



Diffie-Hellman Key Exchange

Color Mixing Example




The Problem of Key Exchange

- One of the main problems of symmetric key encryption is it requires a secure & reliable channel for the shared key exchange.
- The Diffie-Hellman Key Exchange protocol offers a way in which a public channel can be used to create a confidential shared key.



Modular what?

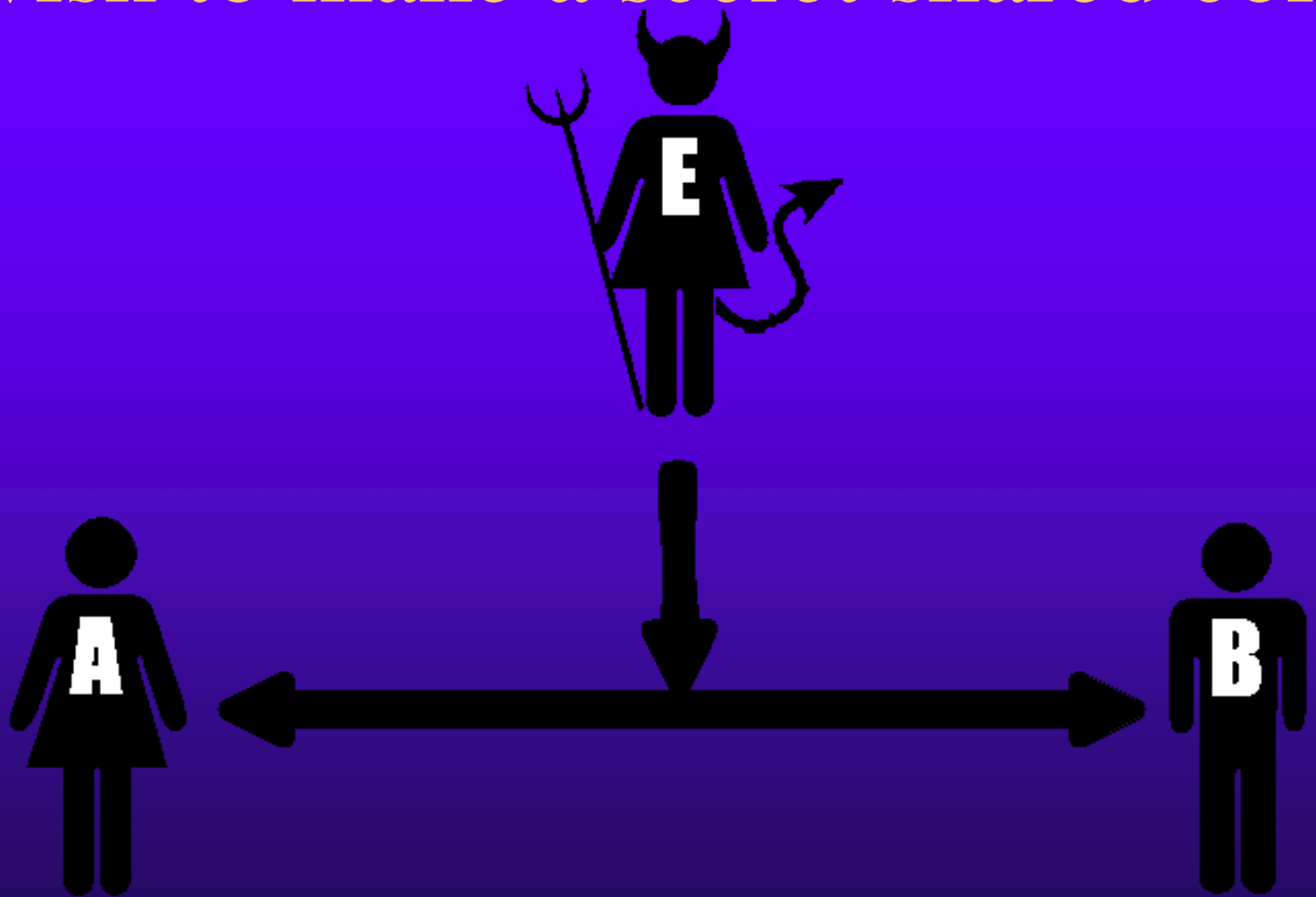
- In practice the shared encryption key relies on such complex concepts as *Modular Exponentiation*, *Primitive Roots* and *Discrete Logarithm Problems*.
- Let's see, we can explain the Diffie-Hellman algorithm with no complex mathematics.



