Module 2B: Modular exponentiation and cryptographic applications

MTH 225
16 Sept 2020

Agenda

- Review of Daily Prep activity + Q/A time
- Minilecture with practice: Using the repeated squaring algorithm; application to key generation using the Diffie Hellman algorithm
- Wrap up with ungraded quiz + feedback time

The purpose of the repeated squaring algorithm is

To find square roots of integers in an efficient way

To find integers raised to exponents mod n, in an efficient way

To find the product of two integers mod n, in an efficient way

To draw squares on a canvas over and over again, in an efficient way

When using repeated squaring to calculate $5^{65}\%100$, how many squaring operations would be required?

- 5
- 6
- 7
- 10
- 65
- 99

When using repeated squaring to calculate $5^{65}\%100$, the numbers involved

- Will never be greater than 5
- Will never be greater than 1000
- Will never be greater than 65
- Will never be greater than 100

$5 \% 100 = 5$

$5^2 \% 100 = 25$

$5^4 \% 100 = (5^2)^2 \% 100 = 25^2 \% 100 = 625 \% 100 = 25$

$5^8 \% 100 = (5^4)^2 \% 100 = 25^2 \% 100 = 25$

$5^{16} \% 100 = (5^8)^2 \% 100 = 25^2 \% 100 = 25$

$5^{32} \% 100 = (5^{16})^2 \% 100 = 25^2 \% 100 = 25$

$5^{64} \% 100 = (5^{32})^2 \% 100 = 25^2 \% 100 = 25$

Number of squaring steps to compute a^b mod n ≈ log(b)

$5^{65} \% 100 = 5^{64+1} \% 100$

$= 5^{64} \cdot 5^1 \% 100$

$= (25)(5) \% 100$

$= 125 \% 100$

$= 25$.

