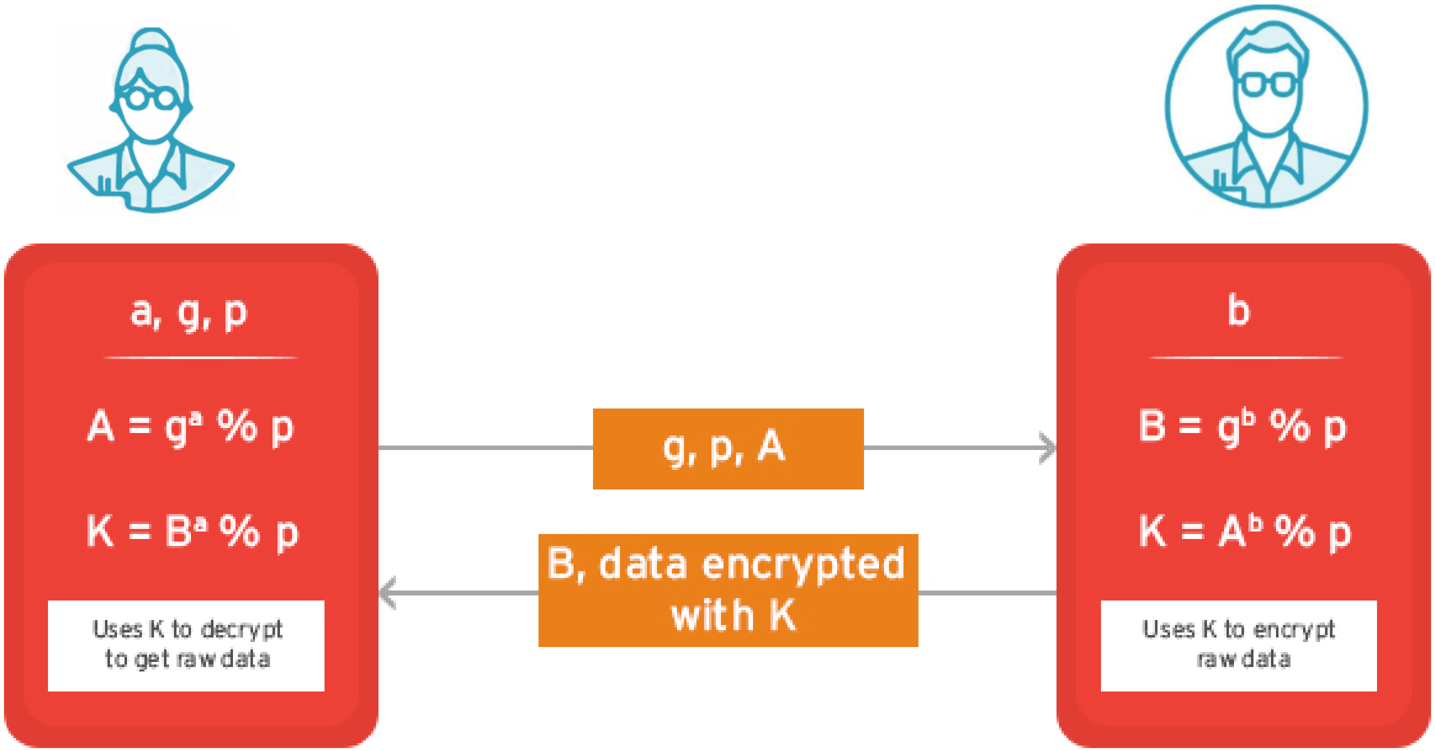
Diffie Hellman Key Exchange



a, g, p

$A = g^a \% p$

$K = B^a \% p$

Uses K to decrypt to get raw data

g, p, A

B, data encrypted with K

b

$B = g^b \% p$

$K = A^b \% p$

Uses K to encrypt raw data

$K = A^b \% p = B^a \% p$

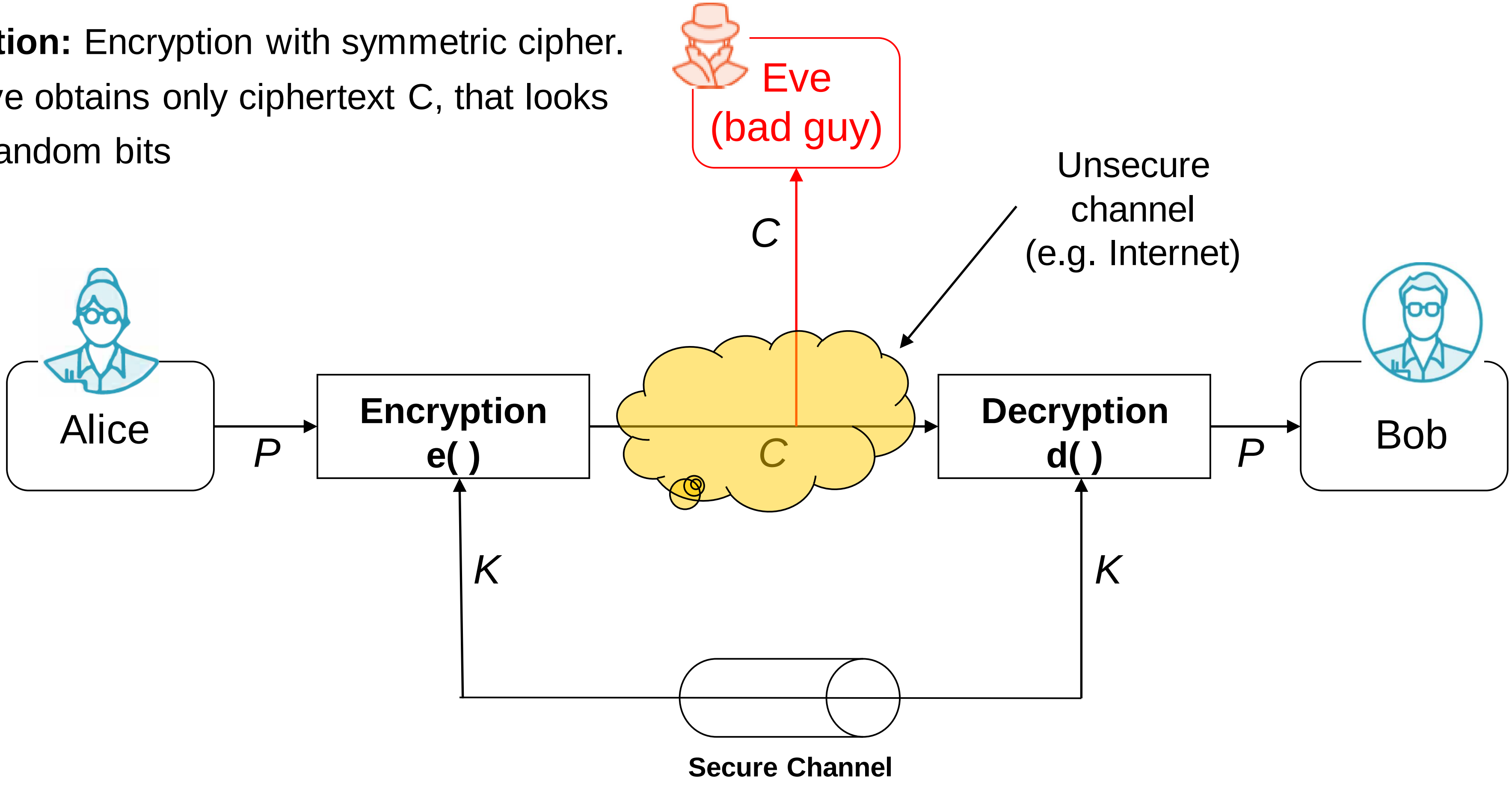
Symmetric Key Crypto Problem

- Symmetric key crypto lets two parties exchange secret messages *as long as they already have a shared key*
- How do you exchange secret messages with someone when you don’t already have a shared key?

Symmetric Cryptography Revisited

Solution: Encryption with symmetric cipher.

⇒ Eve obtains only ciphertext C, that looks like random bits



=> Alice and Bob can use symmetric encryption to securely exchange messages

But how do they can share a key?

⇒ **The problem of secure communication is reduced to secure transmission and storage of the key K**

Eve the Eavesdropper

- Eve is an attacker who can see Alice and Bob’s messages
- Eve can’t modify them
- Eve is a *passive attacker*
- Real-world examples
  - Internet provider
  - Government
  - Anyone nearby if your Wi-Fi is unencrypted
  - Someone else on the same network
  - Lots of potential people…