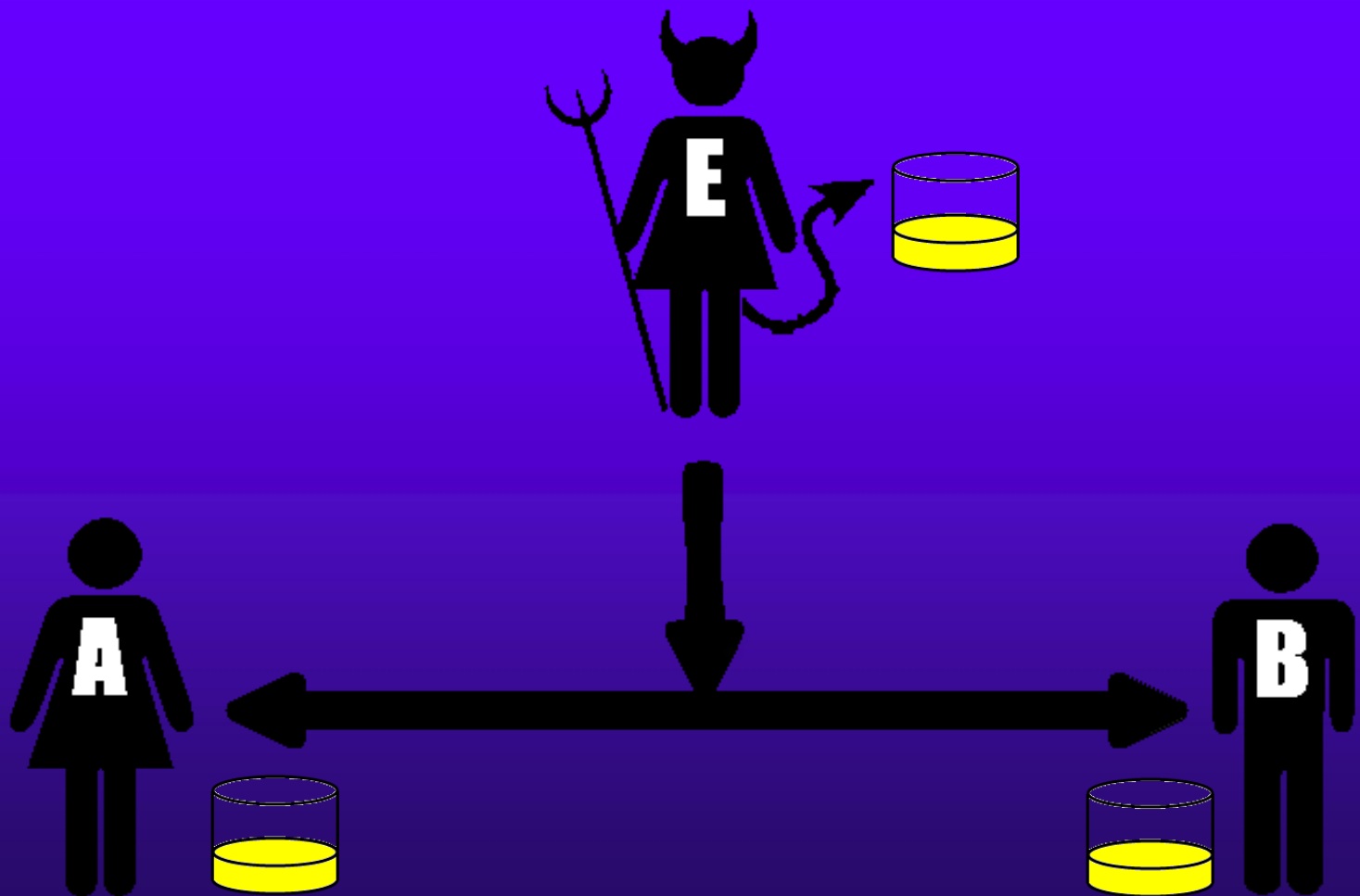
A Difficult One-Way Problem

- The first thing we require is a simple real-world operation that is easy to *Do* but hard to *Undo*.
 - You can ring a bell but not unring one.
 - Toothpaste is easy to squeeze out of a tube but famously hard to put back in.
- In our example we will use *Mixing Colors*.
 - Easy to mix 2 colors, hard to unmix

