Digital Signature

a digital code (generated and authenticated by public key encryption) which is attached to an electronically transmitted document to verify its contents and the sender's identity.

Digital Signatures

- Used to “Sign” messages to validate the source and integrity of the contents.
- Taking a digital picture of a written signature does not provide adequate security
- Digitized written signature could easily be copied from one electronic document to another with no way to determine whether it is legitimate
- Electronic signatures, on the other hand, are unique to the message being signed and will not verify if they are copied to another document
- When you add them to a document, you are "signing" that document as a way of endorsing or agreeing with what the document says.

- Unlike handwritten signatures, digital signatures are used only with computers.
- They are electronic signatures that ca be used to sign electronic documents like word processing files or spreadsheets
- A digital signature is a kind of ID.
- 'You can use it on the Internet to identify yourself in a secure manner.
- This is extremely useful in areas such as electronic commerce
- For instance, when making a credit card purchase on the Internet, you can use your digital signature to "sign" that purchase.
- This helps to ensure that only you can make purchases with your credit card number.

Digital Signatures

Requirements

- Signature → bit pattern —depends→ signed message

- Signature —use→ information unique to the sender
  —prevent→ forgery and denial
- Easy —produce→ digital signature
- Easy —recognize & verify→ digital signature
- Easy —retain→ copy of digital signature in storage

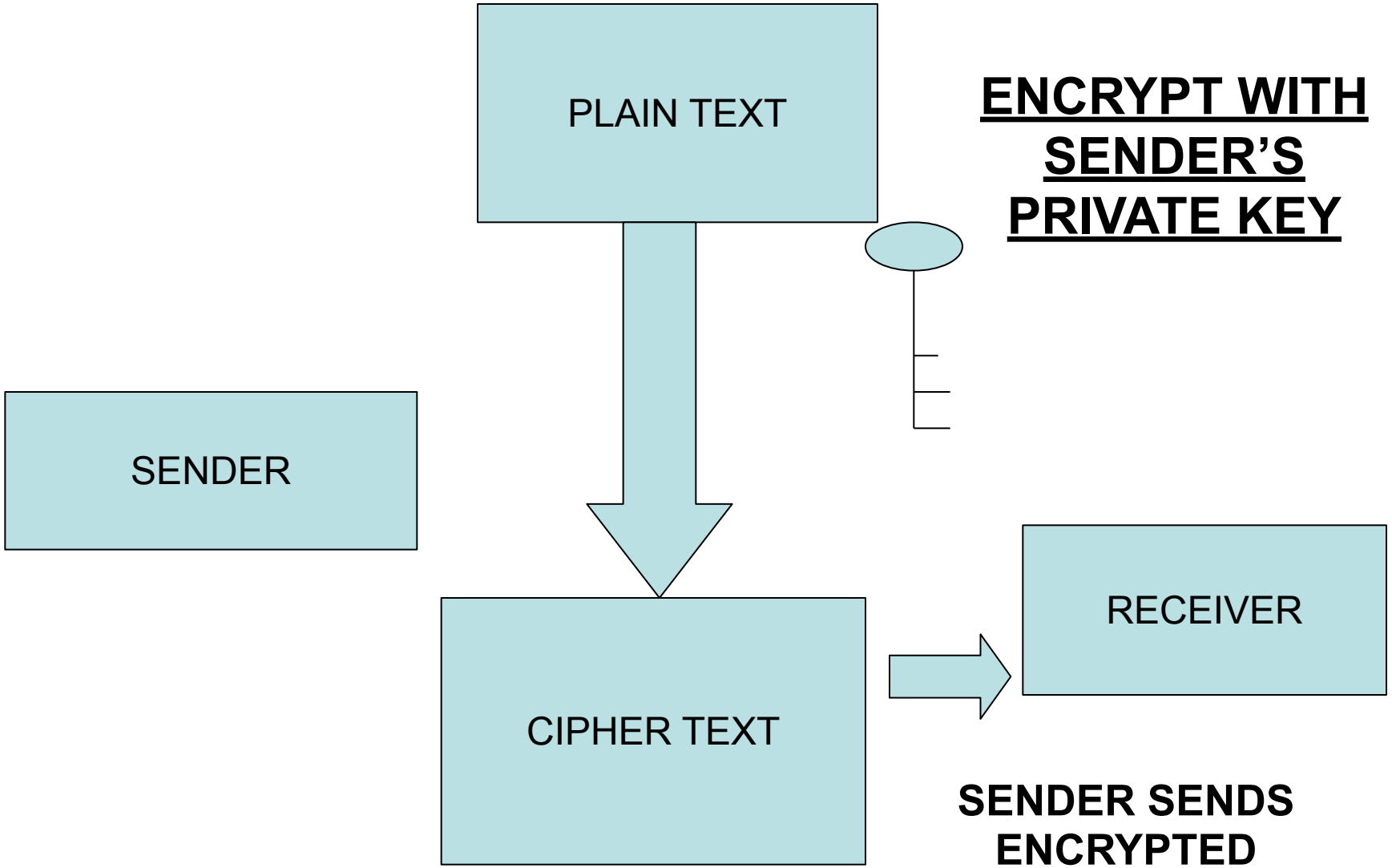
- Infeasible computation —forge→ digital signature