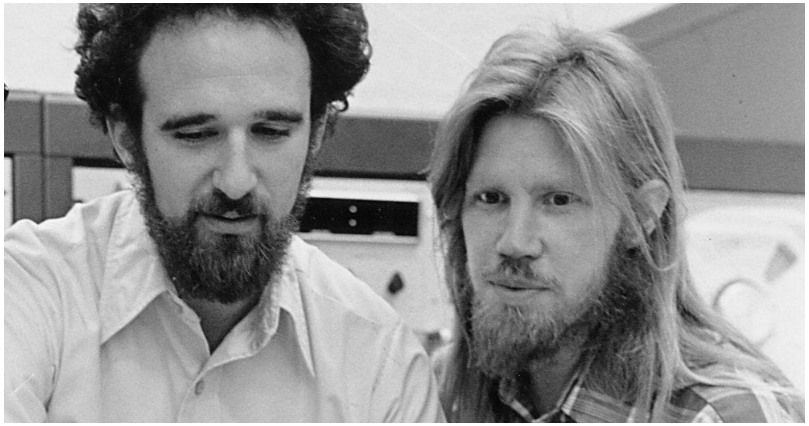
Ok, so…

- We can’t pick a key and send it => Eve will know it !
- We could pick a key together offline
  - Not feasible in the general case
  - You want to use encrypted communication with a lot of different services on the internet…

Diffie-Hellman Key Exchange

- Invented by Whitfield Diffie and Martin Hellman in 1976
- Allows Alice and Bob to exchange a key without Eve learning it
- Better name for it is Diffie-Hellman key **agreement** protocol
- **Revolutionary Idea**: no need for any prior secret agreement in order to communicate securely

1976

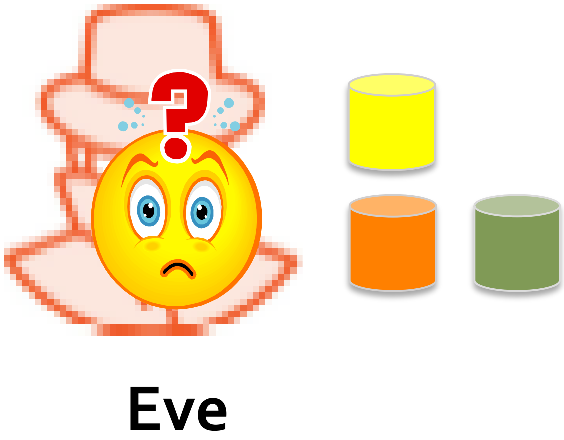


2015

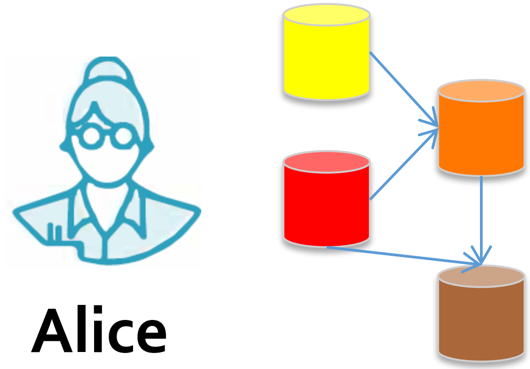
Encryption wizards **Whitfield Diffie** and **Martin Hellman** won $1m 2015 ACM Turing Award (CS Nobel Prize) for their **Diffie Hellman** Key Exchange protocol

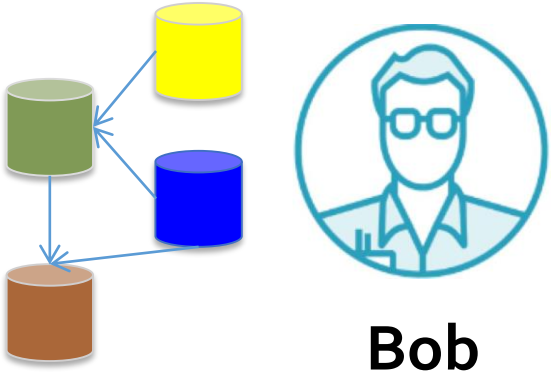
DH in Colors

How can Alice and Bob agree on a **secret color** without Eve finding it out?



