# 패스워드 기반 인증 프로토콜

박익수*, 오병균

*목포대학교 정보공학부 정보보호전공

# Design of User Authentication Protocol based on Human Memorable Password

Ik-Su Park*, Byeong-Kyun Oh

*Major of Information Security Division of Information Engineering, MokPo National University.

## 요 약

지금까지 제안된 패스워드를 이용하는 인증 프로토콜은 오프라인 추측 공격이나 패스워드 파일 컴프로마이즈에 대하여 안전하지 않으므로 이에 대한 연구가 요구되고 있다.

본 논문에서는 패스워드를 이용하는 인증 프로토콜인 스키마 PAP(Password based Authentication Protocol)을 적용하여 새로운 인증 프로토콜 PAPRSA를 제안하였다. PAP는 패스워드를 표현하는 많은 값들 중에서 임의로 선택한 한 값을 처리하는 것을 특징으로 한다. PAPRSA는 패스워드를 표현하는 값을 처리하기 위하여 RSA를 이용하는 PAP기반 인증 프로토콜이다. 제안된 PAPRSA는 오프라인 추측 공격과 패스워드 파일 컴프로마이즈를 포함한 공격들로부터 안전하였으며, 패스 수와 계산량을 측정한 결과 효율성 면에서 매우 우수하였다.

## Ⅰ. 서론

인증이란 시스템 접근자가 인가된 사용자인가를 확인하는 절차이며, 인증 프로토콜은 통신로와 서버 상에서 비밀정보가 안전하도록 하면서 인증하는 프로토콜이다.

현재까지 제안된 인증 프로토콜에는 패스워드 기반 방식과 시도-응답 방식, 영-지식 기반 방식 등이 있다. 패스워드 기반 인증 프로토콜은 일방향 함수나 UNIX의 Salt를 이용한다[5][10]. OTP(One Time Password)는 일방향 함수를 이용하는 패스워드 기반 방식으로 증명자는 이전 전송된 메시지에 일방향 함수(일방향 해쉬 함수)를 한 번 더 적용한 결과를 검증자에게 전송한다. OTP는 검증자에 전송되는 메시지가 매번 다르므로 재전송 공격에 안전하지만, 선시도와 오프라인 추측 공격, 서버 컴프로마이즈 등으로부터 안전하지 않다[17]. UNIX에서 Salt는 사용자가 패스워드를 등록할 때 패스워드와 접합되는 정보로 사용자들의 패스워드가 동일하더라도 Salt값에 의해 구분되어지도록 하는 역할을 한다. Salt를 이용한 인증 프로토콜은 재전송 공격과 오프라인 추측 공격, 패스워드 파일 컴프로마이즈 공격에 안전하지 않다.

시도-응답 프로토콜에는 대칭 키 암호 알고리즘을 이용하는 방식과 공개 키 암호 시스템을 이용하는 방식 등이 있다[13][14]. 대칭 키 암호 알고리즘을 이용하는 프로토콜에서 검증자는 시도(난수)를 증명자에게 전송하고, 증명자는 시도에 대응하는 응답(난수를 대칭키로 암호화한 암호문)을 검증자에게 전송하는 과정을 반복한다. 대칭 키 암호 기반 인증 프로토콜은 한 트랜잭션에 대한 속도는 빠르지만, 패스 수가 많이 발생하기 때문에 키 관리 문제가 발생할 뿐만 아니라 평문 선택 공격에 노출이 될 수 있고, 서버 컴프로마이즈로부터 안전하지 않으며, 사용자가 기억하기 어려운 키를 사용하므로 키를 저장하는 부가매체가 필요하다. 공개 키 암호 시스템을 이용하는 프로토콜에서 검증자는 시도(난수)를 증명자에게 전송하고, 증명자는 시도에 대응하는 응답(난수를 개인키로 암호화한 결과)을 검증자에게 전송하는 과정을 반복하는 방식이다[6][16][18]. 공개 키 암호 시스

템 기반 인증 프로토콜은 공개 키 암호 시스템을 이용하기 때문에 키의 관리가 용이한 반면, 속도가 느리며 평문 선택 공격에 노출이 될 수 있다. 그리고, 사용자가 기억하기 어려운 개인키를 사용하므로 개인키를 저장하기 위한 부가 매체가 필요하다.