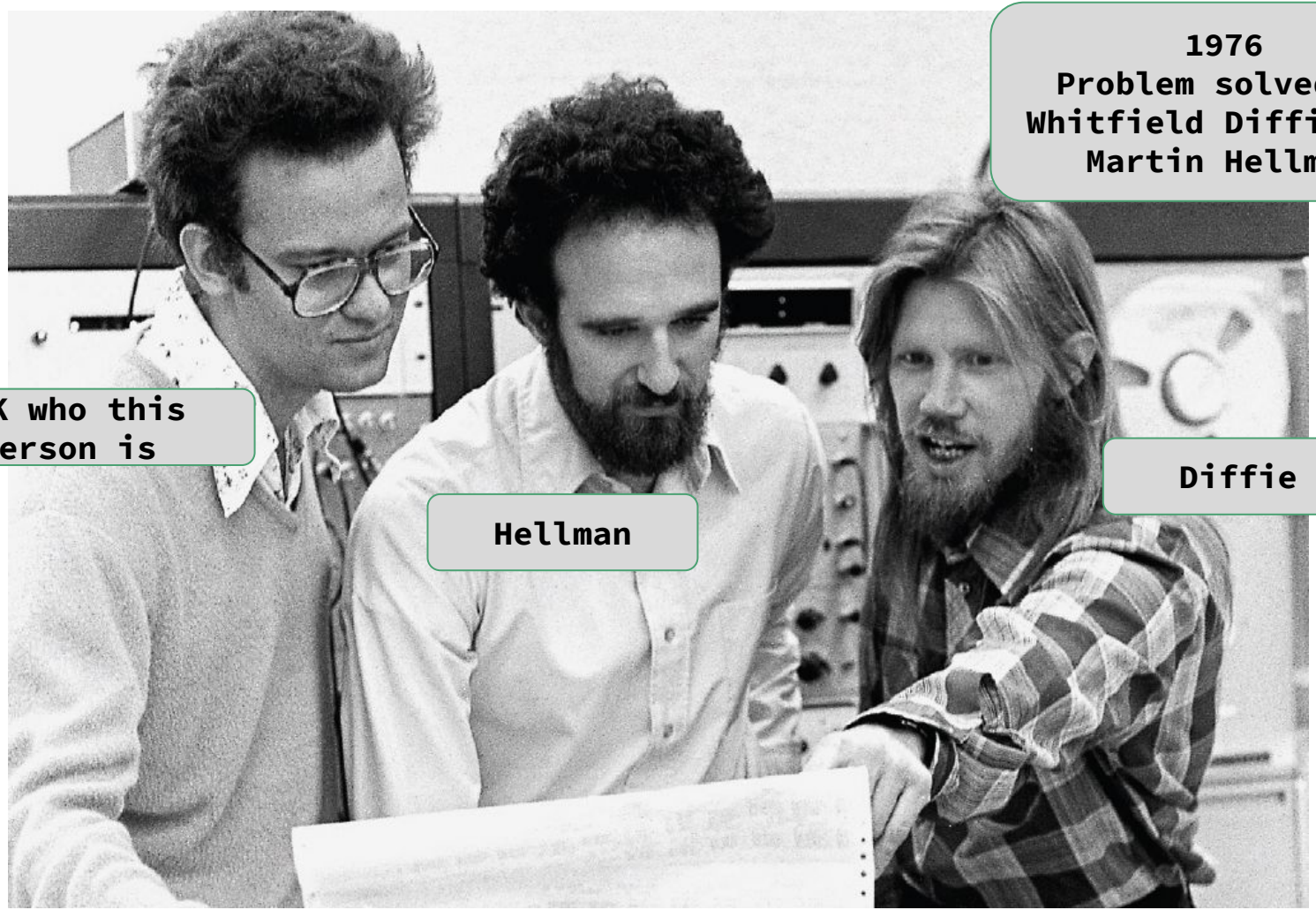
Q&A time

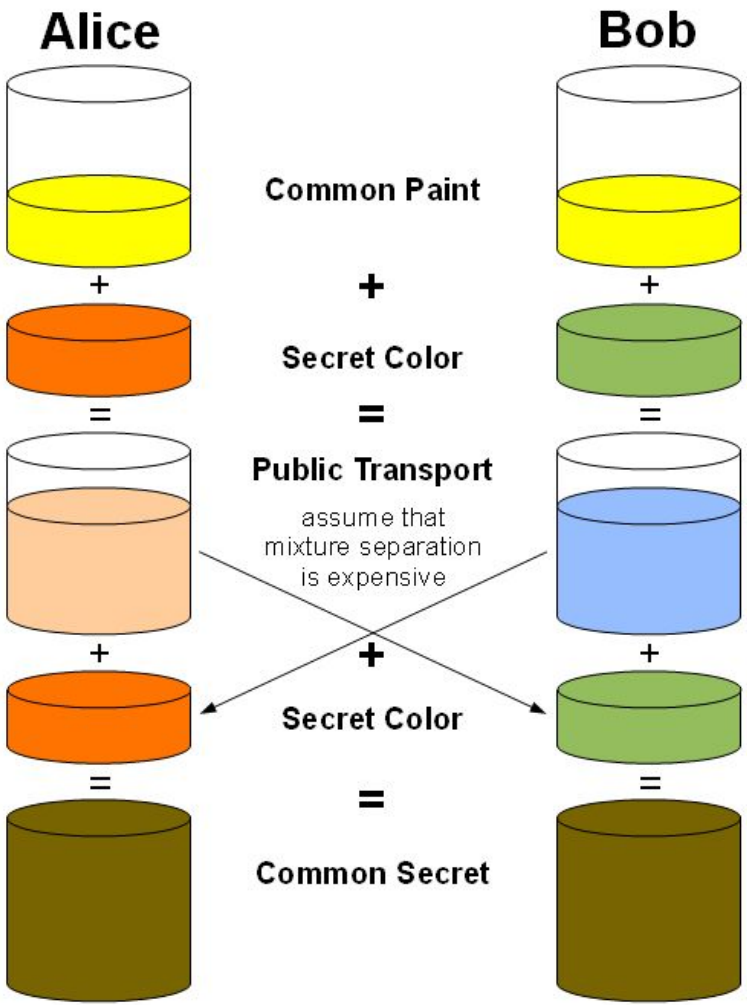
Practice with the repeated squaring algorithm: Go to http://gvsu.edu/s/1sU

Application: The Diffie-Hellman algorithm

A major problem with cryptosystems

- When using a cryptosystem to send and receive secret information, we need a key to encrypt and decrypt.
- Typically both the sender and the recipient need the same key.
- **Problem**: How to communicate the key to both sender and recipient without the key being intercepted?


1976
Problem solved by Whitfield Diffie and Martin Hellman
IDK who this person is
Hellman
Diffie



https://commons.wikimedia.org/wiki/File:Diffie-Hellman_Key_Exchange.png

The Diffie-Hellman Key Exchange Protocol

- Alice, Bob: Choose a prime number *p* and a random integer *g* having no common factors with *p*-1. They make these **public** (website, email signature, etc.)
- Alice, Bob: Each choose random integers less than *p* and keep those secret. Alice chooses *a*, Bob chooses *b*.
- Alice: Computes A = $\mathbf{g^a \% p}$ and sends to Bob
- Bob: Computes B = $\mathbf{g^b \% p}$ and sends to Alice
- Alice computes $\mathbf{B^a \% p}$ and Bob computes $\mathbf{A^b \% p}$
- **These are the same number.** Can be used as a key.

