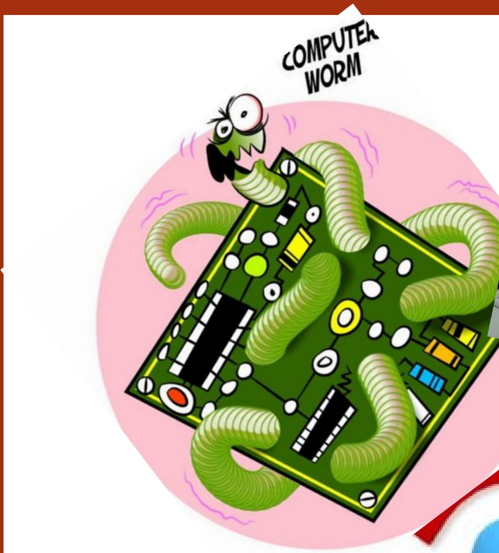
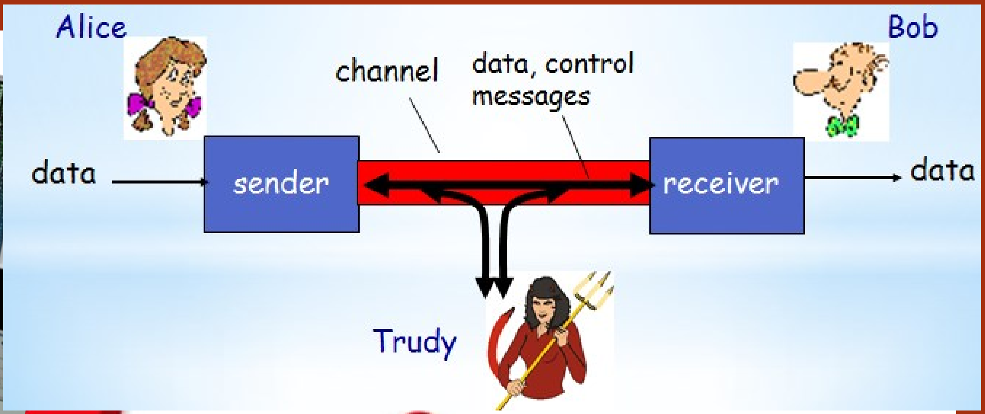
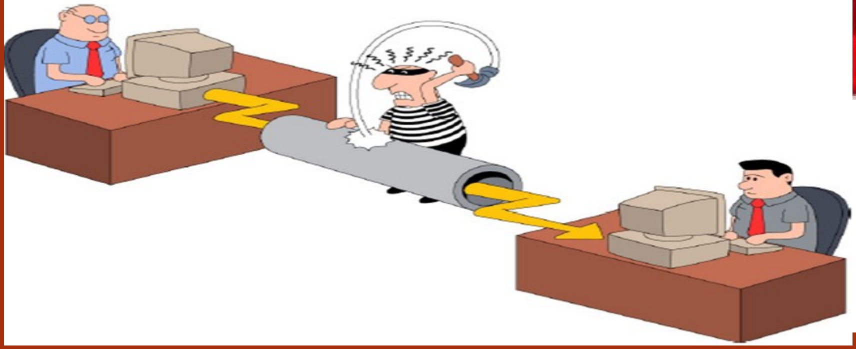
# Cryptography Primitives for Blockchain

Presented by,
Dr. M. Brindha
Assistant Professor
Department of CSE
NIT, Trichy-15

# Cyber World


COMPUTER WORM


Alice
Bob
channel
data, control messages
data
sender
receiver
data
Trudy



i-Virus Automatic Alert
VIRUS DETECTED
virus-de
Repair
Delete
Canc

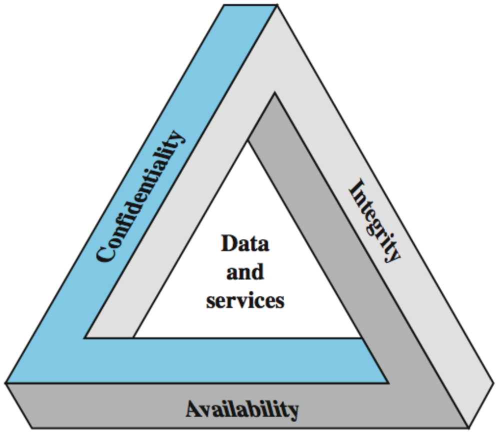
# Contents

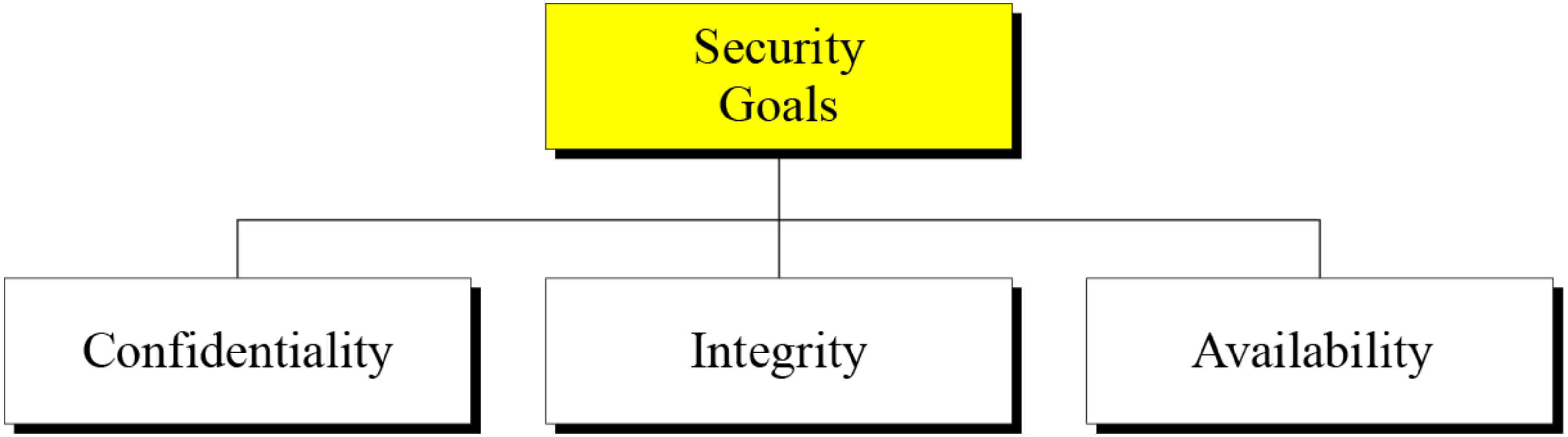
| | |
|---|---|
| Part 1 | Cryptography and its types |
| Part 2 | Public key Cryptography |
| Part 3 | Digital Signature and Hashing |
| Part 4 | How Cryptography primitives are applied in Blockchain? |

# Introduction

- We are living in the information age.
- We need to keep information about every aspect of our lives. In other words, information is an asset that has a value like any other asset.
- As an asset, information needs to be secured from attacks.
- To be secured, information needs to be hidden from unauthorized access (confidentiality), protected from unauthorized change (integrity), and available to an authorized entity when it is needed (availability).

# Taxonomy of security goals





Security Goals

Confidentiality

Integrity

Availability

# Confidentiality

Confidentiality is probably the most common aspect of information security. We need to protect our confidential information. An organization needs to guard against those malicious actions that endanger the confidentiality of its information.

# Integrity

Information needs to be changed constantly. Integrity means that changes need to be done only by authorized entities and through authorized mechanisms.

# Availability

The information created and stored by an organization needs to be available to authorized entities. Information needs to be constantly changed, which means it must be accessible to authorized entities.

# Examples of Security Requirements

- Confidentiality – student grades
- Integrity – patient information
- Availability – authentication service
- Authenticity – admission ticket
- Non-repudiation – stock sell order

# TECHNIQUES

- The actual implementation of security goals needs some techniques

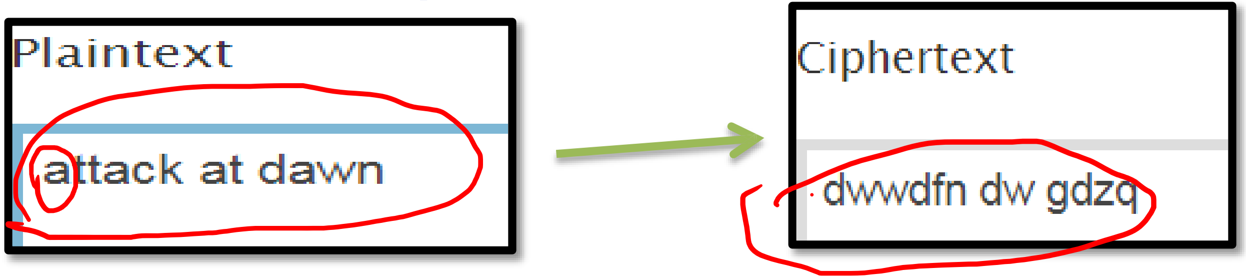
Cryptography

# Cryptography

- Cryptography, a word with Greek origins, means “secret writing”
- However, we use the term to refer to the science and art of transforming messages to make them secure and immune to attacks.

# Historical Example

- The Caesar cipher is one of the earliest known and simplest ciphers.
- It is a type of substitution cipher in which each letter in the plaintext is 'shifted' a certain number of places down the alphabet.
- For example, with a shift of 1, A would be replaced by B, B would become C, and so on.
- The method is named after Julius Caesar, who apparently used it to communicate with his generals.
- Caeser acctually used a shift of 3

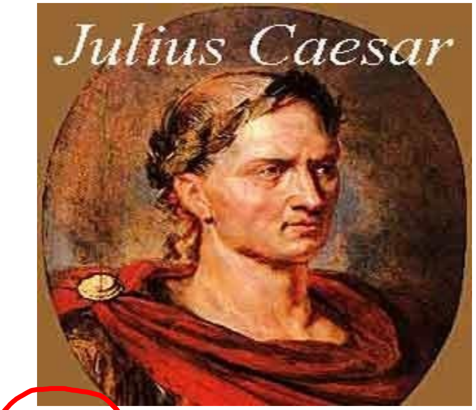


Plaintext

attack at dawn

Ciphertext

dwwdfn dw gdzq



| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

# Cryptography: Basic Terminology

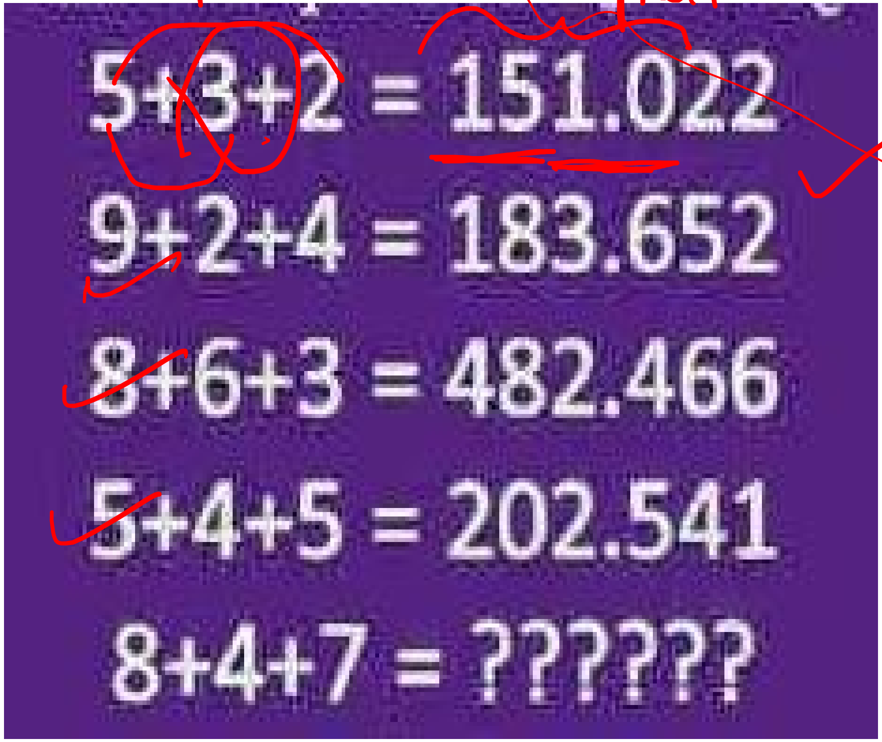
- Plaintext (or cleartext)
  - The message.
- Encryption (encipher)
  - Encoding of message.
- Ciphertext
  - Encrypted message.
- Decryption (decipher)
  - decoding of ciphertext

# Cryptography

known plaintext

Plaintext

ciphatext



5+3+2 = 151.022

9+2+4 = 183.652

8+6+3 = 482.466

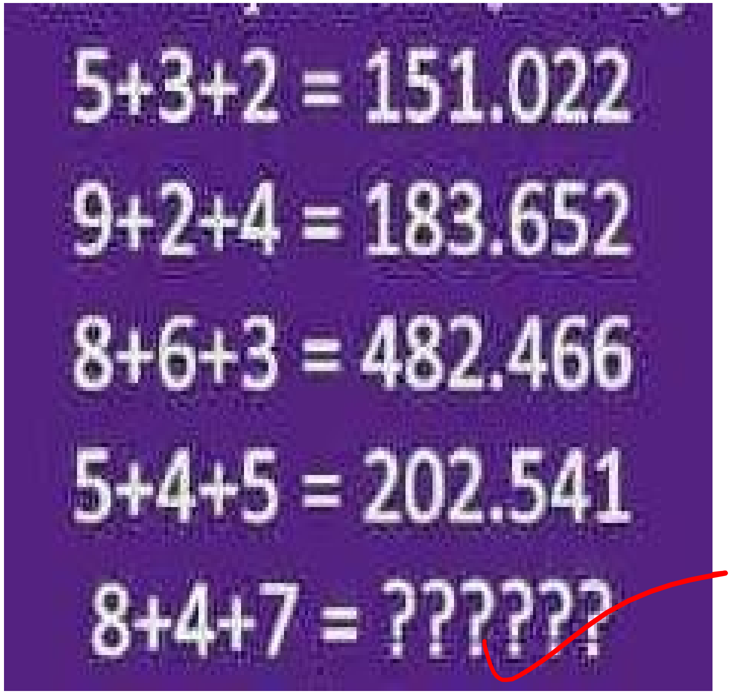
5+4+5 = 202.541

8+4+7 = ??????