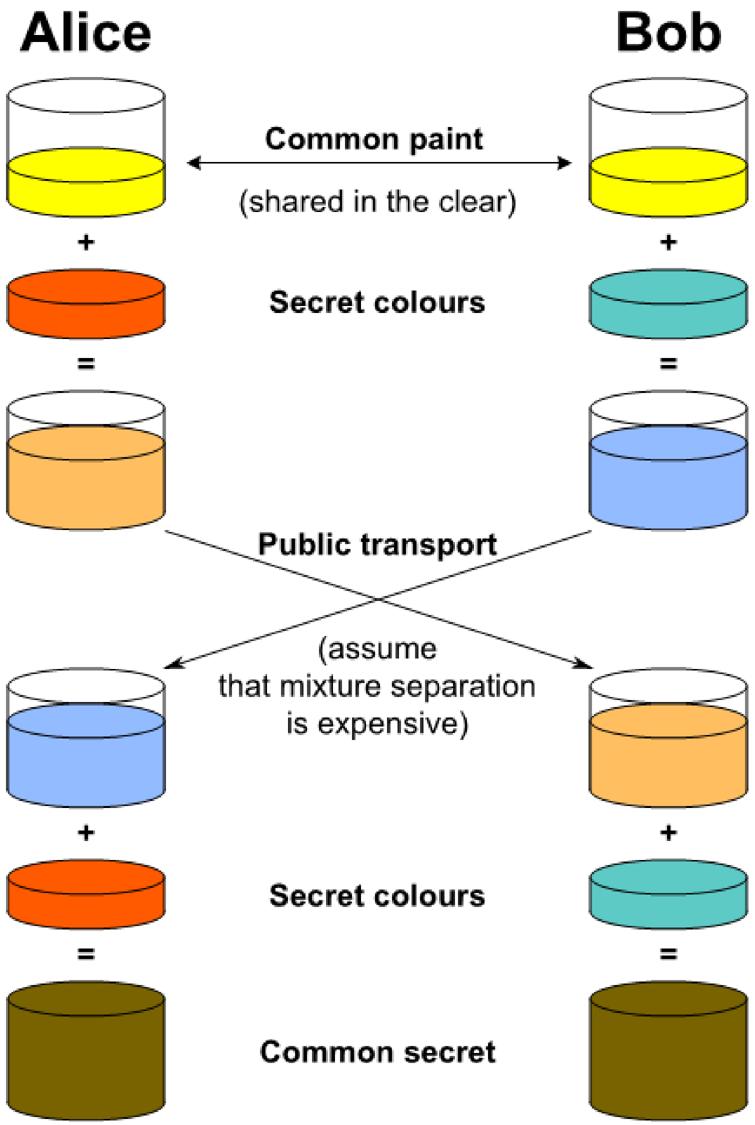
Eve unable to determine the secret color because she doesn't have the right colors to mix together



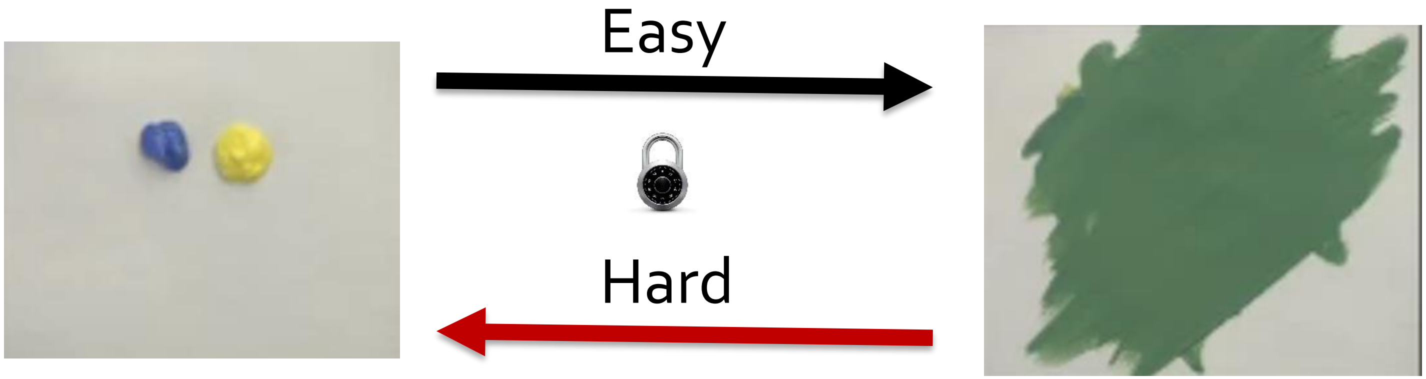


This is just an analogy, the actual algorithm uses mathematics

DH Colors - Summary



One Way Function



Easy

Hard

- Easy to mix 2 colors to produce a 3rd one
- Given a mixed color, it is hard to unmix it to get the exact original colors

DH in Math Example

1. Alice and Bob agree on a prime number **p** and a base value **g**. Here, p=23 and g=5
2. Alice chooses a secret number, **a**, and sends Bob $A=g^a \bmod p$. Here, **a**=**6**

