

---

—

# SECURITY (COMP0141): AUTHENTICATION



# DIGITAL CERTIFICATES

The screenshot shows a digital certificate viewer interface. On the left, a tree view displays the certificate chain: DigiCert Global Root CA, DigiCert SHA2 Secure Server CA, and \*.duckduckgo.com. The main panel shows the details for the \*.duckduckgo.com certificate, which is issued by DigiCert SHA2 Secure Server CA and expires on Wednesday, 10 November 2021 at 00:00:00 Greenwich Mean Time. A green checkmark indicates it is valid. The certificate is a Standard certificate. The subject name is \*.duckduckgo.com, and the issuer name is DigiCert SHA2 Secure Server CA. The certificate includes fields for Subject Name, Country, State/Province/County, Locality, Organisation, and Common Name. The issuer also has fields for Country, Organisation, and Common Name.

Subject Name	Country	US
State/Province/County	State/Province/County	Pennsylvania
Locality	Locality	Paoli
Organisation	Organisation	Duck Duck Go, Inc.
Common Name	Common Name	*.duckduckgo.com

Issuer Name	Country	US
Organisation	Organisation	DigiCert Inc
Common Name	Common Name	DigiCert SHA2 Secure Server CA

**q: does the client authenticate itself to the server?**

**a: no! we'll see client authentication later on.**

Summer Time  
Not Valid After Wednesday, 10 November 2021 at 00:00:00 (Greenwich Mean Time)

Algorithm RSA Encryption ( 1.2.840.113549.1.1.1 )  
Parameters None  
Public Key 256 bytes: AE 25 F8 F2 28 B4 61 93 4D 41 AA  
75 5E 23 6E 17 6C 5C 11 3F 5B F3 1C 83 0B BF

OK

Not After 10/11/2021, 00:00:00 (Greenwich Mean Time)

Subject Alt Names  
DNS Name \*.duckduckgo.com  
DNS Name duckduckgo.com

# AUTHENTICATION

---

Authentication is:

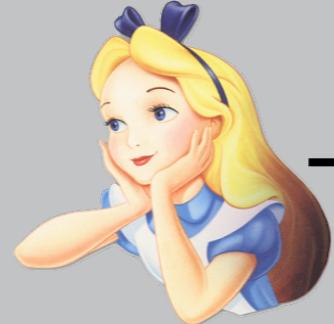
**What you know**

What you have

What you are

# PASSWORDS

---



uname,passwd



simple? how do you make them:

easy to memorise?

easy to enter?

hard to leak/steal?

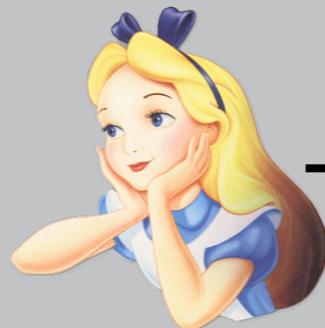
hard to guess?

resettable?

retrievable?

# STORING PASSWORDS

---



uname,passwd



uname,passwd here?

(RockYou breach leaked 32M passwords)

**problem:** password is revealed

**solution:** use a hash function

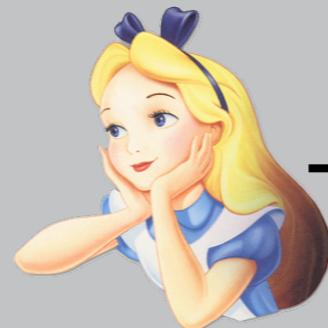
u1	p1
u2	p2
...	...
un	pn



passwords.csv

# STORING PASSWORDS

---



uname,passwd



uname,**H(passwd)** here?