15
10
25
3
22

**Decode this …**

# Cryptography



5+3+2 = 151.022

9+2+4 = 183.652

8+6+3 = 482.466

5+4+5 = 202.541

8+4+7 = ??????

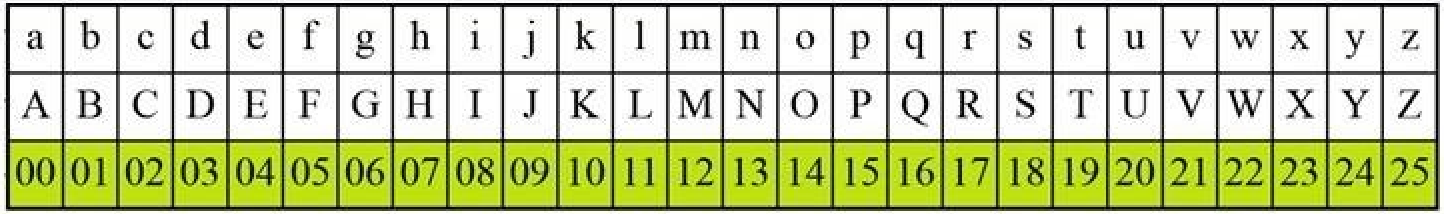
5 3 2 -->151022

9 2 4 --> 183652

8 6 3 --> 482466

5 4 5 --> 202541

8 4 7 --> 325684



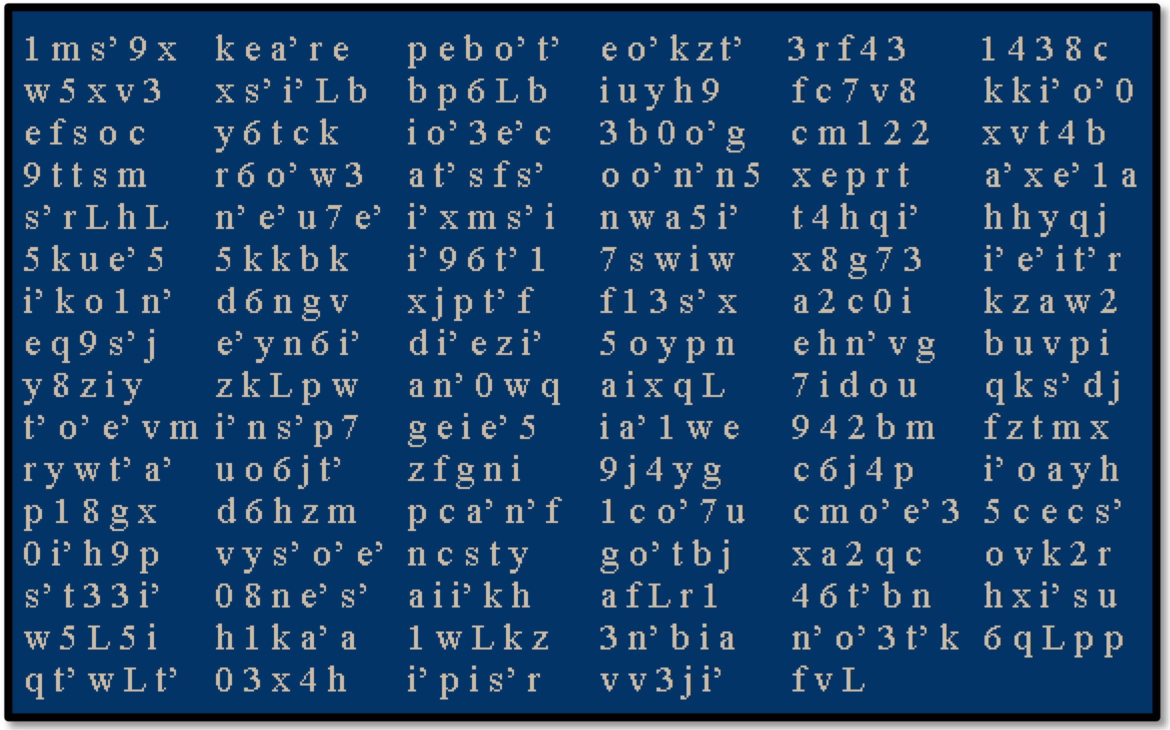
| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

If I say my computer password is 345

Decode this …

121523
bcbfcd

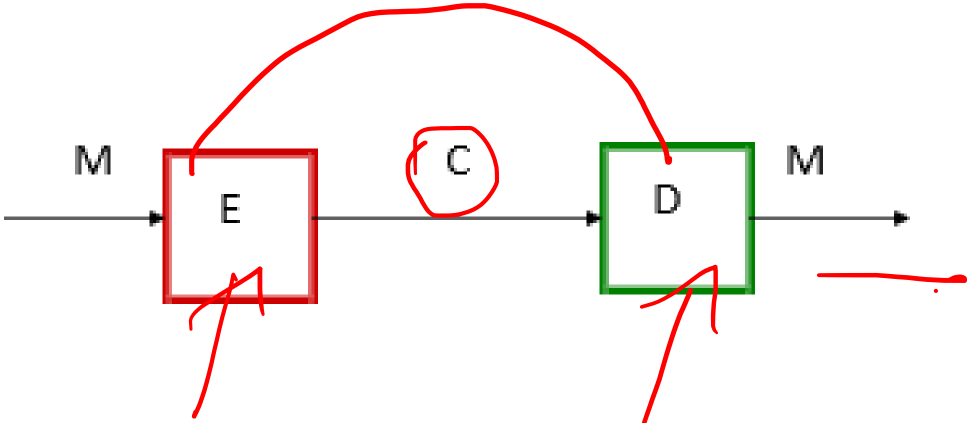
# Cryptography



```
1ms’9x   kea’re   pebo’t’  eo’kzt’  3rf43    1438c
w5xv3    xs’i’Lb  bp6Lb    iuyh9    fc7v8    kki’o’0
efsoc    y6tck    io’3e’c  3b0o’g   cm122    xvt4b
9ttsm    r6o’w3   at’sfs’  oo’n’n5  xeprt    a’xe’1a
s’rLhL   n’e’u7e’ i’xms’i  nwa5i’   t4hqi’   hhyqj
5kue’5   5kkbk    i’96t’1  7swiw    x8g73    i’e’it’r
i’ko1n’  d6ngv    xjpt’f   f13s’x   a2c0i    kzaw2
eq9s’j   e’yn6i’  di’ezi’  5oypn    ehn’vg   buvpi
y8ziy    zkLpw    an’0wq   aixqL    7idou    qks’dj
t’o’e’vm i’ns’p7  geie’5   ia’1we   942bm    fztmx
rywt’a’  uo6jt’   zfgni    9j4yg    c6j4p    i’oayh
p18gx    d6hzm    pca’n’f  1co’7u   cmo’e’3  5cecs’
0i’h9p   vys’o’e’ ncsty    go’tbj   xa2qc    ovk2r
s’t33i’  08ne’s’  aii’kh   afLr1    46t’bn   hxi’su
w5L5i    h1ka’a   1wLkz    3n’bia   n’o’3t’k 6qLpp
qt’wLt’  03x4h    i’pis’r  vv3ji’   fvL
```

**Decode this …**

# Encryption and Decryption



**The following identity must hold true:**

**D(C) = M, where C = E(M)**

**M = D(E(M))**

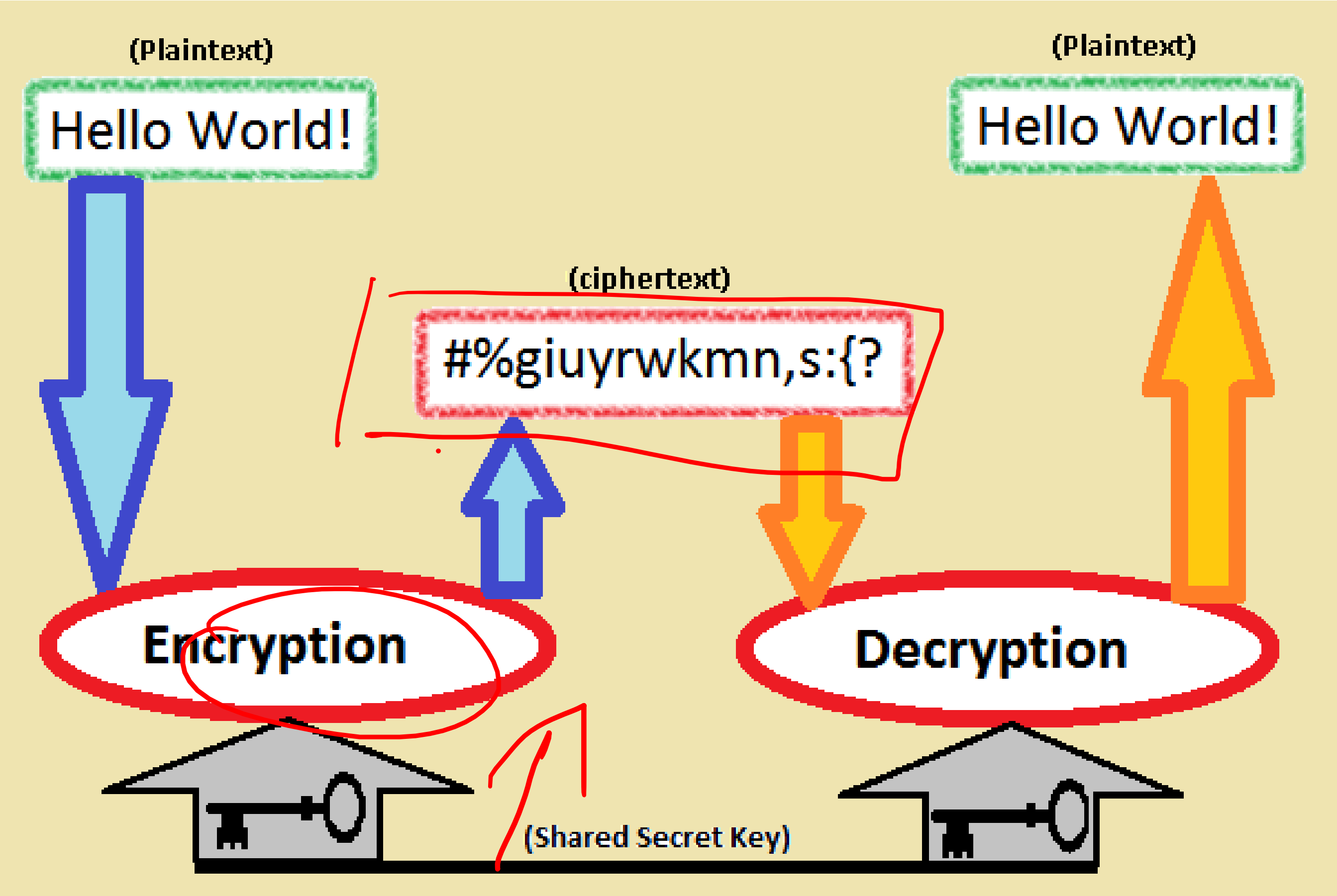
# Cryptography: Algorithms and Keys

- A method of encryption and decryption is called a cipher.
- Generally there are two related functions: one for encryption and other for decryption.
- Some cryptographic methods rely on the secrecy of the algorithms.
  - Such methods are mostly of historical interest these days.
- All modern algorithms use a key to control encryption and decryption.
- Encryption key may be different from decryption key.

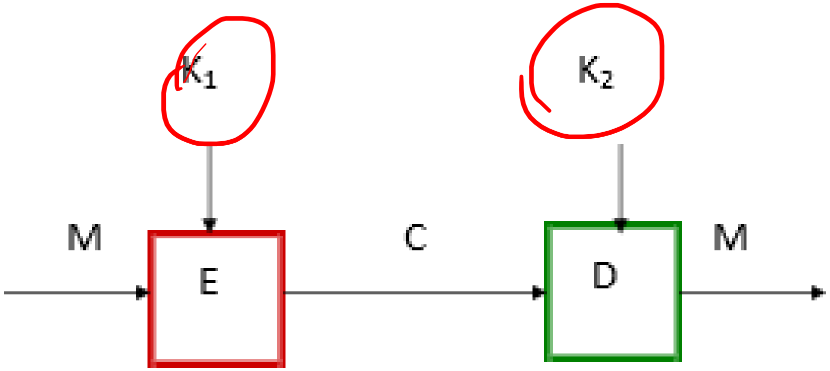
# Cryptography

- An encryption scheme has five ingredients:
  - Plain text
  - Encryption algorithm
  - Key
  - Cipher text
  - Decryption algorithm
- **In the modern world, Security depends on the secrecy of the key, not the secrecy of the algorithm**

