stallings, "CRYPTOGRAPHY AND NETWORK SECURITY PRINCIPLES AND PRACTICE 2nd", 1999

[7] 이정배, 이두원, "임베디드 시스템 연구 동향", 정보처리 학회지 특집 제9권 1호, 2002. 1

[8] 김기천, "모바일 서비스 기술 동향", 정보처리 학회지 특집 제9권 2호, 2002. 3

[9] 최용락, 소우영, 이재광, 이임영 역, 통신망 정보보호, 도서출판 그린, 1996. 2

[10] 이은주, "RSA 암호의 구현", 고려대학교 교육대학원, 2001. 11

[11] Douglas Boling, "Programming Windows CE 2nd", 정보문화사, 2002. 1

[12] 고재관, "Mobile PDA Programming", 삼각형프레스, 2001. 8