

# The Evolution of Authentication

---

Andy Li

OWASP Vancouver 2020

# Agenda

- What is authentication?
- Passwords
- Multi-Factor Authentication (MFA)
- Beyond Passwords

# About me



- Security Engineer at Segment
- Software Engineer Intern at Bouncer
- Software Engineer Intern at Amazon
- Competitive Tetris player

# What is Authentication?

- Authentication (AuthN)
  - **“Who are you?”**
  - Basically, logging in
- Not to be confused with Authorization (AuthZ)
  - **“What are you allowed to do?”**
  - i.e. should you be able to click this button?

# How do you authenticate?

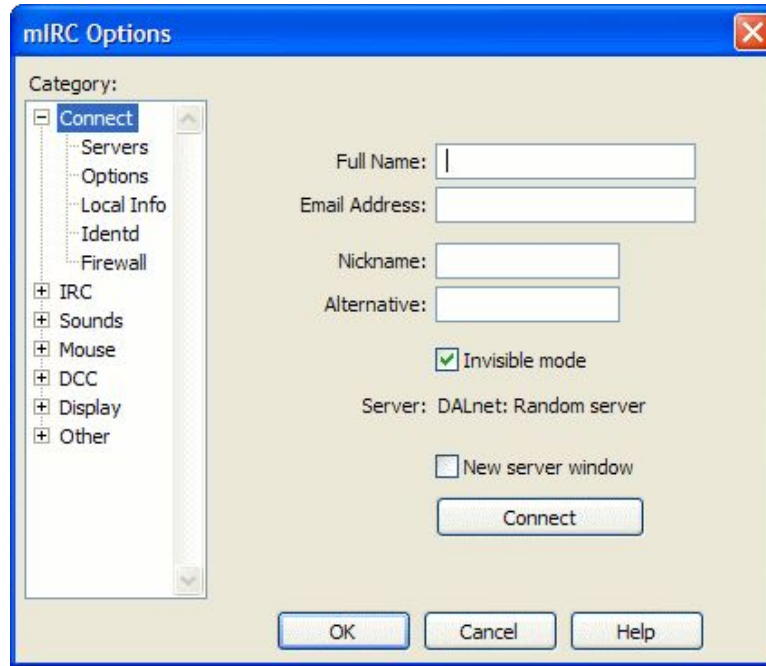
**Authentication is the process of generating a specific sequence of bytes**



# How do you authenticate?

- 3 categories:
  - Something you know (password, PIN)
  - Something you have (phone, security token)
  - Something you are (biometrics)