# What is Cryptography?



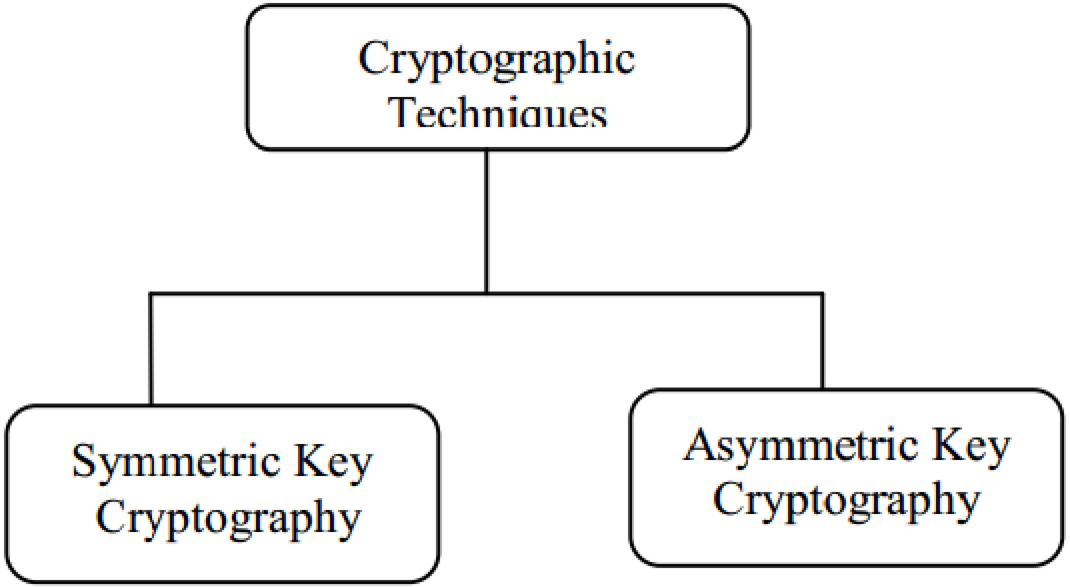
# Key Based Encryption/Decryption



**Symmetric Case:** both keys are the same or derivable from each other. $K_1 = K_2$

**Asymmetric Case:** keys are different and not derivable from each other. $K_1 \mathrel{!}= K_2$

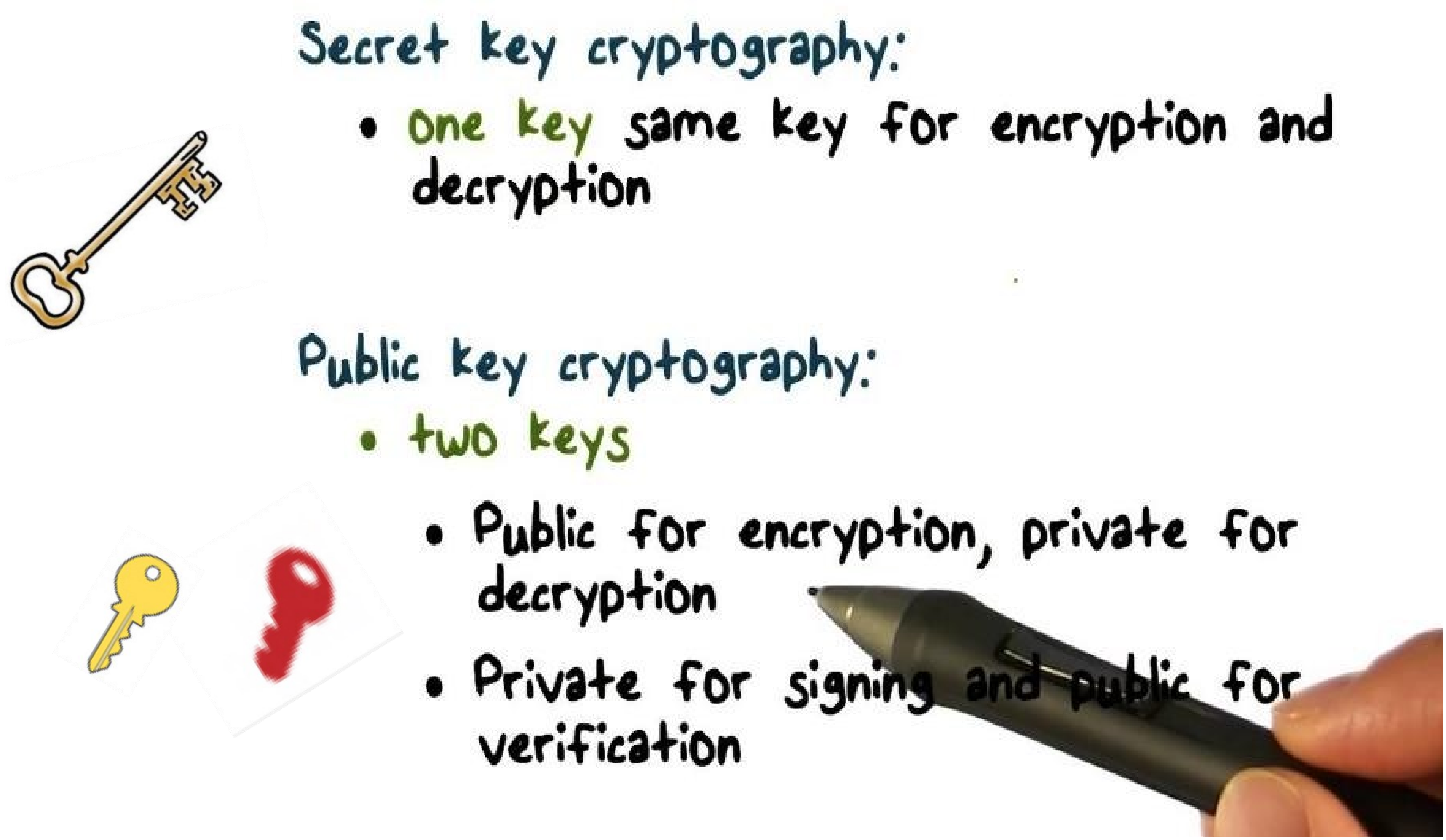
# Types of  Cryptography?


Cryptographic Techniques

Symmetric Key Cryptography

Asymmetric Key Cryptography

# Types of Cryptography?



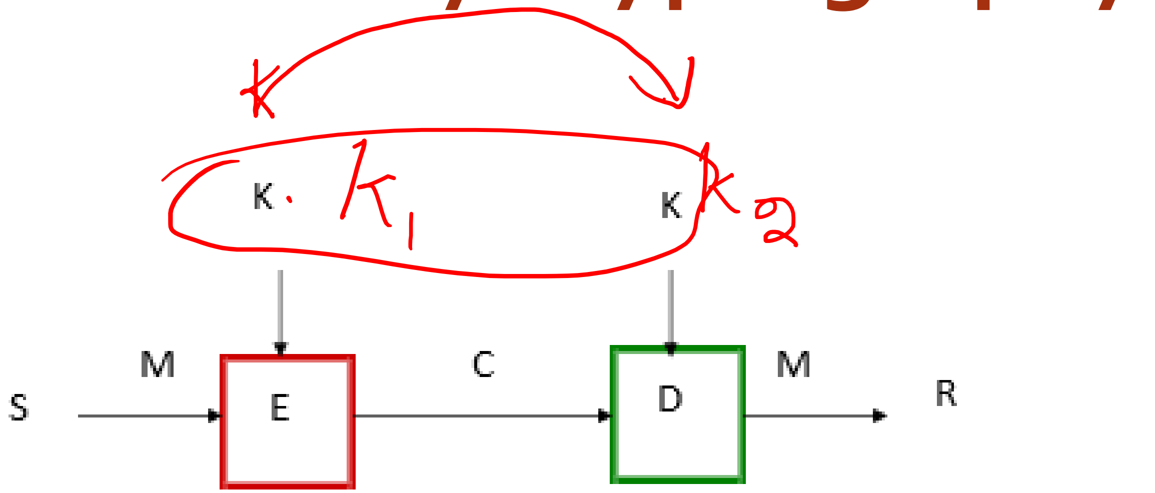
Secret key cryptography:

- one key same key for encryption and decryption

Public key cryptography:

- two keys
  - Public for encryption, private for decryption
  - Private for signing and public for verification

# Secret Key Cryptography

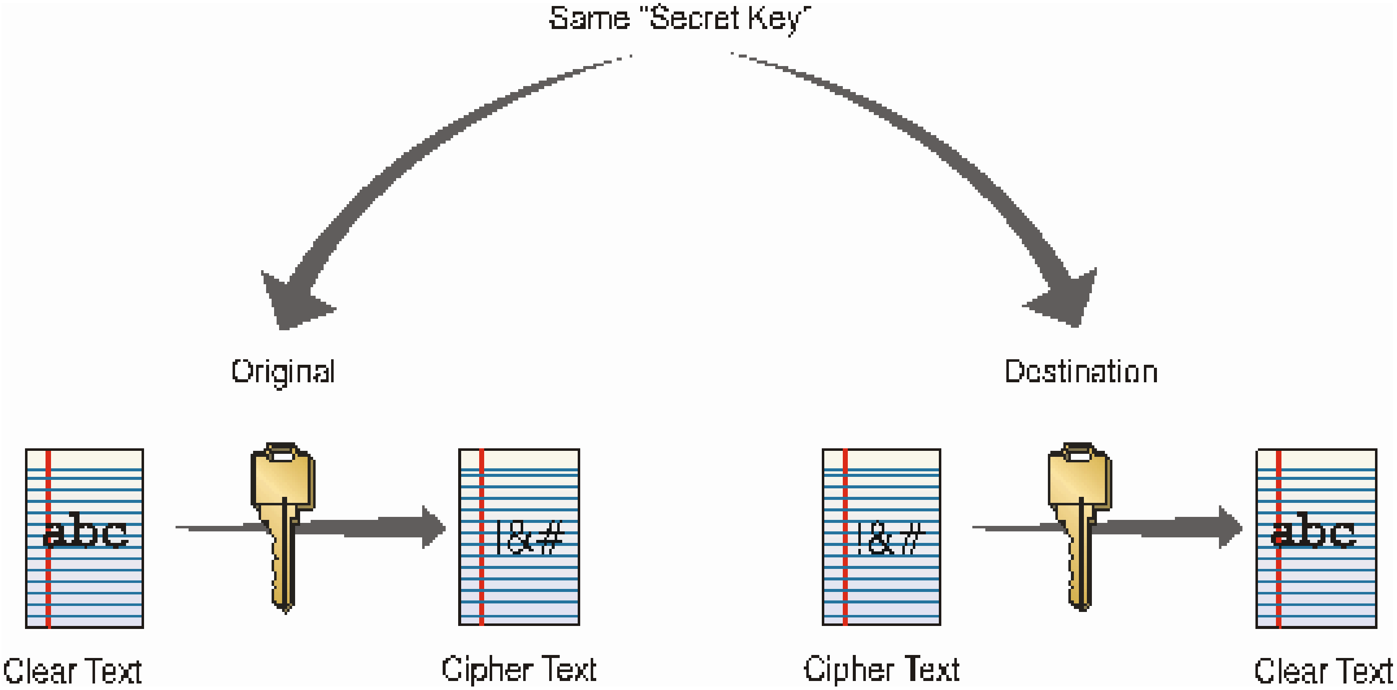


K is the secret key shared by both the sender (S) and receiver (R)

# Secret Key Cryptography

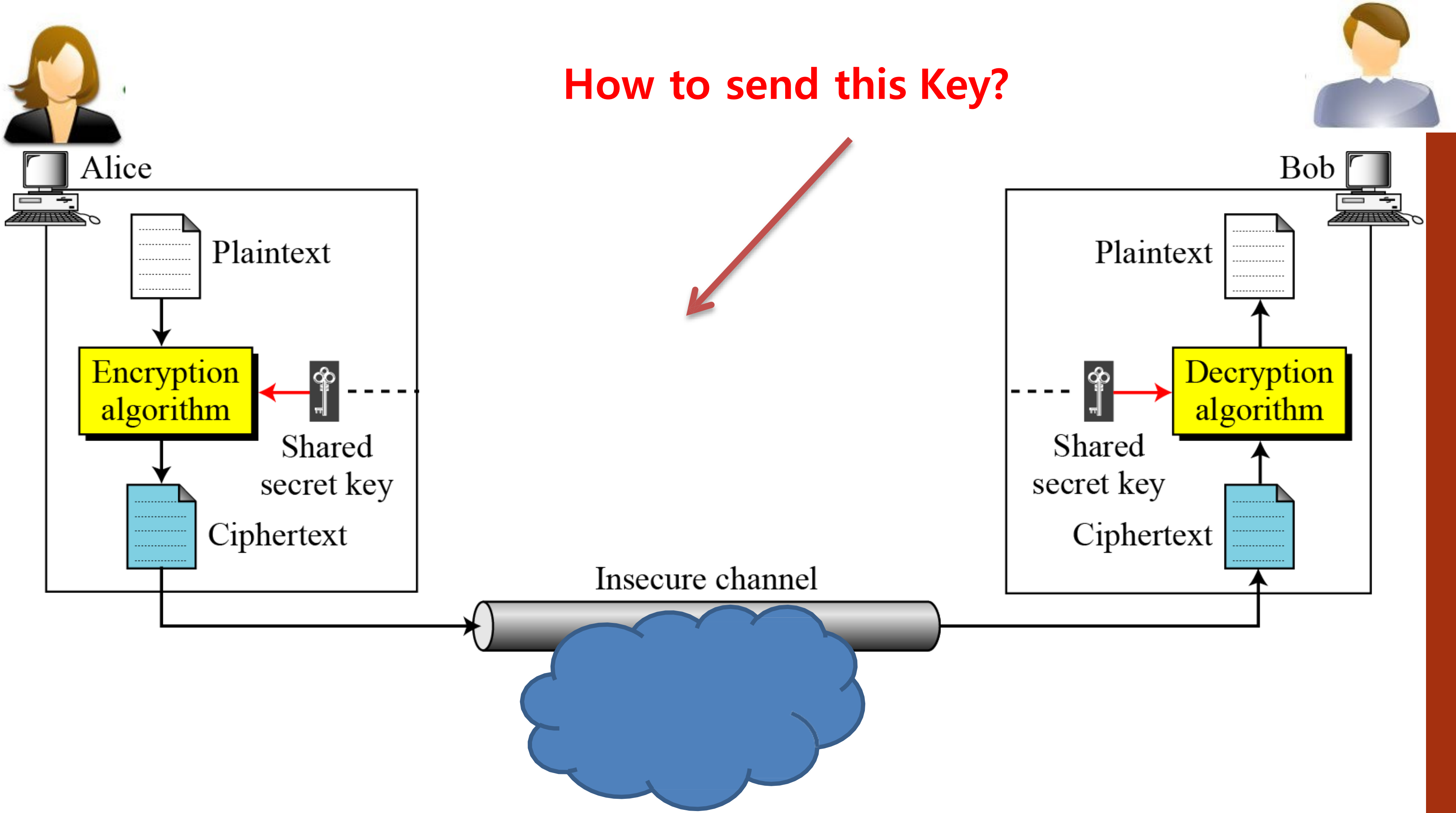
- Also called symmetric or single-key algorithms.
- The encryption and the decryption key are the same.
- Techniques based on a combination of substitution and permutation.
- Stream ciphers: operate on single bit or byte.
- Block ciphers: operate on blocks (typically 64/128/256… bits)
- Advantage: simple, fast.
- Disadvantage: *key exchange, key management*.
- Examples: DES,RC4, IDEA, Blowfish, AES, etc.