problem: common passwords

( $H(p1) = H(p2)$  for  $p1 = p2$ )

u1	$H(p1)$
u2	$H(p2)$
...	...
un	$H(pn)$



passwords.csv

# COMMON PASSWORDS

# general purpose

# password(1)

123456

**qwerty(123)**

<names>

# service specific

hotmail

gmail

# dreamweaver

**macromedia**

# linkedin



# COMMON PASSWORDS

Adobe: 38 million password records stolen

asdx2      asd  
usual      asdasdd  
normal      letters

## Across

- ▶ 6: zk8NJgAOqc4=
- ▼ 7: WIMTLimQ5b4= asd; dsadsa; asdasdasd; a; qweqw; as; 123123; dsa; asdas; 123456; asdx2; asdasda; lol; asdasdasdasd; asdasd1; asd asd; 123; usual; ad; asd2; aaa; qwe; asd?; same; das; asdasdd; asd123; ???; ??; sss; none; dasdas; asddsa; ???; zxczxc; no; hi; aaaaaa; aa; qwerty; normal; la de siempre; easy; d; asdfgh; asdasd123; sdf; letters; keyboard; asda
- ▶ 8: FTeB5SkrOZM=
- ▶ 9: WqflwJFYW3+PszVFZo1Ggg==
- ▶ 10: yxzNxPlsFno=
- ▶ 11: L3uQHNDf6Mw=

## Down

- ▼ 1: 2aZl4Ouarwm52NYYI936YQ== adobe; adobex2; adobe2; adobe twice; twice; adobetwice; adobe2x; site; ?????; name; software; 2x; company; 2xadobe; programa; adobe x 2; program; adobe x2; ???; ad; adobe\*2; ???; Adobe; double; namename; 2adobe; ?????; x2; a; name twice; photoshop; company name; adobe adobe; adobe?; ado; aa; company twice; 2; marca; website; none; adobe 2x; product; company name twice; adobeX2; this; logiciel; ??; ???????; what is this
- ▶ 2: L8qbAD3jl3jSPm/keox4fA==
- ▶ 3: 7Z6uMyq9bpxe1EB7HijrBQ==
- ▶ 4: vp6d18mfGL+5n2auThm2+Q==
- ▶ 5: dA8D8OYD55E=

8

# COMMON PASSWORDS

Adobe: 38 million password records stolen

Reveal   Check   Hide

adobex2 adobe  
name company  
adobe twice

## Across

- ▶ 6: zk8NJgAOqc4=
- ▼ 7: WIMTLimQ5b4=
- asd; dsadsa; asdasdasd; a; qweqwe; as; 123122; dsa; asdas; 123456; asdx2; asdasda; lol; asdasdasdasd; asdasd1; asd asd; .23; usual; ad; asd2; aaa; qwe; asd?; same; das; asdasdd; asd123; ???; ??; sss; none; dasdas; asddsa; ???; zxczxc; no; hi; aaaaaa; aa; qwerty; normal; la de siempre; easy; d; asdfgh; asdasd123; sdf; letters; keyboard; asda
- ▶ 8: FTeB5SkrOZM=
- ▶ 9: WqflwJFYW3+PszVFZo1Ggg==
- ▶ 10: yxzNxPlsFno=
- ▶ 11: L3uQHNDf6Mw=

## Down

- 1: 2aZl4Ouarwm52NYYI936YQ==  
adobe; adobex2; adobe2; adobe twice; twice; adobetwice; adobe2x; site; ?????; name; software; 2x; company; 2xadobe; programa; adobe x 2; program; adobe x2; ???; ad; adobe\*2; ????; Adobe; double; namename; 2adobe; ?????; x2; a; name twice; photoshop; company name; adobe adobe; adobe?; ado; aa; company twice; 2; marca; website; none; adobe 2x; product; company name twice; adobeX2; this; logiciel; ??; ???????; what is this
- 2: L8qbAD3jl3SPm/keox4fA==
- 3: 7Z6uMyq9bpxe1EB7HijrBQ==
- 4: vp6d18mfGL+5n2auThm2+Q==
- 5: dA8D8OYD55E=

9

# PASSWORD CRACKING

---



goal: recover common passwords

$H(\text{password})$   
 $H(123456)$   
 $H(asdasd)$   
 $H(qwerty)$   
 $H(john)$

equal?