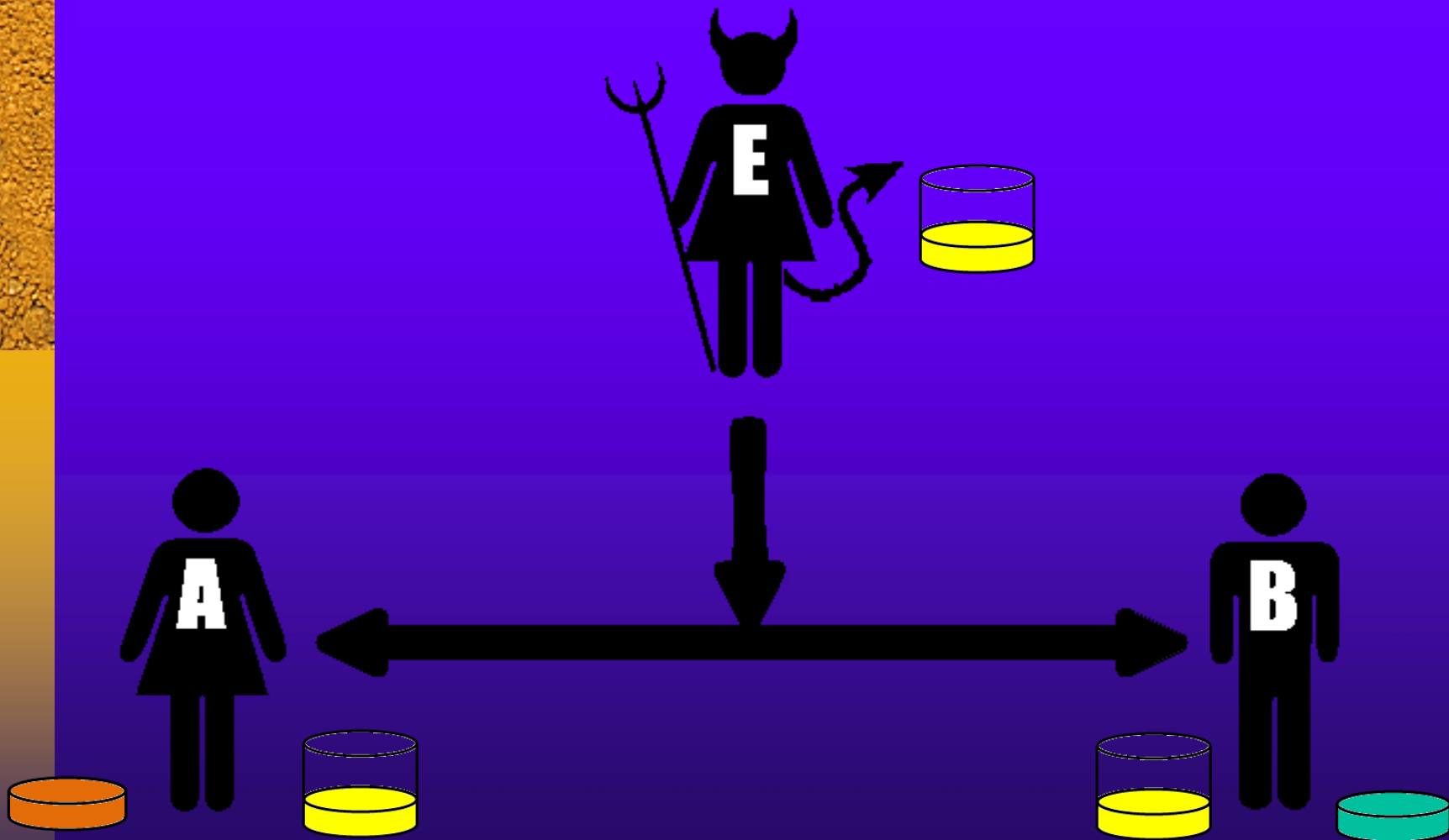
Alice & Bob with Eve listening
wish to make a secret shared color