# Private Key Cryptosystem (Symmetric)



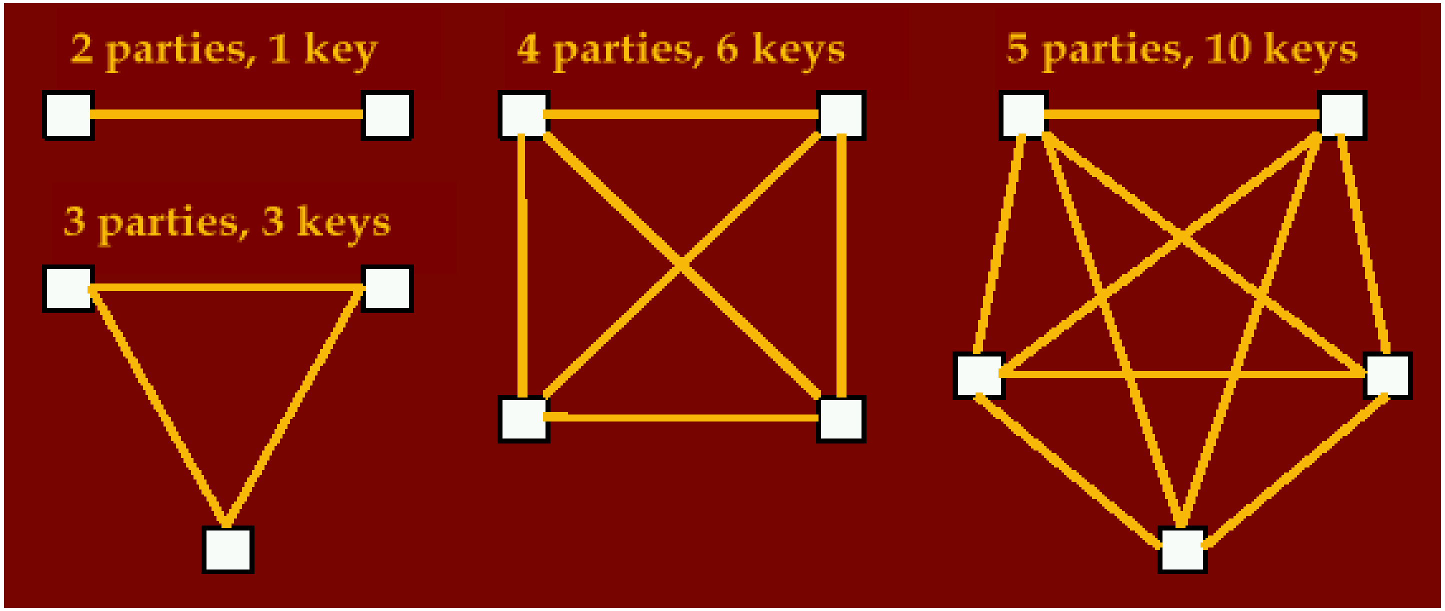
# Symmetric Key Cryptography

How to send this Key?



# Symmetric Key - Issues

**Key management, keys required = (p*(p-1))/2 or:**



# Secret Key Assurances

secret

key. (iljmn)

- Confidentiality
  - is assurance that only owners of a shared secret key can decrypt a message that has been encrypted with the shared secret key
- Authentication
  - is assurance of the identify of the person at the other end of the line (use challenge and response protocols)
- Integrity
  - is assurance that a message has not been changed during transit and is also called message authentication (use message fingerprint)
- Non-repudiation
  - is assurance that the sender cannot deny a file was sent. This cannot be done with secret key alone (need trusted third party or public key technology)

# Example: non-repudiation

- Scenario 1:
  - Alice sends a stock buy request to Bob
  - Bob does not buy and claims that he never received the request
- Scenario 2:
  - Alice sends a stock buy request to Bob
  - Bob sends back an acknowledge message
  - Again, Bob does not buy and claims that he never received it
  - Alice presents the ack message as proof
- Can she prove that the ack message was created by him?

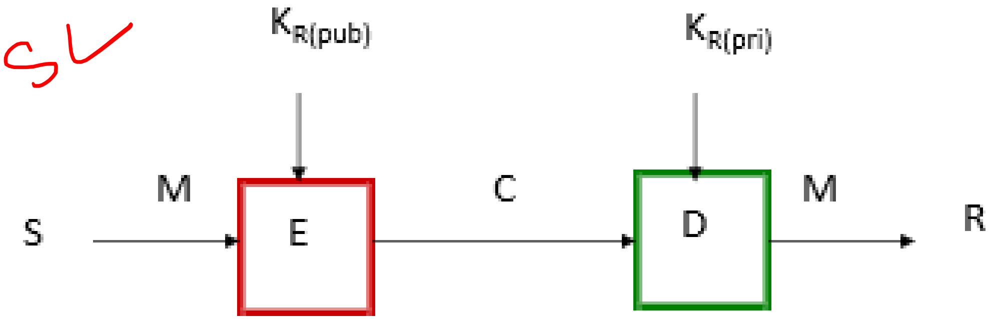
session key public

# Public Key Cryptography

data → symmetric

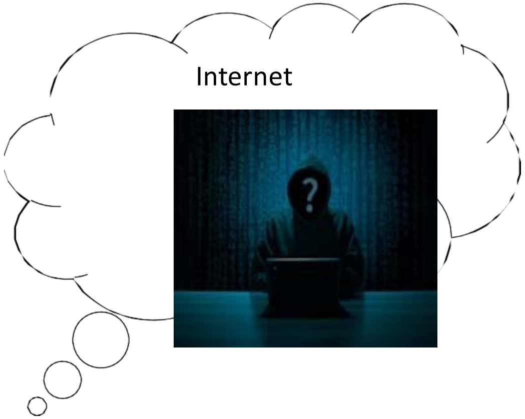
https

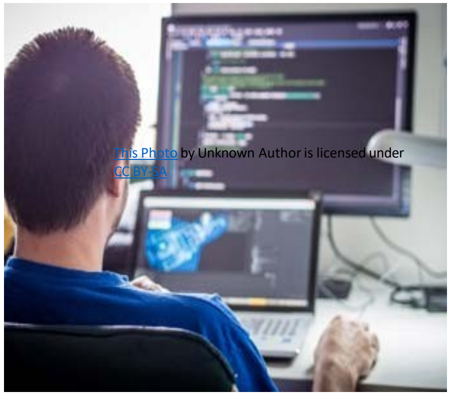
SSL



**$K_{R(pub)}$ is Receiver's public key and $K_{R(pri)}$ is Receiver's private key.**

# Establishing Shared Secret





This Photo by Unknown Author is licensed under CC BY-SA

# Problem Statement

- Suppose Alice has a channel for communicating with Bob.
- Alice and Bob wish to use this channel to establish a shared secret.

attacker

- However, Eve is able to learn everything sent over the channel.
- If Alice and Bob have no other channel to use, can they establish a shared secret that Eve does not know?

# Public Key Cryptographic Algorithms

***Find a hard math problem, that is easy to compute in the forward direction, but is difficult to solve in the reverse direction, unless you have some special knowledge.***

$n =$ ✓ $p \times q$ ✓.

$y = x^e \bmod n$.

$x$.

$e'$

$y, e, n$.

Bob → Alice

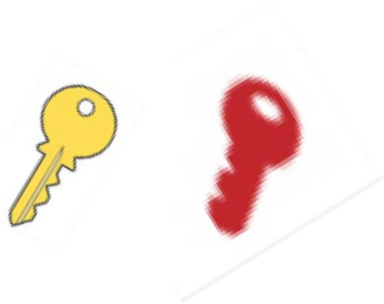
# General Strategy

- A public key is used to encrypt a message that can be decrypted only by the matching private key.
- Bob can use Alice's public key to encrypt messages. Only Alice can decrypt the message.
- Similarly, Alice can also use Bob's public key.
- Alice and Bob exchange information, each keeping a secret to themselves.
- The secrets that they keep allow them to compute a shared secret.
- Since Eve lacks either of these secrets she is unable to compute the shared secret.

# Asymmetric Key Cryptography

Every user has two keys.

Public Key

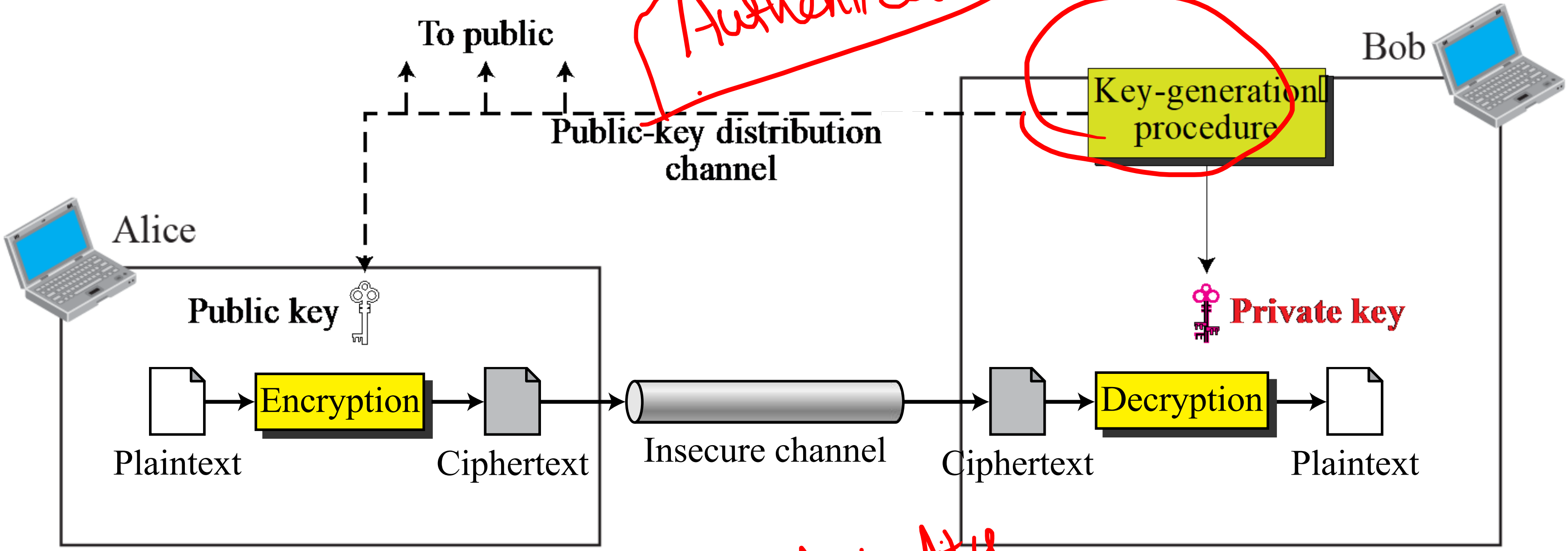


Private Key

User announces this key to entire public

Only user knows about it

# General idea of asymmetric-key cryptosystem



Authentication.

Confidentiality.

Sender

private Spr

public Spu

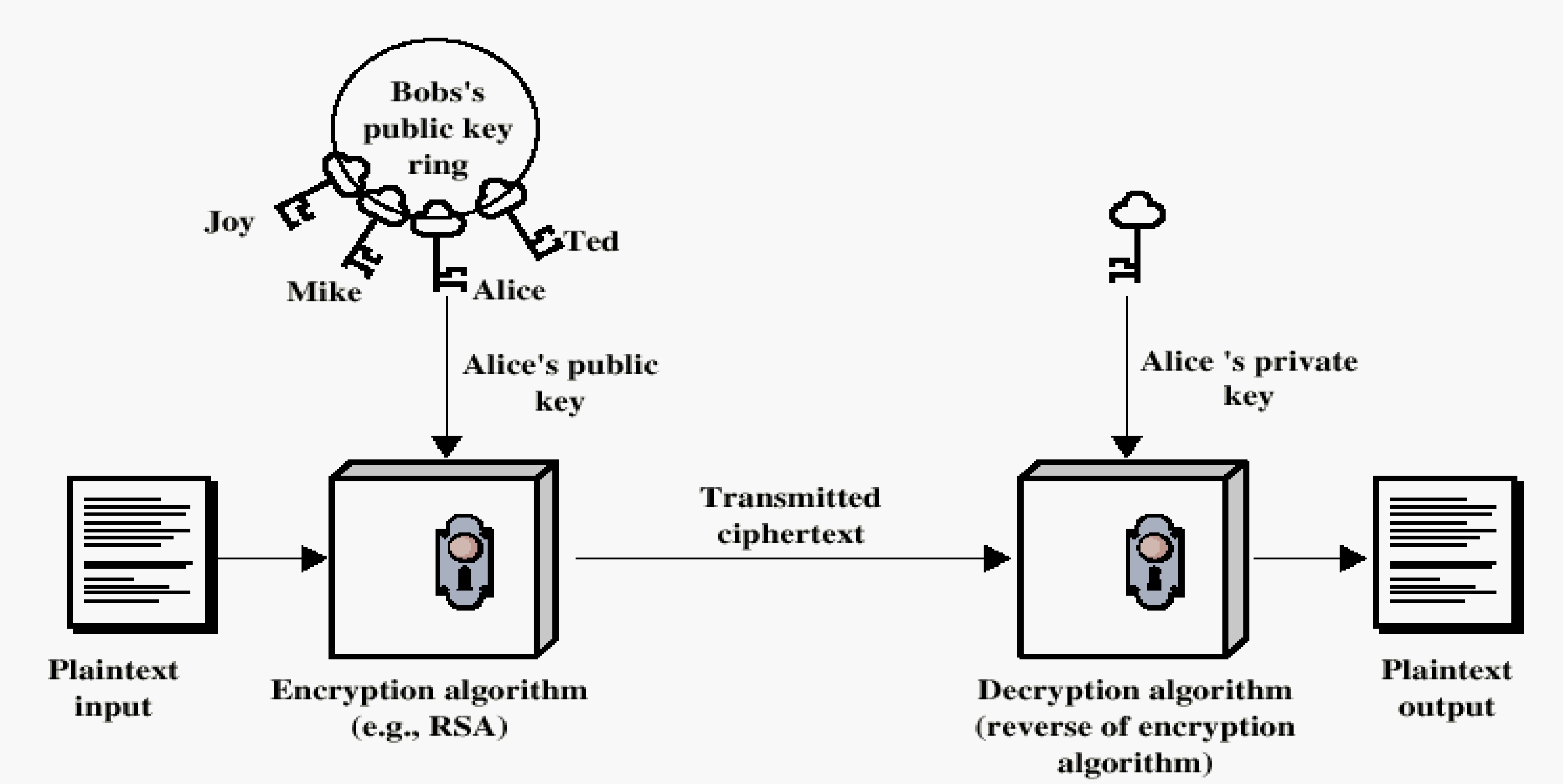
Receiver.

private Rpr

public Rpu.

