안전한 키 전송을 위한 좌표기법 OTP 설계

이길헌**, 전문석*
**숭실대학교 일반대학원 컴퓨터학과
e-mail: clarkent@ssu.ac.kr

A Design of Coordinates Techniques OTP for a Secure Transmission with Key

KIL-HUN LEE**, Moon-Seog Jun*
**Dept of Computer Science, Soongsil University

요 약

본 논문에서는 좌표기법을 사용한 PAD 형식의 OTP를 제안하였다. 기존에 이용되던 SEED 기반의 OTP가 아닌 사용자의 기기 정보와 조합 동기화 방식을 통해 키 재료를 생성하여 OTP의 PAD 속에 좌표를 사용한 암호화를 이루는 알고리즘을 제안했기 때문에 기존의 Password 방식의 OTP보다 보안성에 우수한 특성을 나타낸다. 또한 키 교환 과정에 신개념의 Digest 방식을 도입하여 차별화를 이뤘다.

1. 서론

IT 정보통신이 발전하며 네트워크에서 금융, 게임, 그룹웨어 등 비즈니스 활동 시 화폐기능을 제공하는 IT 서비스가 등장하기 시작했다. 이에 따라 IT 서비스에 관련된 범죄가 증가하고 범행수법 역시 고도화되고 있는 가운데 부가적 보안수단의 중요성이 부각됐다.

이차적 보안장치로서 보안카드, 지문인식, 음성인식 등 많은 기술들이 연구되어 사용하고 있지만, 실제 활용에 있어 OTP 만큼의 보안기능을 제공하는 장치는 드문 현실이다. 특히 이차적인 보안장치로서 고정 Password가 적용되던 기존의 보안 취약성을 해결하기 위한 가장 강력한 본인인증 수단으로 매번 Password가 변하게 되는 OTP는 그 중요성이 더욱 부각되고 있다.

하지만 절대보안이라 여겨오던 OTP는 2006년 7월경 미국 시티뱅크에 이어 2011년 RSA 사의 Secure ID가 해킹되는 등의 사례가 발생함으로써 기존 OTP의 보안위협이 증대되었으며 이에 따른 OTP 인증에 대한 추가적인 연구를 필요로 하게 됐다.

본 논문에서는 이에 대한 해결책으로 OTP를 생성하기 위해 좌표기법을 도입한 Digest를 생성하여 본인인증을 하는 알고리즘을 제안하였다.

2. 관련연구

2.1. OTP(One-Time PAD)

OTP는 1회에 한해 사용할 수 있는 암호 시스템으로 매번 다른 암호를 이용하여 사용자를 인증하는 방식이다. 흔히 우리가 알고 있는 OTP란 One-Time Password를 말하지만 본 논문에서 뜻하는 OTP는 One-Time PAD라는 숫자형식이 아닌 패드형식의 일회용 암호이다.

OTP는 위의 두 형식에 상관없이 서버와 클라이언트 사이에 미리 약속된 규칙에 의해서 주고받는 암호를 통해 일회용 암호의 유효성을 검사하게 된다. 서버와 클라이언트 간에 주고받는 암호는 따로 사용자 데이터가 들어있는 데이터베이스에서 비밀키 값을 가져온 후, 시스템에 접근하여 OTP를 생성 후 사용자에게 인정하게 된다. 다음의 [그림 1]은 One-time PAD를 간략화한 예이다.

ZDXWIW	EJFLAS	BAZNME	AKSRIG	JZTLBC
HTSVTV	SSKDPP	ANSMTM	SJANFV	BZXQDQ
RKFKLS	SANNDK	BCBDWZ	VVSOES	DGGIAK
EKDJFL	ANSTA	TPQWID	VIQGBK	UCVXTW
ASLDKW	ANKKGN	FFLMMM	BAZSTN	SBFGZC
BIRLQK	APFKTZ	APPTAA	IJCWHG	DWHPER
MLQPSD	ZMGQWR	ZLKFIG	JKFXIO	HGAAPR
QMPDIW	ASKDIR	ASKDIW	DWHPER	GRRSSE

[그림 1] One-Time PAD의 예

One-Time PAD의 각 문자는 출현 빈도가 모두 같은 것을 사용하고, 평문의 길이와 비밀키의 길이가 같은 n개의 임의 문자열로 이루어져 있다.