   $A = 5^6 \bmod 23 = 15625 \bmod 23 = 8$

3. Bob chooses a secret number, **b**, and sends Alice $B=g^b \bmod p$. Here, **b**=**15**

   $B = 5^{15} \bmod 23 = 30{,}517{,}578{,}125 \bmod 23 = 19$

Example from http://en.wikipedia.org/wiki/Diffie-Hellman_key_exchange

DH in Math Example

4. Alice computes $s = B^a \bmod p$

   $s = 19^6 \bmod 23$

   $s = 47{,}045{,}881 \bmod 23$

   $s = 2$

5. Bob computes $s = A^b \bmod p$

   $s = 8^{15} \bmod 23$

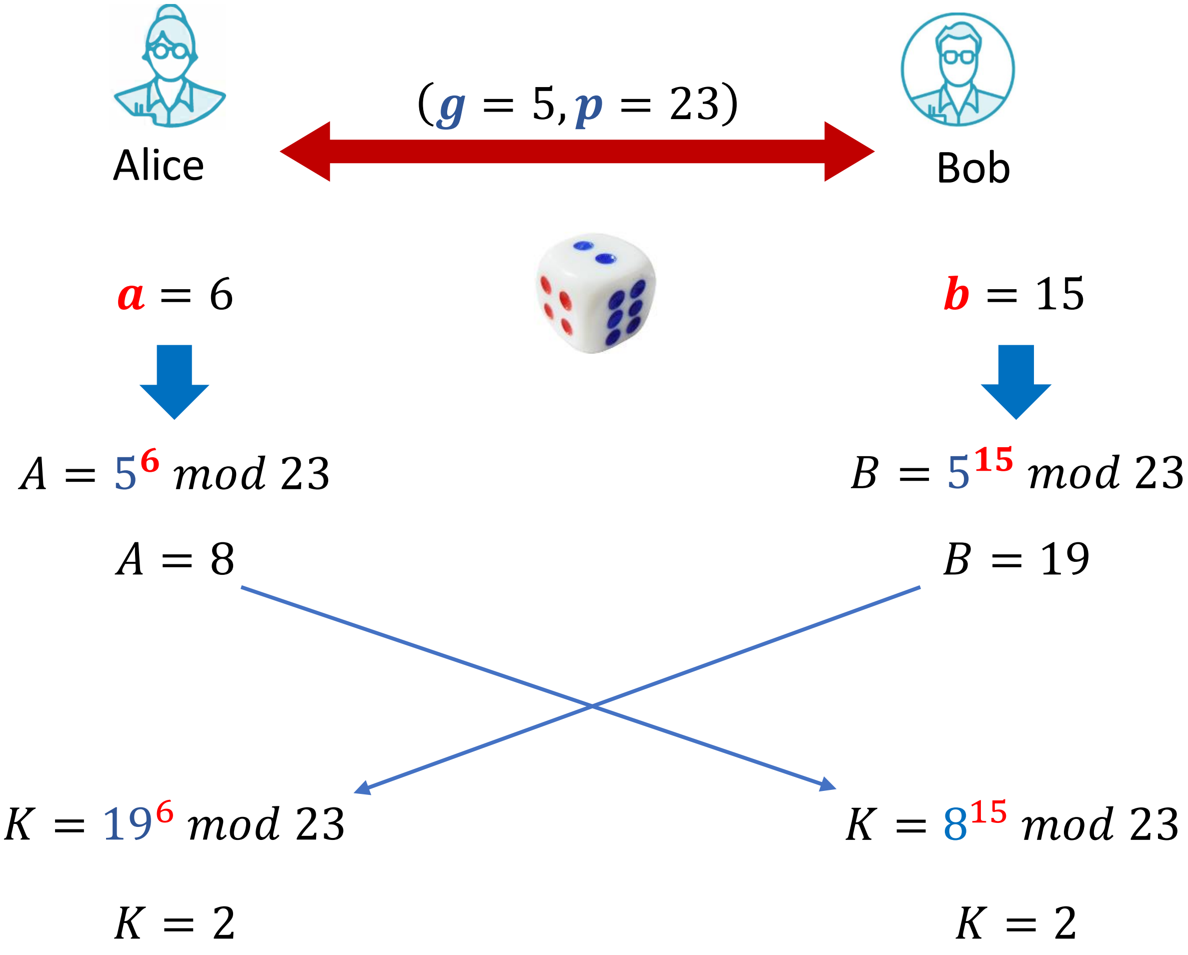
   $s = 35{,}184{,}372{,}088{,}832 \bmod 23$

   $s = 2$

=> Alice and Bob now share a secret, s=2, that can’t be derived from the public information