영지식 기반 인증 프로토콜에서 검증자는 증거를 이용하여 증명자에게 여러 가지 질문을 시도하는 방식이다[7]. 증명자는 검증자가 시도하는 모든 질문에 정확하게 응답함으로써 정당한 증명자임을 확인한다. 영지식 기반 인증 프로토콜에서 증명자는 자신이 가지고 있는 정보를 검증자에게 노출시키지 않으면서 그 정보를 가지고 있다는 사실을 검증자에게 확인시킬 수 있지만, 패스 수가 많기 때문에 처리 속도가 늦으며 사용자가 기억하기 어려운 정보를 이용하므로 정보저장을 위한 부가매체가 필요하다[7][8][11].

앞에서 기존에 제안된 각 인증 프로토콜을 검토한 결과 다음과 같은 연구의 제안 점을 도출할 수 있었다. 앞으로 인증 프로토콜은 다음과 같은 특성을 요구할 것이다. 첫째, 패스 수와 계산량이 작아야 하고, 둘째, 선시도 공격과 오프라인 추측 공격 및 패스워드 파일 컴프로마이즈 등과 같은 인증 프로토콜 공격들로부터 안전하며, 셋째, 사용자가 기억 가능한 정보를 처리하기 때문에 비밀정보를 저장하기 위한 부가매체가 필요 없는 인증 프로토콜이 필요함을 알 수 있다.

본 논문에서는 패스워드를 이용하는 인증 프로토콜을 제안하기 위하여 스키마 PAP( Password based Authentication Protocol)를 정의한다. PAP는 기본적으로 사용자가 기억 가능한 패스워드라 하더라도 패스워드를 표현하는 가지 수가 무한할 수 있다는 성질을 이용하고, 패스워드를 표현하는 많은 값들 중에서 임의로 선택한 한 값을 처리하기 위하여 공개 키 암호 시스템을 이용하는 것을 특징으로 한다.

PAP기반 인증 프로토콜에서 공격자는 1) 공개 키 암호 시스템을 분석하여 패스워드를 알려고 하거나 2) 패스워드 추측이나 패스워드 전사적 공격을 이용하여 패스워드를 알려고 하는 시도를 할 수 있다. PAP기반 인증 프로토콜에 대하여 공개 키 암호 시스템의 안전성과 공개 키 암호 시스템을 분석하지 않고 도청한 메시지로부터 패스워드를 유추하려는 공격에 대한 안전성을 동시에 고려한 용어로 Powerfully secure를 정의한다.

패스워드를 표현하는 값을 처리하기 위하여 RSA를 이용하는 PAP기반 인증 프로토콜 PAPRSA를 제시한다. PAPRSA는 선시도 공격과 오프라인 추측 공격으로부터 Powerfully secure하며, 패스워드 파일 컴프로마이즈로부터 안전하다. PAPRSA는 패스 수가 1 이고 계산량이 작아 효율성 면에서도 우수하다.

논문의 구성은 다음과 같다. Ⅲ장에서는 PAP를 정의하고, Ⅳ장에서는 PAPRSA를 제안한다.

## Ⅱ. 인증 프로토콜에 대한 공격

인증 프로토콜에 대한 공격은 도청을 전제로 한다.

1) 재전송(Replay): 공격자가 사용자간 통신 중에 획득한 메시지를 그대로 재사용하여 정당한 사용자로 위장하는 공격이다.

2) 선시도(Pre-paly): 공격자가 현재 통신 메시지를 이용해서 다음 통신 메시지를 결정하고, 결정된 메시지를 이용하여 정당한 사용자로 위장하는 공격이다.