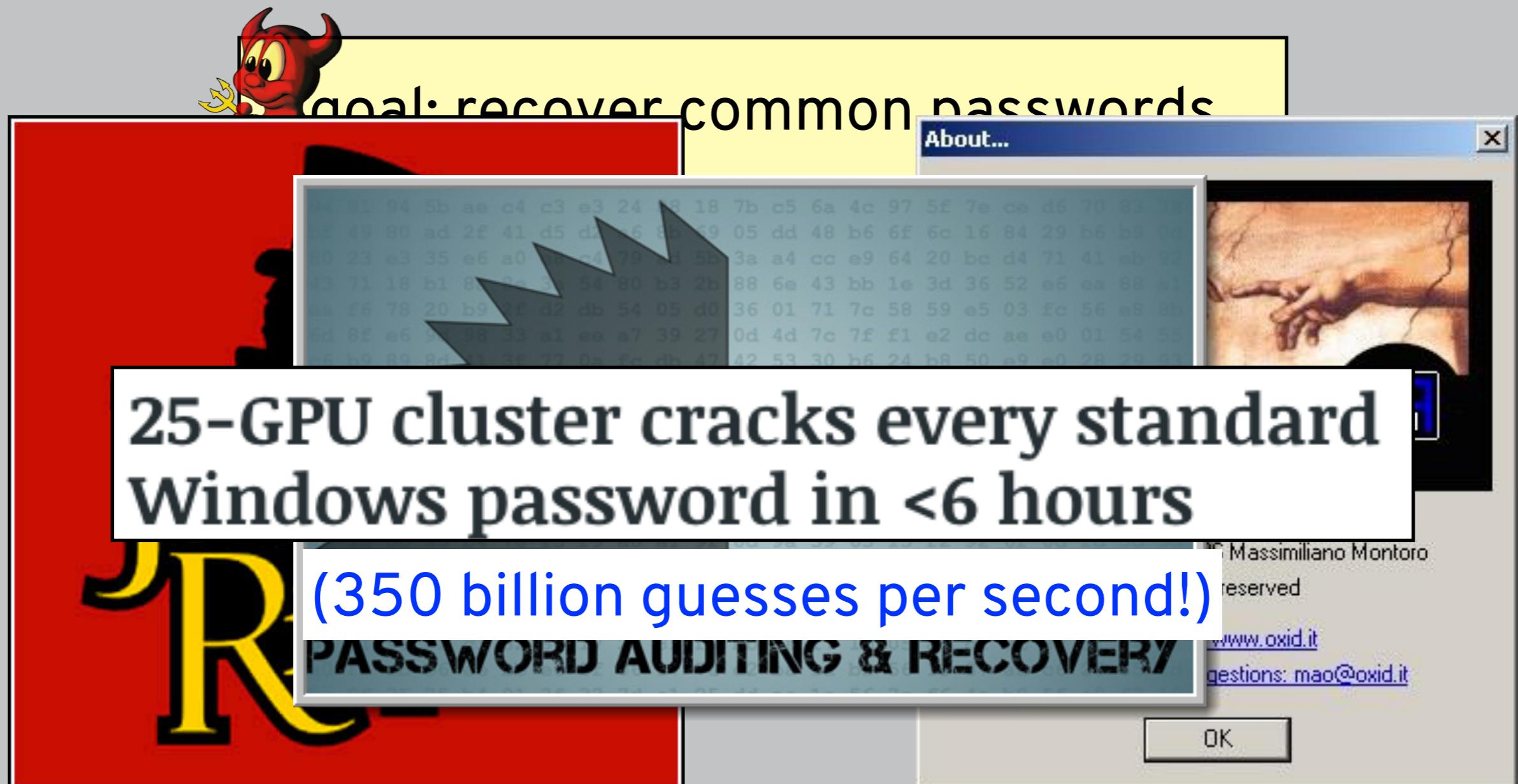
$u_1$	$h_1$
$u_2$	$h_2$
...	...
$u_n$	$h_n$