# No Authentication

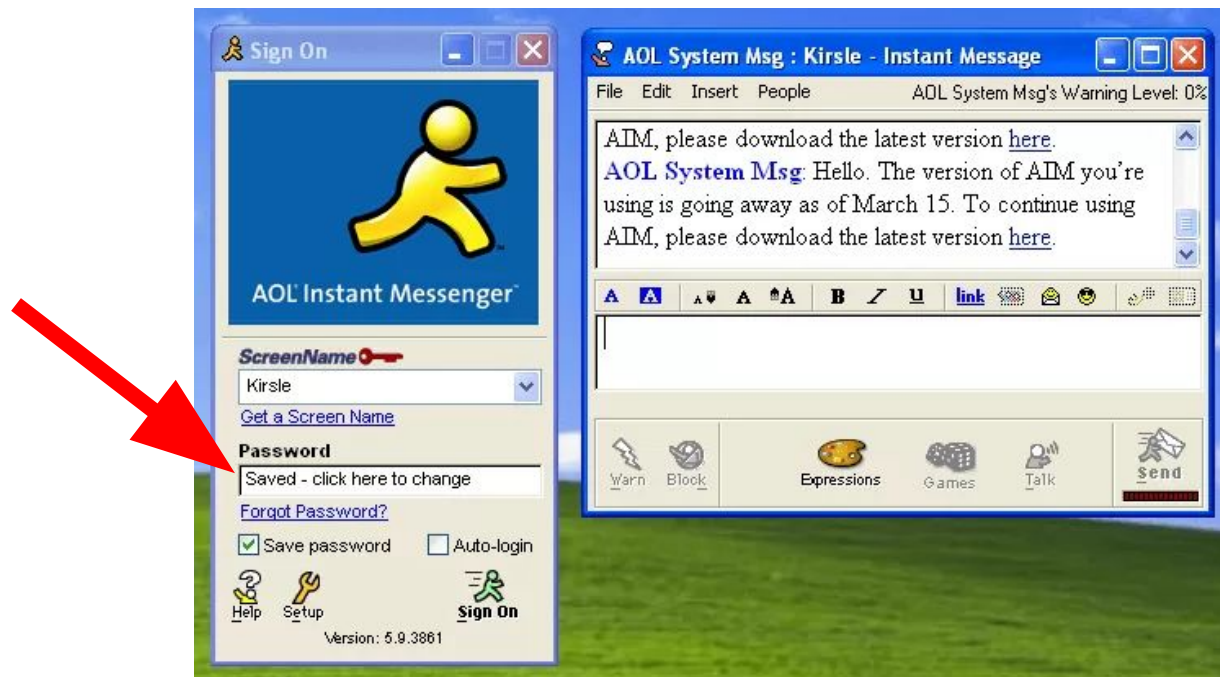


# How can you authenticate?

- 3 categories:
  - **Something you know (password, PIN)**
  - Something you have (phone, security token)
  - Something you are (biometrics)



# Passwords



# Plaintext

- Simple way to validate “something you know”
- Better ways to do it

# Plaintext: How the database looks



Username	Password
pro1337hacker	t0psecr3t

# Plaintext

## **21** Facebook Stored Hundreds of Millions of User Passwords in Plain Text for Years

MAR 19

### Zynga 2019 Hack Update: 26M Plaintext Passwords Exposed

Dec 21st, 2019

## Google stored some passwords in plain text for fourteen years

*Only affects some G Suite customers*

By [Dieter Bohn](#) | [@backlon](#) | May 21, 2019, 7:16pm EDT

# Plain Text Offenders

Follow plaintextoffenders

tumblr

Did you just email me  
back my own password?!

About

FAQ

Developers FAQ

Offenders List

3rd Party Tools

Reformed Offenders

Archive

Talk To Us

Submit a post

NOTE: Tumblr's search feature  
is broken and therefore  
disabled. Please use the list at  
plaintextoffenders.com/offenders  
to search for any domain.

June 5th, 2020 at 6:01PM

## AuctionZip Login Information



**Lost Login** <lostpass@auctionzip.com>

to: [redacted]

=====

This e-mail was sent in response to your request for your  
Lost/Forgotten AuctionZip login information.

Click on the link below to go to the AuctionZip Login Screen.

=====

<http://www.auctionzip.com/cgi-bin/azlogin.cgi>

Your login information is as follows:

E-Mail: [redacted]

Password: [redacted]

auctionzip.com

Auctions

# Hashing

One-way operation

- Not reversible

```
hash = hash_function(plaintext)
```

# Hash: How the database looks

Username	Password ( <u>not stored</u> )	Hash
pro1337hacker	t0psecr3t	d195111a97345cb8ac
Carl	t0psecr3t	d195111a97345cb8ac
sloth	TopSecret	c970a8bc56b8509de4

# Hash **with Salt**: How the database looks

Username	Password ( <u>not stored</u> )	Hash
pro1337hacker	t0psecr3t	d195111a97345cb8ac
Carl	t0psecr3t	d195111a97345cb8ac
sloth	TopSecret	c970a8bc56b8509de4

Username	Password ( <u>not stored</u> )	Salt	Hash
pro1337hacker	t0psecr3t	oJP1c6AkgZJO	a4b57836f9b23691ba
Carl	t0psecr3t	n1PkV660ZrKV	258c863292aafaecdc
sloth	TopSecret	pxDt4pUopu71	e1d811f51ea87b2e70



# Hashing — Key Derivation Function (KDF)

Cryptographic hash function

- **Cost factor** can scale with computational advancements
- Use these!
  - PBKDF2, bcrypt

```
hash = hash_function(plaintext)
```

```
hash = key_derivation_function(plaintext, cost_factor)
```