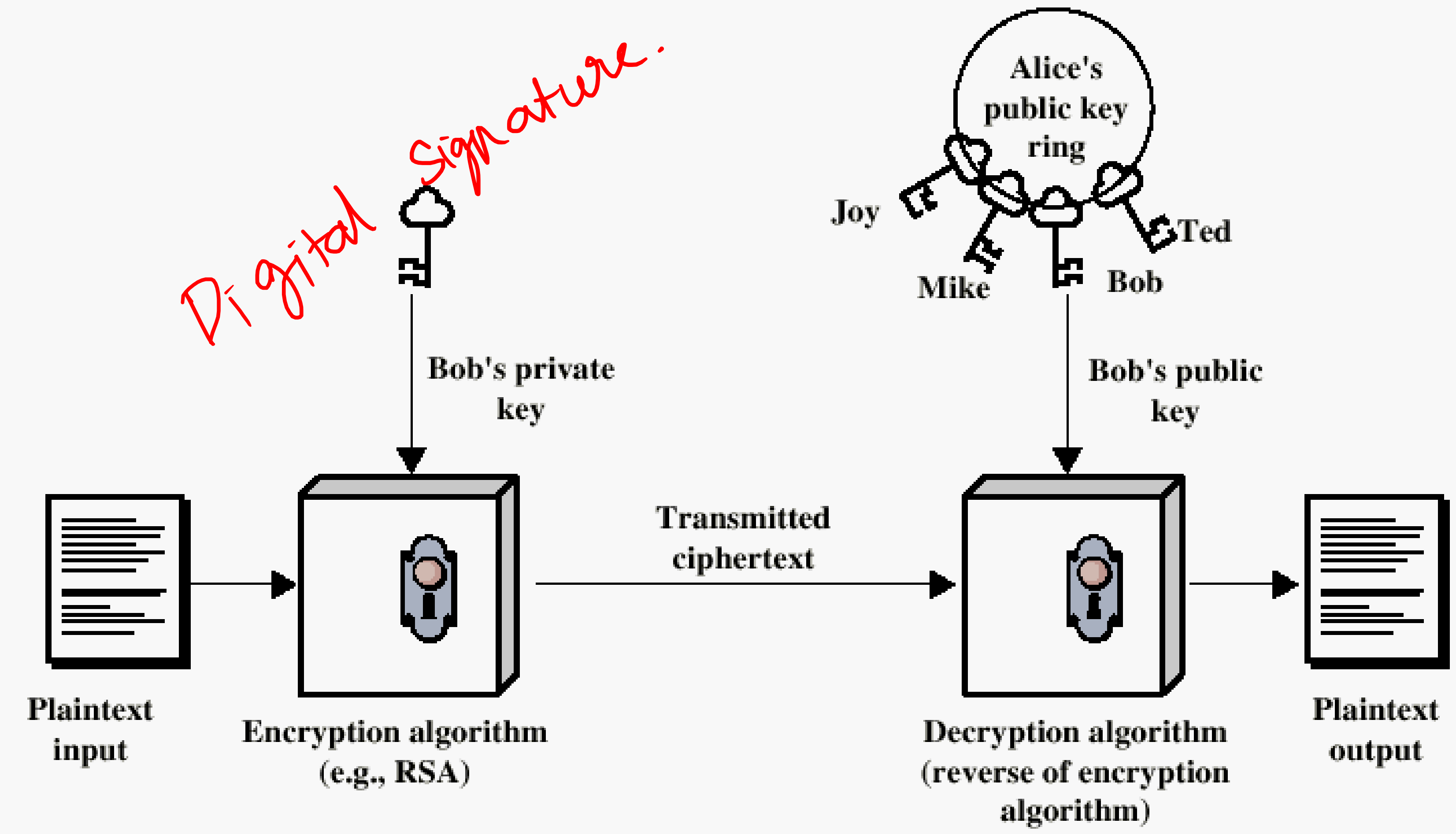
Rpu

# Asymmetric Key Cryptography



Result is confidentiality .

# Asymmetric Key Cryptography



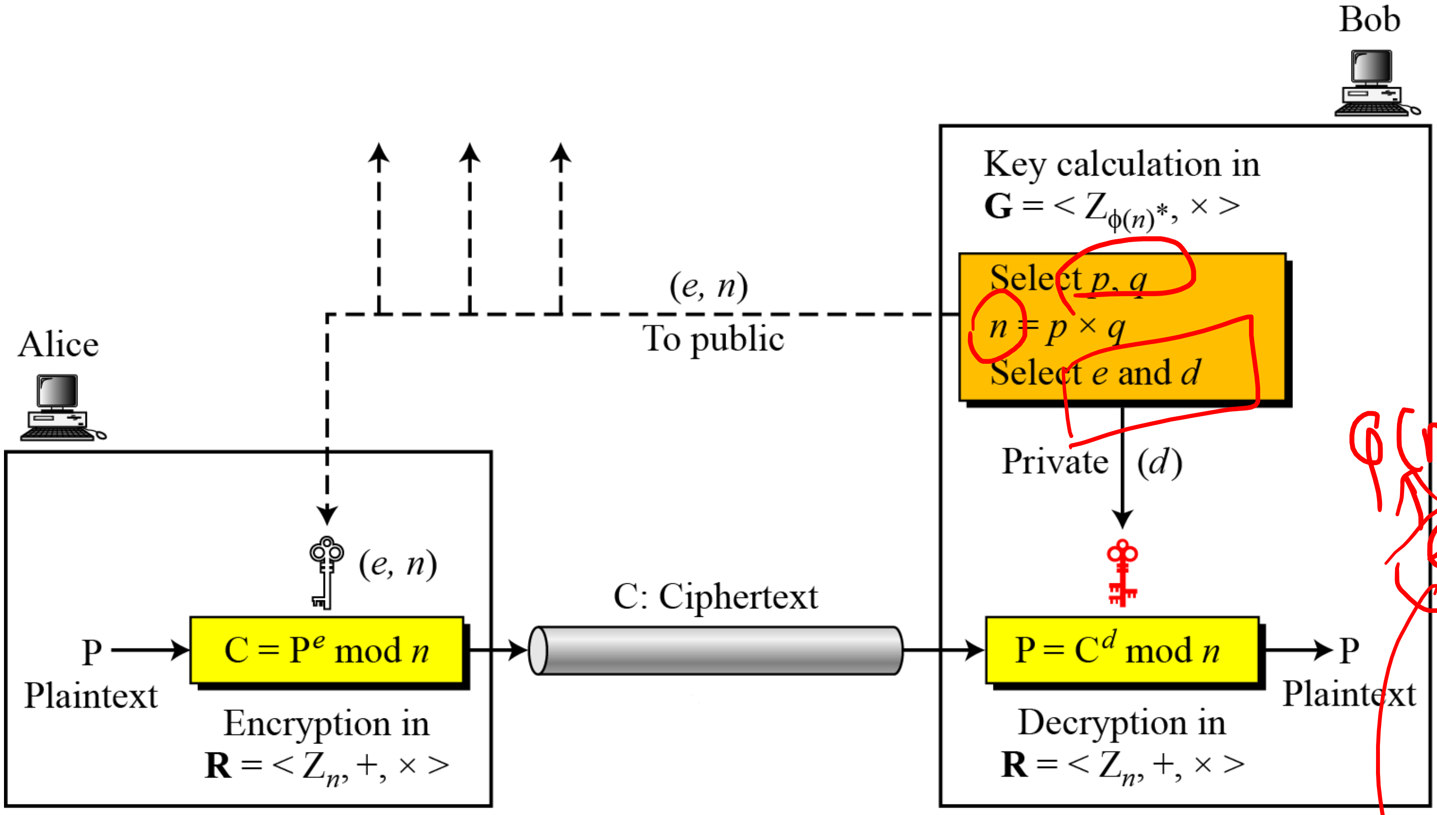
Result is Authentication.

# Asymmetric Algorithms

- Also called public-key algorithms.
- Encryption key is different from decryption key.
- Furthermore, one cannot be calculated from other.
- Encryption key is often called the public key and decryption key is often called the private key.
- Advantages: better key management.
- Disadvantages: slower, more complex.
- Both techniques are complementary.
- Examples: RSA, Diffie-Hellman, El Gamal, etc.

Dillie Hellman

# Encryption, decryption, and key generation in RSA


Bob
Key calculation in
$\mathbf{G} = < Z_{\phi(n)}^*, \times >$
Select $p, q$
$n = p \times q$
Select $e$ and $d$
Private $(d)$
$(e, n)$
To public
Alice
$(e, n)$
P
Plaintext
$\mathrm{C} = \mathrm{P}^e \bmod n$
Encryption in
$\mathbf{R} = < Z_n, +, \times >$
C: Ciphertext
$\mathrm{P} = \mathrm{C}^d \bmod n$
P
Plaintext
Decryption in
$\mathbf{R} = < Z_n, +, \times >$

$\phi(n) = p-1 \times q-1$

$e$

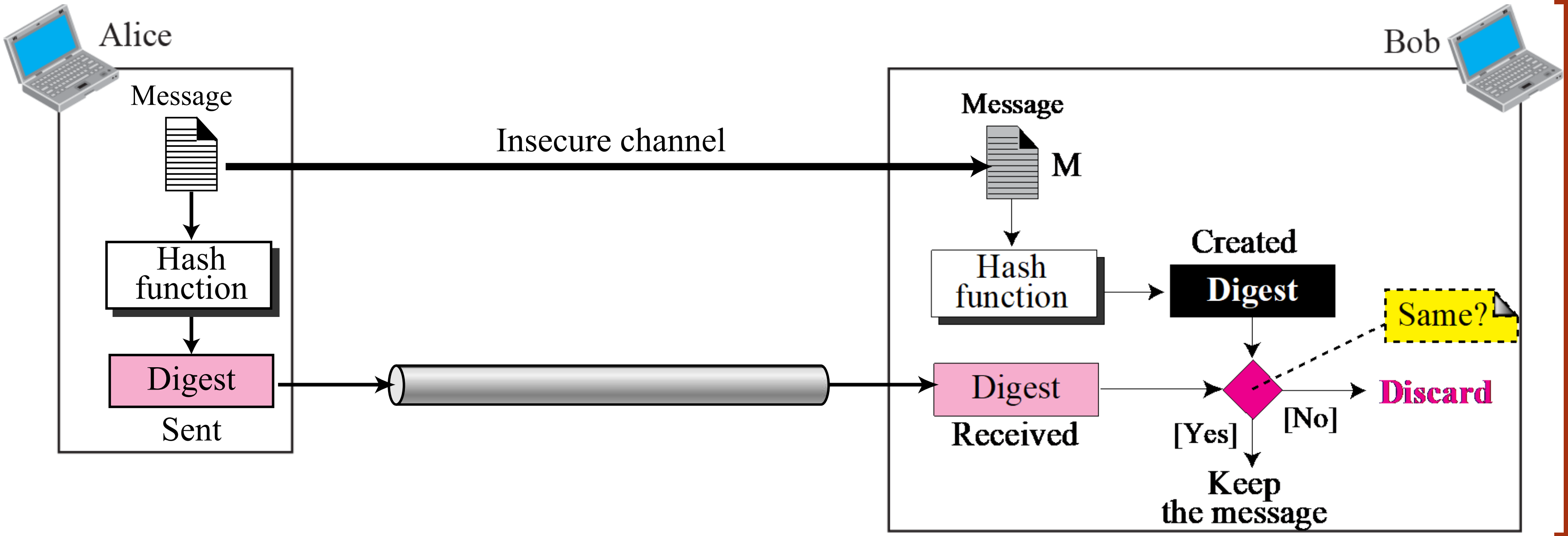
$d = e^{-1} \bmod \phi(n)$

$(e \times d) \bmod \phi(n) = 1$

# Message Integrity

- The cryptography systems that we have studied so far provide secrecy, or confidentiality, but not integrity.
- However, there are occasions where we may not even need secrecy but instead must have integrity.
- For example, Alice may write a will to distribute her estate upon her death.
- The will does not need to be encrypted.
- After her death, anyone can examine the will.
- The integrity of the will, however, needs to be preserved.
- Alice does not want the contents of the will to be changed.

# Message and digest



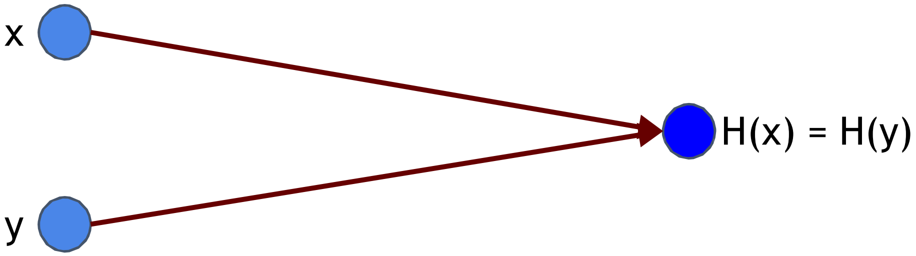
# Cryptographic Hash Functions

- **Hash function:**
  - takes an arbitrary length string as input
  - produces a fixed-size output (e.g 256 bits)
    - Easy to compute
    - Almost impossible to reverse

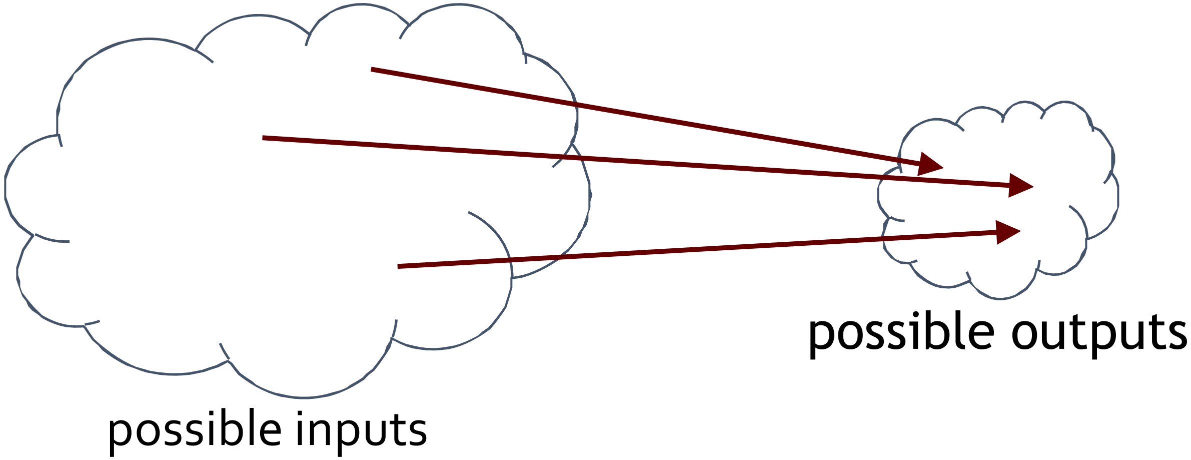
- **Security properties:**
  - collision-resistant
  - Hides the original String
    - Almost impossible to get the original string from the output
  - puzzle-friendly

# Hash property 1: Collision-resistance

It is computationally NOT feasible to find x and y such that
x != y and H(x)=H(y)



# However, for a weak hash function collisions may be feasible to find



**Examples of hash functions for which collisions are found feasibly: MD-5, SHA-1**

# Brute-forcing collision

- Try $2^{130}$ randomly chosen inputs
  99.8% chance that two of them will collide

- This works no matter what H is …
… but it takes too long to matter

- Even if each input checking takes 100 ms, we are talking about $2^{130} \times 0.1$ seconds which is $1/5 \times 2^{129}$ seconds = $4.3 \times 10^{30}$ years

# Application: Hash as message digest

- If we know H(x) = H(y),
  - it's safe to assume that x = y
- To recognize a file that we saw before,
  - just remember its hash.
- Useful because the hash is small.