rainbow table

this is a dictionary attack

# PASSWORD CRACKING

---



this is a dictionary attack

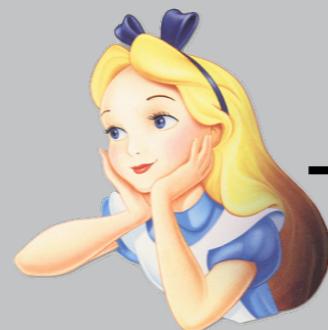
# JOHN THE RIPPER DEMO

---

```
smeiklej@noccia... demos % john --wordlist=rockyou.txt --stdout | head
Using default input encoding: UTF-8
Press 'q' or Ctrl-C to abort, almost any other key for status
123456
12345
123456789
password
iloveyou
princess
1234567
rockyou
12345678
abc123
smeiklej@noccia... demos %
```

# STORING PASSWORDS

---



uname,passwd



uname,H(passwd) here?

**problem:** common passwords

**solution:** use a salt

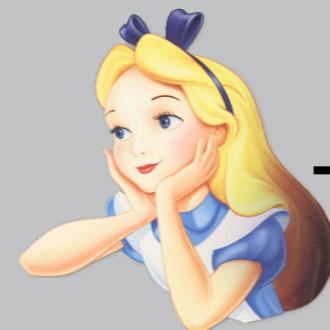
u1	H(p1)
u2	H(p2)
...	...
un	H(pn)



passwords.csv

# STORING PASSWORDS

---



uname,passwd



uname,**s,H(passwd||s)** here?

u1	s1	H(p1  s1)
u2	s2	H(p2  s2)
...	...	...
un	sn	H(pn  sn)



passwords.csv

# PASSWORD CRACKING

---



goal: recover common passwords

$H(\text{password})$   
 $H(123456)$   
 $H(\text{asdasd})$   
 $H(\text{qwerty})$   
 $H(\text{john})$

equal?		
u1	s1	h1
u2	s2	h2
...	...	...
un	sn	hn

dictionary attack won't work!

$(H(\text{password} \parallel s1) \neq H(\text{password} \parallel s2)) \text{ if } s1 \neq s2$

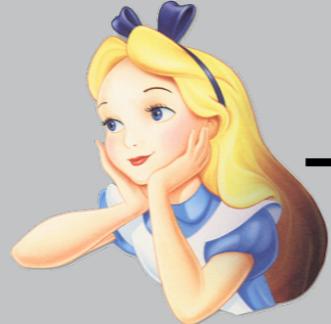
# JOHN THE RIPPER DEMO

---

```
smeiklej@noccia... demos % john --wordlist=rockyou.txt --stdout | head
Using default input encoding: UTF-8
Press 'q' or Ctrl-C to abort, almost any other key for status
123456
12345
123456789
password
iloveyou
princess
1234567
rockyou
12345678
abc123
smeiklej@noccia... demos %
```

# PASSWORDS

---



uname,pword



simple? how do you make them:

easy to memorise?

easy to enter?

hard to leak/steal?