암호화의 과정을 살펴보면 평문 $a_1a_2 \cdots a_n$와 비밀키 $k_1k_2 \cdots k_n$를 준비한다. 다음으로 각 i번째 비트를 XOR 연산하여 암호문자 $c_i = a_i \oplus k_i$를 생성하여 암호문 $c_1c_2 \cdots c_n$를 만든다. 식으로 나타내면 다음과 같다.

$$c_1c_2 \cdots c_n = a_1a_2 \cdots a_n \oplus k_1k_2 \cdots k_n$$

즉, 이 암호는 평문과 비밀키의 길이가 같게 된다. 그러므로 대량의 비밀키를 안전하게 보내는 것은 상당히 어려운 일이지만, 짧은 키로 의사난수를 발생하여 암호학적으로 안전한 비밀키를 매번 다르게 생성할 수 있다면, 암/복호화가 아주 경제적이고 효과적일 것이다. 본 논문에서는 그 방법에 대해 연구하고 제안한다.

2.1. OTP(One-Time Password)

OTP는 1회에 한해 사용할 수 있는 비밀번호 시스템으로 매번 다른 비밀번호를 이용하여 인증하는 방식이다. 일반 패스워드와는 달리 단방향 해쉬함수를 사용하여 세션이 끝나면 폐기되므로 재사용이 불가능한 특징이 있고 각각 장단점이 존재하는 동기화 기술을 사용하여 생성된다. 그 동기화 기술은 OTP 생성 방법에 따른 분류로 나뉜다.

2.1.1 Challenge-Response 방식

질의-응답 방식이라고도 부르며 OTP 토큰과 OTP 인증서버 간의 동기화되는 기준값이 없어, 인증 요청 시 사용자가 직접 임의 난수값을 OTP 토큰에 입력하여 OTP 값을 생성하는 방식이다. 서버가 제시하는 질의값을 사용자가 알고리즘에 입력해 출력되는 값을 얻고 이를 응답값으로 서버에 전송하여 자신을 인증하게 된다. 하지만 공격자가 네트워크를 모니터링하여 사용자의 비밀키를 찾아낼 위험이 있다.

2.1.2 Time-Synchronous 방식

시간-동기화 방식은 OTP 토큰과 OTP 인증서버 간에 동기화된 시간정보를 기준으로 특정시간 간격(보통 30초~1분)마다 변하는 OTP를 생성하는 방식으로, 일정 시간마다 하나씩의 난수를 생성하기 위해 난수 발생 알고리즘과 비밀키가 필요하다. 각각의 사용자에게는 특정키가 할당되고, 사용자가 기관에 사용자 정보를 전송하면 기관에서 로그인 정보를 확인하여 사용자에게 OTP값을 전송하고 응답값을 비교하여 사용자 인증을 수행한다.

특정시간 간격마다 비밀번호가 변하기 때문에 입력 가능한 시간 간격을 너무 길게 잡으면 공격의 가능성이 커지게 되며 사용자가 비밀번호를 잘못 입력하면 다시 생성될 때까지 기다려야 한다는 단점이 있다.

2.1.3 Event-Synchronous 방식

이벤트-동기화 방식은 OTP 토큰과 OTP 인증서버 간에 동일한 카운트 값을 기준으로 OTP를 생성하는 방식이다. 사용자가 OTP를 생성할 경우, OTP 알고리즘에서 해쉬함수의 입력값으로 비밀키와 인증횟수를 사용하여 그 기록을 인증서버와 공유, OTP를 생성한다. OTP를 생성한 후에는 카운터 값을 증가시켜 저장해두고 다음번에 사용하게 된다.