Example from http://en.wikipedia.org/wiki/Diffie-Hellman_key_exchange

Diffie-Hellman key exchange protocol



1. One-time Setup

Alice $(g = 5, p = 23)$ Bob

2. Random a and b

$a = 6$

$b = 15$

3. Compute A and B

$A = 5^{6} \; mod \; 23$

$A = 8$

$B = 5^{15} \; mod \; 23$

$B = 19$

4. Send A and B

5. Shared key

$(g^{b})^{a} \; mod \; p = (g^{a})^{b} \; mod \; p$

$K = 19^{6} \; mod \; 23$

$K = 2$

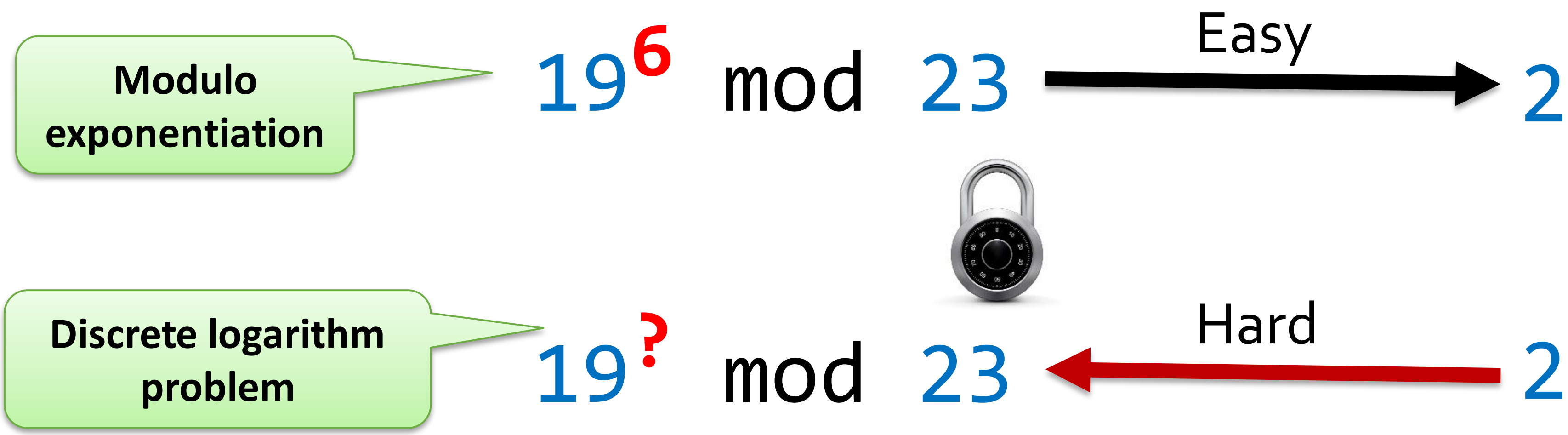
$K = 8^{15} \; mod \; 23$

$K = 2$

DH in Practice

- **a**, **b**, and **p** would need to be MUCH larger in practice
  - 100s of digits long
- This works because Eve can’t use A and B to figure out the secret numbers **a** and **b** chosen by Alice and Bob
- DH doesn’t prove *who* you share the key with, just that the key isn’t known by anyone else