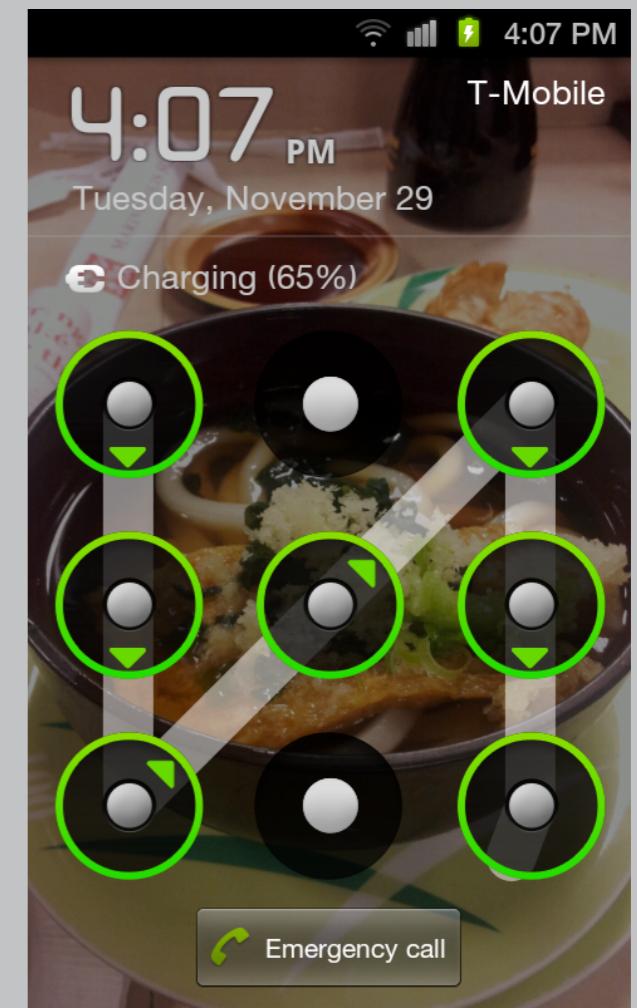
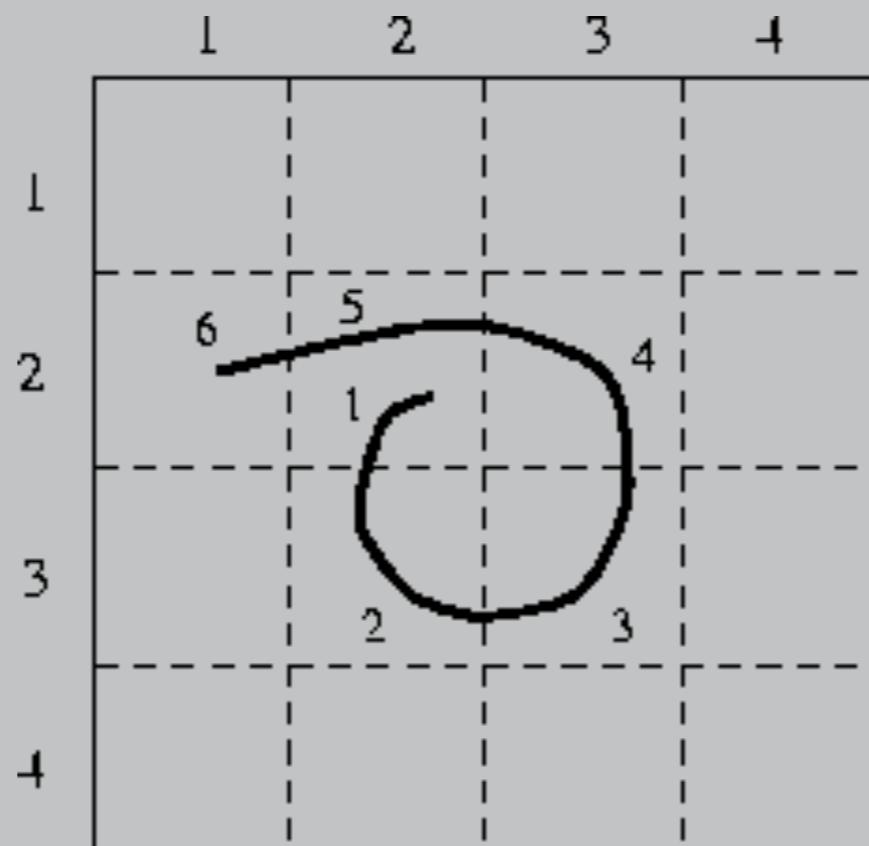
hard to guess?

resettable?

retrievable?

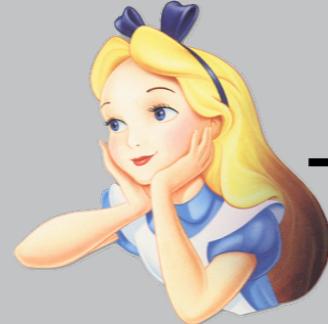
how to retrieve hashed and salted password?

# GRAPHICAL PASSWORDS



# PASSWORDS

---



uname,pword



simple? how do you make them:

easy to memorise?

hard to guess?

easy to enter?

resettable?

hard to leak/steal?

retrievable?