# 2012 LinkedIn Breach had 117 Million Emails and Passwords Stolen, Not 6.5M

---

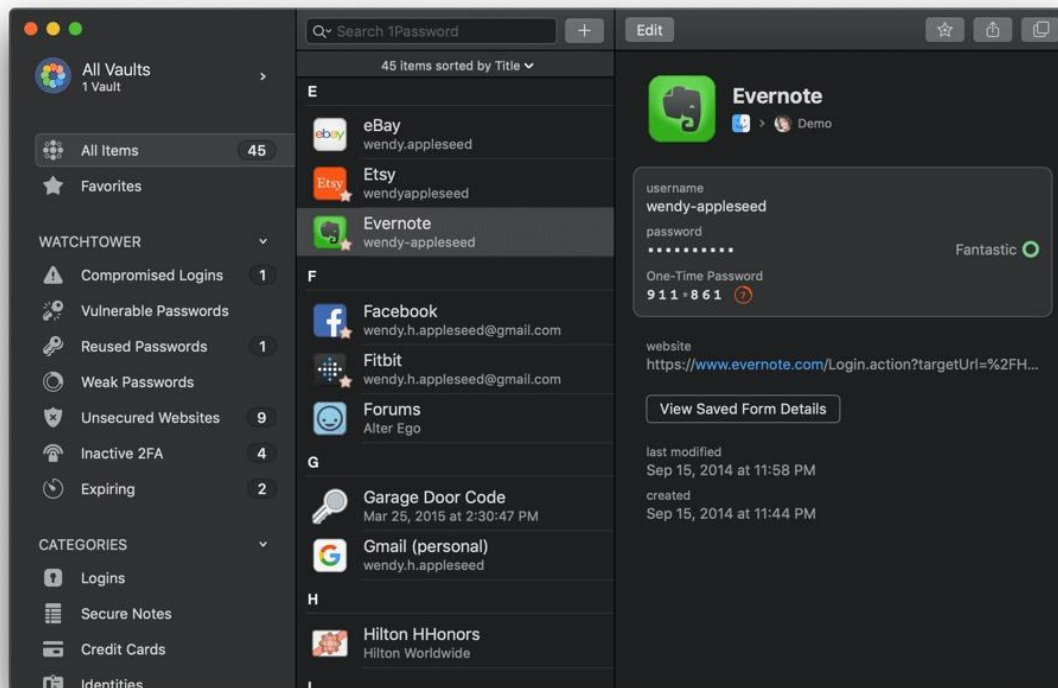
May 18, 2016

“Passwords were stored in SHA1 with no salting,” the password-selling site claims. “This is not what internet standards propose. Only 117m accounts have passwords and we suspect the remaining users registered using FaceBook or some similarity.”

# NIST Password guidelines

1. Users no longer have to use special characters
2. Users should be able to use all characters
3. It is reasonable to copy and paste passwords
4. Password policies should not require employees to change passwords on a regular basis
5. Increased character allowance

