# Asymmetric Cryptography

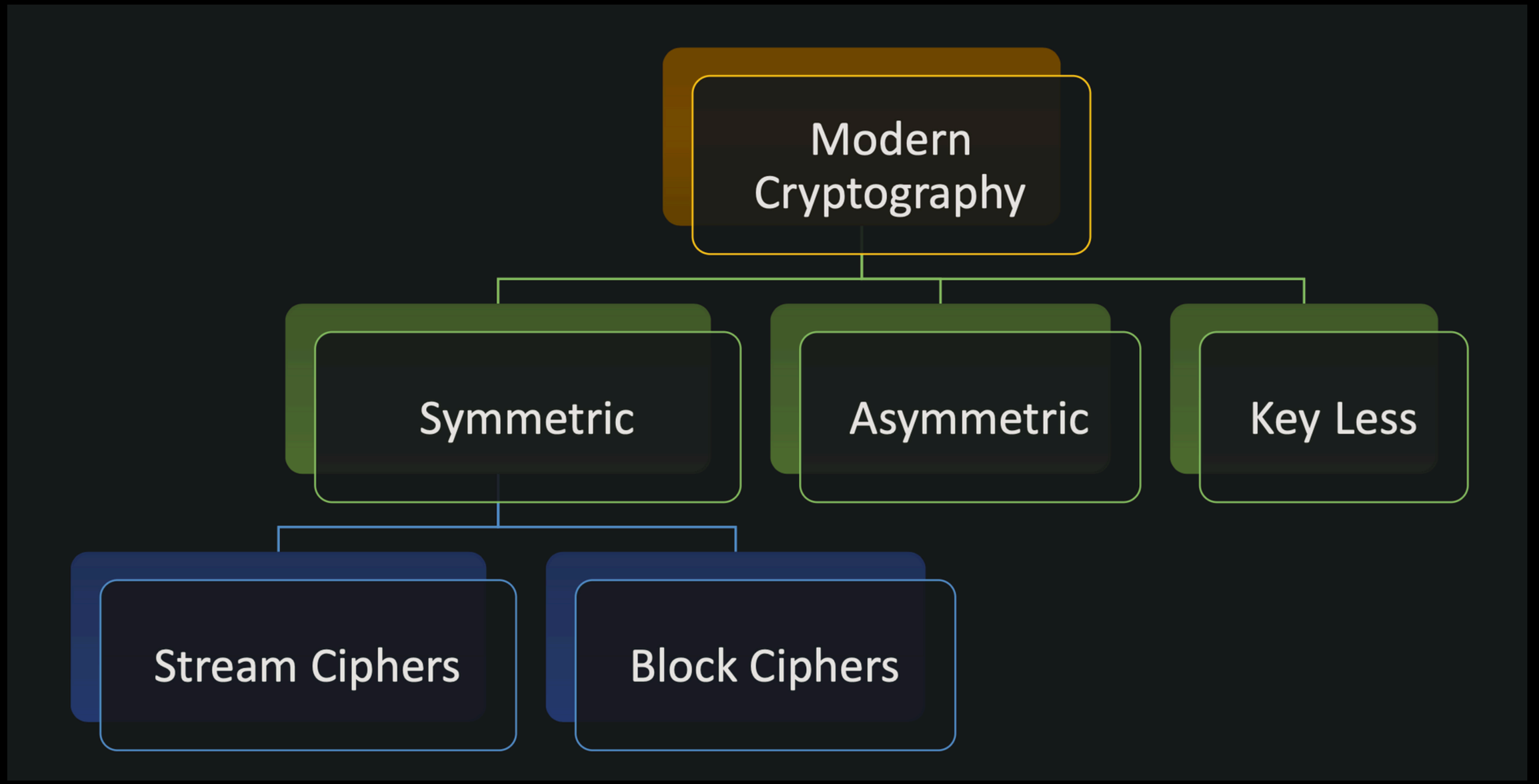
“When math becomes trust.”

**CЯ4CK1NG-THΞ-C0D3**

# Today’s Roadmap

- What is Asymmetric Cryptography?
- Public & Private Keys
- RSA – Math & Intuition

# Firstly recap of Modern Cryptography

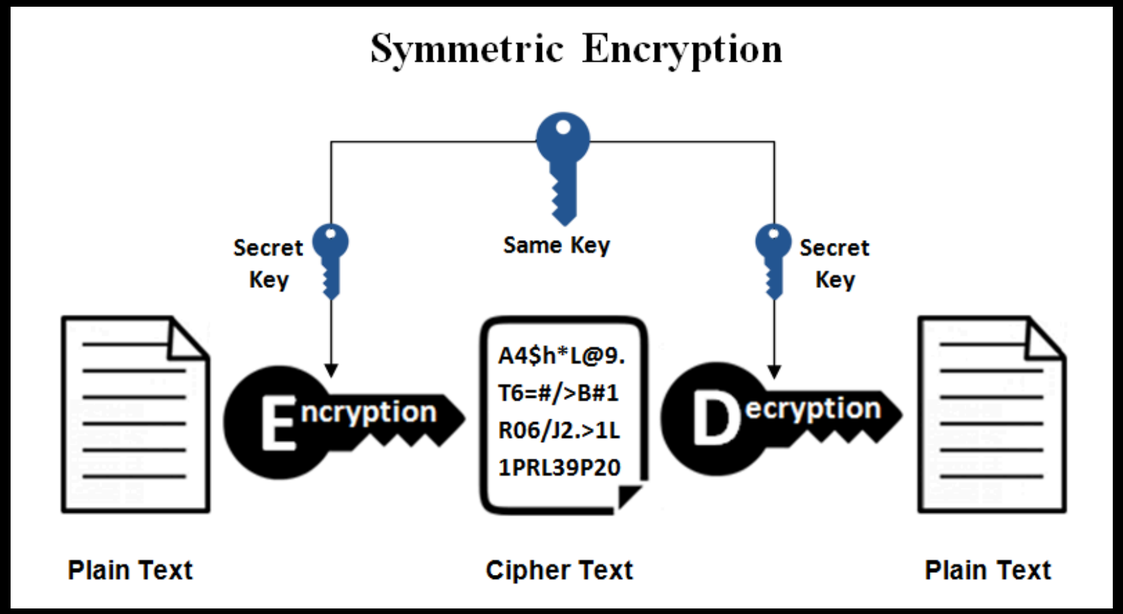

Modern Cryptography
Symmetric
Asymmetric
Key Less
Stream Ciphers
Block Ciphers

# What is Symmetric Cryptography?

# Symmetric Cryptography

*Think: One lock, one key — shared between trusted parties.*

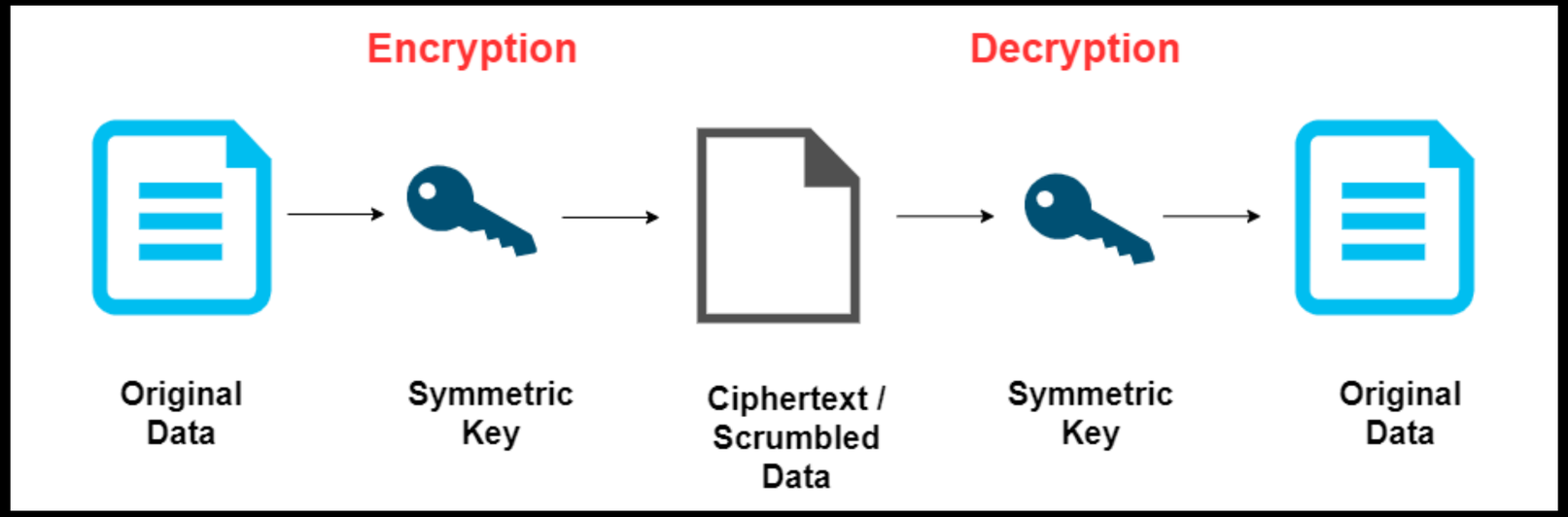
- Both encryption and decryption use the same secret key
- Fast, lightweight, and great for bulk data
- Used in: File encryption, Wi-Fi, ZIPs, and even disk drives