3) Man-in-the-middle: 송수신 메시지를 이용하여 증명자에게는 검증자로 위장하고 검증자에게는 증명자로 위장하는 공격이다. 주로 상호 인증 프로토콜이나 패스 수가 일정 횟수 이상인 프로토콜에 적용되는 공격이다.

4) 패스워드 추측 공격: 공격자는 기본적으로 패스워드일 가능성이 높은 것들을 모아 놓은 사전을 이용하며, 온라인 추측 공격과 오프라인 추측 공격으로 구분된다.

· 오프라인 추측 공격은 사용자간 통신 메시지를 가로채어 패스워드 사전에 있는 값과 비교하고, 일치되는 값을 유도하는 패스워드를 이용하여 사용자로 위장하는 공격이다[1-4][9][10]. 오프라인 추측 공격을 위하여 공격자는 추측된 패스워드와 추측된 패스워드를 처리한 결과(추측한 패스워드를 프로토콜에서 암호 알고리즘이나 함수를 이용하여 처리한 결과)를 사전에 저장해 놓는 방법이 사용될 수 있다.

· 온라인 추측 공격에서 공격자는 사전에 있는 패스워드를 하나씩 차례로 선택하여 시도하는 과정을 유효한 패스워드가 나타날 때까지 반복한다. 현실에서 온라인 추측 공격은 패스워드 유효기간을 설정하거나 실패 수의 제한, 사용자 로긴 시도의 규칙을 만들어 방어가 가능하므로 본 논문에서는 온라인 추측 공격을 고려하지 않는다[12][15].

5) 서버 컴프로마이즈 : 공격자는 기본적으로 서버에 저장된 비밀정보를 이용하여 공격자가 서

**Registering procedure**
**Input** : $id$, 패스워드 $P$.
(1) 공개 키 암호 시스템의 공개키와 개인키를 구한 후 공개키를 공개한다;
(2) $x_1$을 랜덤하게 선택하고, $x_1 \odot x_2 = P$ 또는 $x_1 \odot_1 x_2 \odot_2 x_3 = P$인 $x_2$를 결정한다;
($\odot, \odot_1, \odot_2$는 연산자이며 $x_3$은 증명자와 검증자가 공유하고 있는 값이다.)
(3) 암호 함수 또는 알고리즘을 이용하여 반드시 $x_1$을 다른 값으로 사상하고, $x_2$를 선택적으로 사상한다;
(4) (3)에서 사용하는 암호 함수 또는 알고리즘을 $F$라 할 때 $(F(x_1), x_2)$ 또는 $(F(x_1), F(x_2))$를 $id$에 저장한다.

**Authenticating procedure**
**Input** : $id$, 패스워드 입력시 사용자에 의해 입력된 값 $PC$.
(1) $y_1$을 랜덤하게 선택하고, $y_1 \odot y_2 = P0$ 또는 $y_1 \odot_1 y_2 \odot_2 y_3 = P0$인 $y_2$를 결정한다;
(2) 증명자는 검증자의 공개키를 이용하여 반드시 $y_1$을 암호화하고, $y_2$를 선택적으로 암호화한다;
(3) (2)에서 $y_2$를 증명자로 암호화하는 경우에는 두개의 암호문을 검증자에게 전송하고, $y_1$만을 암호화한 경우에는 $y_1$에 대응하는 암호문과 $y_2$을 전송한다.
(4) 검증자는 두 정보를 비교하여 $P = P0$ 인지 아닌지 결정한다;

그림 1: 패스워드 기반 인증 프로토콜 스킴

버로 위장하거나, 공격자가 사용자로 위장하는 경우가 있다[12][17][19][21].

## Ⅲ. 패스워드 기반 인증 프로토콜 스킴(PAP)

PAP는 등록 프로시저(Registering procedure)와 인증 프로시저(Authenticating procedure)로 구성된 인증 프로토콜 스킴이다.