# Password manager

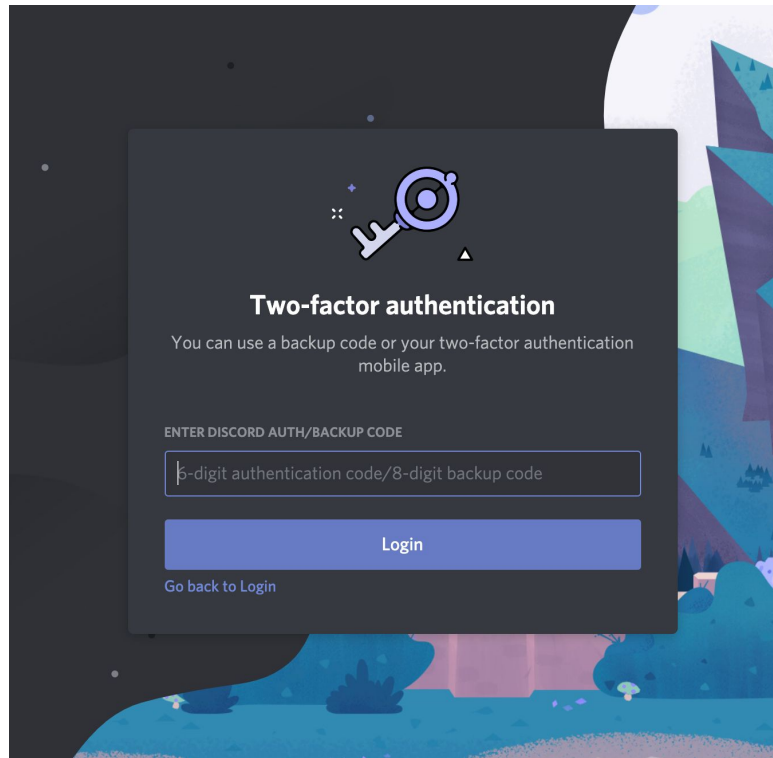


# How can you authenticate?

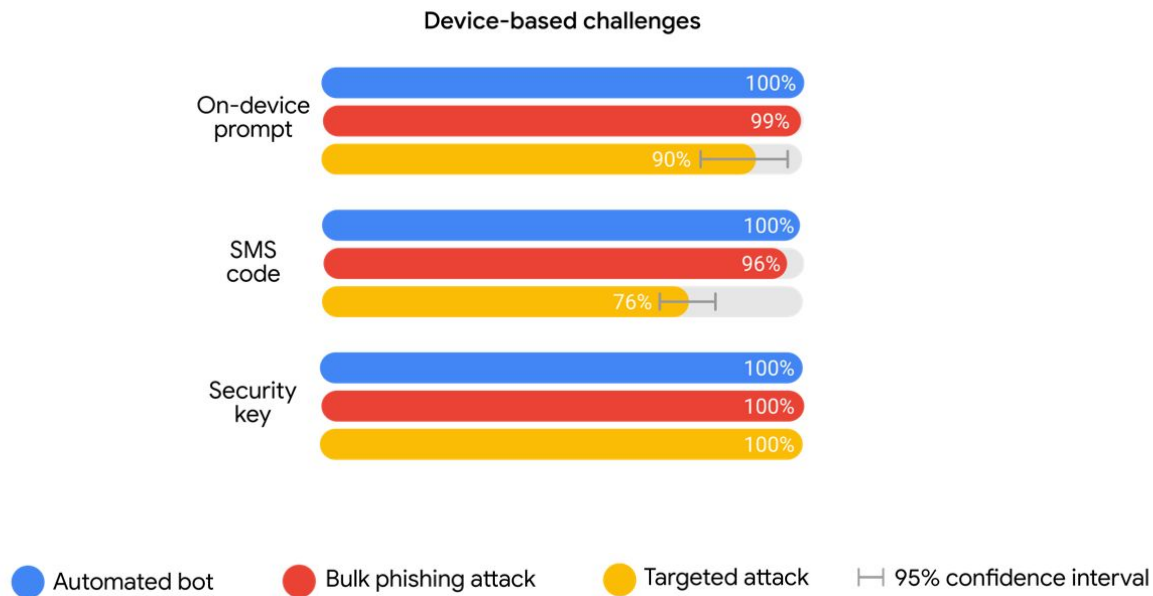
- 3 categories:
  - Something you know (password, PIN)
  - **Something you have (phone, security token)**
  - Something you are (biometrics)

# Multi-Factor Authentication (MFA)

- A way to supplement passwords
- Typically a prompt to enter a code (after entering your password)



# Account takeover stats by challenge type



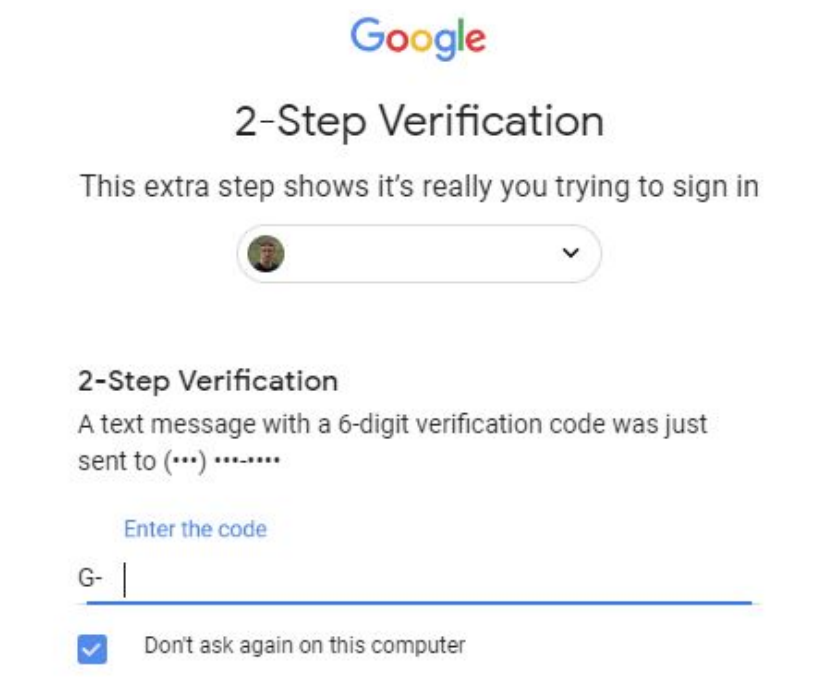
# Short-Message Service (SMS)