# What are challenges and why use symmetric key cryptography?

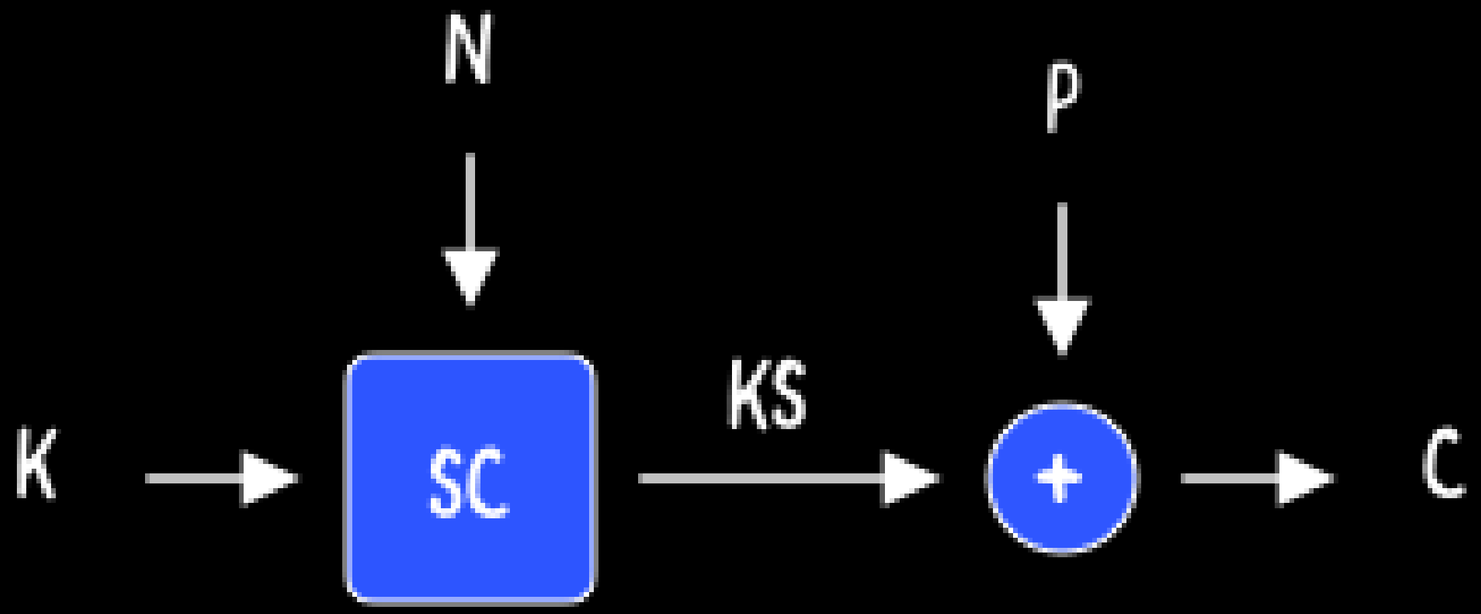
# Advantages of SKC

- Encryption is speed.
- Less processing power.
- it’s built in.
- Keys always remain local