# SECURITY QUESTIONS

---

What is your pet's name?

In what year was your father born?

In what county where you born?

What is the color of your eyes?

What is the first name of the person you first kissed?

What is the last name of the teacher who gave you your first failing grade?

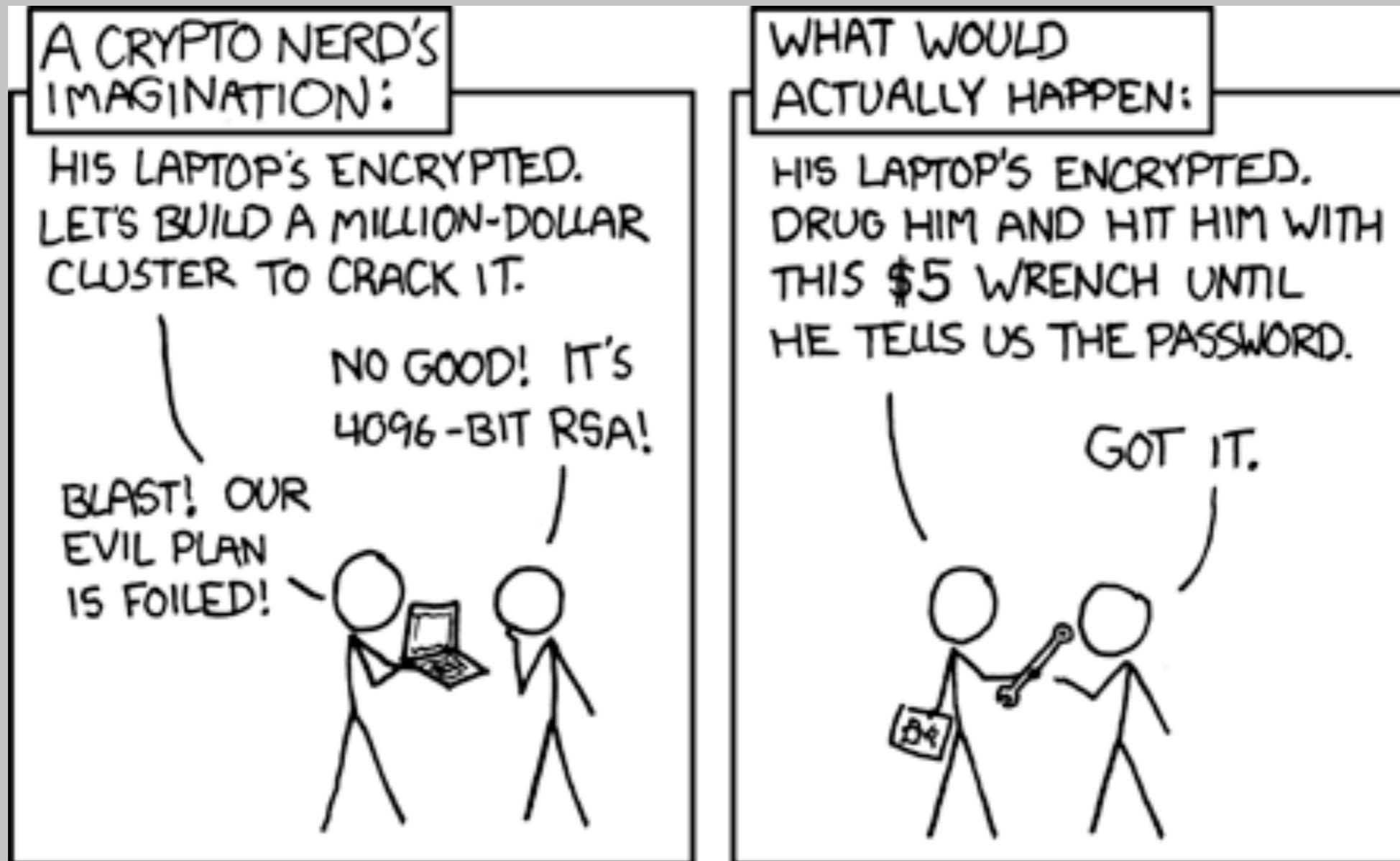
What is the name of the place your wedding reception was held?