- p = 197; g = 5
- a = 100, b = 72
- A = 5^{100} % 197 = 172 ➔ Bob
- B = 5^{72} % 197 = 81 ➔ Alice
- Alice: 81^{100} % 197 = **60.**
- Bob: 172^{72} % 197 = **60.**

```
Information that was interceptable: p,
g (made completely public); A and B.

Private information: a, b.
```

How hard is this to break?

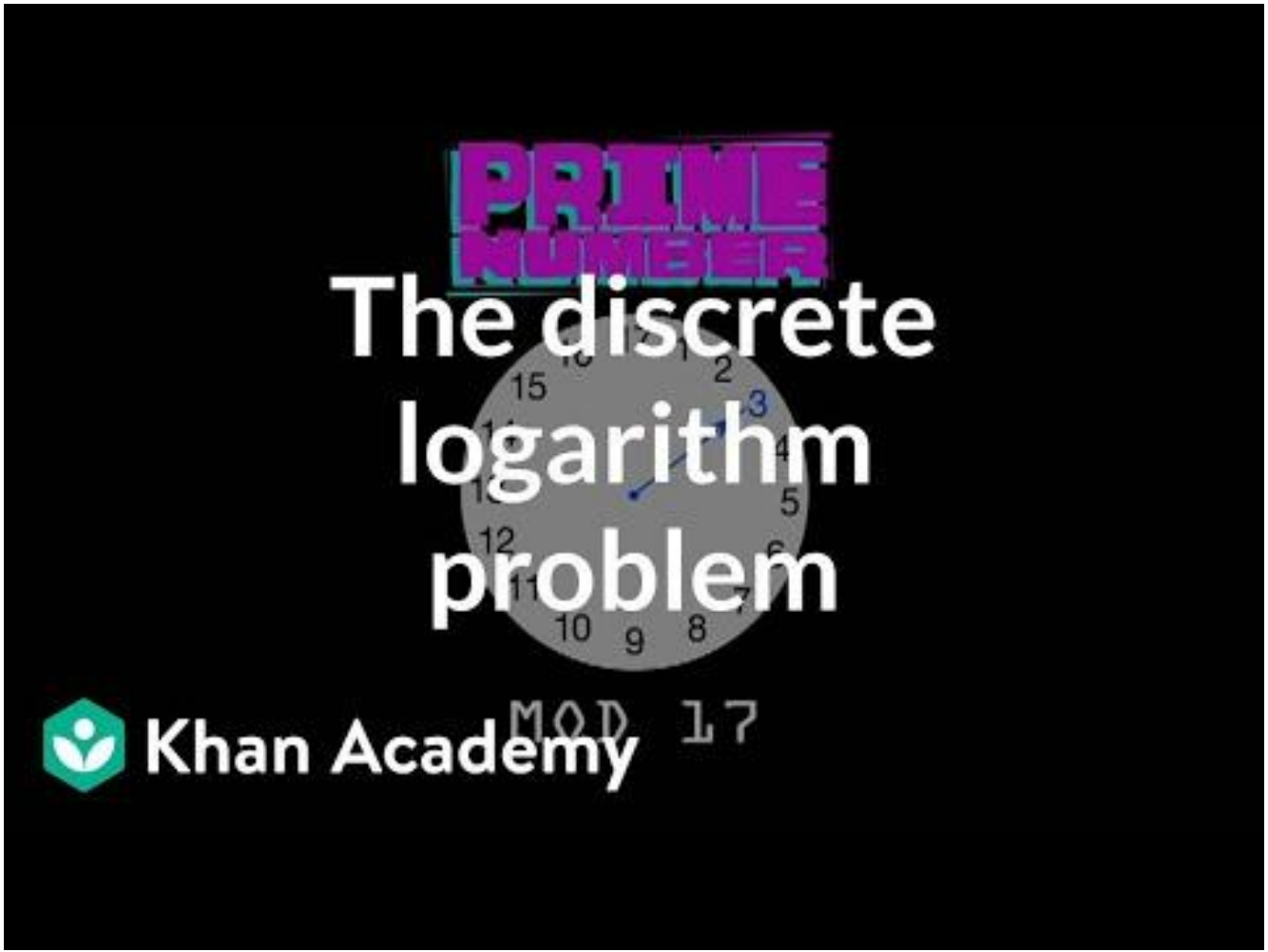
“Break” = A third party could determine Alice or Bob’s private info from the public info.

$$A = g^a \% p$$

$$B = g^b \% p$$

There is currently (2020) no known efficient algorithm for extracting the exponent from an expression like this.

The “discrete logarithm problem”


PRIME NUMBER
The discrete logarithm problem
MOD 17
Khan Academy

Feedback:
http://gvsu.edu/s/1rF

Add sticky notes for comments, ideas, and questions.