# Types of SKC

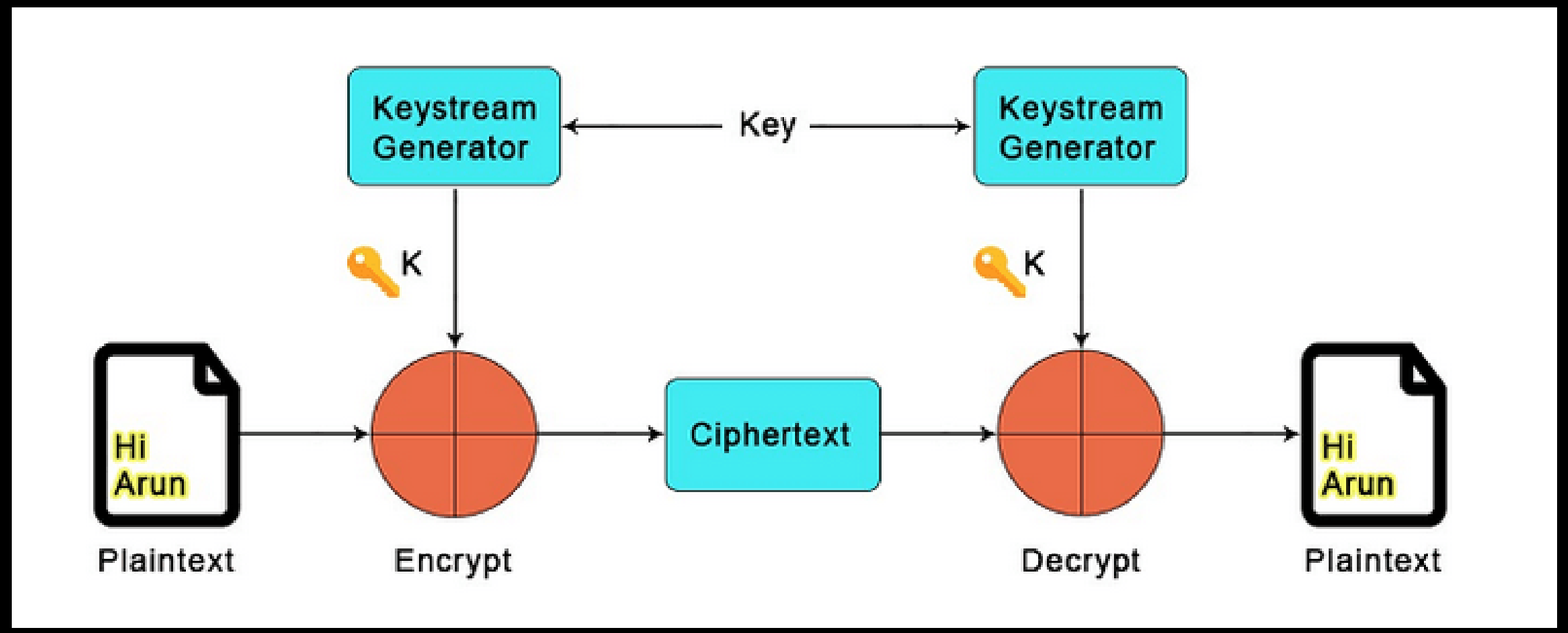
# Stream Ciphers



# Why Stream Ciphers?

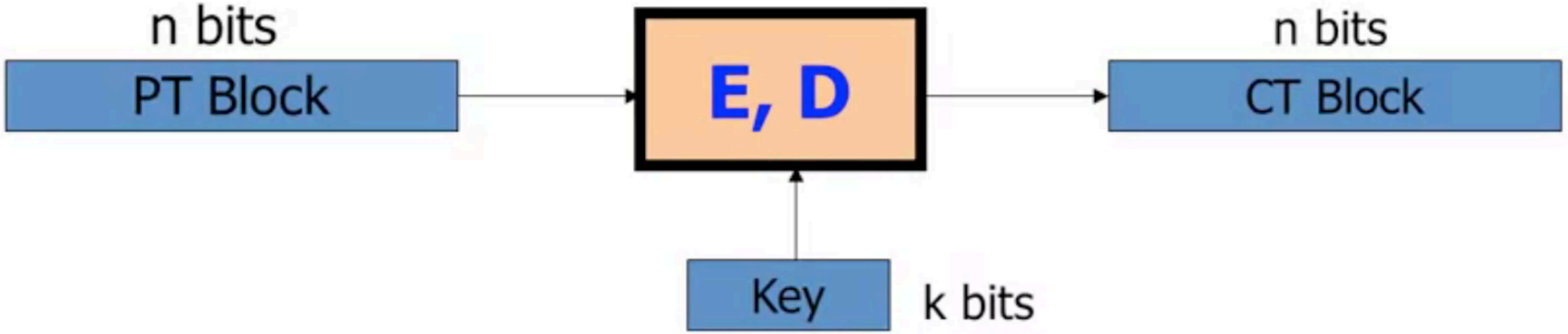
Stream ciphers come with plenty of benefits, including:

- Speed
- Low complexity
- Serial nature
- Ease of use



# Most majorly used second type is?

# Block ciphers: crypto work horse




Canonical examples:

1. 3DES: n= 64 bits, k = 168 bits
2. AES: n=128 bits, k = 128, 192, 256 bits

# What are some types of Block cipher schemes?

# Block Cipher Schemes

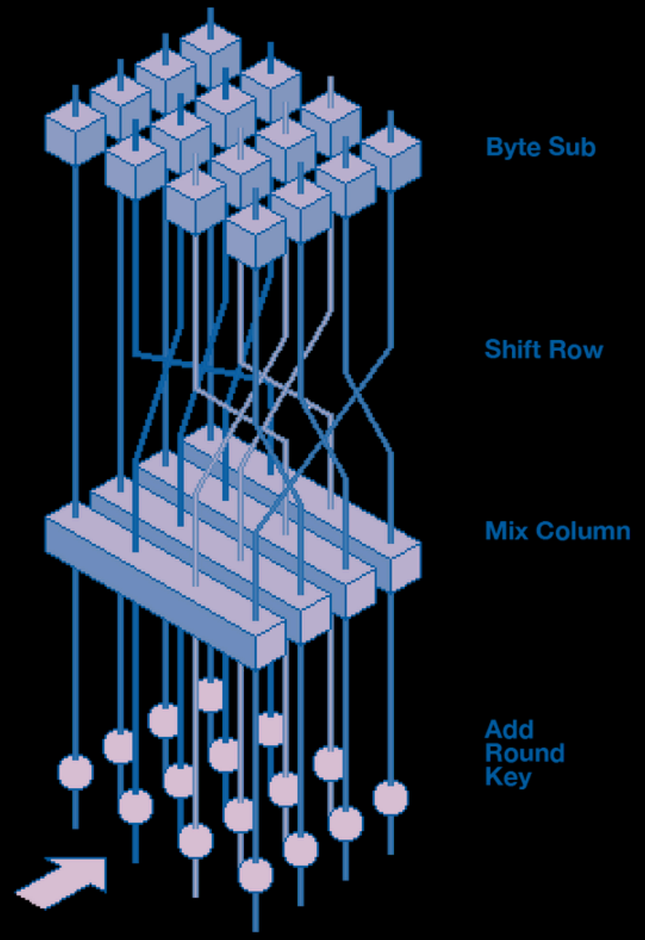
Block ciphers have many schemes to operate. Some of the widely used and popular ones are as follows:

- DES: Digital Encryption Standard. It’s broken and no more used.
- Triple DES: Repeated DES. Much stronger algos are available than this.
- AES: Advanced Encryption Standard.
- Serpent: Has block size of 128 bits and keys of length 128,196 or 256 bits. Slower but very secure than other block ciphers.
- TwoFish: With a block size of 128 bits, it is based on an earlier block cipher called Blowfish which uses a block of 64 bits

# Finally AES

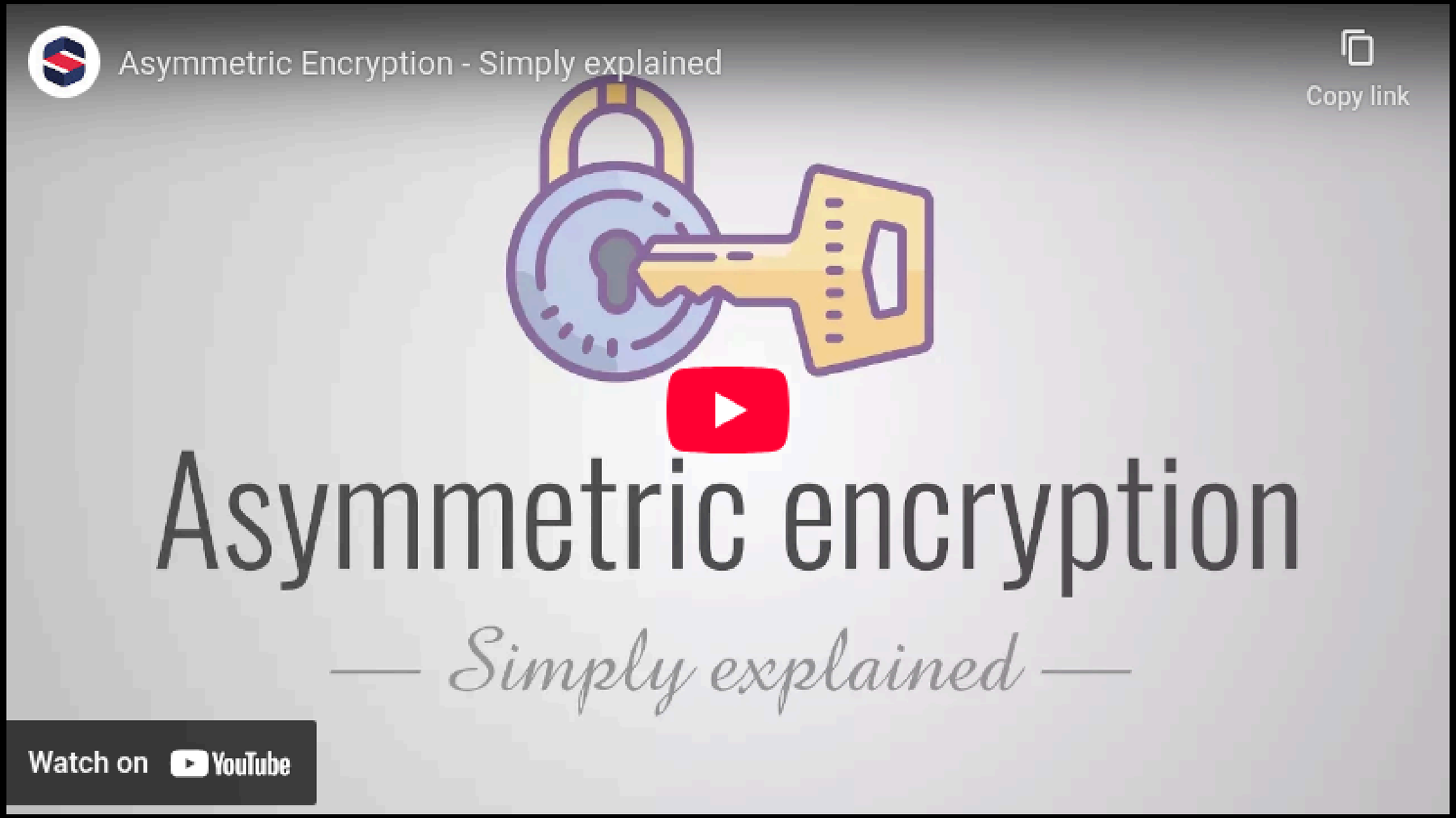
# AES: The Transformation Process

- SubBytes → Non-linear substitution.
- ShiftRows → Rearranges data.
- MixColumns → Diffuses info across bytes.
- AddRoundKey → Combines with secret key.