In what city or town did you meet your spouse/partner?

What was the make and model of your first car?

# COERCION (“RUBBER-HOSE”) ATTACK

---



# COPING STRATEGIES

---

**try to enforce:**

**users will:**

different passwords → use same password

long passwords → add ‘123’ (or other padding)

random passwords → write passwords down

letters and numbers → add ‘1’ (or ‘123’ or other)

regular passwords changes → add, e.g., ‘spring’ or ‘june’

more complicated stuff → use reset functionality



**MilFlip Logon Details**

Username: 222005

Password: Seaking5

Logon:



**MilFlip Logon  
Details**

Username: 22385  
Password: Seaking5



ATTENTION ALL  
ORMS STUDENTS

PRIVACY: TO LEARN MORE ABOUT HOW WE COLLECT, USE AND DISCLOSE YOUR INFORMATION, PLEASE VISIT OUR PRIVACY POLICY WHICH IS AVAILABLE AT WWW.ORMS.COM/PRIVACY. THIS POLICY APPLIES TO INDIVIDUALS IN CANADA, EXCEPT FOR THE PROVINCE OF QUEBEC, WHERE THE INFORMATION CONTAINED HEREIN IS PROVIDED PURSUANT TO THE PROVINCIAL LAW OF QUEBEC.

PRIVACY: TO LEARN MORE ABOUT HOW WE COLLECT, USE AND DISCLOSE YOUR INFORMATION, PLEASE VISIT OUR PRIVACY POLICY WHICH IS AVAILABLE AT WWW.ORMS.COM/PRIVACY. THIS POLICY APPLIES TO INDIVIDUALS IN CANADA, EXCEPT FOR THE PROVINCE OF QUEBEC, WHERE THE INFORMATION CONTAINED HEREIN IS PROVIDED PURSUANT TO THE PROVINCIAL LAW OF QUEBEC.



MilFlip Logon Details

Username: 22222  
Password: Seaking1



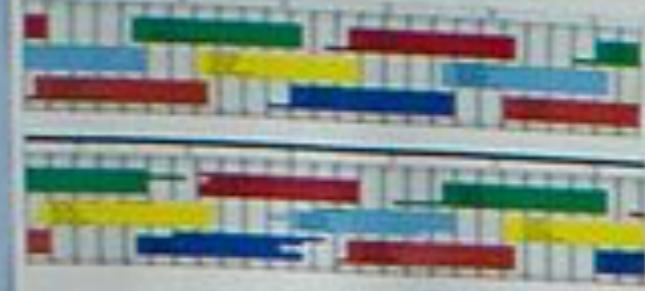
ATTENTION ALL  
ORMS STUDENTS

PROGRESSIVE FLIGHT FACULTY CLASS  
COMPUTER SCIENCE, INTEGRATION, AND  
INNOVATION ARE NOW OFFERING THE  
PROGRESSIVE FLIGHT FACULTY CLASS  
PROGRAM. LEARN MORE AT [www.oms.org](http://www.oms.org)  
FOR MORE INFORMATION.  
OR CALL 800.800.8000.



MilFlip Logon  
Details

Username: 22200  
Password: Seaborg5



# AUTHENTICATION AS A USER TASK

---

Properties of password-based authentication:

- Unaided recall (violation of “recognition rather than recall”)
- Recall and entry have to be 100% correct
- No corrective feedback on failure

Coping strategies include:

- Re-using the same password (or small set of passwords)
- Writing passwords down
- Relying on password reset

# PASSWORDS

---



uname,pword



**security:** want long and random passwords

**s** **usability:** humans can't generate or remember these

easy to memorise?

hard to guess?

easy to enter?

resettable?

hard to leak/steal?

retrievable?

# AUTHENTICATION

---

Authentication is:

**What you know**

text passwords

graphical passwords

personal details

**What you have**

**What you are**