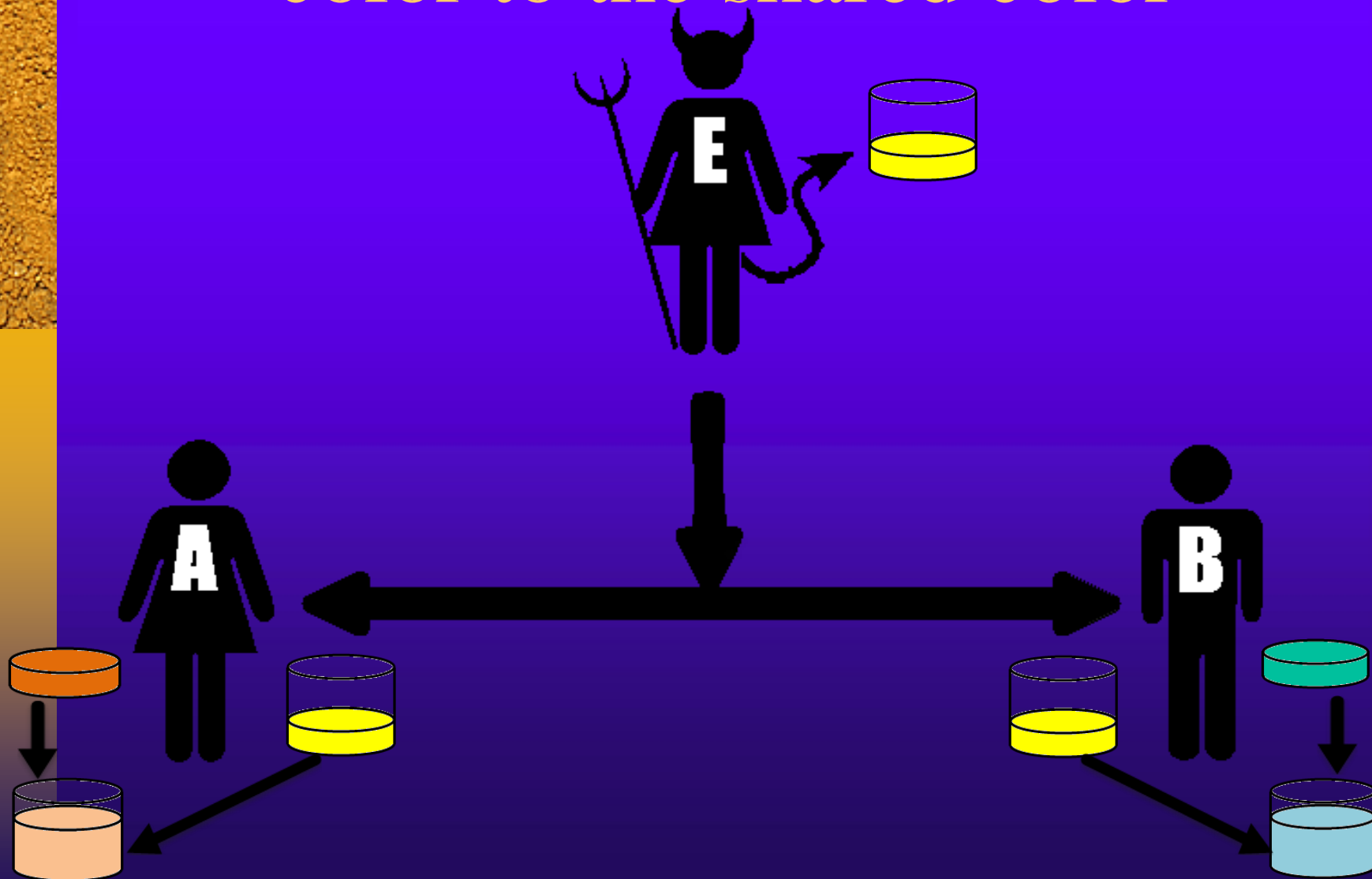
Step 1 - Both publicly agree to a shared color