# Hands on for implementing the same

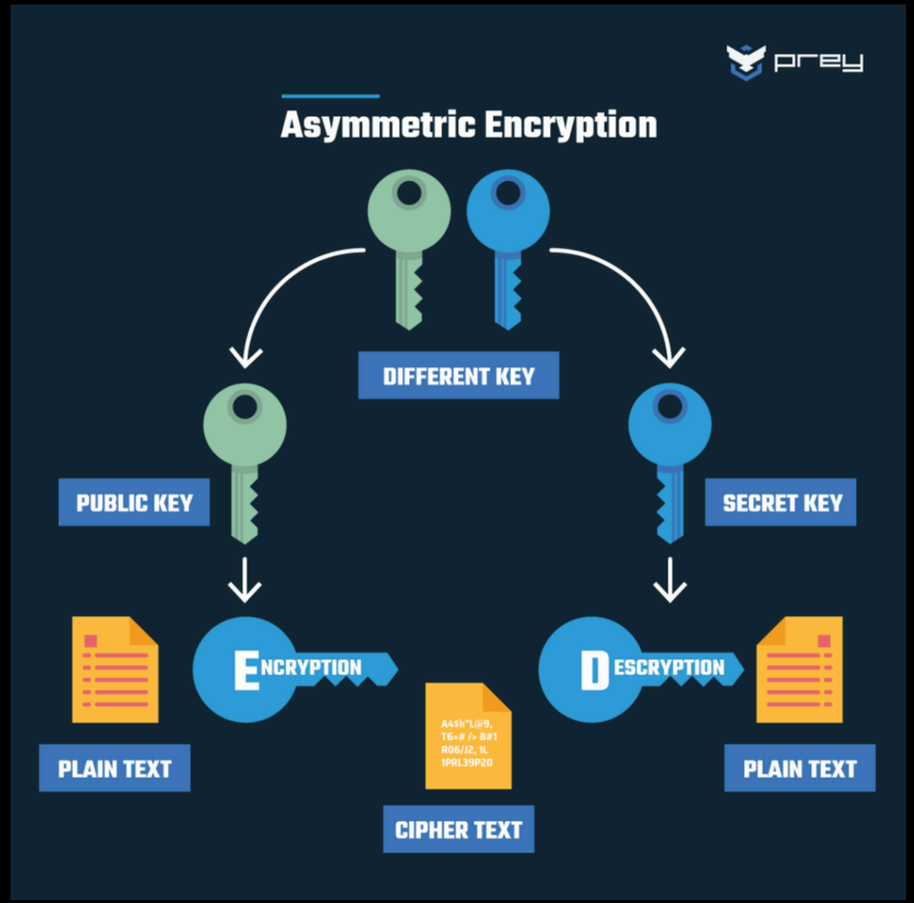
Now lets come to todays class


Asymmetric Encryption - Simply explained
Copy link
Asymmetric encryption
— Simply explained —
Watch on YouTube

# So, What is Asymmetric Cryptography?

*two keys — mathematically related, but not identical.*

- Also called Public Key Cryptography (PKC).
- Uses a key pair:
  - Public Key: shared openly.
  - Private Key: kept secret.
- Encrypt with one → decrypt with the other.



# Lets rewind a little.. Firstly why AKC?

# Imagine this scenario

N users are connected in a network and any two of them may want to communicate.

☞ This would require each user to securely store n − 1 different

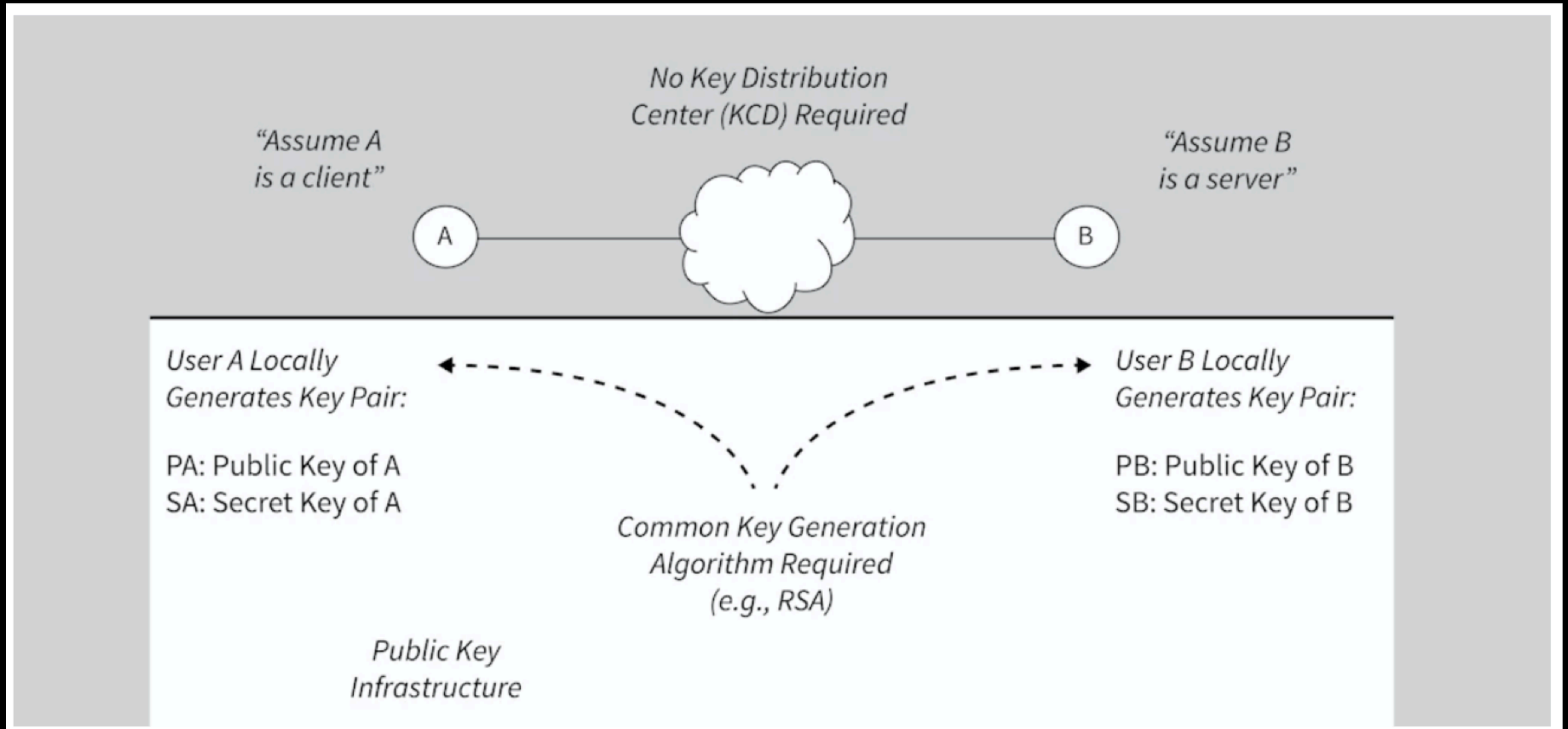
symmetric keys (one for each other user), resulting in a total of n(n − 1)/2 keys.

☞ If the network is connecting 2000 university students, then there will be roughly 2 million different keys.

☞ A huge key management problem with questions like:

- How do you add add a new user to the system?
- What if a user’s key is compromised?
- How long should a key be considered valid and how should we refresh them?