PAP에서 패스워드로부터 유도되는 값은 $(x_1, x_2)$와 $(y_1, y_2)$이다. 그리고, $(x_1, x_2)$또는 $x_1$은 검증자에게 전송되기 전에 공개 키 암호 시스템에 의해 암호화된다. 따라서 공격자가 메시지를 도청한 후 패스워드를 알기 위해서는 1) 공개 키 암호 시스템을 분석하거나 2) 패스워드 추측이나 패스워드에 대한 전사적 공격과 같은 시도를 하여야 한다. 이러한 이유로, PAP기반 인증 프로토콜에서는 공개 키 암호 시스템을 분석하여 패스워드를 알려고 하는 경우와 공개 키 암호 시스템을 분석하지 않고 도청한 메시지로부터 패스워드를 유추하려는 공격에 대한 안전성을 동시에 고려하여야 한다.

[정의 2] 도청한 메시지를 이용하는 공격에 대하여, PAP기반 인증 프로토콜의 안전성이 공개 키 암호 시스템의 안전성에 의존한다면 PAP기반 인증 프로토콜을 강력한 보안(powerfully secure)이라고 한다.

## Ⅳ. 패스워드 기반 인증 프로토콜 응용(PAPRSA)

### 1. 기호 표기

- $id$: 사용자 신원.
- $P$: 패스워드.
- $PC$: 검증자가 증명자에게 패스워드 입력을 유도할 때, 사용자에 의해 입력된 값.
- $p, q$: RSA에 적합한 두 소수.
- $N$: $N = pq$
- $\phi(n)$: $(p-1)(q-1)$.
- $e$: $\phi(n)$과 서로 소인 정수.
- $d$: $ed \equiv 1 \bmod \phi(n)$.
- $Z$: $\{1, 2, ..., N-1\}$.
- $t$: 타임스탬프.
- $Z$의 원소: $x_1$, $x_2$, $y_1$, $y_2$.

### 2. PAPRSA 프로토콜

(그림 2)는 PAPRSA에 대한 프로토콜을 기술한 것이며 이에 대한 특성은 다음과 같다.

[성질 1] PAPRSA에서 검증자는 인가된 증명자인지를 정확하게 결정 가능하다.

(증명) 검증자는 증명자로부터 수신한 암호문을 개인키를 이용하여 복호화하여 $(x_1, x_2)$를 얻을 수 있다. 증명자는 $x_1 - x_2 = y_1 - y_2$ 라는 사실을 알고 있으므로, $(x_1-x_2)^2 \bmod N = (y_1-y_2)^2 \bmod N$ 인지의 여부를 검사하여 현재 시스템 접근자가 인가된 검증자인가를 결정할 수 있다.

**Registering procedure**
**Input** : $id$, $P$.
(1) Get $N = pq$ and $(e, d)$;
(2) Publish $e$ and $N$;
(3) Choose $y_1$ in random and then determine $y_2$ so that $y_1 - y_2 = P$;
(4) Get $y_1^2 \bmod N$, $y_2^2 \bmod N$ and then store them at $id$.

**Authenticating procedure**
**Input** : $id$, $P$.

| Prover | | Verifier |
|---|---|---|
| $id$, $P$, $e$, $N$ | | $(e,d)$, $(p,q)$, $N$, $y_1^2 \bmod N$, $y_2^2 \bmod N$ |
| Pick $x_1$ in random | | |
| Determine $x_2$ so that $x_1 - x_2 = P$ | | |
| Get $t$ | | |
| Compute $(x_1+t)^e \bmod N$, $x_2^e \bmod N$ | | |
| | $t$, $(x_1+t)^e \bmod N$, $x_2^e \bmod N$ → | |
| | | $x_1 + t = (x_1+t)^{ed} \bmod N$, $x_2 = x_2^{ed} \bmod N$ |
| | | $x_1 = x_1 + t - t \bmod N$ |
| | | $(x_1-x_2)^2 \bmod N \stackrel{?}{=} (y_1-y_2)^2 \bmod N$ |

그림 2: PAPRSA 프로토콜

## V. PAPRSA 프로토콜 분석

본 장에서는 PAPRSA의 프로토콜에 대한 안전성과 효율성을 분석한다.