Requirements for a Digital Signature

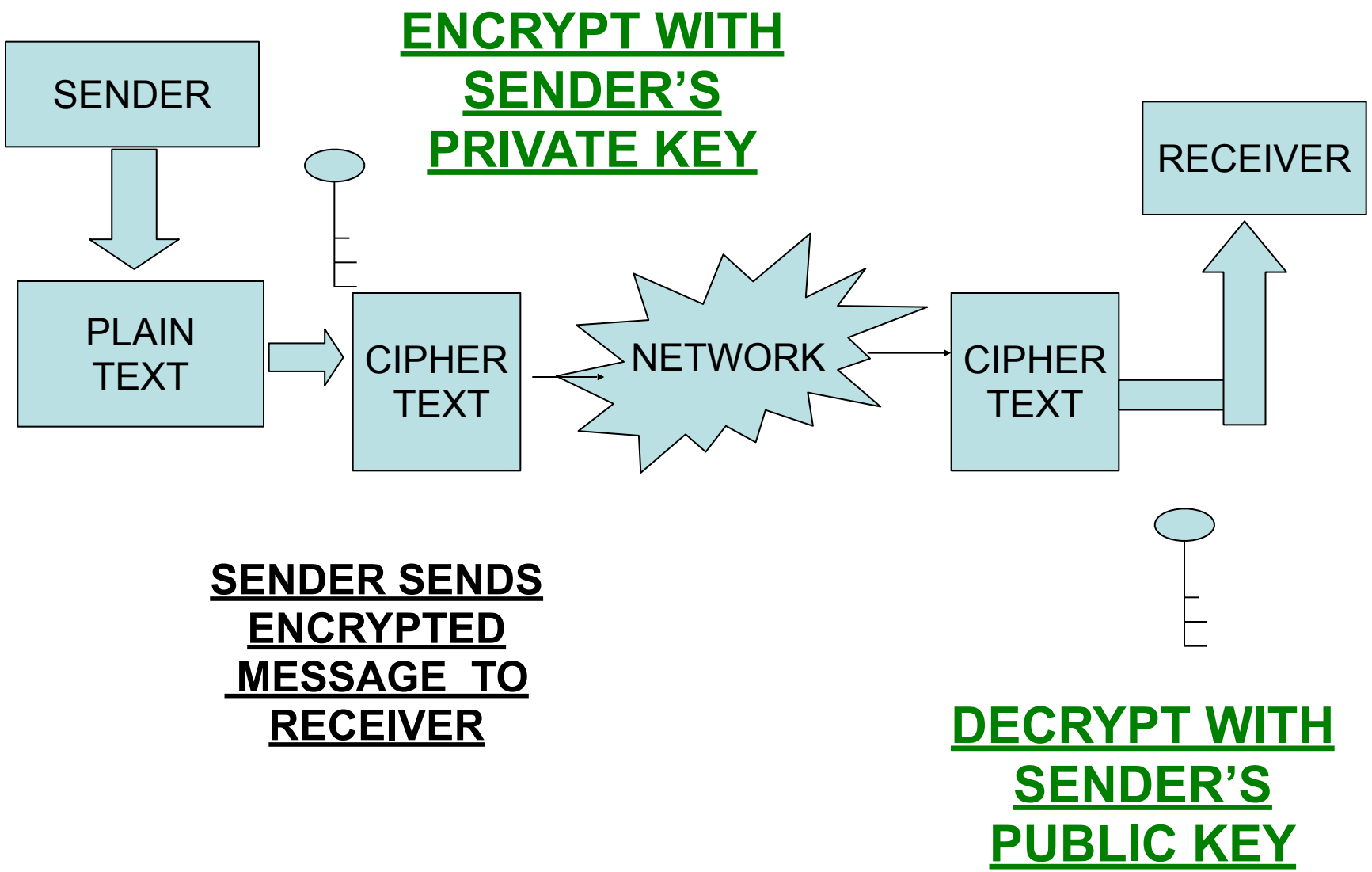
- The signature must be a **bit pattern** that depends on the message being signed
- The signature must use some information **unique to the sender**, to prevent both forgery and denial.
- It must be **relatively easy to produce** digital signature.
- It must be **relatively easy to recognize** and verify the digital signature.
- It must be **computationally infeasible to forge a digital signature**, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.
- It must be **practical to retain a copy of the digital signature** in storage.