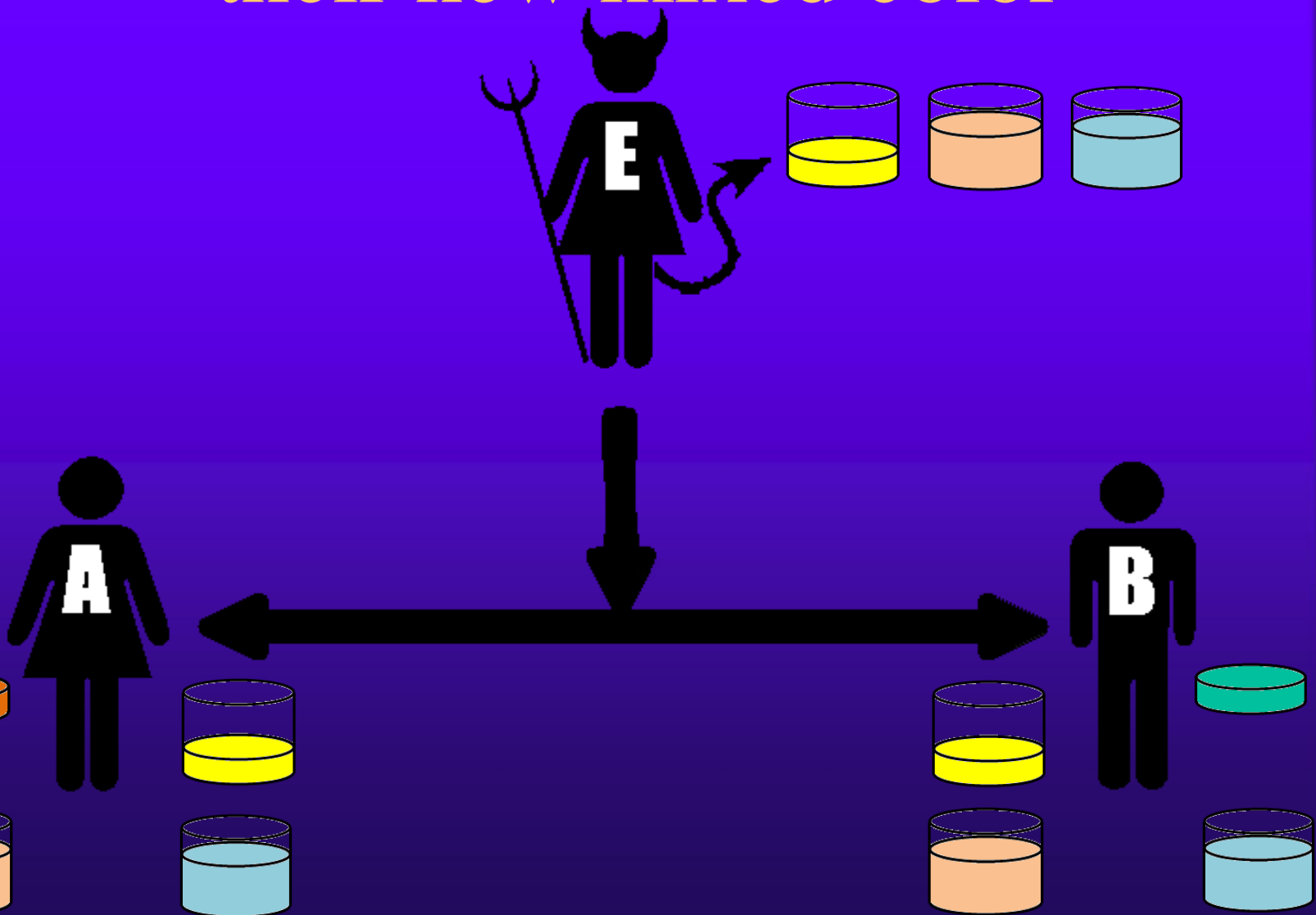
Step 2 - Each picks a secret color



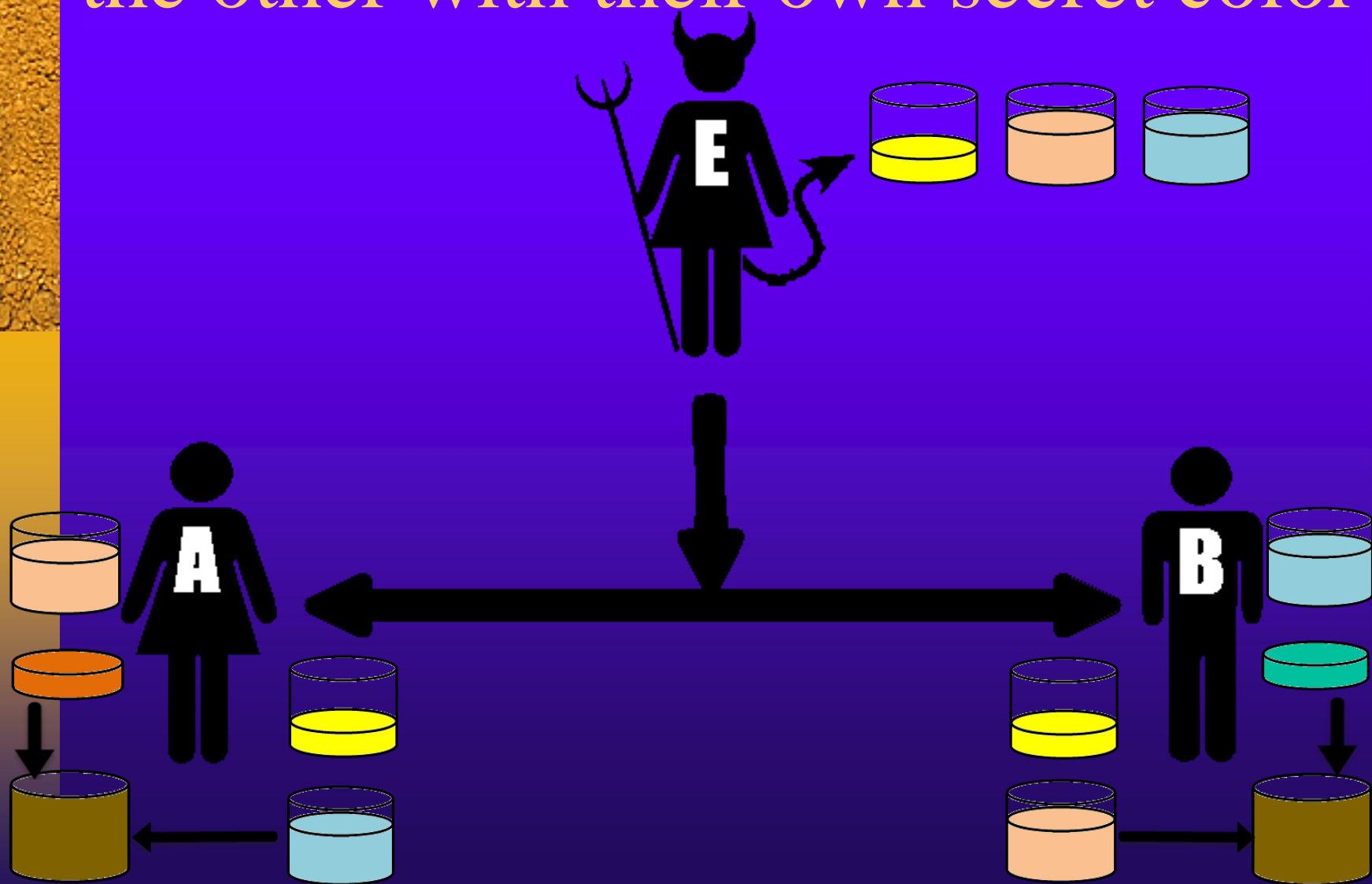
Step 3 - Each adds their secret color to the shared color




Step 4 - Each sends the other
their new mixed color




Each combines the shared color from the other with their own secret color





Alice & Bob have agreed to a shared color unknown to Eve

- How is it that Alice & Bob's final mixtures are identical?
- Alice mixed
 - $[(\text{Yellow} + \text{Teal})_{\text{from Bob}}] + \text{Orange}$
- Bob mixed
 - $[(\text{Yellow} + \text{Orange})_{\text{from Alice}}] + \text{Teal}$



Alice & Bob have agreed to a shared color unknown to Eve

- How is it that Alice & Bob's final mixture is secret?
- Eve never has knowledge of the secret colors of either Alice or Bob
- Unmixing a color into its component colors is a hard problem



Diffie-Hellman Key Exchange

Adding Mathematics



Diffie-Hellman Key Exchange:

- It is a protocol that enables two users to establish a secret key using a public-key scheme based on discrete logarithms. The protocol is secure only if the authenticity of the two participants can be established.