# Birth of Public Key Cryptography


No Key Distribution
Center (KCD) Required
"Assume A
is a client"
"Assume B
is a server"
A
B
User A Locally
Generates Key Pair:
PA: Public Key of A
SA: Secret Key of A
User B Locally
Generates Key Pair:
PB: Public Key of B
SB: Secret Key of B
Common Key Generation
Algorithm Required
(e.g., RSA)
Public Key
Infrastructure

# Public Key Cryptography Early History

☞ Proposed by Diffie and Hellman, in “New Directions in Cryptography” (1976)

• Public-key encryption schemes

• Key distribution systems

• Diffie-Hellman key agreement protocol

• Digital signature

☞ Public-key encryption was proposed in 1970 in a classified paper by James Ellis

☞ paper made public in 1997 by the British Governmental Communications Headquarters

☞ Concept of digital signature is still originally due to Diffie & Hellman Executive MTech in Cybersecurity Engineering 11

# So why is this secure?

# Mechanisms of Public-Key Cryptography

Here are main mechanisms that can be realized with asymmetric cryptography:

- Key Distribution (e.g., Diffie-Hellman key exchange, RSA) without a pre-shared secret (key)
- Nonrepudiation and Digital Signatures (e.g., RSA, DSA or ECDSA) to provide message integrity.
- Identification using challenge-response protocols with digital signatures
- Encryption (e.g., RSA / ElGamal)
- Disadvantage: Computationally very intensive (1000 times slower than symmetric Algorithms)

# To summarise

☞ Public key ciphers consist of:

- Key generation algorithm
- Encryption algorithm
- Decryption algorithm

☞ Designed around computationally hard mathematical problems

☞ Very hard to solve without key, i.e. trapdoor functions

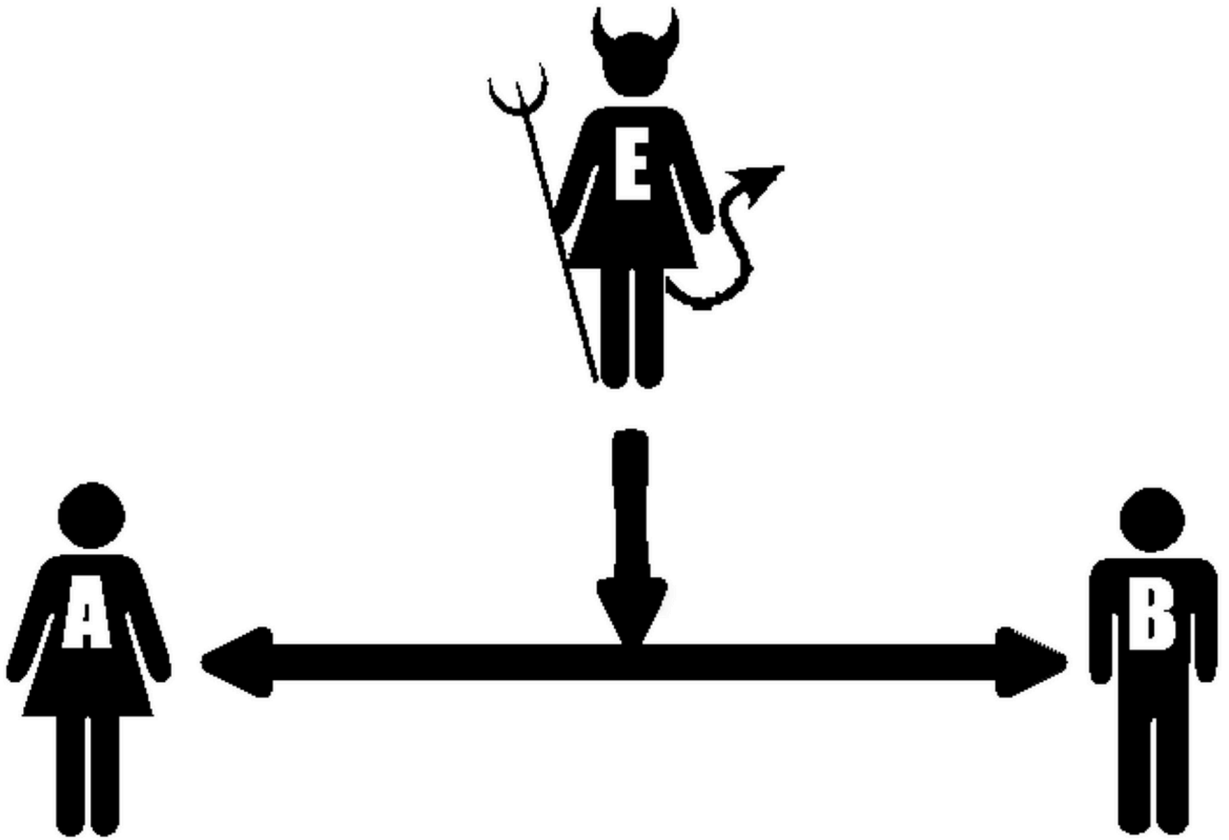
- Finding prime factors of large integers
- Solving logarithms in modulo arithmetic
- Solving logarithms on elliptic curves