DH in Math



Modulo exponentiation

$19^{6} \bmod 23 \xrightarrow{\text{Easy}} 2$

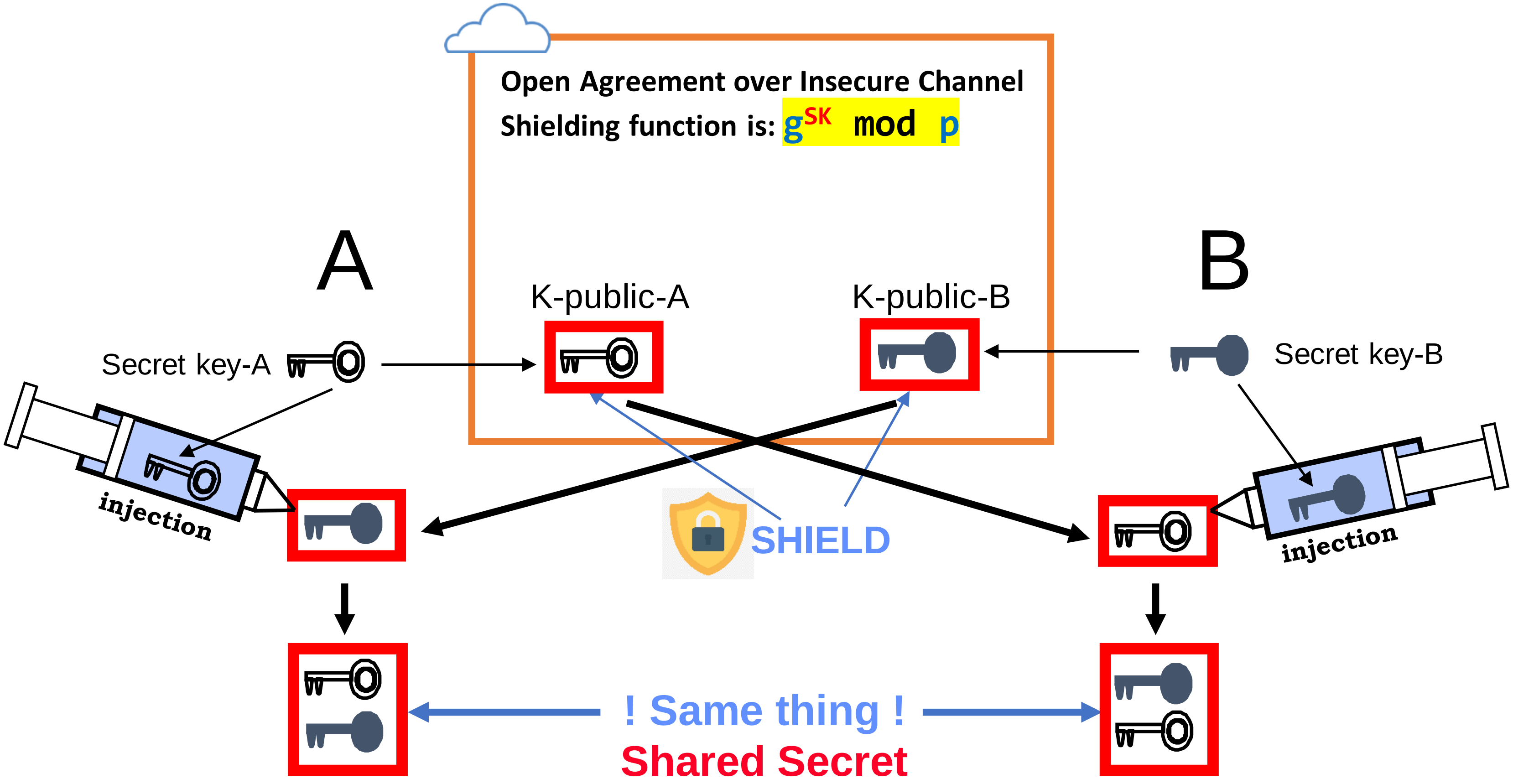
Discrete logarithm problem

$19^{?} \bmod 23 \xleftarrow{\text{Hard}} 2$

- The mathematics of DH is based on **Modulo exponentiation** + makes use of **big prime numbers** (100s of digits long)
- Security relies on the difficulty of computing **Discrete logarithm problem**. No efficient algorithm is known to solve it.

=> can only be solved through trial and error to find matching exponent but it will take a very long time (thousands of years with the world’s computing power to run through all possibilities!)

DH Breakthrough
Shared Secret without exchange of secrets



Open Agreement over Insecure Channel

Shielding function is: g^{SK} mod p

A

B

K-public-A

K-public-B

Secret key-A

Secret key-B

injection

injection

SHIELD

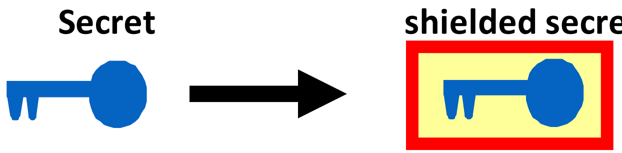
! Same thing !

Shared Secret

How to “publicly” hide (shield) a secret ?

Key idea: by using the so called *One-way Function*

Diffie-Hellmann used a One-Way function:



SHIELD = One Way Function

$$6 \longrightarrow 2$$

How:

$$g^{a} \bmod p$$

$$19^{6} \bmod 23$$

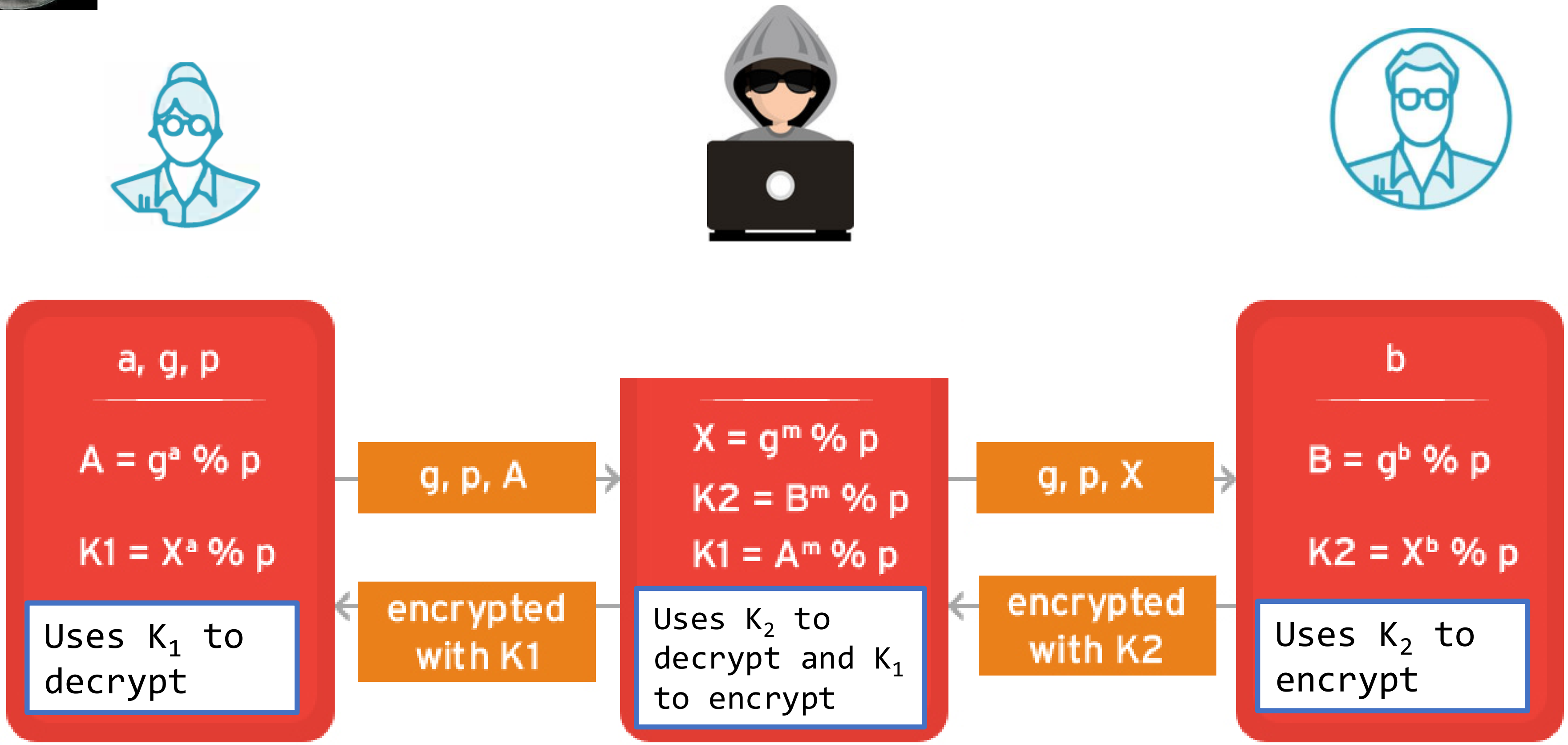
Discrete Logorithm Problem (DH Lock)

$$g,\ a \longrightarrow \boxed{g^{a} \bmod p} \longrightarrow A$$

$$A,\ g \longrightarrow \boxed{g^{?} \bmod p} \longrightarrow a$$

No efficient algorithm is known to compute the secret ? and remove the shield!

Man-in-the-middle (MITM) attack



a, g, p

$A = g^a \% p$

$K1 = X^a \% p$

Uses K_1 to decrypt

g, p, A

encrypted with K1

$X = g^m \% p$

$K2 = B^m \% p$

$K1 = A^m \% p$

Uses K_2 to decrypt and K_1 to encrypt

g, p, X

encrypted with K2

b

$B = g^b \% p$

$K2 = X^b \% p$

Uses K_2 to encrypt

$$K1 = A^m \% p = X^a \% p$$

$$K2 = X^b \% p = B^m \% p$$

Source: https://blog.trendmicro.com/trendlabs-security-intelligence/how-exploit-kit-operators-are-misusing-diffie-hellman-key-exchange/

Man-in-the-middle (MITM) attack

- Mallory (active attacker) intercepts the DH exchanged parameters in both directions
- Mallory and Alice use DH algorithm to calculate a shared key K_1

$$K_1 = A^m \% p = X^a \% p$$

- Mallory and Bob use DH algorithm to calculate a shared key K_2

$$K_2 = B^m \% p = X^b \% p$$

- Now Alice and Bob correspond through Mallory who can read all their messages
  - Mallory will use K_2 to decrypt messages from Bob then Encrypt them with K_1 before forwarding them to Alice

Summary

- Symmetric Key crypto has a major problem: How do two people who don't know each other share a key?
- A Diffie-Hellman key exchange lets them compute a shared key even in the presence of an eavesdropper, Eve.
- DH is vulnerable to Man-in-the-middle (MITM) attack

Resources

- DH original paper

https://ee.stanford.edu/~hellman/publications/24.pdf

- DH Wikipedia page

http://en.wikipedia.org/wiki/Diffie-Hellman_key_exchange

- Play with color mixing

https://trycolors.com/