Primitive Root

- A primitive root of a prime number **p** is one whose powers modulo generate all the integers from **1** to **p-1**. That is, if **a** is a primitive root of the prime number **p**, then the numbers

$$a \bmod p, a^2 \bmod p, \dots, a^{p-1} \bmod p$$

- are distinct and consist of the integers from **1** through **p - 1** in some permutation.



Diffie-Hellman Key Exchange

Algorithm:

- For this scheme, there are two publicly known numbers: a prime number q and an integer α that is a primitive root of q .
- Suppose the users A and B wish to exchange a key K . User A selects a random integer $X_A < q$ and computes Y_A . Similarly, user B independently selects a random integer $X_B < q$ and computes Y_B .
- Each side keeps the X value private and makes the Y value available publicly to the other side.



Algorithm



Global Public Elements

q	prime number
α	$\alpha < q$ and α a primitive root of q

User A Key Generation

Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{X_A} \bmod q$

User B Key Generation

Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{X_B} \bmod q$

Calculation of Secret Key by User A

$$K = (Y_B)^{X_A} \bmod q$$

Calculation of Secret Key by User B

$$K = (Y_A)^{X_B} \bmod q$$



Step 1 –Publicly shared information

- Alice & Bob publicly agree to a large prime number called the modulus, or q .
- Alice & Bob publicly agree to a number called the generator, or α , which has a primitive root relationship with q .
- In our example we'll assume
 - $q = 17$
 - $\alpha = 3$
- Eve is aware of the values of q or α .




Step 2 – Select a secret key

- Alice selects a secret key, which we will call X_a .
- Bob selects a secret key, which we will call X_b .
- For our example assume:
 - $X_a = 54$
 - $X_b = 24$
- Eve is unaware of the values of X_a or X_b .



Step 3 – Combine secret keys with public information

- Alice combines her secret key of X_a with the public information to compute Y_a .
 - $Y_a = \alpha^{X_a} \bmod q$
 - $Y_a = 3^{54} \bmod 17$
 - $Y_a = 15$



Step 3 – Combine secret key with public information

- Bob combines his secret key of X_b with the public information to compute Y_b .
 - $Y_b = \alpha^{X_b} \bmod q$
 - $Y_b = 3^{54} \bmod 17$
 - $Y_b = 16$




Step 4 – Share combined values

- Alice shares her combined value, Y_a , with Bob. Bob shares his combined value, Y_b , with Alice.
- Sent to Bob
 - $Y_a = 15$
- Sent to Alice
 - $Y_b = 16$
- Eve is privy to this exchange and knows the values of Y_a and Y_b .



Step 5 – Compute Shared Key

- Alice computes the shared key.
 - $K = (Yb)^{Xa} \bmod q$
 - $K = (16)^{54} \bmod 17$
 - $K = 1$
- Bob computes the shared key.
 - $K = (Ya)^{Xb} \bmod q$
 - $K = (15)^{24} \bmod 17$
 - $K = 1$



Alice & Bob have a shared encryption key, unknown to Eve

- Alice & Bob have created a shared secret key, K , unknown to Eve
- In our example $K=1$
- The shared secret key can now be used to encrypt & decrypt messages by both parties.