하지만 OTP 단말기에서 OTP만 여러번 생성하고 해당 OTP를 서버에 입력하지 않으면 단말기와 서버 사이에 카운트 값이 달라져 단말기를 초기화해야 하는 단점이 있다.

2.1.4 조합 방식

OTP 토큰과 OTP 인증서버 간에 시간과 이벤트 혹은 다른 값끼리의 연계를 통해 비밀번호가 생성되며 재시도 시에는 관련값을 변화시키도록 하는 방식이다. 현재 많이 사용되고 있는 것으로 시간+이벤트 동기화 방식이 있다.

2.2. Padding Algorithm

패딩이란 “채워 넣는다”는 뜻으로 현재 패딩 알고리즘은 파일의 끝 부분이 어디인지를 인식시켜주기 위해서 사용되고 있다. 비트열의 앞부분 혹은 끝부분에 NULL 문자나 패딩 문자열을 넣어 문자열의 시작과 끝을 인식하도록 인식시킨다. 또한 임의의 데이터를 삽입하여 일정한 용량의 비트열로 변환하는 방식이다. 기존의 패딩 방식에 대해 알아보면, 다음과 같이 4가지 부류로 나뉜다.

·standard padding: 일반적인 패딩 방식으로서 처리할 블록의 남는 공간에 임의의 문자를 채워 넣은 후

맨 끝에 padding 한 개 수를 집어넣어 패딩된 데이터임을 표시하는 방식

·space padding: 처리할 블록의 남는 공간을 공백으로 채우는 방식

·null padding: 처리할 블록의 남는 공간을 null로 채우는 방식

·one-and-zeros padding: end-characrter-padding 이라고도 불리며 처리할 데이터 블록의 마지막에 원문의 끝임을 나타내는 문자를 넣은 후 남은 공간을 임의의 문자로 채우는 방식

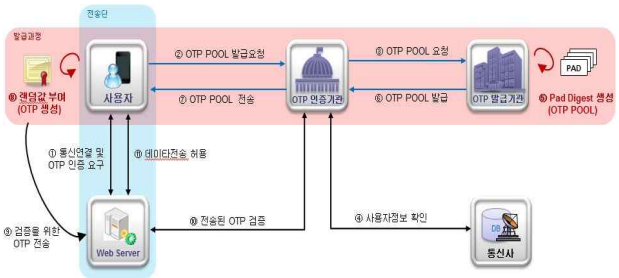
3. 제안방법

3.1. 제안된 One-Time PAD

기존 OTP 토큰은 절대 보안에 대한 잘못된 맹신과 OTP 사용 프로세스의 가장 취약한 연결고리인 사람이 공략되는 해킹방법을 통해 OTP의 SEED가 유출되거나 30초~1분의 유휴시간을 이용한 해킹이 이루어지곤 했다.

이와 같은 문제는 기존 OTP의 생성 구조가 6자리의 질의/응답값으로 이루어져 있다는 단순성과 유휴시간의 존재로 인해 1회성 암호로서의 보안 취약성을 제공했기 때문이라고 할 수 있다. 그러므로 원천적인 생성 구조에서의 Digest 구조를 변경함으로써 이런 취약점을 막아보고자 한다.

본 논문에서는 이러한 문제점을 막아보고자 OTP POOL 이라는 Digest를 생성하여 기존의 SEED방식을 벗어난 키 공유를 이뤄보고자 하였다. OTP POOL이란 일종의 OTP Materials Digest로서 여러 추가적인 인증 정보를 조합하여 키 역할로서 생성되는 비밀 패드이다. OTP POOL을 생성하여 주고 받음으로써 기존 방식과 다른 키 정보의 송/수신이 가능하게 된다.