### 1. 안전성

PAPRSA 프로토콜 안전성은 기본적으로 패스워드를 표현하는 가능한 가지 수가 Z개라는 사실을 이용하며, 패스워드를 표현하기 위하여 선택한 값의 안전성은 RSA의 안전성에 의존한다.

[정리 2] PAPRSA는 선시도 공격과 오프라인 추측 공격, Man-in-the-Middle 공격으로부터 Powerfully secure하다.

(증명) 집합 $S$를 $\{ (z_1^i, z_2^i) \mid 1 \le i \le N-1,$ $z_1^i + z_2^i = P\}$라 하자. 그러면 $(x_1, x_2) \in S$ 이고 $x_1$이 집합 $j \mid 1 \le j \le N-1$에서 임의로 선택된 값이므로 $(x_1, x_2)$는 $S$에서 임의로 선택된 두 수로 이루어진 순서쌍이다. 공격자가 증명자로부터 검증자에게 전송되는 메시지를 도청하였다고 가정 할 때, 공격자가 사용자로 위장하기 위해서는 공격자는 RSA에 의해 암호화된 암호문으로부터 $(x_1, x_2)$을 결정 가능하여야 한다. 그런데, RSA에서 $p$와 $q$를 결정하기 위한 공간은 $N-1$보다 작으므로 공격자가 $(x_1, x_2)$을 결정하기 위한 공간은 RSA에서 $p$와 $q$를 결정하기 위한 공간보다 크다.

▪ 재전송: PAPRSA에서 타임스탬프는 현재 메시지와 이전 메시지가 다르도록 한다.

▪ 패스워드 파일 컴프로마이즈: 제곱근을 찾는 문제(SQROOT problem)는 정수 $n$과 $a^2 \bmod n$이 주어졌을 때 $a$를 찾는 문제이다[15]. 그리고, RSA 문제(주어진 두 소수의 곱으로부터 두 소수를 결정하는 문제)는 $n$이 두 소수의 곱일 때의 SQROOT 문제로 다항시간에 변환 가능하다(polynomial time reducible)[15]. 그러므로, RSA 문제와 SQROOT 문제는 계산적으로 동치이다

(Computationally equivalent). 따라서 PAPRSA는 공격자가 1) ($y_1^2 \bmod N$, $y_2^2 \bmod N$)만을 컴프로마이즈하는 경우에는 안전하고, 2) 도청을 하지 않는 것을 전제로 개인키의 컴프로마이즈에 안전하며, 3) 도청을 하지 않는 것을 전제로 $(e,d)$, $(p,q)$, $N$, $y_1^2 \bmod N$, $y_2^2 \bmod N$의 컴프로마이즈에 안전하다.

다음 (표 1)은 PAPRSA 프로토콜의 효율성을 분석한 것이다.

PAPRSA는 사용 환경에 따라 증명자가 검증자의 공개키를 저장하고 있는 경우와 그렇지 않은 경우로 분류할 수 있다. 그림 3은 증명자가 검증자의 공개키를 저장하고 있지 않은 환경에서 PAPRSA가 어떻게 변형 가능한지를 나타내고 있다. 증명자가 검증자의 공개키를 저장하고 있지 않은 경우의 PAPRSA 패스 수는 3이다.

**Registering procedure**
**Input** : $id$, $P$.
(1) Get $N = pq$ and $(e,d)$;
(2) Publishes $e$ and $N$;
(3) Choose $y_1$ in random and then determine $y_2$ so that $y_1 - y_2 = P$;
(4) Get $y_1^2 \bmod N$, $y_2^2 \bmod N$ and then store them at $id$.

