What is your password strategy?

clicker.cs.illinois.edu

Q1

~Code~
340

CS 340

Authentication

Updates

1. MP 8 Due today
2. MP 9 out!
    a. MP 9 autograded portion DUE TUESDAY
    b. MP 9 - in class checkoff Thursday 11/20
3. HW 7 Due Thursday at 2:00 pm

4%

2%

2%

Agenda

1. Logging into a website
2. Randomness
3. Hiding Information

Goals:

- Help you start realizing the complexities of security. This is not a security class and you shouldn’t try implementing these yourself.

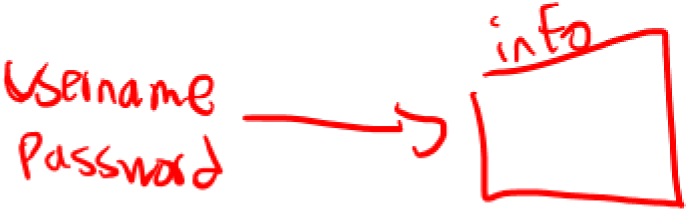
Storing Passwords Idea I

Create an account

username: schatz

password: pass123

users	passwords
schatz	pass123

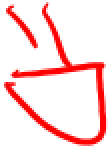


Hash Function

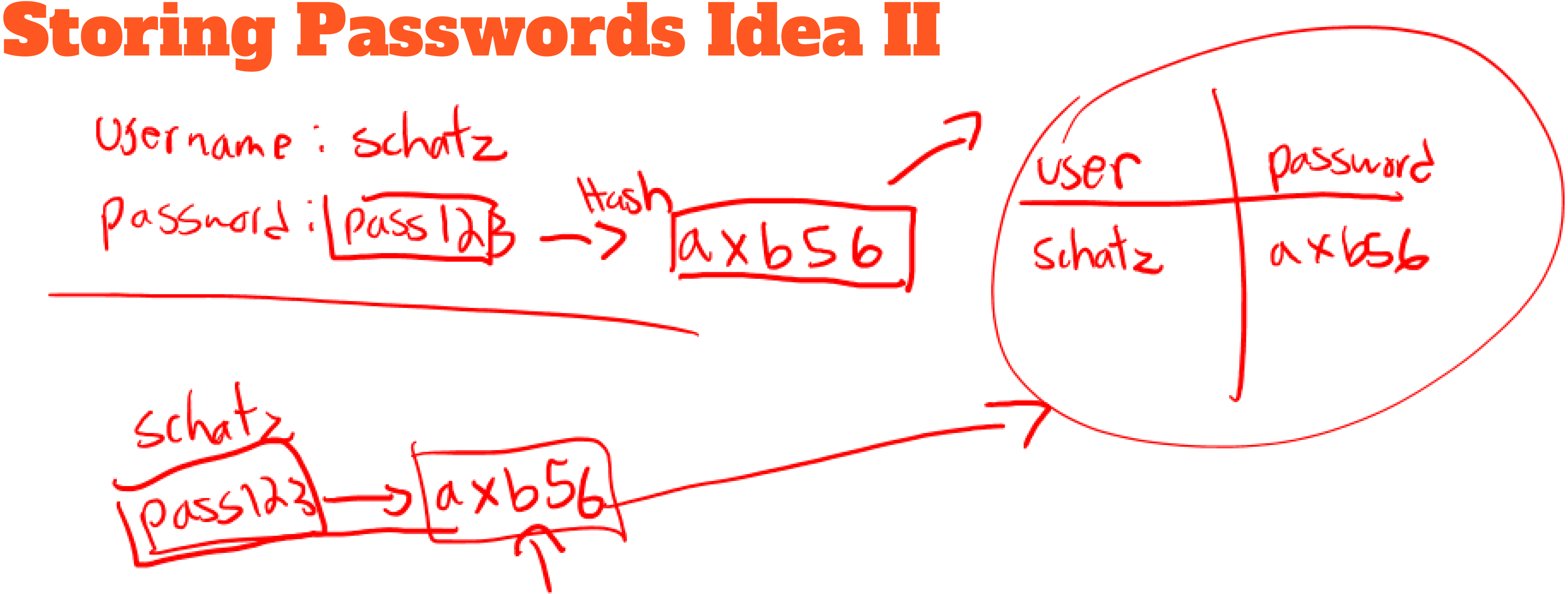
$h(m) = H$

m = message 0-X bytes

H = hash fixed size

given H very hard to find m 

Storing Passwords Idea II



user	password
schatz	axb56

Which of the following is a vulnerability of just using hashing for password storing?

clicker.cs.illinois.edu

Q3

~Code~
340

A. ~~If someone gets access to the database of passwords your password is automatically compromised~~.

B A. If another user has the same password as you and someone figures out that password, your password is compromised.