# Lets understand this better-
# The Diffie-Hellman Key exchange

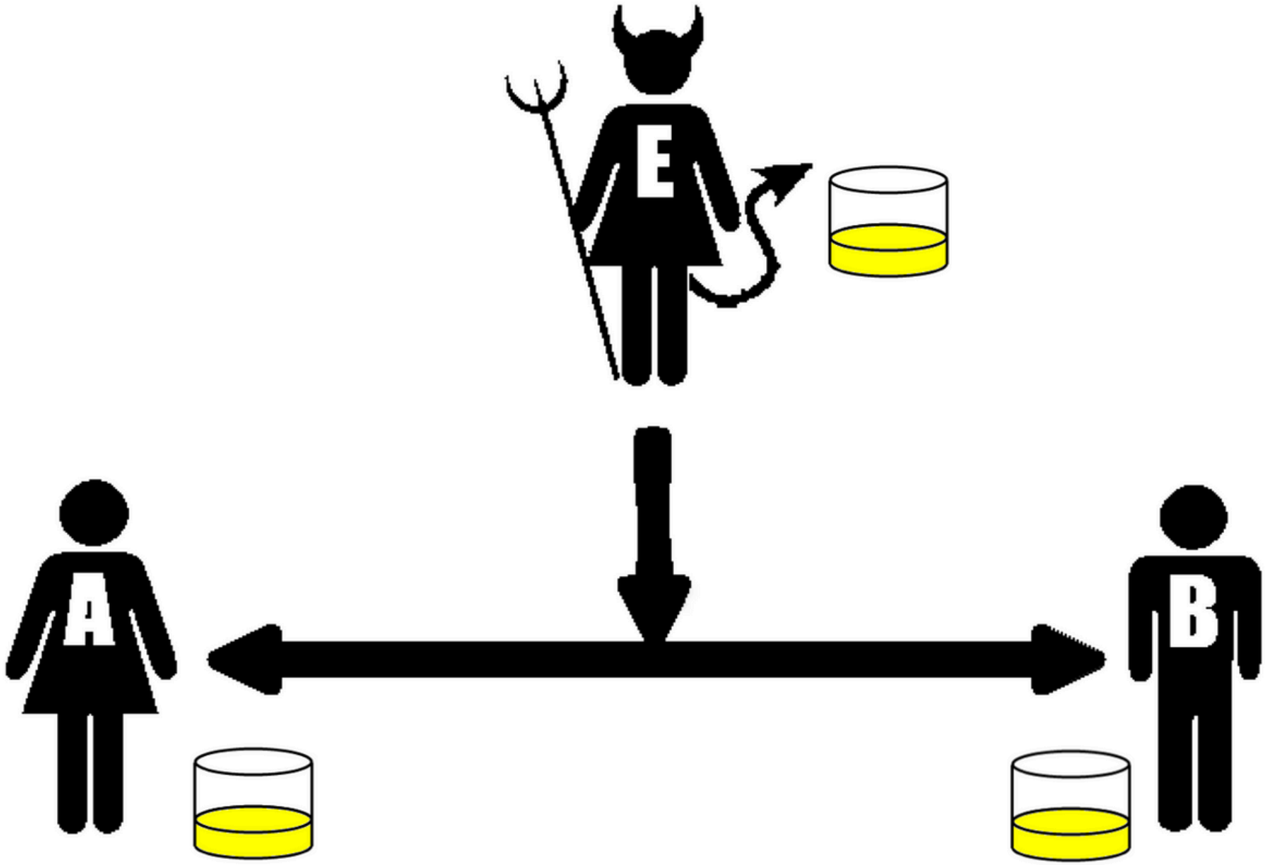

Diffie-Hellman Key Exchange
Copy link
Watch on YouTube

# What did that explainer show us?

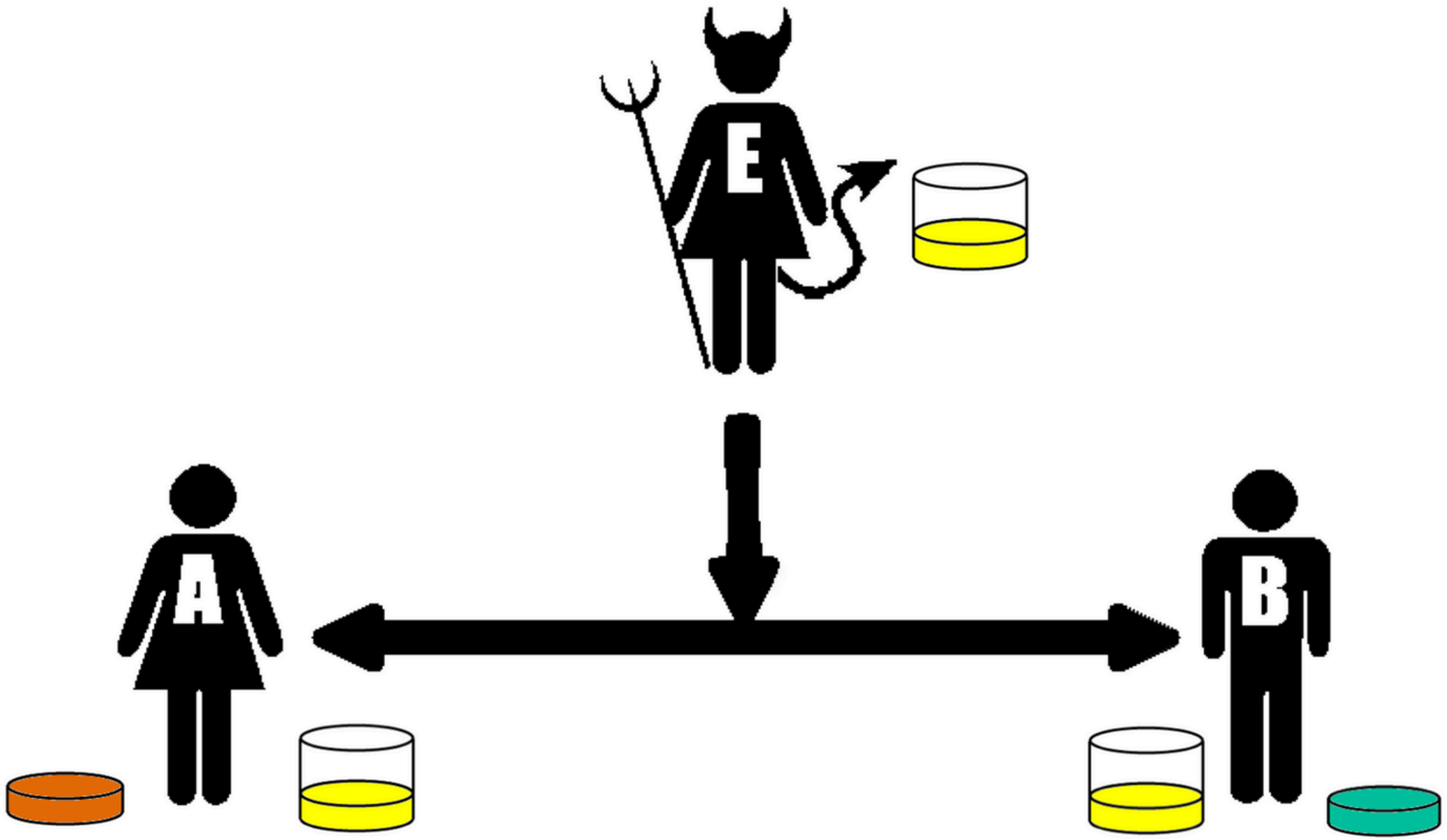
Alice & Bob with Eve listening wish to make a secret shared color



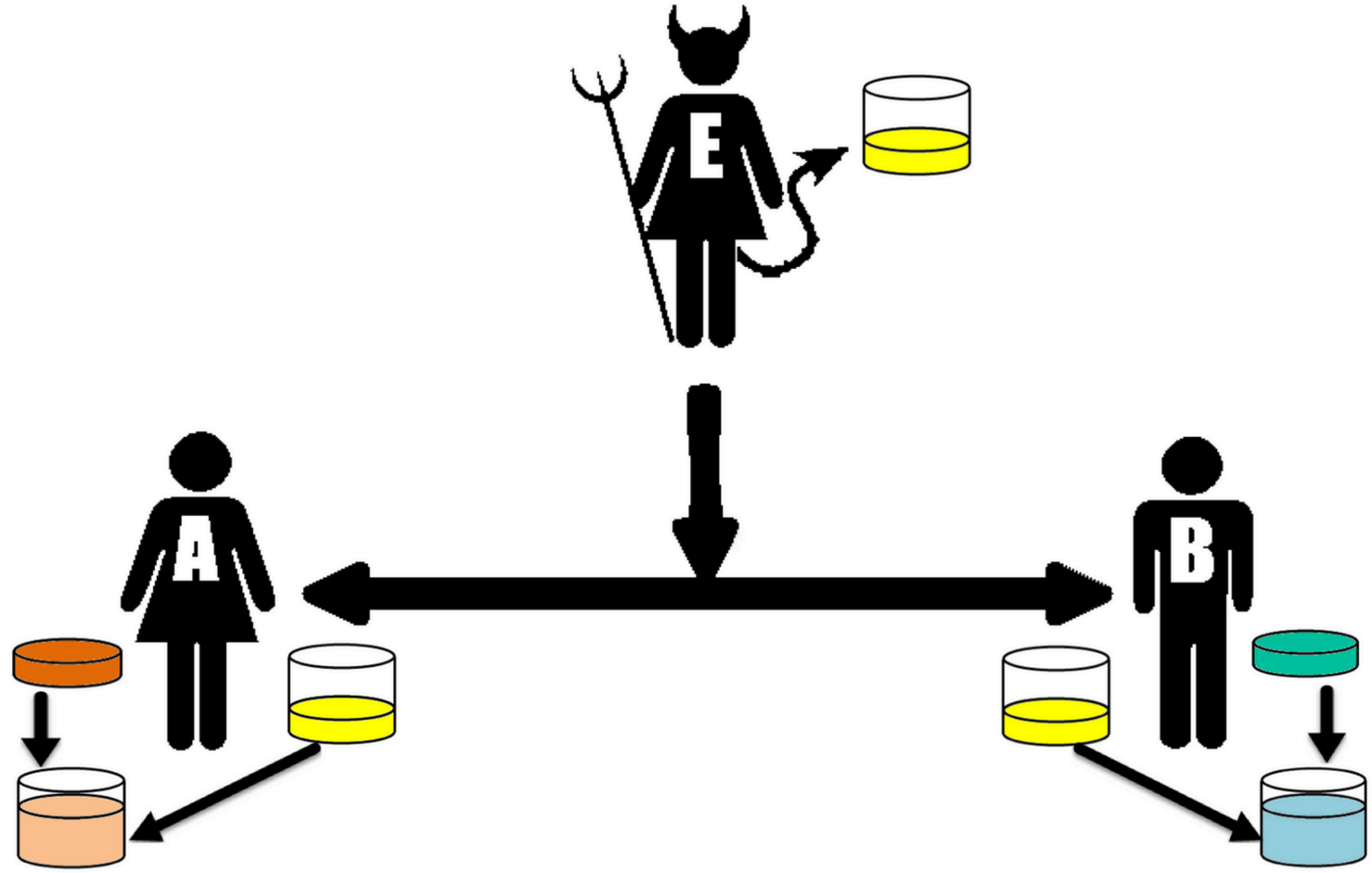
# Step 1 - Both publicly agree to a shared color



