A multi-round bilinear-map-based secure password hashing scheme

Csanád Bertók, Andrea Huszti, Tamás Kádek, Zsanett Jámbor

University of Debrecen, Faculty of Informatics

IEEE 2nd Conference on Information Technology and Data Science

Debrecen, 2022

Topics of the presentation

Introduction

- *Password usage*
  - *Authentication:* password (+salt)$\longrightarrow$hash (+salt)
  - *Key generation:* PAKE, PBKDF

Multiple attacks against weak or not correctly stored passwords.

- 1Password (2017)
- Tesla SolarCity Solar Monitoring Gateway (2019)
- Passwordstate (2021)

Our contribution

To protect users, services,... against attacks, several password hashing schemes/functions have been proposed and used.

- PBKDFv2
- Argon2 (winner of PHC 2015)
- bcrypt

We construct a secure PHS based on bilinear pairing with the following properties:

- Multi-round
- Adjustable cost factor

Off-line attacks

- (Mostly) salt (and hash) independent attacks
  - Brute force
  - Dictionary
- Attacks against hashes (mostly mitigated by salt)
  - Rainbow-tables

$$p_{i,1} \xrightarrow{H} c_{i,1} \xrightarrow{R} p_{i,2} \xrightarrow{H} c_{i,2} \xrightarrow{R} p_{i,3} \rightarrow \cdots \rightarrow p_{i,k} \xrightarrow{H} c_{i,k}$$

Preliminaries

Admissible bilinear map

Let $\mathbb{G}$ be an additive and $\mathbb{G}_{\mathbb{T}}$ a multiplicative group of order p for some large prime p. A map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{\mathbb{T}}$ is an admissible bilinear map if it satisfies the following properties:

1. Bilinear: We say that a map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{\mathbb{T}}$ is bilinear if $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}$ and all $a, b \in \mathbb{Z}$.
2. Non-degenerate: The map does not send all pairs in $\mathbb{G} \times \mathbb{G}$ to the identity in $\mathbb{G}_{\mathbb{T}}$. Since $\mathbb{G}$, $\mathbb{G}_{\mathbb{T}}$ are groups of prime order, if P is a generator of $\mathbb{G}$ then $\hat{e}(P, P)$ is a generator of $\mathbb{G}_{\mathbb{T}}$.
3. Computable: There is an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in \mathbb{G}$.

Preliminaries

For elliptic curve based cryptography usually

- $\mathbb{G}$ is an elliptic curve group (a subgroup of the r-torsion)
- $\mathbb{G}_T$ is the roots of unity in a finite field

Associated problem:

Computational Diffie-Hellman Problem

Let $\mathbb{G}$ be a cyclic group with generator $G \in \mathbb{G}$ and let xG, $yG \in \mathbb{G}$. The Computational Diffie-Hellman Problem is to compute xyG.

Preliminaries

Mapping into elliptic curves

- $q \equiv 3 \pmod 4$ prime
- $E: y^2 = x^3 + ax$ over $\mathbb{Z}_q$

$$tr : \mathbb{Z}_q \longrightarrow E(\mathbb{Z}_q)$$
$$x \mapsto \left(\varepsilon(x) \cdot x, \varepsilon(x)\sqrt{\varepsilon(x) \cdot (x^3 + ax)}\right),$$

where $\sqrt{\cdot}$ is the square root over $\mathbb{Z}_q$ and $\varepsilon(x) = \left(\frac{x^3+ax}{q}\right)$, where $\left(\frac{\cdot}{q}\right)$ is the Legendre symbol.

The proposed scheme

Requirements based on PHC

- Password length between 0 and 128 bytes
- Salt length 16 bytes
- Output length minimum 32 bytes
- Configurable time and/or memory cost

Our algorithm fulfills all the criteria, the configurable parameter is the time (t_cost) which can be adjusted by increasing / decreasing the number of rounds.

The proposed scheme

Algorithm The proposed algorithm

INPUT: password

OUTPUT: PswStore, S

Initialize $E(\mathbb{Z}_q)$

Initialize S

$PswStore \leftarrow Convert(password)$

for $i = 0$ up to number of rounds **do**

    $R \leftarrow hashToCurve(PswStore)$

    $PswStore \leftarrow TatePairing(R, S + iG)$

    $PswStore \leftarrow Convert(PswStore)$

return $(PswStore, S)$

Security analysis

The following security requirements were considered:

- Pre-image resistance (bilinear pairing is one-way)
- Second pre-image resistance
- Collision resistance

Pre-image resistance

Let $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map. Let $\langle G \rangle = \mathbb{G}$ and $\langle g \rangle = \mathbb{G}_T$ be any elements such that $\hat{e}(G, G) = g$. If the CDH problem is infeasible for $g, g^a, g^b \in \mathbb{G}_T$ with any $a, b \in \mathbb{Z}_q$, then $\hat{e}$ is a one-way pairing.

Thus CHD hard $\Longrightarrow$ pre image resistance.

Security analysis

Collision resistance $\Longrightarrow$ second pre-image resistance

Collision resistance

- Bilinear pairing considered over torsion groups of E
- The r-torsion has $r + 1$ cyclic groups
- Same subgroup$\longrightarrow$same result
- Probability of collision for our curve and prime $\sim 10^{-48}$

Efficiency analysis - running time

Comparing with bcrypt and RSA (running time measured in seconds)

# of rounds	bcrypt	Our algorithm	RSA
16	0,0030458	0,5346845	0,009764
32	0,0037977	0,8726219	0,0090346
64	0,0069453	1,7379774	0,0251674
128	0,0130193	1,5386831	0,0334561
256	0,023243	3,5953085	0,0638214
512	0,0431535	5,3515215	0,1371731
1024	0,087049	10,3966082	0,2013071
2048	0,167253	20,9222832	0,452279
4096	0,3439718	46,5067361	0,7515071
8192	0,6667411	86,7408044	1,3365767

Efficiency analysis - memory usage, LoC

Memory usage - limited to 1 second of runtime

Python memory profiler module

Argon2 $\longrightarrow$ $20,1$ MiB

bcrypt $\longrightarrow$ $20,2$ MiB

Our algorithm $\longrightarrow$ $22,0$ MiB

For the number of lines of code (LoC) our algorithm is between bcrypt and Argon2, however this is not a factor which can be measured precisely.

THANK YOU FOR YOUR ATTENTION!