**Authenticating procedure**
**Input** : $id$, $P$.

| **Prover** | | **Verifier** |
|---|---|---|
| $id$, $P$ | | $(e,d)$, $(p,q)$, $N$, $y_1^2 \bmod N$, $y_2^2 \bmod N$ |
| | $id$ → | |
| | $e, N$ → | |
| Pick $x_1$ in random | | |
| Determine $x_2$ so that $x_1 - x_2 = P$ | | |
| Get $t$ | | |
| Compute $(x_1+t)^e \bmod N$, $x_2^e \bmod N$ | | |
| | $t$, $(x_1+t)^e \bmod N$, $x_2^e \bmod N$ → | |
| | | $x_1 + t = (x_1+t)^{ed} \bmod N$, $x_2 = x_2^{ed} \bmod N$ |
| | | $x_1 = x_1 + t - t \bmod N$ |
| | | $(x_1 - x_2)^2 \bmod N \stackrel{?}{=} (y_1 - y_2)^2 \bmod N$ |

그림 3: 공개키를 저장하지 않은 PAPRSA 프로토콜

표 1: PAPRSA 프로토콜의 효율성 분석

| 인자 | Prover | Verifier |
|---|---|---|
| 패스 | 1 | 0 |
| 난수 생성 | 1 | 0 |
| RSA 암호화 | 2 | 0 |
| RSA 복호화 | 0 | 2 |
| 모듈러 곱셈 | 0 | 2 |

## 2. 효율성

## Ⅵ. 결론 및 향후 연구과제

정보시스템에서는 특정 시스템에 로그인하기 위하여 사용자의 신원을 확인하는 인증 과정과 통신로와 서버 상에서 비밀정보를 안전하게 전송하는 인증 프로토콜을 이용한다.

현재까지 제안된 인증 프로토콜들은 패스워드 기반 방식과 시도-응답 방식, 영-지식 증명 방식 등이 있다. 패스워드 기반 방식은 사람이 기억 가능한 패스워드를 사용하지만 선시도 공격, 오프라

인 추측 공격 및 패스워드 파일 컴프로마이즈와 같은 인증 프로토콜 공격에 안전하지 않다. 시도-응답 방식과 영지식 증명 방식은 랜덤값에 의한 시도를 사용하므로 선택 평문(암호문)공격에 노출될 수 있고, 사용자가 기억하기 어려운 키를 사용함으로 키를 저장하는 부가 매체가 필요하다. 이처럼 기존에 제안된 인증 프로토콜로부터 패스 수와 계산량이 적고, 선시도 공격과 오프라인 추측 공격 및 패스워드 파일 컴프로마이즈 등과 같은 공격들로부터 안전하며, 사람이 기억 가능한 정보를 이용하기 때문에 비밀정보를 저장하기 위한 부가매체가 필요 없는 인증 프로토콜이 필요함을 알 수 있다.

본 논문에서는 PAP라는 인증 프로토콜 스킴을 정의하였다. PAP는 패스워드를 표현하는 공간(Space)에서 임의의 값을 선택하여 패스워드를 표현하며, 공격자에게 어떤 정보도 주어지지 않았을 때 어떤 값이 선택되었는지를 결정하는 것은 계산상 불가능한 (Computationally infeasible) 공간에서 임의로 선택되는 것을 특징으로 한다. PAP는 공개 키 암호 시스템의 안전성과 공개 키 암호 시스템을 분석하지 않고 도청한 메시지로부터 패스워드를 유추하려는 공격에 대한 안전성을 동시에 고려하였다. PAP기반으로 패스워드를 표현하는 값을 암호화하기 위하여 RSA를 이용하는 인증 프로토콜 PAPRSA를 제안하였다. PAPRSA는 선시도 공격과 오프라인 추측 공격 등으로부터 강력히 안전하고, 도청을 하지 않는 컴프로마이즈로부터 안전하다. PAPRSA는 패스 수가 1이며, RSA 암호화와 복호화를 2번 수행한다. 그리고 난수를 1회 생성하며 모듈로 제곱을 2회 수행을 통하여 안전성과 효율성을 보장할 수 있었다.