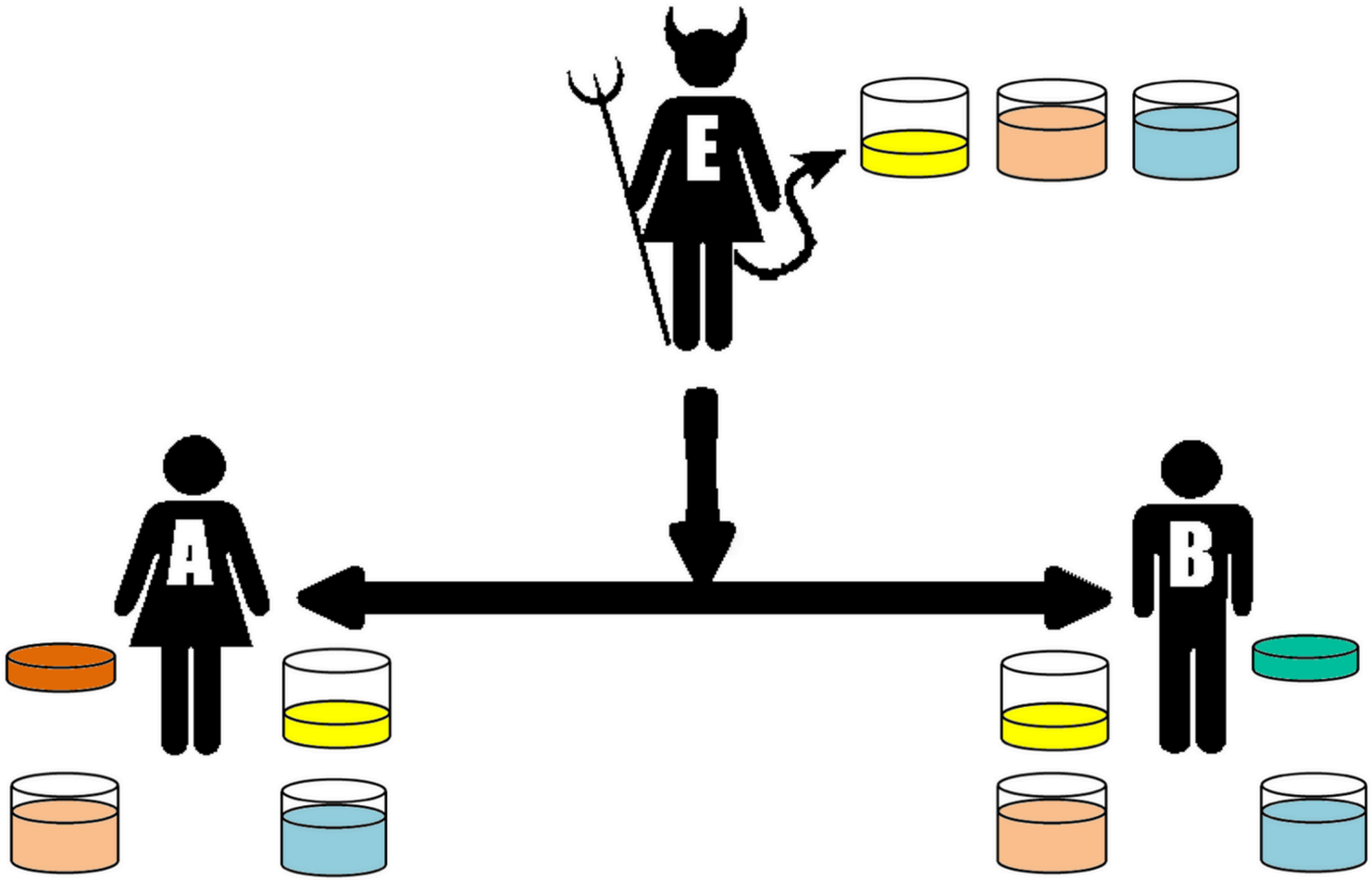
# Step 2 - Each picks a secret color



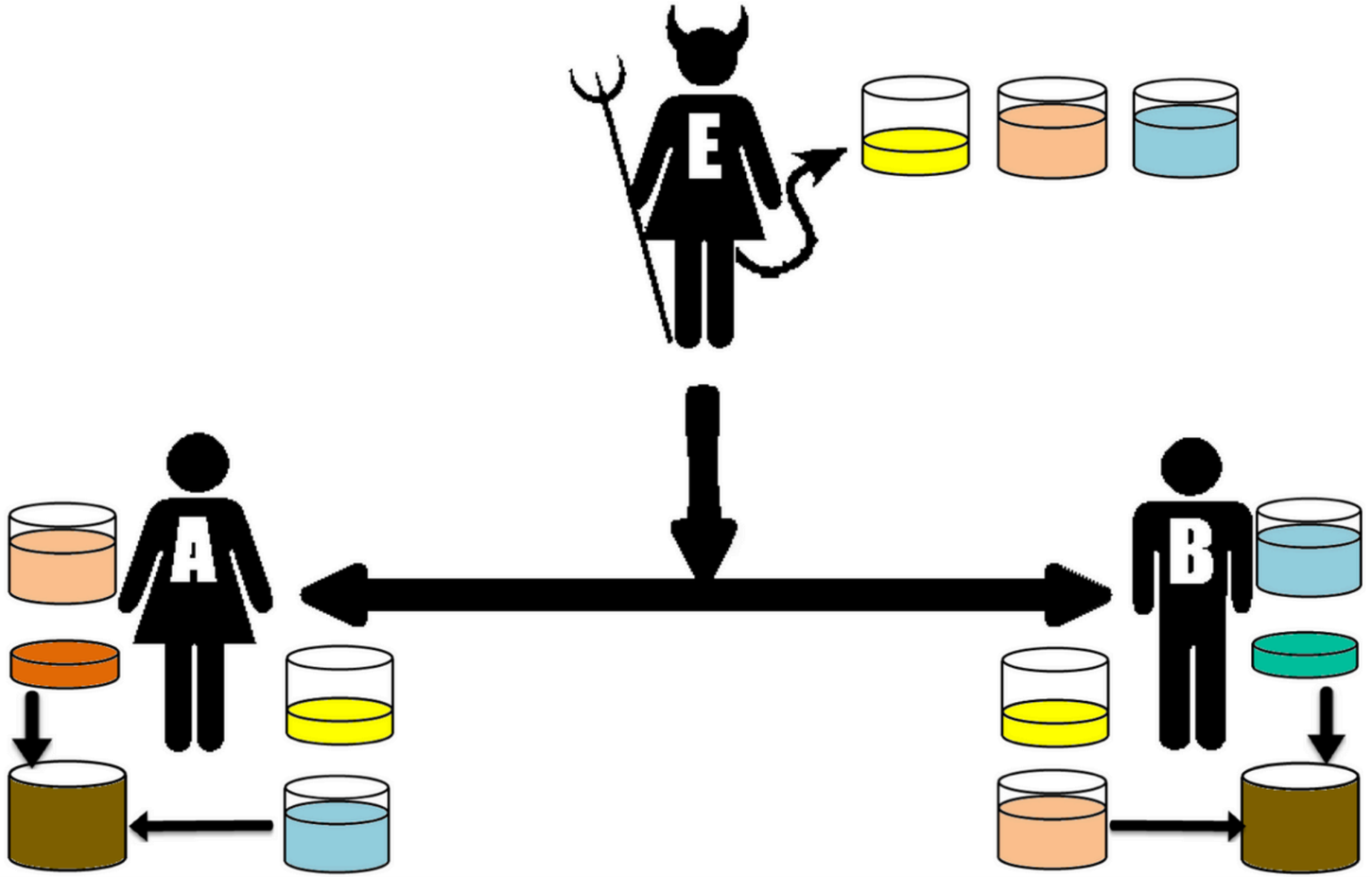
# Step 3 - Each adds their secret color to the shared color



# Step 4 - Each sends the other their new mixed color



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



# Lets take an example

**Example:**

Step 1: Alice and Bob get public numbers P = 23, G = 9

Step 2: Alice selected a private key a = 4 and

Bob selected a private key b = 3

Step 3: Alice and Bob compute public values

Alice: x =(9^4 mod 23) = (6561 mod 23) = 6

Bob: y = (9^3 mod 23) = (729 mod 23) = 16

Step 4: Alice and Bob exchange public numbers

Step 5: Alice receives public key y =16 and

Bob receives public key x = 6

Step 6: Alice and Bob compute symmetric keys

Alice: ka = y^a mod p = 65536 mod 23 = 9

Bob: kb = x^b mod p = 216 mod 23 = 9

Step 7: 9 is the shared secret.