Luther pass123 → axb56

C ~~A. If the hashing function is very slow.~~

Storing Passwords Idea II Vulnerability

have same password

- Figure them out

luther123

rockyou.txt ← 10^6

Rainbow table

pre compute hashes

8 characters random

10^{15} combinations

Which best describes a rainbow table attack?

clicker.cs.illinois.edu

Q4

~Code~
340

[✓ A. Hashes are pre-computed for popular passwords, then compared to hashes stored for users]

~~A.~~ B Passwords are stored in plain text and then the database is compromised

~~A.~~ C Hashes are compared across users to find matches

Storing Passwords Idea III

Salt and pepper

↓

random series of bytes

- every user gets a salt

user	hash password	salt
schatz	akxB5	ab5777
luther	b7662	b7889

↑ pass123

pass123 →

$(\text{pass123} + \text{Salt}) \xrightarrow{\text{Hash}} \text{akxB5}$

pe

Which is not important for a salt?

A. It needs to be unique for each user

B A. It needs to be random

C ~~A. It needs to be stored outside the database~~

clicker.cs.illinois.edu

Q5

~Code~
340

Storing Passwords Idea III

pepper

↓

random value outside of the database

↓

same for every user

(password + salt + pepper)

↓

Hash

Which is not important for a pepper?

[✓ A. It needs to be unique for each user]

~~A.~~ B. It needs to be random

~~A.~~ C. It needs to be stored outside the database

clicker.cs.illinois.edu

Q6

~Code~
340

Randomness

generate a random password

salt and pepper need to be random

Which sequence is least likely to be generated by a truly random number generator?

clicker.cs.illinois.edu

Q7

~Code~
340

A. 1, 2, 3, 4

A. 5, 2, 1, 8

~~A.~~ C 1, 1, 1, 1

~~A.~~ D A or C

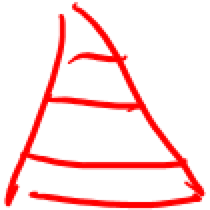
[✓ ~~A.~~ E All the same likelihood]

Which idea could generate good random numbers?

A) Analyzing an image of 100 live lava lamps.

~~A~~ B) Ask random people on the street for a random number.

~~A~~ C) Count how many cache misses there are on your OS every hour.



What are some strategies YOU can do for protecting your accounts?

