CSE 20
DISCRETE MATH

Fall 2020

http://cseweb.ucsd.edu/classes/fa20/cse20-a/

Today's learning goals

- Use the equivalence relation of congruence modulo integers and apply its properties
- Trace the algorithms involved in Diffie-Helman key exchange

Review: Relations

Definition: When A and B are sets, we say any subset of $A \times B$ is a **binary relation**.

Definition: When A is a set, we say any subset of $A \times A$ is a (binary) **relation** on A.

Definition: (*Rosen 9.1*) A relation R on a set A is called **reflexive** means $(a, a) \in R$ for every element $a \in A$.

Definition: (*Rosen 9.1*) A relation R on a set A is called **symmetric** means $(b, a) \in R$ whenever $(a, b) \in R$, for all $a, b \in A$.

Definition: (*Rosen 9.1*) A relation R on a set A is called **transitive** means whenever $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$, for all $a, b, c \in A$.

Definition: (*Rosen 9.5*) A relation is an **equivalence relation** if it is reflexive, symmetric, and transitive.

Definition: (*Rosen 9.5*) An **equivalence class** of an element $a \in A$ for an equivalence relation R on the set A is the set $\{s \in A | (a, s) \in R\}$. We write this as $[a]_R$.

Review: Congruence modulo n

Definition: Let $R_{(\mathbf{mod}\ n)}$ be the set of all pairs of integers (a, b) such that $(a\ \mathbf{mod}\ n = b\ \mathbf{mod}\ n)$. We say a is **congruent to** b **mod** n means $(a, b) \in R_{(\mathbf{mod}\ n)}$. A common notation is to write this as $a \equiv b (\mathbf{mod}\ n)$.

Which of the following are true?

A. $(102 + 48)\ \mathbf{mod}\ 10 = (102\ \mathbf{mod}\ 10) + (48\ \mathbf{mod}\ 10)$

B. $(7 \cdot 10)\ \mathbf{mod}\ 5 = (7\ \mathbf{mod}\ 5) \cdot (10\ \mathbf{mod}\ 5)$

C. $(2^5)\ \mathbf{mod}\ 3 = \left((2\ \mathbf{mod}\ 3)^{(5\ \mathbf{mod}\ 3)} \right)\ \mathbf{mod}\ 3$

D. More than one of the above

E. None of the above

Modular arithmetic

Lemma (Section 4.1, page 241): For $a, b \in \mathbb{Z}$ and positive integer n, $(a, b) \in R_{(\mathbf{mod}\ n)}$ if and only if $n|a - b$.

Modular arithmetic

Lemma (Section 4.1 Theorem 5): For $a, b \in \mathbb{Z}$ and positive integer n, if $(a, b) \in R_{(\mathbf{mod}\ n)}$ and $(c, d) \in R_{(\mathbf{mod}\ n)}$ then $(a + c, b + d) \in R_{(\mathbf{mod}\ n)}$ and $(ac, bd) \in R_{(\mathbf{mod}\ n)}$.

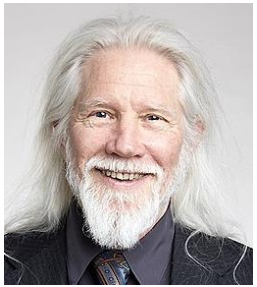
Application: Cryptography

How do parties communicate private key securely across insecure channel?

- Caesar cipher: need to agree on shift
- AES: private key, needs to be generated for each session

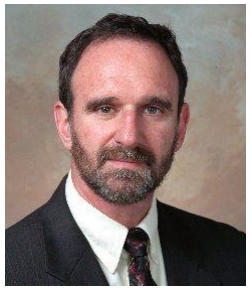
Application: Cryptography

Rosen, 4.6, p302



a: public

p: public



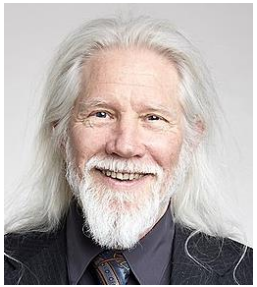
k1: secret

k2: secret

Definition: Let a be a positive integer and p be a large[1] prime number, both known to everyone. Let k_1 be a secret large number known only to person P_1 (Alice) and k_2 be a secret large number known only to person P_2 (Bob). Let the **Diffie-Helman shared key** for a, p, k_1, k_2 be $(a^{k_1 \cdot k_2} \textbf{ mod } p)$.

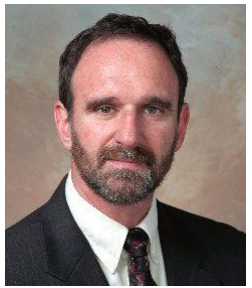
Application: Cryptography

Rosen, 4.6, p302



a: public

p: public



k1: secret

k2: secret

Suppose that Alice and Bob want to share a common key. The protocol follows these steps, where the computations are done in $\mathbf{Z}_p$.

(1) Alice and Bob agree to use a prime p and a primitive root a of p.

(2) Alice chooses a secret integer k_1 and sends a^{k_1} **mod** p to Bob.

(3) Bob chooses a secret integer k_2 and sends a^{k_2} **mod** p to Alice.