## Pros ✓

- Simple
  - Good for adoption
- Pretty good UX

## Cons ✗

- Weak against targeted attacks
  - Simjacking + message interception



The screenshot displays the Google 2-Step Verification process. At the top, the Google logo is centered. Below it, the heading "2-Step Verification" is shown, followed by the text "This extra step shows it's really you trying to sign in". A dropdown menu contains a profile picture and a downward arrow. Further down, the heading "2-Step Verification" appears again, followed by the text "A text message with a 6-digit verification code was just sent to (...) .....". A link "Enter the code" is provided. Below this is a text input field with a "G-" label and a blue underline. At the bottom, there is a checked checkbox and the text "Don't ask again on this computer".



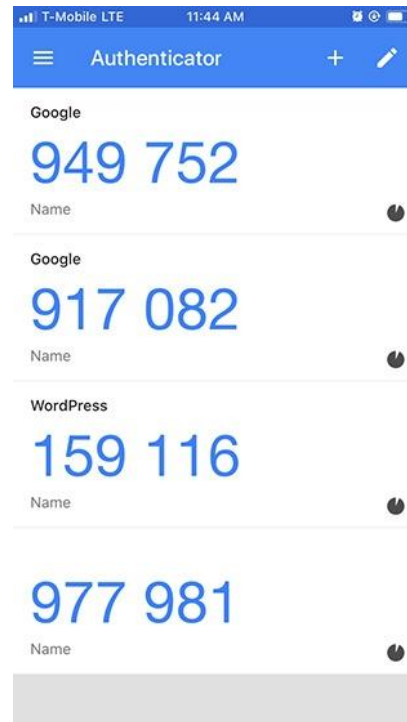
# Time-based One-Time Password (TOTP)

## Pros ✓

- Better protection than SMS against targeted attacks

## Cons ✗

- More complex
  - May need to hire more staff to troubleshoot
- Don't lose your phone



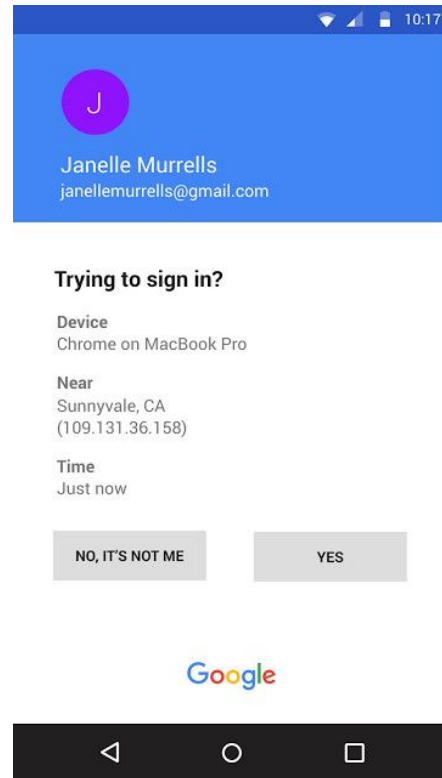
# Push Notifications

## Pros ✓

- Great UX
  - No codes to enter!

## Cons ✗

- Requires service to have a mobile app installed



# Hardware token

## Pros ✓

- Very secure
- Great UX
  - Just press the key!

## Cons ✗

- Requires additional hardware



# How can you authenticate?

- 3 categories:
  - Something you know (password, PIN)
  - Something you have (phone, security token)
  - **Something you are (biometrics)**