[그림 2] 제안시스템 전체구조도

위의 그림은 제안하는 OTP의 전체구조도이다. 사용자의 접속수단은 상관없으나, 본 논문에서는 모바일을 통한 접속을 배경으로 설정하겠다.

① 통신 연결 및 OTP 요구: 서버로 접속하기위한 사용자의 요청에 OTP 인증을 요구함

② OTP POOL 발급요청: 사용자→ OTP 인증기관

③ OTP POOL 발급요청: OTP인증기관→ 발급기관

④ 사용자 정보의 확인: 모바일 사용자의 정보를 통신사로부터 확인함

⑤ PAD Digest 생성: 발급기관에서 취합된 정보를 통해 OTP POOL을 생성함

⑥ OTP POOL 발급: OTP발급기관→ OTP인증기관

⑦ OTP POOL 전송: OTP인증기관→ 사용자

⑧ OTP 생성: 사용자의 단말기에서 모듈을 통해 랜덤값이 부여된 OTP를 생성함. 여기서의 랜덤값은 OTP 동기화 방식 중 조합 방식을 이용한다.

⑨ 검증: 상대방 서버로 인증을 위해 OTP를 전송함

⑩ OTP 검증: OTP 인증기관에게 발급된 OTP를 전송하여 OTP 적합성을 점검한다.

⑪ 데이터 전송 허용: 데이터의 송/수신을 시작한다.

3.2. 좌표 기법의 OTP PAD

위에서 제안된 구조도를 보면 OTP POOL이 생성될 때 발급에 관련된 정보를 발급기관에서 취합하게 되는데, 이때 생성에 사용되는 방법을 제안한다.

One-Time PAD의 특성상 암호화 및 복호화의 과정에서 파일의 크기와 같은 용량의 키가 사용되게 된다. 하지만 좌표를 부여한 패딩을 PAD에 적용하면 빠른 암/복호화 과정과 PAD 수준의 보안성 유지가 가능하다.

우선 키 생성에 필요한 재료로는 사용자의 기기 정보(USIM-PIN)와 조합 동기화 방식에 의한 값(Time+Count)이 들어간다. 조합 동기화 방식에는 생성된 시간과 OTP가 발급된 순서번호가 함께 사용되며 좌표값의 부여를 위한 알고리즘이 필요하다.

본 논문에서 제안하는 것은 바로 이 좌표기법 알고리즘의 생성을 위해 OTP PAD를 만드는 방식이며 키 정보를 암호화하기 위한 패딩이 필요하다.

일반적인 패딩 방법으로는 파일과 파일 사이를 구분짓기 위하여 임의적인 정보를 부여하게 된다. 그 예를 들어보면, [표 3]처럼 파일이 끝나는 지점에 들어가는 임의정보를 송신자와 수신자만이 아는 방법으로 구분지어 전송하는 방법을 들 수 있다.

제안하는 패딩방법도 크게 다르지는 않지만, 주체가 되는 사항이 파일이 아니라 패딩으로 바뀐다는 특징을 가진다.

일정한 알고리즘으로 구성된 패딩 파일 속에 정보를 감춰두고 송신자와 수신자가 가진 모듈을 통한 과정을 거쳐야만 관련 정보의 취득이 가능하게 되는 것이다.