(4) Alice computes $(a^{k_2})^{k_1}$ **mod** p.

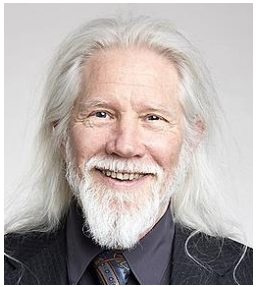
(5) Bob computes $(a^{k_1})^{k_2}$ **mod** p.

At the end of this protocol, Alice and Bob have computed their shared key, namely

$$(a^{k_2})^{k_1} \bmod p = (a^{k_1})^{k_2} \bmod p.$$

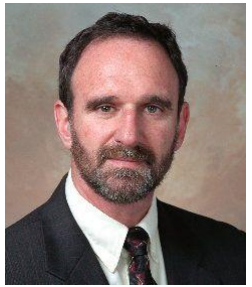
Application: Cryptography

Rosen, 4.6, p302



a: public

p: public



k1: secret

$a^{k1} \bmod p$

k2: secret

Computes

$(a^{k2} \bmod p)^{k1} \bmod p$

$= (a^{k1\,k2}) \bmod p$

$a^{k2} \bmod p$

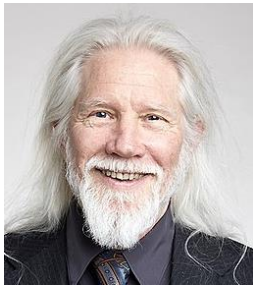
Computes

$(a^{k1} \bmod p)^{k2} \bmod p$

$= (a^{k1\,k2}) \bmod p$

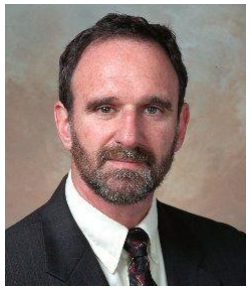
Application: Cryptography

Rosen, 4.6, p302



a: public

p: public



k1: secret

k2: secret

$a^{k1} \bmod p$

Computes
$(a^{k2} \bmod p)^{k1} \bmod p$
$= (a^{k1\,k2}) \bmod p$

$a^{k2} \bmod p$

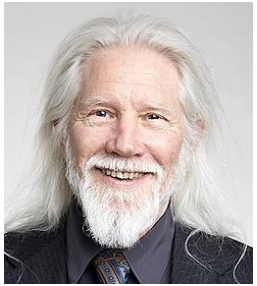
Computes
$(a^{k1} \bmod p)^{k2} \bmod p$
$= (a^{k1\,k2}) \bmod p$

Public: $a, p, a^{k1} \bmod p, a^{k2} \bmod p$

Common secret: $a^{k1\,k2} \bmod p$

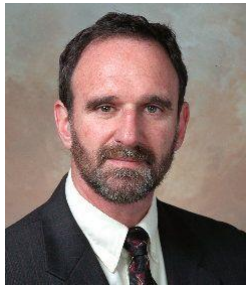
Application: Cryptography

Rosen, 4.6, p302



a: public

p: public



k1: secret

k2: secret

San Diego connection

key cryptosystem they need to share a common key. The protocol we will describe is known as the **Diffie-Hellman key agreement protocol**, after Whitfield Diffie and Martin Hellman, who described it in 1976. However, this protocol was invented in 1974 by Malcolm Williamson in secret work at the British GCHQ. It was not until 1997 that his discovery was made public.

Modular exponentiation

Rosen p. 254

Calculate $3^8 \bmod 7$

Approach 1: Directly

$3^1 \bmod 7 =$

$3^2 \bmod 7 =$

$3^3 \bmod 7 =$

$3^4 \bmod 7 =$

$3^5 \bmod 7 =$

$3^6 \bmod 7 =$

$3^7 \bmod 7 =$

$3^8 \bmod 7 =$

Modular exponentiation

Rosen p. 254

Calculate 3^8 **mod** 7

Modular Exponentation; Algorithm 5 in Section 4.2 (page 254)

```
1  procedure modular exponentiation(b: integer;
2                   n = (a_{k-1}a_{k-2}...a_1a_0)_2, m: positive integers)
3  x := 1
4  power := b mod m
5  for i:= 0 to k-1
6     if a_i = 1 then x:= (x · power) mod m
7     power := (power · power) mod m
8  return x {x equals b^n mod m}
```

Approach 2: Using Algorithm 5

$b =$ ____, $n =$ _____, $k =$ ____, $m =$ ____

i	a_i	x	$power$
		1	b **mod** $m =$
0			
1			
2			
3			

Application: Cryptography

Multiplication is computationally cheap, but
Factoring is computationally expensive
Given a 400 digit number that is a product of two 200 digit primes, can't efficiently find these primes. *Basis of security of RSA*

Modular exponentiation is computationally cheap, but
Discrete logarithm is computationally expensive:
Given a 300 digit prime and the result of exponentiation mod this prime, can't efficiently find the logarithm, i.e. find k when given a^k mod p
Basis of security of Diffie-Hellman

Peter Shor 1994 https://ieeexplore.ieee.org/document/365700