How is a Digital Signature Produced?

- Very briefly, a typical digital signature works like this:
  - A signature in the form of a code is generated by applying an algorithm, such as RSA, and the sender's private key to some or all of the message contents.
  - The recipient verifies the signature by decrypting it using the sender's public key.


PLAIN TEXT
ENCRYPT WITH SENDER'S PRIVATE KEY
SENDER
CIPHER TEXT
RECEIVER
SENDER SENDS ENCRYPTED MESSAGE TO RECEIVER

Basis for Digital Signature



What purpose will it serve?

- Sender's public key is accessible by anybody . This means that any one can decrypt using public key thus causing **failure to encryption scheme**

- Though this sounds true the intension of sender is **not confidentiality ;**

- If the decryption was successful it is well ensured that it was sent by A only because private key is known only to sender alone. So this scheme achieves **authentication** i.e **identifies and proves** A is the sender

- In case of dispute tomorrow, B can take the cipher text decrypt with A's public key and prove that it came only from A. This achieves **non-repudiation**

If C intercepts?

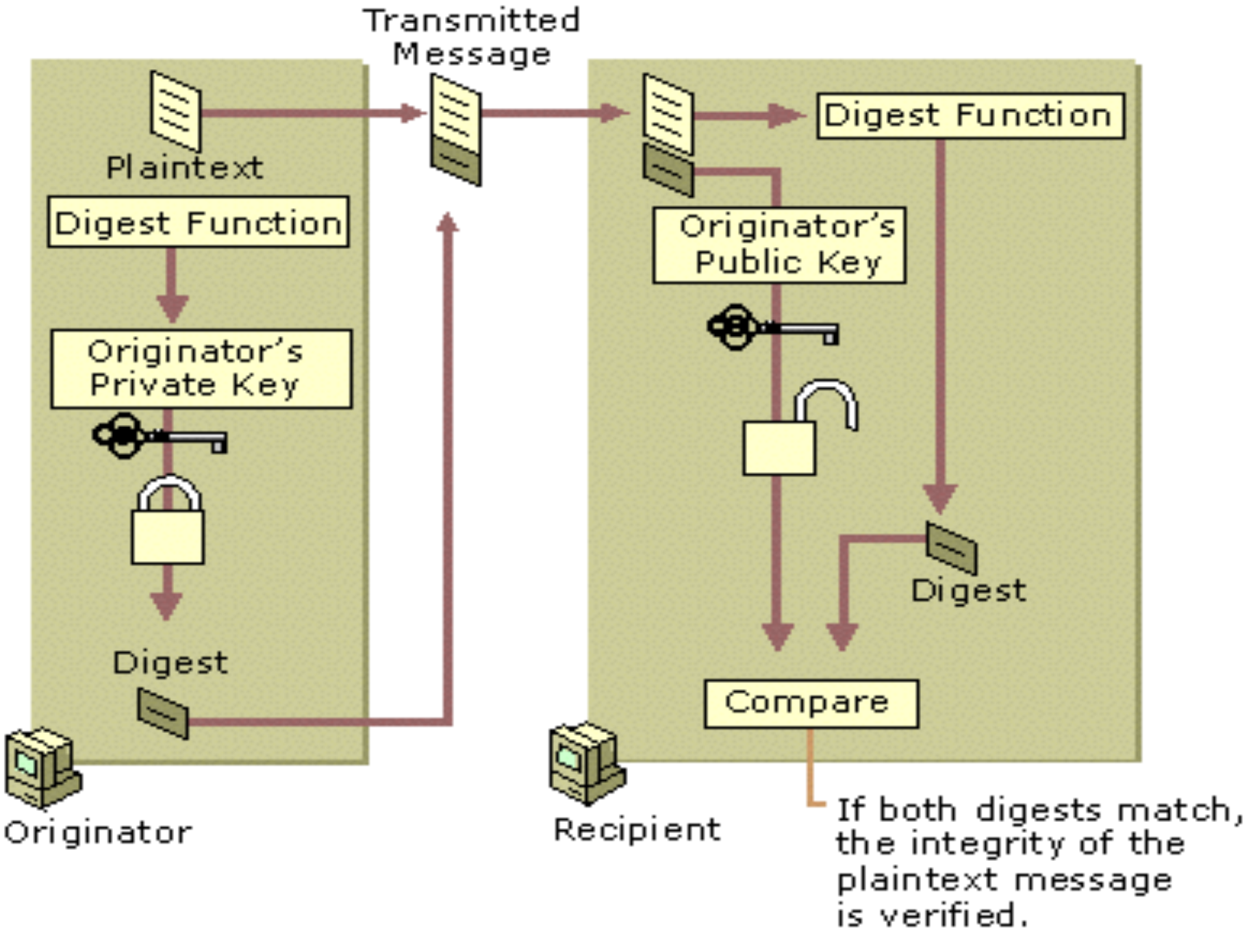
- Let C **intercept** the ciphertext
- Use the public key of A and **reads the message**
- **Changes the message** that would not achieve any purpose
- Because C doesn't have A's private key, so **cannot encrypt with private key again**
- Even if C forwards the message to B he **will not believe** that it came from A as it was not encrypted with private key of A.
- This scheme where you **encrypt with private key** and **decrypt with public key** to ensure **authentication , integrity and non-repudiation,** is called digital signature.