[표 1] 파일 a

주소	0	1	2	3	4	5	6
내용	H	E	L	L	O	0	

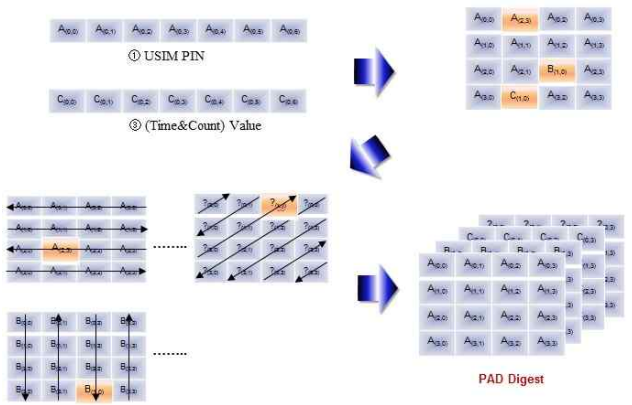
[표 2] 파일 b

주소	0	1	2	3	4	5	6
내용	T	H	E	R	E	0	

[표 3] 일반적인 파일 “a”와“b”를 연결한“C”파일의 모습

주소	0	1	2	3	4	5	6
내용	H	E	L	L	O	0	ab
주소	7	8	9	10	11	12	13
내용	0	T	H	E	R	E	0

특정 파일을 조합한다고 했을 때, 그 파일에는 수많은 PAD가 생성될 것이며 그 패드 중 어느 곳에 실제 정보가 담겨있는지 알기는 힘들게 된다. 또한, 각 패드별 암호화되는 과정을 바꾸게 되어 한 가지 방법만을 사용한 암호의 해독이 불가능하므로 암호의 추측은 더욱 힘들게 된다. [그림 3]과 같은 과정을 거쳐 생성되는 패드 정보는 키 길이가 증대되면 증대될수록 알기 힘들게 되며 OTP 사용자에게 돌아가 OTP를 생성하기 위한 재료로서 사용된다. 또한, 알고리즘을 알고 있는 사람으로서는 암호화와 복호화에 어려움이 많지 않게 되지만, 그 과정을 모르는 이에게는 키 전송 간의 철저한 보안을 가능하게 한다.



[그림 3] 좌표 기법이 도입된 OTP PAD

4. 결론

본 논문에서는 OTP를 사용한 통신에서 기존의 보안 취약성을 해결하기 위한 방법으로 OTP POOL이라는 방법을 제안하였으며, 이는 기존의 짧은 길이의 Password 방식이 아닌 PAD 방식을 통해 생성하여 보안성을 강화하였다. OTP로서는 가장 이상적인 형태지만, 키 길이가 길어져 암/복호화 과정이 길어질 수 있다는 단점이 있지만, 현대사회의 진보된 통신기술을 배경으로 하면 충분히 가능한 방법이라고 결론지었다. 물론 기존에 이용되고 있는 Password 방식의 OTP보다는 속도적인 성능 면에서는 다소 뒤떨어질 수 있으나, 보안성에서만큼은 충분히 우월한 조건을 가진다고 여긴다.

[표 4] OTP 성능 및 보안성 분석

종류	속도	키 보안성	암호 복잡도
Password	빠음	높음	중간
PAD	느림	매우 높음	높음

본 논문에서 제안된 사항에 다소 미진한 감이 있지만 제안 방안의 실제적인 지연시간 단축과 편차에 대한 확인을 위해 추가적인 연구가 병행되고 있다. 또한 호환될 수 있는 인터페이스 간 상세 비교값을 위한 연구와 검증은 향후 과제로 남아있다.

참고문헌

[1] 장승주, “패딩 문자열 길이 정보를 이용한 패딩 알고리즘 설계”, 멀티미디어학회논문지, 제9권 제10호, pp.1371-1379, 10월, 2006.

[2] 김재용, “OTP 기반의 인증서를 이용한 홈 네트워크 사용자 인증에 관한 연구”, 숭실대학교 학위논문(석사), 2010.

[3] 금융보안연구원, “모바일OTP 보안성 분석서 v2.0”, 2월, 2011.

[4] 김태형, 이준호, 이동훈, “피싱 방지 및 가용성 개선을 위한 PKI기반의 모바일 OTP(One Time Password) 메커니즘에 관한 연구”, 정보보호학회논문지, 제21권 제1호, pp.15-26, 2월, 2011.

[5] DIRK RIJMENANTS, “THE COMPLETE GUIDE TO SECURE COMMUNICATIONS WITH THE ONE TIME PAD CIPHER”, Cipher Machines&Cryptology, 11월, 2010.