# Exercise - 1

**Exercise – 1:** $q = 353, \alpha = 3, X_A = 97, X_B = 233.$

Determine ***Public Key*** and ***Shared key*** for both users using Diffie-Hellman key exchange algorithm.

**Solution:**

1. Here $q = 353, \alpha = 3$
2. Calculate *Public key for user A*. $X_A = 97,$

   $Y_A = \alpha^{X_A} \bmod q = 3^{97} \bmod 353 = 40$ **$Y_A = 40$**
3. Calculate *Public key for user B*. $X_B = 233,$

   $Y_B = \alpha^{X_B} \bmod q = 3^{233} \bmod 353 = 248$ **$Y_B = 248$**
4. Calculation of *secret key by user A*,

   $K = (Y_B)^{X_A} \bmod q = 248^{97} \bmod 353 = 160$ **$K = 160$**
5. Calculation of *secret key by user B*,

   $K = (Y_A)^{X_B} \bmod q = 40^{233} \bmod 353 = 160$ **$K = 160$**

# RSA: Rivest–Shamir–Adleman algorithm

*Difficult math is the solution to our problems*

- Invented: 1977 by Rivest, Shamir, Adleman
- Based on: difficulty of factoring large primes.

Core idea:

- Choose two large primes: p, q.
- Compute $n = p \times q$.
- Compute $\varphi(n) = (p-1)(q-1)$.
- Choose public key e.
- Compute private key $d = e^{-1} \bmod \varphi(n)$.

https://youtu.be/4zahvcJ9glg?si=mFt-L9C1S7x1AIrK

# The Power of One-Way Math

- Easy to multiply primes.
- Hard to factor the product.
- This “trapdoor” property keeps RSA secure.
- Example: 2048-bit keys = 617 digits → practically unbreakable.

# Lets try it in action

# Real-World Cryptosystems are hybrid

- Real-world systems (SSL/TLS, PGP) use both:

- Asymmetric → for key exchange.
- Symmetric → for data encryption.

- Example: HTTPS handshake uses RSA to exchange AES session keys.

# The Web Needs Trust - SSL/TLS

*“The internet isn’t just about secrecy — it’s about knowing who you’re talking to.”*

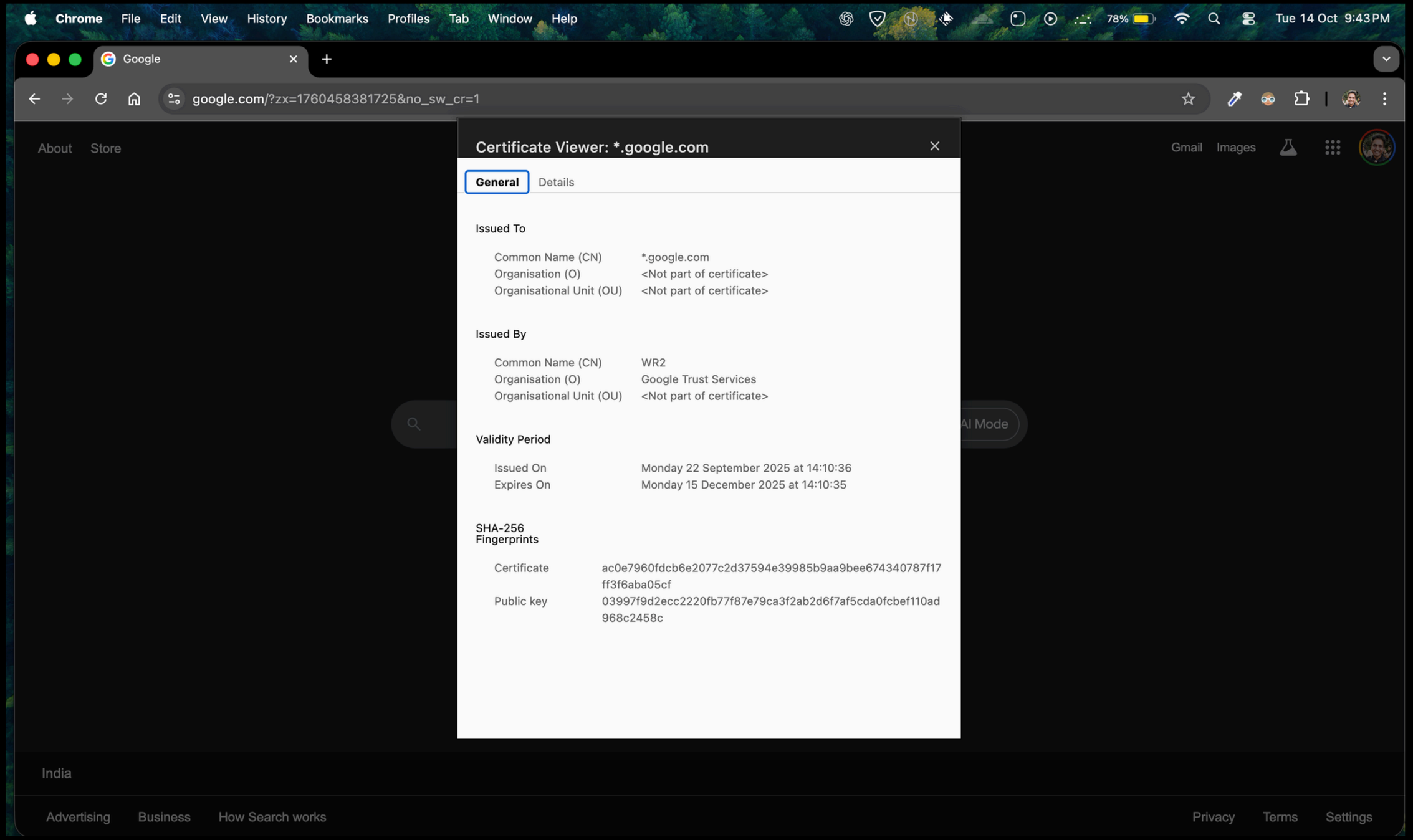
- Protocol securing communication between browser & server.

  Ensures:

- 🔒 Encryption
- 🧾 Authentication
- ✅ Integrity

- Used in HTTPS, email, and secure APIs.


https://


Chrome File Edit View History Bookmarks Profiles Tab Window Help
78%
Tue 14 Oct 9:43 PM
Google
google.com/?zx=1760458381725&no_sw_cr=1
About Store
Gmail Images
Certificate Viewer: *.google.com
General
Details
Issued To
Common Name (CN) *.google.com
Organisation (O) <Not part of certificate>
Organisational Unit (OU) <Not part of certificate>
Issued By
Common Name (CN) WR2
Organisation (O) Google Trust Services
Organisational Unit (OU) <Not part of certificate>
Validity Period
Issued On Monday 22 September 2025 at 14:10:36
Expires On Monday 15 December 2025 at 14:10:35
SHA-256 Fingerprints
Certificate ac0e7960fdcb6e2077c2d37594e39985b9aa9bee674340787f17ff3f6aba05cf
Public key 03997f9d2ecc2220fb77f87e79ca3f2ab2d6f7af5cda0fcbef110ad968c2458c
AI Mode
India
Advertising Business How Search works
Privacy Terms Settings

# Hands-On Task

Try this:

- Visit https://badssl.com/
- Explore invalid/expired certificates.
- Check valid ones using browser “lock” → “Certificate.”
- Optional: View in Wireshark how TLS hides content.

# Takeaways

- Asymmetric cryptography lets two people share secrets — without ever sharing their keys.
- Public keys encrypt, private keys decrypt — together they build trust.
- The power comes from math that’s easy one way, impossible the other.
- This idea forms the base for SSL, digital IDs, secure email, and blockchain.
- “You don’t need to hide your lock — just protect your key.”