향후에는 1) 다양한 컴프로마이즈 공격으로부터 안전한 인증 프로토콜을 설계하고, 2) PAP 스킴에 기반한 식별 프로토콜을 설계하며, 3) 패스워드 확장 개념을 상호 인증 및 세션 키 공유 프로토콜에 적용한 스킴과 프로토콜을 설계하고자 한다.

## 참고문헌


[1] M. Bellare, D. Pointcheaval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks", Advances in Cryptology Eurocrypt'00, LNCS Vol. 1807, Springer-Verlag, pp. 139-155, 2000.

[2] S. M. Bellovin and M. Merrit, "Augmented encrypted key exchange: Password-based protocol secure against dictionary attack and password file compromise", In ACM Security (CCS'93), pp. 244-250, 1993.

[3] S. M. Bellovin and M. Merrit, "Encrypted key exchange: Password-based protocols secure against dictionary attack", In Proceedings of IEEE Security and Privacy, pp. 72-84, 1992.

[4] V. Boyko, P. MacKenzie, and S. Patal, "Provably secure password authenticated key exchange using Diffie-Hellman", In B. Prenel, editor, Advances in Cryptology Eurocrypt'00, LNCS Vol. 1807, Springer-Verlag, pp. 156-171, 2000.

[5] W. Diffie and H. E. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, 22, pp. 644-654, 1976.

[6] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," IEEE Transactions on Information Theory, v. IT-31, n. 4, pp. 469-472, 1985.

[7] U. Feige, A. Fiat and A. Shamir, "Zero knowledge proof of identity", Journal of Cryptology, Vol. 1, pp. 77-94, 1983

[8] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems", Advances in Cryptology-CRYPTO' 86, LNCS 263, pp. 186-194, 1987.

[9] L. Gong, "Optimal authenticationprotocols resistant to password guessing attacks", In 8th IEEE Computer Security Foundations Workshop, pp. 24-29, 1995.

[10] L. Gong, T. M. A. Lomas, R. M. Needham, and J. H. Saltzer, "Protecting poorly chosen secrets from guessing attacks", IEEE Journal on Selected Areas in Communications, 11(5), pp. 648-656, June 1993.

[11] L. C. Guillou and J. J. Quisquater, "A practical zero-knowledge protocol to security microprocessor minimizing both transmission and memory", Advances in Cryptology-EUROCRYPT ' 88, LNCS 330, pp. 123-128, 1988.

[12] S. Halevi and H. Krawczyk, "Public-key cryptography and password protocols,"ACM Security (CCS' 98), pp. 122-131.

[13] ISO/IEC 9798-2, "Information technology-Security techniques-Entity authentication-Part 2: Mechanisms using symmetric encipherment algorithms", International Organization for Standardization, Geneva, Switzerland, 1994.

[14] ISO/IEC 9798-4, "Information technology-Security techniques-Entity authentication-Part 4: Mechanisms using a cryptographic check function", International Organization for Standardization, Geneva, Switzerland, 1995.

[15] D. Jablon, "Strong password-only authenticated key exchange", ACM Computer Communication Review, ACM SIGCOMM, Vol. 26, No. 5, pp. 5-20, October 1996.

[16] N. Koblitz, "Eliptic curve cryptosystems,"Mathematics of Computation, v. 48, n. 177, pp. 203-209, 1987.

[17] L. Lamport, "Password authentication with insecure communication", Communications of the ACM, Vol. 24, pp. 770-772, 1981.

[18] R. J. McEliece, "A public key cryptosystem based on algebraic coding theory," Deep Space Network Progress Report 42-44, Jet Propulsion Laboratory, California Institute of Technology, pp. 42-44, 1978.

[19] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, Applied Cryptography, CRC press, 1997.

[20] R. C. Merkle, Secrecy, Authentication, and Public Key Systems, UMI Research Press, Ann Arbor,

Michigan, 1979.

[21]R. Morris and K. Thompson, "Password security: a case history", Communications of the ACM, Vol. 22, pp. 594-597, 1979.