# Hash property 2: Hiding

We want something like this:

Given H(x), it is infeasible to find x

# Hash property 2: Hiding

- **Hiding property:**
    - If r is chosen from a probability distribution that has *high min-entropy*, then given $H(r \mid x)$, it is infeasible to find x.

- **High min-entropy means**
    - the distribution is "very spread out", so that no particular value is chosen with more than negligible probability.

# Application: Commitment

Want to “seal a value in an envelope”, and “open the envelope” later.

Commit to a value, reveal it later.

# Hash property 3: Puzzle-friendly

**Puzzle-friendly:**

**For every possible output value y, if k is chosen from a distribution with high min-entropy,**

**then it is infeasible to find x such that H(k | x) = y.**

# Application: Search puzzle

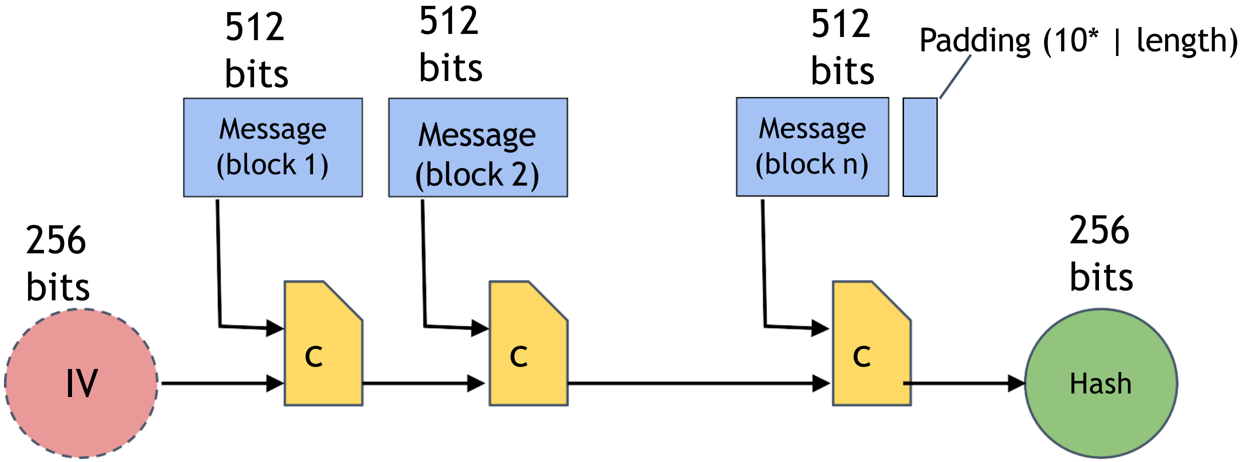
Given a "puzzle ID" *id* (from high min-entropy distrib.), and a target set *Y*:

Try to find a "solution" *x* such that

$$H(id \mid x) \in Y$$

Puzzle-friendly property implies that no solving strategy is much better than trying random values of *x (Brute-force)*

# SHA-256 hash function

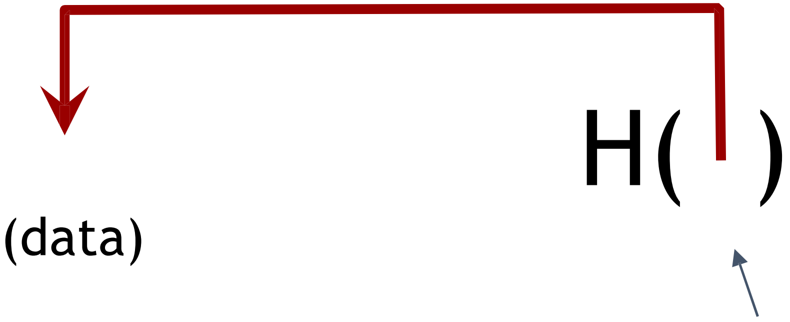


**Theorem: If c is collision-free, then SHA-256 is collision-free**

# Hash Pointers

- hash pointer is:
  - pointer to where some info is stored, and
  - (cryptographic) hash of the info
- if we have a hash pointer, we can
  - ask to get the info back (locate)
  - verify that it hasn't changed (integrity)

# Pictorial Representation of Hash Pointers



H( )

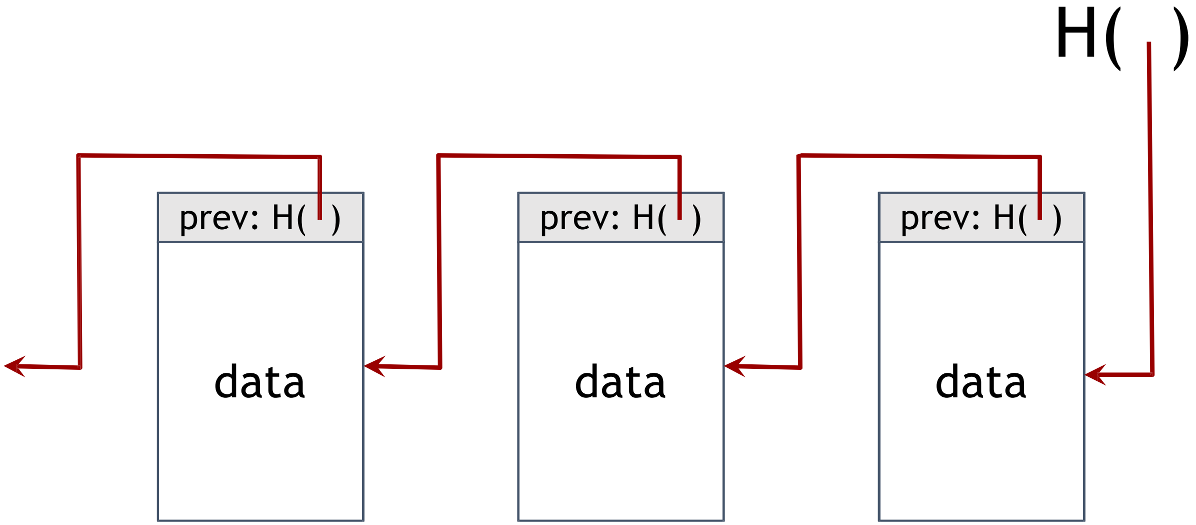
(data)

Represent hash pointers like this

## Key Idea

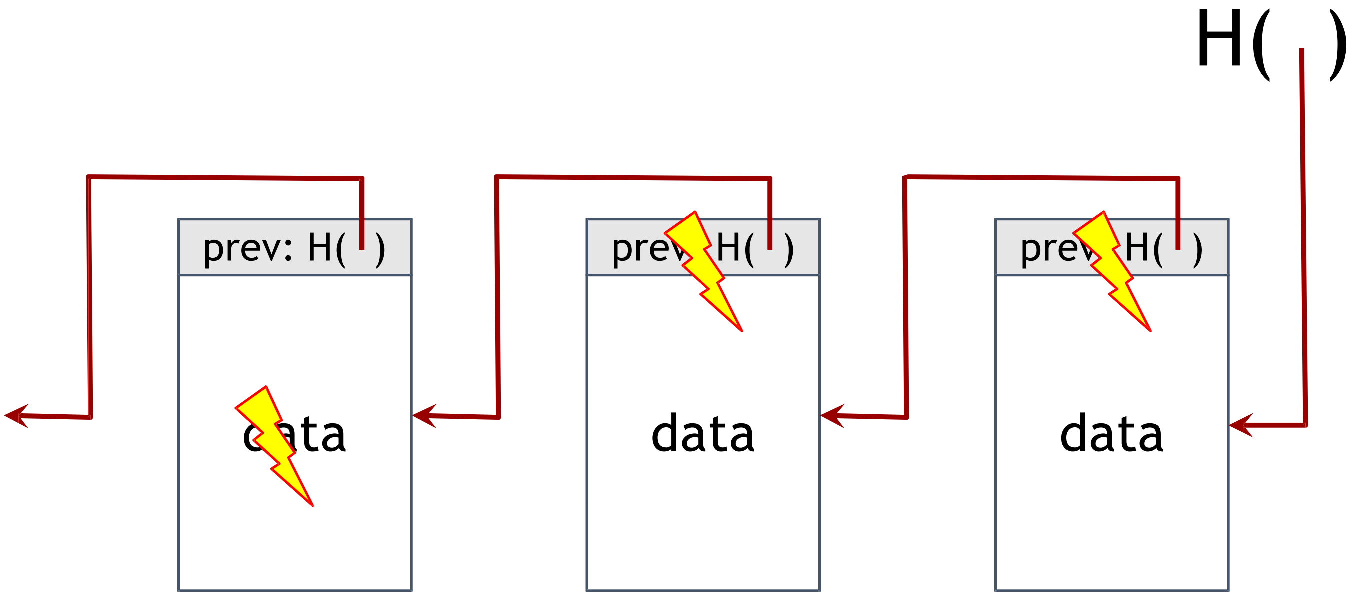
## Build data structures with hash pointers

## linked list with hash pointers = "block chain"



use case: tamper-evident log

# Detecting tampering



use case: tamper-evident log

# Digital Signatures

Only you can sign, but anyone can verify

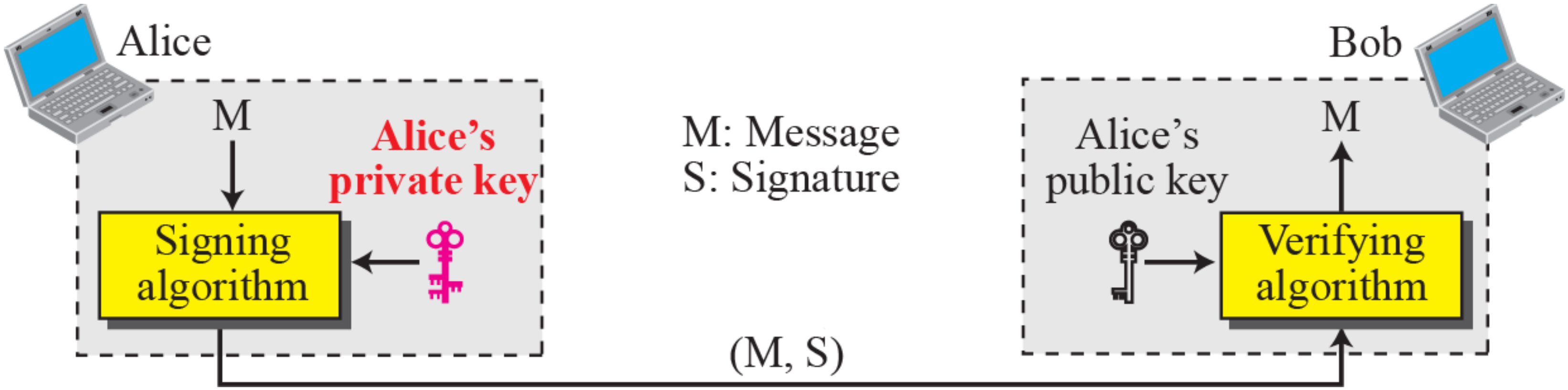
Signature is tied to a particular document
  can’t be cut-and-pasted to another doc

# Digital Signature

Another way to provide message integrity and message authentication is a digital signature.

A digital signature uses a pair of private-public keys.

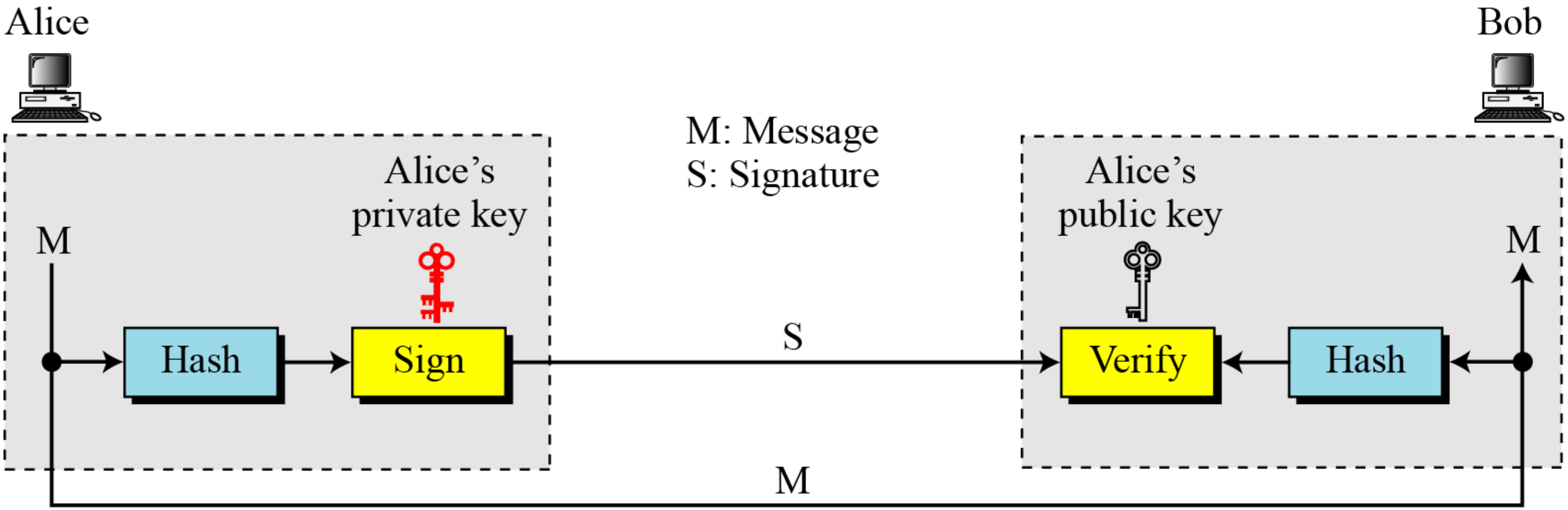
# Digital signature process



A digital signature needs a public-key system.
The signer signs with her private key; the verifier verifies with the signer's public key.