Strategies the server does

- Hashing
- salt pepper
- others

Hiding Information

uijt dmbtt jt uif cftu

↑ ↑

1

key

A	B	C	D	E	F	G	H	I	J
1	2	3	4	5	6	7	8	9	10

K	L	M	N	O	P	Q	R	S	T
11	12	13	14	15	16	17	18	19	20

U	V	W	X	Y	Z
21	22	23	24	25	26

Hashing vs. Encryption

↓

one
way

↓

passwords

↓

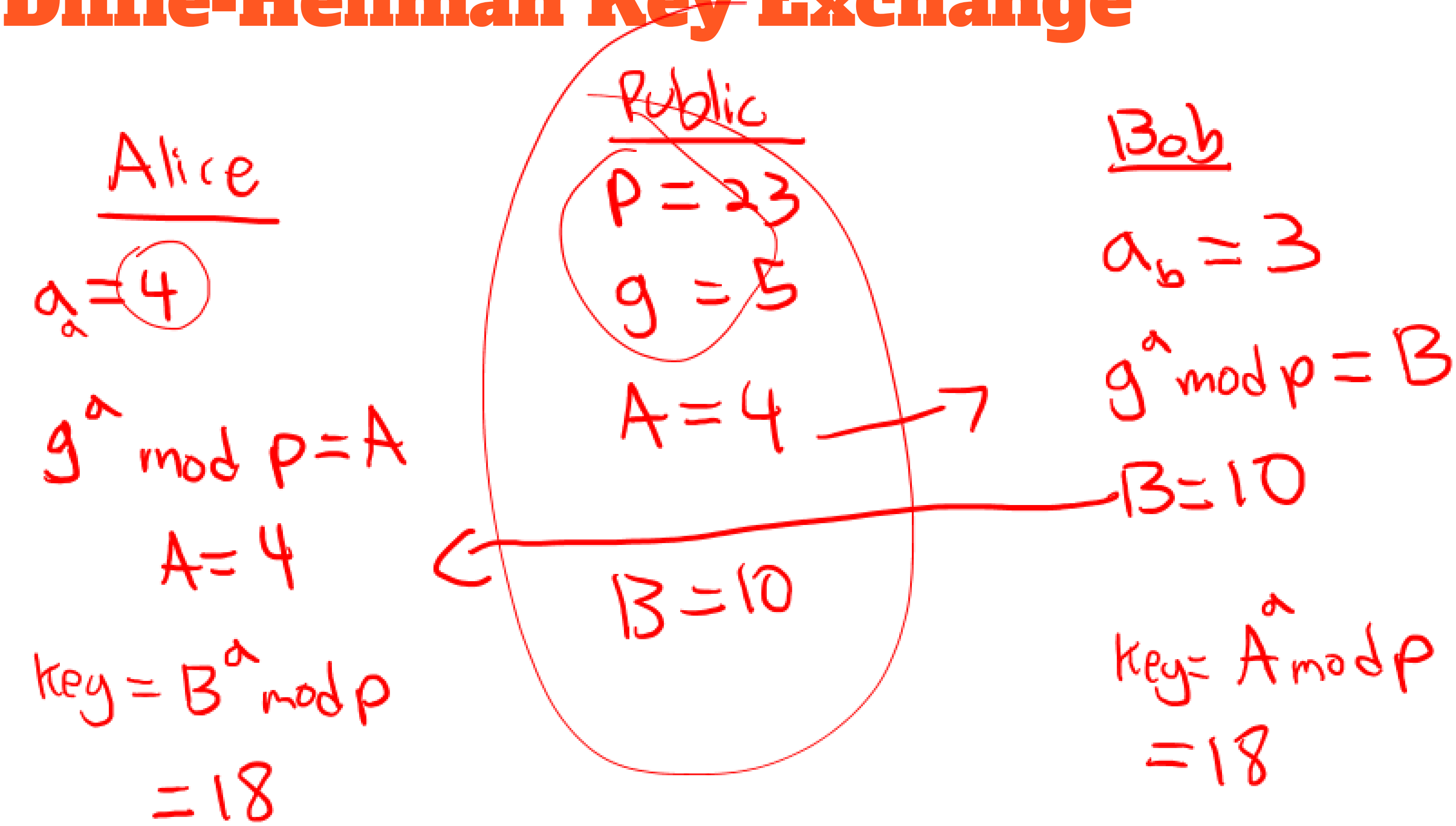
reversible
with
a key

Symmetric Encryption

1 key that locks and unlocks

How do we agree on a key

Diffie-Hellman Key Exchange



Alice

$a_a = 4$

$g^a \bmod p = A$

$A = 4$

key $= B^a \bmod p$

$= 18$

Public

$p = 23$

$g = 5$

$A = 4$

$B = 10$

Bob

$a_b = 3$

$g^a \bmod p = B$

$B = 10$

key $= A^a \bmod p$

$= 18$

What is our key?

clicker.cs.illinois.edu

Q8

~Code~

340

$a=30$

27

$p=89$

$g=7$

$a_0=12$

$g^{a} \bmod p = A = 47$

$A=47$

$B=\boxed{81}$

$B^{a} \bmod p$

$81^{a} \bmod p = \text{key} = 4$

7

A - D

A	2
B	3
C	4
D	5

Asymmetric Encryption

2 Keys

1 encrypts — private key

1 dycrpts — public key

Signature — Verify person A sent document B .

A - Hash document B

A - encrypte hash with private key → documnt B encrypted Hash

C

decrypt Hash using public key

Hash(document B) = hash

I am who I am!

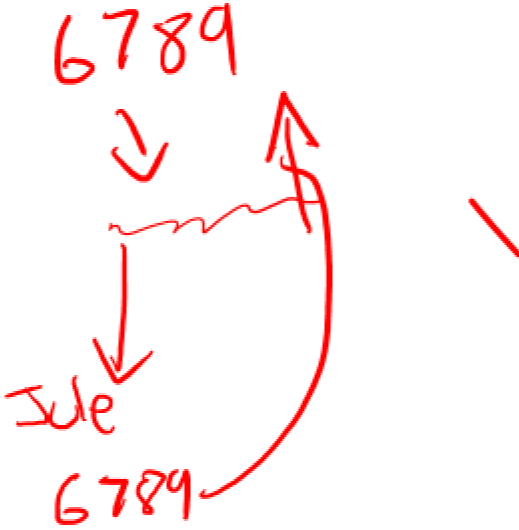
a) central authority – sign a doc
a persons public key

b) Give that to skeptic

- encrypts number with public key
- give that to you

c) You decrypt with private key

number


6789
Jule
6789