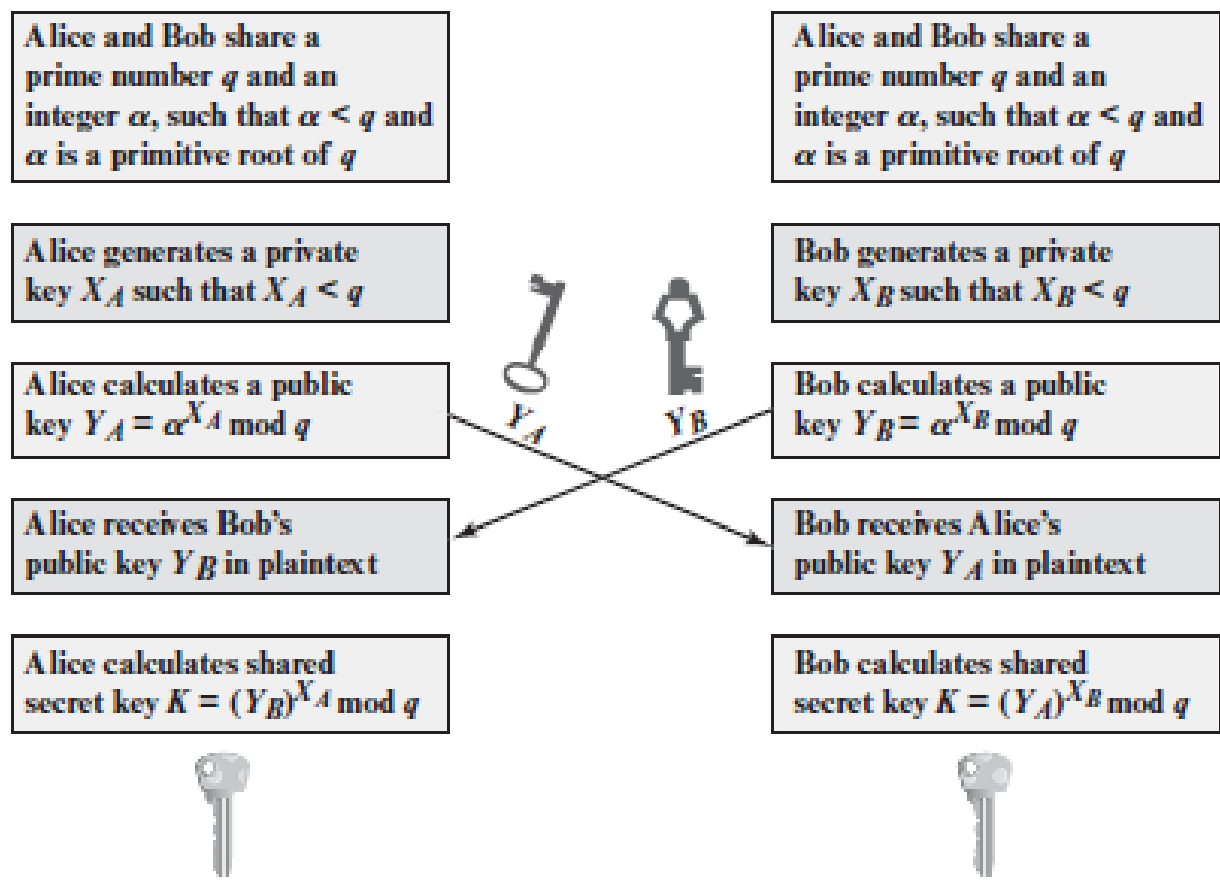
Key Exchange Protocol Scenario using Diffie-Hellman



Alice



Bob



Man-in-the-Middle Attack:

The protocol depicted in the previous figure is insecure against a man-in-the-middle attack. Suppose Alice and Bob wish to exchange keys, and Darth is the adversary. The attack proceeds as follows.

1. Darth prepares for the attack by generating two random private keys X_{D1} and X_{D2} and then computing the corresponding public keys Y_{D1} and Y_{D2} .
2. Alice transmits Y_A to Bob.
3. Darth intercepts Y_A and transmits Y_{D1} to Bob. Darth also calculates $K2 = (Y_A)^{X_{D2}} \bmod q$.
4. Bob receives Y_{D1} and calculates $K1 = (Y_{D1})^{X_B} \bmod q$.
5. Bob transmits Y_B to Alice.
6. Darth intercepts Y_B and transmits Y_{D2} to Alice. Darth calculates $K1 = (Y_B)^{X_{D1}} \bmod q$.
7. Alice receives Y_{D2} and calculates $K2 = (Y_{D2})^{X_A} \bmod q$.

At this point, Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key **K1** and Alice and Darth share secret key **K2**. All future communication between Bob and Alice is compromised in the following way.

1. Alice sends an encrypted message M : $E(K2, M)$.
2. Darth intercepts the encrypted message and decrypts it to recover M .
3. Darth sends Bob $E(K1, M)$ or $E(K1, M')$, where M' is any message. In the first case, Darth simply wants to eavesdrop on the communication without altering it. In the second case, Darth wants to modify the message going to Bob.

The key exchange protocol is vulnerable to such an attack because it does not authenticate the participants. This vulnerability can be overcome with the use of digital signatures and public-key certificates.