# Biometrics

Pros ✓

- Convenience

Cons ✗

- Probabilistic = margin of error
- Cannot be changed
- Expensive to implement



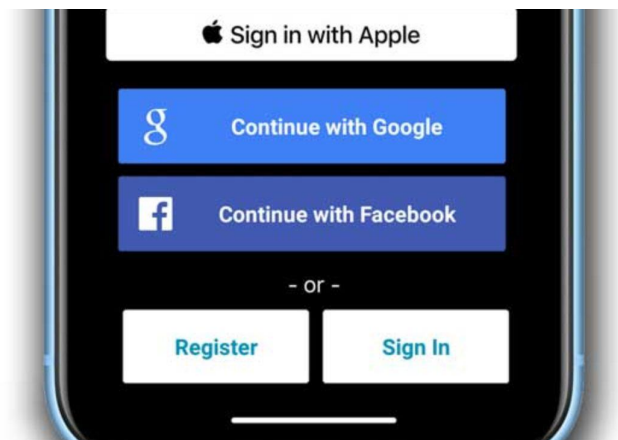
# Beyond Passwords

## Goals

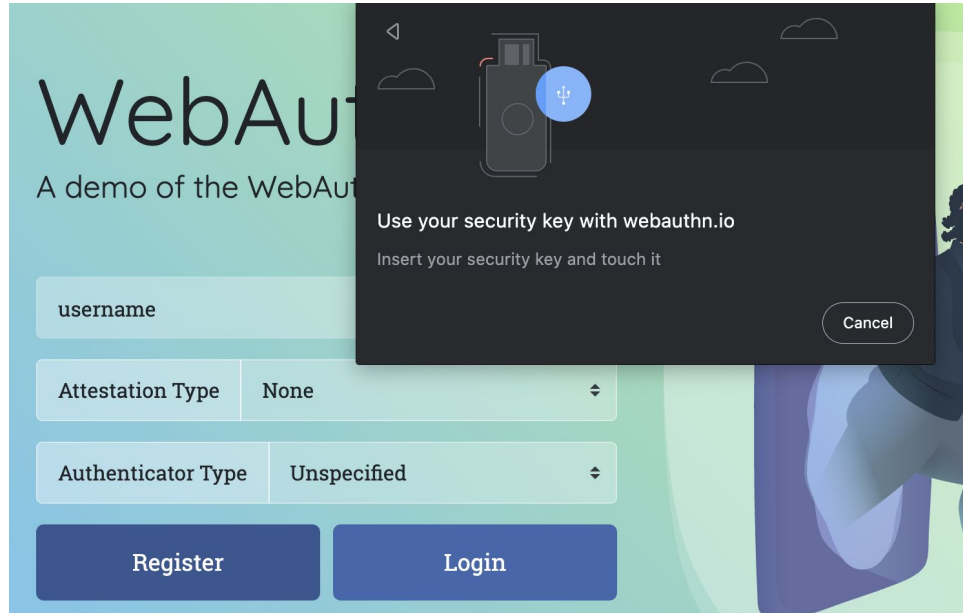
- Good UX
- Minimize bad practices for passwords
- Avoid phishing

# Single Sign-On (SSO)

- Authenticate once using an Identity Provider (IdP)
- Removes opportunity for poor password practices (creating + storing)



# WebAuthn





# Magic link

## Sign in to Hello

hello.slack.com

What is your password?

Sign in

☒ Remember me

[Forgot password?](#)

or

Long password? Hard to type?

We can email you a magic link so you can sign in without having to type your password.

 Email Me A Link



## An email is on its way!

We sent an email to **foorbar@mailinator.com**.

If this email address has an account on the **hello.slack.com** Slack workspace, you'll find a magic link that will sign you into that workspace.

The link expires in 24 hours, so be sure to use it soon.

## Go check your email!

Want to sign in to another workspace? Head [here](#).

# QR Code

**Welcome back!**  
We're so excited to see you again!


EMAIL

PASSWORD


[Forgot your password?](#)

Login


Need an account? [Register](#)




**Log in with QR Code**  
Scan this with the **Discord mobile app** to log in instantly.



11:25





**Are you trying to log in on the computer?**

Only scan QR codes taken directly from your browser. Never use a code sent to you by another user.

Yes, log me in

Cancel

# Summary

- Store passwords with
  - ✓ Key-Derivation Function
- Use MFA
  - ✓ Any MFA is vastly better than no MFA
- Use SSO
  - ✓ Good UX and avoids bad password practices



# We're Hiring

---

<https://segment.com/careers/>

# Appendix

# Encryption

Two-way function that relies on a secret

```
ciphertext = encrypt(plaintext, secret)
```

```
plaintext  = decrypt(ciphertext, secret)
```

# Encryption: How the database looks

Username	Encrypted Password
pro1337hacker	potOgmxâp

Key*	Value
database_encryption_key	MhKgkHnq0MJROUhYjWrS

\*key does not necessarily live in database

# Encryption

## **Adobe confirms stolen passwords were encrypted, not hashed**

In a breach **first announced on this blog Oct. 3, 2013**, Adobe said hackers had stolen nearly 3 million encrypted customer credit card records, as well as login data for an undetermined number of Adobe user accounts.