A cryptosystem uses the private and public keys of the receiver:
a digital signature uses the private and public keys of the sender.

# Signing the digest



# Digital Signature- Services

A digital signature can directly provide message authentication, message integrity, and nonrepudiation;

# Message Authentication

**A secure digital signature scheme, like a secure conventional signature can provide message authentication.**

*Note*

**A digital signature provides message authentication.**

# Message Integrity

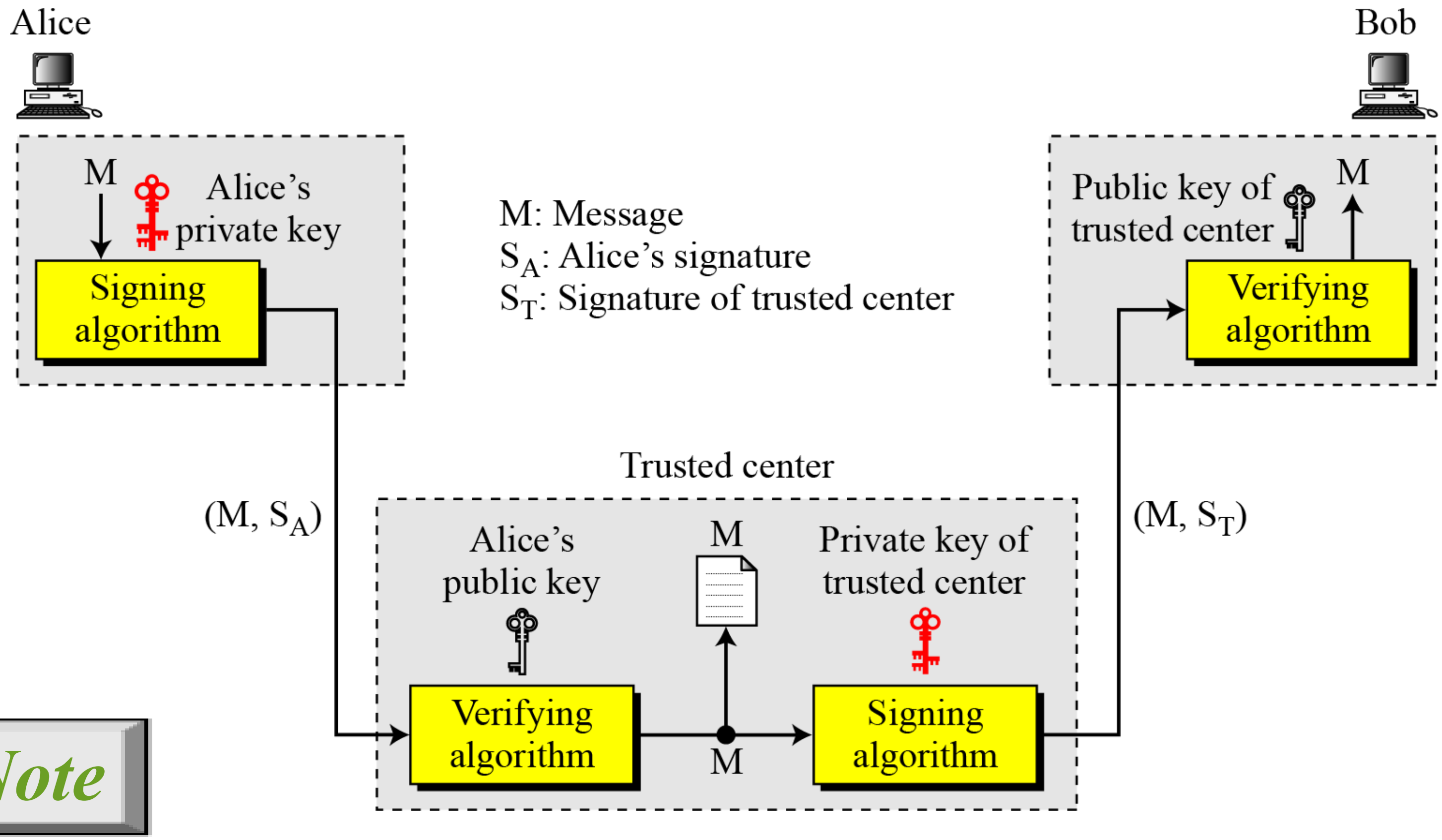
The integrity of the message is preserved even if we sign the whole message because we cannot get the same signature if the message is changed.

*Note*

**A digital signature provides message integrity.**

# Nonrepudiation

*Using a trusted center for nonrepudiation*

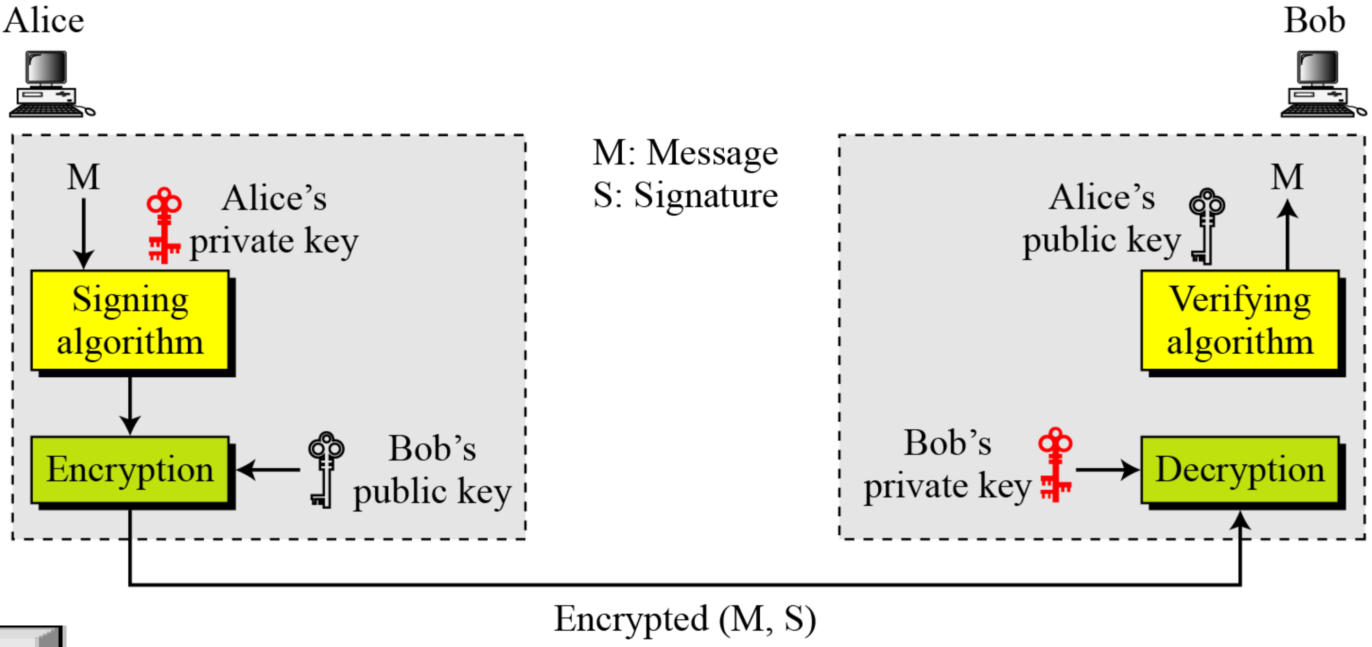


*Note*

**Nonrepudiation can be provided using a trusted party.**

# Confidentiality

*Adding confidentiality to a digital signature scheme*



*Note*

**A digital signature does not provide privacy. If there is a need for privacy, another layer of encryption/decryption must be applied.**

# Requirements for signatures

“valid signatures verify”

verify(pk, message, sign(sk, message)) == true

“can’t forge signatures”

adversary who: knows pk

gets to see signatures on messages of his choice

can’t produce a verifiable signature on another message

# Practical stuff…

- algorithms are randomized
  - need good source of randomness
- limit on message size
  - fix: use Hash(message) rather than message
- trick: sign a hash pointer
  - signature “covers” the whole structure

# What is Blockchain?

- A Linked List
  - Replicated
  - Distributed
  - Consistency maintained by Consensus
  - Cryptographically linked
  - Cryptographically assured integrity of data
- Used as
  - Immutable Ledger of events, transactions or time stamped data
  - Tamper resistant log
  - Platform to Create and Transact in Cryptocurrency
  - log of events/transactions unrelated to currency

# Why a course on Blockchain?

- Have you seen the news lately?
  - Bitcoin
  - Ethereum
  - Blockchain for E-governance
  - Blockchain for supply chain management
  - Blockchain for energy management ……
- Is it just a hype and hyperbole?
  - Hopefully this course will teach you otherwise
  - Even if you do not care about cryptocurrency and its market volatility

# Let’s First talk about Banking

Regulatory Agency (RBI)

Customers

Bank

Bank Employee

# How do you transact?

- You write a check or do internet transaction to pay a payee
- Bank checks if you have balance > transaction amount
  - If yes, it debits your account by balance = balance - transaction_amount
  - credit's payee's account by payee.balance = payee.balance + transaction_amount
  - If no, the transaction is invalid and rejected.
- You can check your transaction list online, or check the monthly statement
- Who maintains the ledger?
  - Bank Does
  - What if Bank allows an invalid transaction go through
    - Invalid = you did not authenticate the transaction
    - Invalid = your balance was not sufficient but transaction was made

# Bank Frauds

- You find a check was used to pay someone but you never wrote the check
  - Someone forged your check and/or signature
- You did sign a check for x amount, but the amount field was modified
  - How do you prove to the bank that an extra o was not there in your signing time?
- The monthly statement says that you did a transaction but you did not recall or the amount of a transaction is different from what you had done
  - Someone got your password, and possibly redirected OTP to another SIM (SIM Fraud)
  - Bank employees themselves might have done something
- How do you argue to the bank? (Non-repudiation)
- How do you argue that the amount was modified? (Integrity)
- Finally, do you tally your transactions when you receive your monthly statement? Most people do not

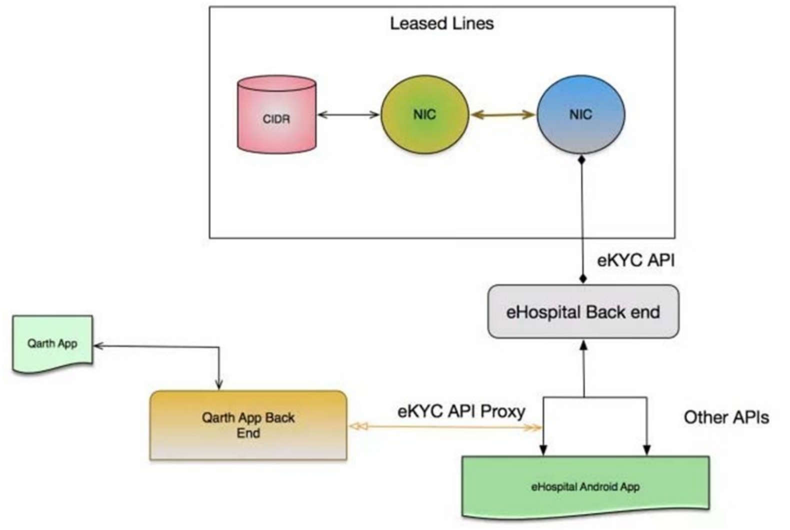
# Supply chain and provenance

- Your buy ice cream for your restaurant from supplier B
- Supplier B actually transports ice cream made in Company C's factory
- Upon delivery, you have been finding that your ice cream is already melted
- Who is responsible?
  - Supplier B is keeping it too long on the delivery truck?
  - Supplier B's storage facility has a temperature problem?
  - Supplier C says it's supplier B's fault as when picked up – ice cream was frozen
  - Supplier B says that when received, the temperature was too high, so C must have stored it or made it wrong
  - How do you find the truth?
    - Put temperature sensors in B's truck and storage, C's factory and storage, and sensor data is digitally signed by the entity where the sensor is placed and put in a log
    - You check the log – but B and C both have hacked the log and deleted some entries?
  - What to do?

# Land Record

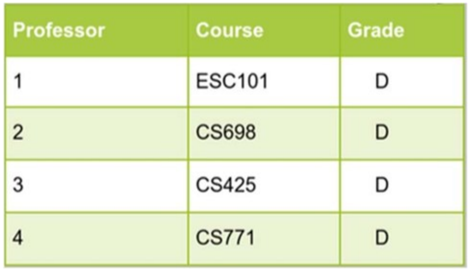
- You buy a piece of land
- Someone else claims to own the land
- But the one who sold you the land showed you paper work
- Land registry office earlier said that the owner was rightful
- Now they say that they made a mistake – it was owned by the other person
- You already paid for the land – to the first person
- First person goes missing
    - How does any one prove who changed the land record?
        - The government employees?

# Then there is Aadhaar



- E-KYC Logs
- Shown to you by UIDAI
- How do you know they did not delete important log events?
- Do you Trust UIDAI?

# A Student Online Grade Submission and Management System



| Professor | Course | Grade |
|---|---|---|
| 1 | ESC101 | D |
| 2 | CS698 | D |
| 3 | CS425 | D |
| 4 | CS771 | D |

# Again, What is a blockchain?

- Blockchain technology is a digital innovation that has the potential to significantly impact trusted computing activities and therefore cybersecurity concerns as a whole.
- Attractive properties of Blockchain
  - Log of data with digital signature
  - Immutable (once written – cryptographically hard to remove from the log)
  - Cryptographically secure – privacy preserving
  - Provides a basis for trusted computing on top of which applications can be built

# Trust Model

- Cyber Security is all about who you trust?
  - Trust your hardware to not leak your cryptographic keys?
  - Trust your O/S to not peek into your computation memory?
  - Trust your hypervisor to not mess up your process memory?
  - Trust your application to not be control hijacked or attack other applications?
- Where is your trust anchor?
  - Hardware?
  - Operating system?
  - Application?
  - Manufacturer?

# Trust Model (2)

- In many real life transactional activities – trust model is the inverse of the threat model
    - Do you trust your bank to not take out small amounts from your balance all the time? (Watch – "Office Space")
    - Do you trust the department of land records to keep your record's integrity?
    - Do you trust UIDAI officials to keep your aadhaar data from unauthorized access?
    - Do you trust your local system admins to not go around your back and change settings, leak passwords, change database entries, and remove their action from system logs?
    - In the patch management system of your enterprise, are the patches being put -- all have digital certificates? Who put them? Do you trust your employees to do the correct thing and not put a malware as patch?

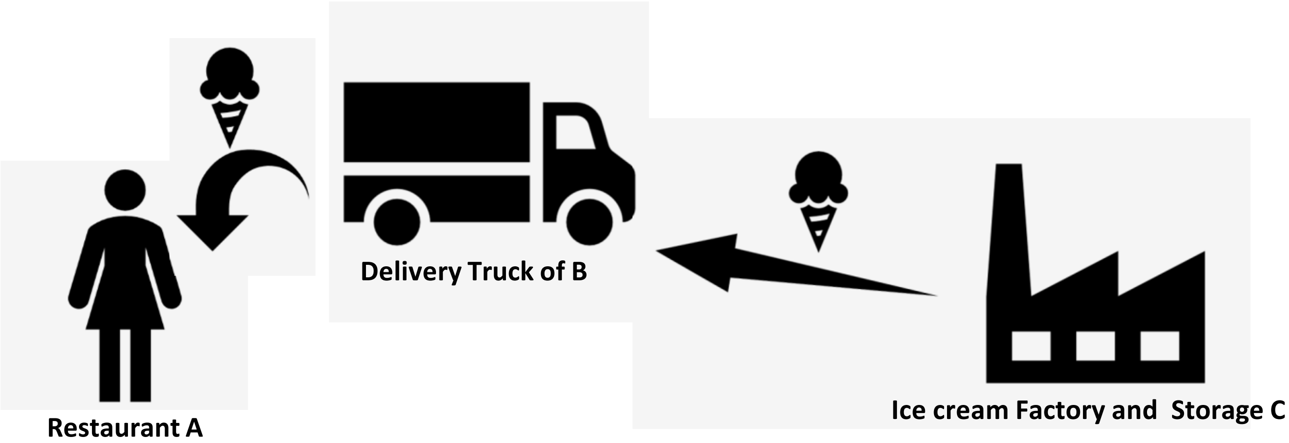
# Trusted Third Party

- Crucial question:

- You become the central Trusted Party keeping track of transaction history, arbitrating validity of transactions

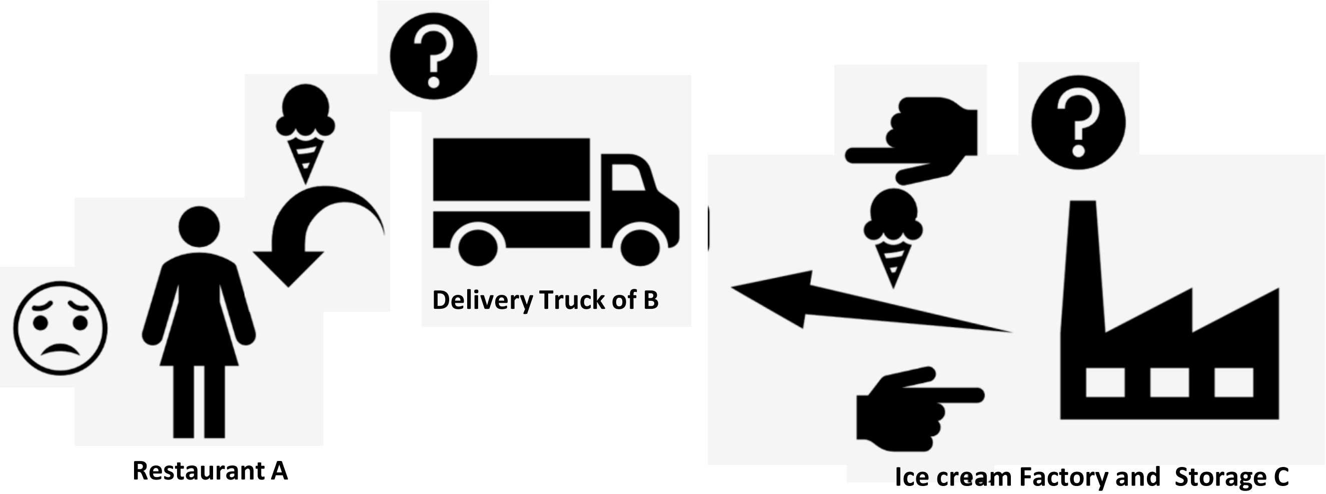
- Why should people trust you??

# Back to the supply Chain Story



Ice Cream Supply Chain to Restaurant A from Factory C via Supplier B

# Ice cream is melted



Ice Cream Supply Chain to Restaurant A from Factory C via Supplier B

# What can go wrong?

- IoT sensor data may be intercepted by a middle man and changed before it reaches the server (data integrity)
- IoT sensors may be stopped and old readings may be replayed (replay attack)
- What the server gets purportedly from factory C, may be manufactured by supplier B (Authenticity)
- If restaurant A claims that C's temperature reading shows that ice cream was melting in the storage, C can say that message you received is not from me – there was an MITM attack (repudiation)
- So restaurant A will not be able to pinpoint any one in the supply chain with full confidence!!

# What can be done?

- Use a message integrity proof (Hashing)
- Use digital signature of the individual IoT devices (Authenticity and non- repudiation)
  - assuming the digital signatures cannot be forged
  - private keys are kept safe
- Use authentic time stamping with the IoT data before hashing for integrity (avoid replay attacks)
- So now factory A can pinpoint with some basic security assumptions
- about this infrastructure

# Concurrency Issue

- A has other suppliers for other goods required for its business (multiple concurrent supply chains)
- B and C has multiple other consumers of their services
- So if there are N suppliers who are also consumers of some of these entities, we have an N$^2$ messaging problem
- A offers that every one can look up their data from my server, so you can get linear number of messaging
- But do you trust A as purveyors of your data?

# Solutions?

- Have a trusted authority or a cloud provider to become a publish- subscribe service provider
- Every supplier sends their IoT data with message integrity, authentication
- code etc., to the cloud server
  - Every consumer subscribes to the events they are interested in on the cloud
  - Every supplier becomes authenticated data generator on the cloud
- What if the cloud provider cannot be trusted?

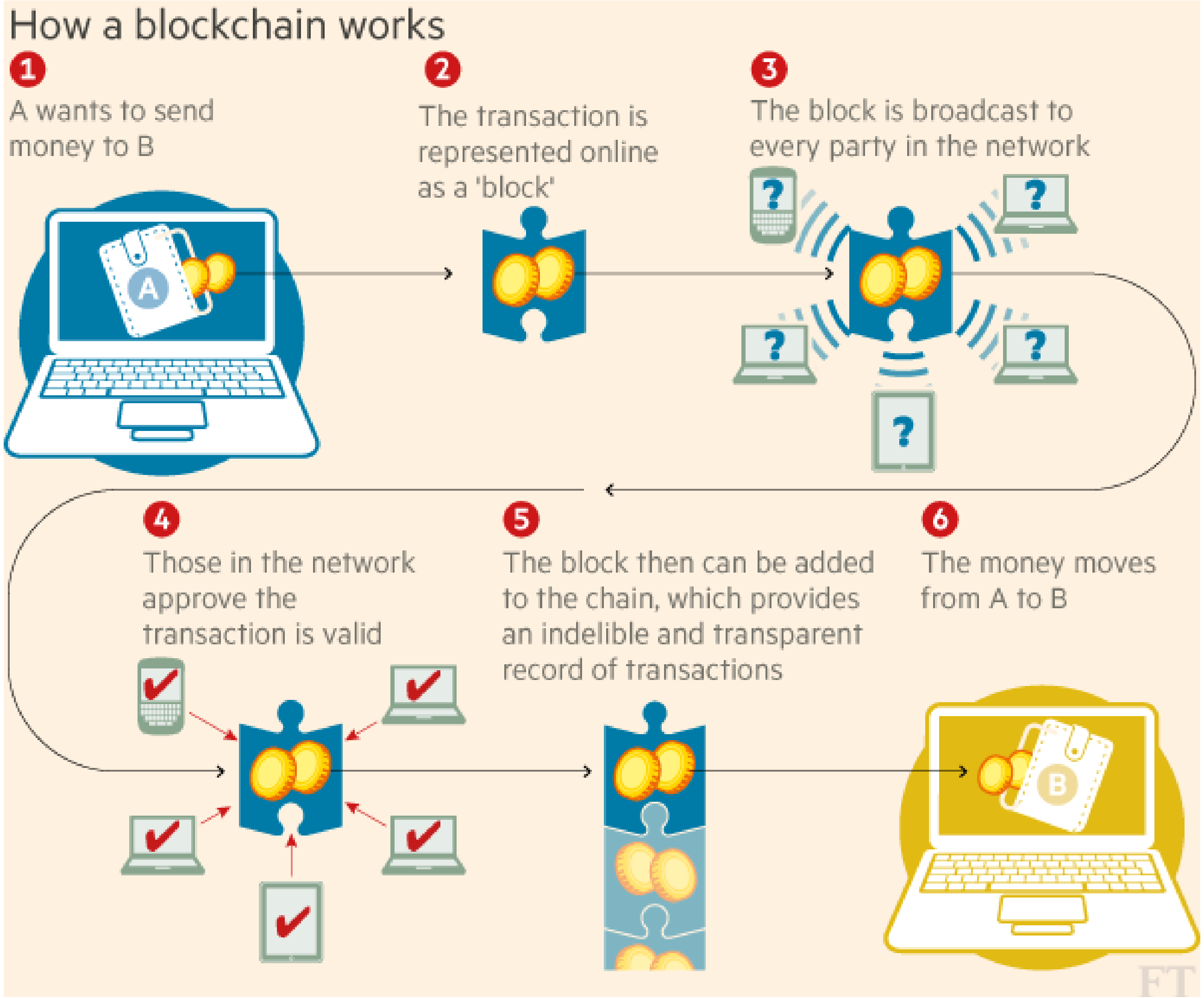
## Create a framework on which data is crowd sourced, validated by the crowd for the crowd?

- You get a block chain
- But now the question is as concurrent messages come in to this framework, how do you order them?

DISTRIBUTED CONCENSUS IS REQUIRED TO DECIDE

1. of all messages coming in concurrently how are they ordered
2. But if some of the crowd are malicious, and tries to allow data that are wrong, or ordered wrong?
3. You need Byzantine fault-tolerant consensus

# How Block chain works



How a blockchain works

1 A wants to send money to B

2 The transaction is represented online as a 'block'

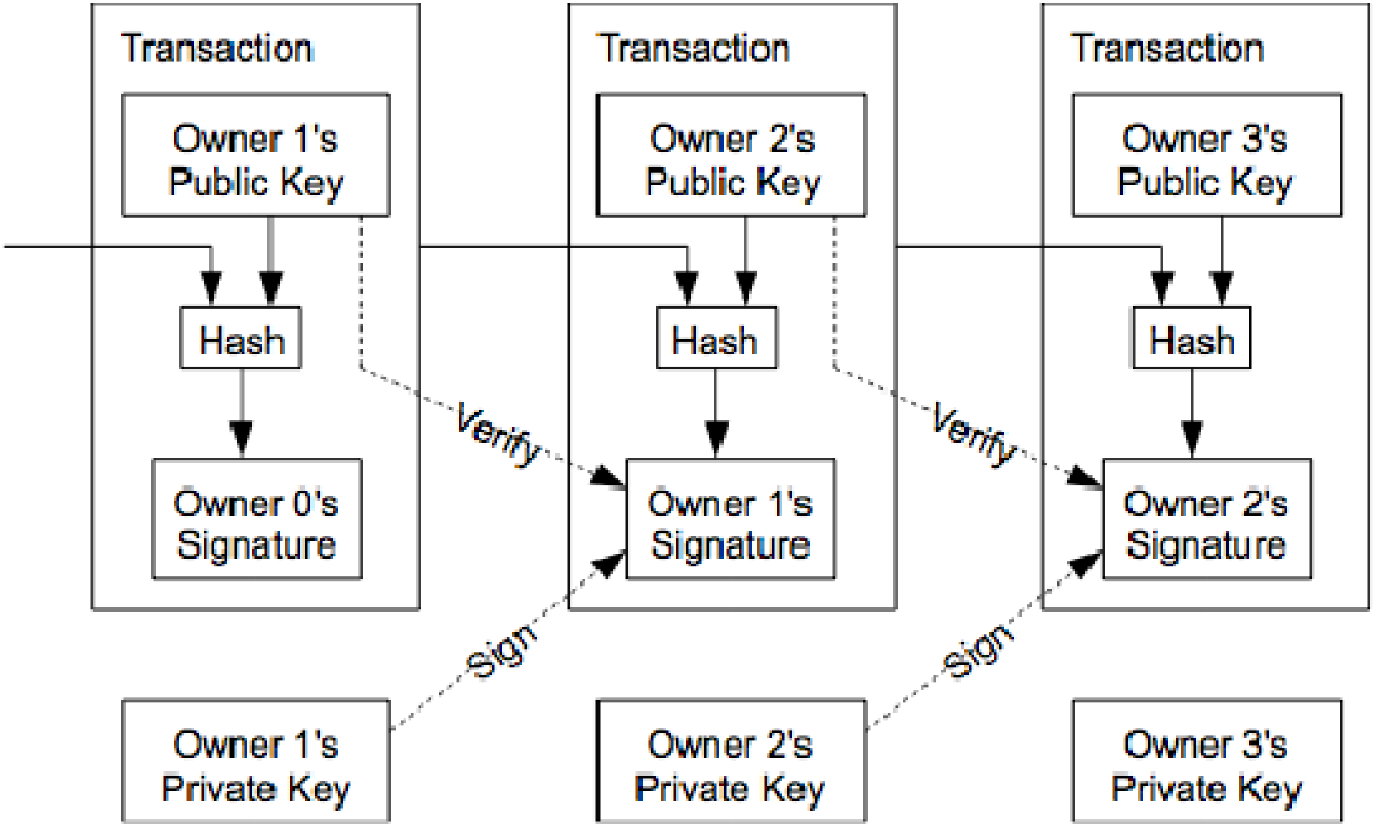
3 The block is broadcast to every party in the network

4 Those in the network approve the transaction is valid

5 The block then can be added to the chain, which provides an indelible and transparent record of transactions

6 The money moves from A to B

# How Block chain works



# Thank You!!!