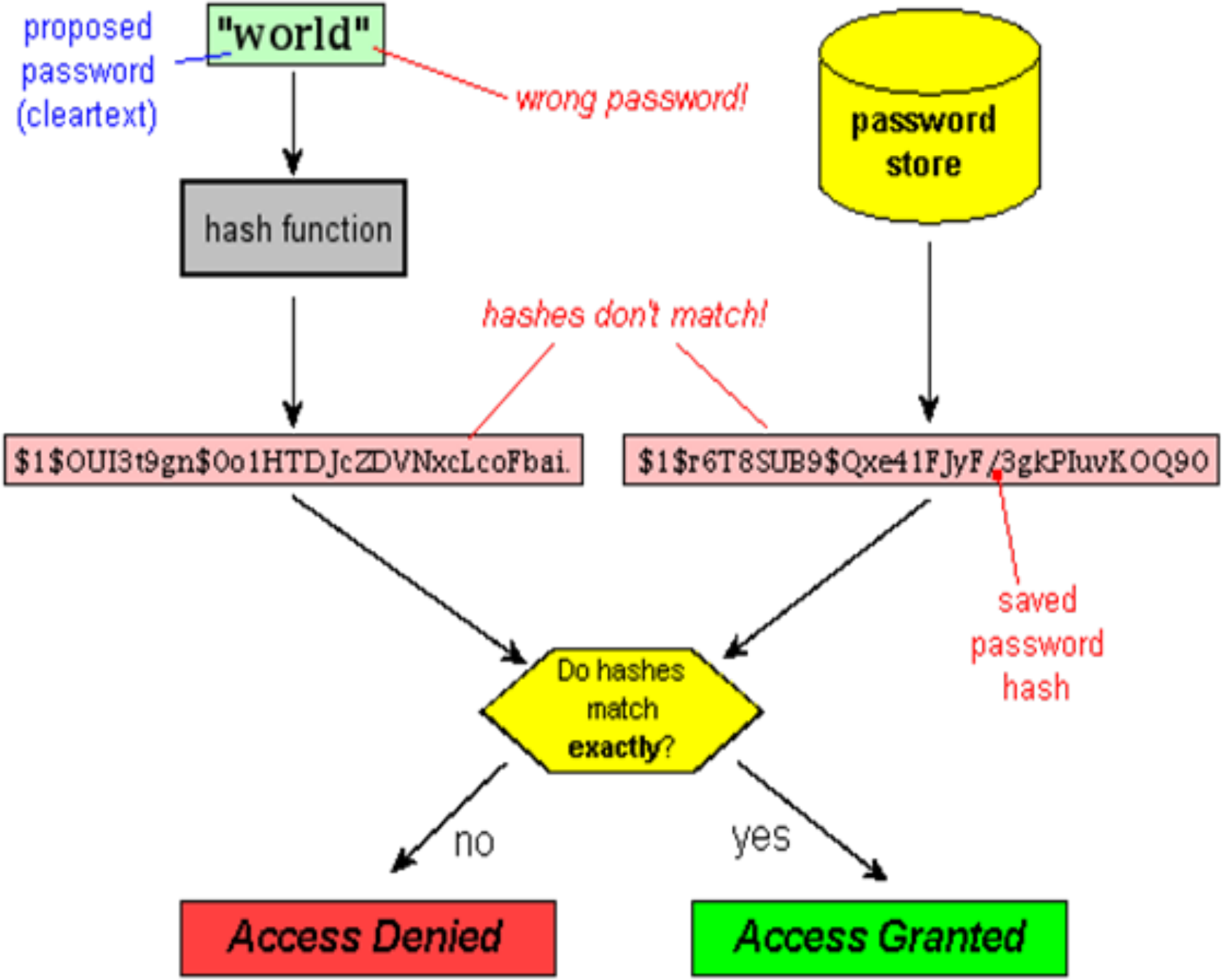
Problems faced

- As it deals with asymmetric scheme, if the plain text is large, **process could become slow.**
- This can be **tackled by digital envelope**
- Yet another Efficient scheme used today is **message digest also called hash**

What is the message digest?

- A **message digest** is a cryptographic hash function containing a string of digits created by a one-way hashing formula. **Message digests** are designed to protect the integrity of a piece of data or media to detect changes and alterations to any part of a **message**.


Transmitted Message
Plaintext
Digest Function
Originator's Private Key
Digest
Originator
Digest Function
Originator's Public Key
Digest
Compare
Recipient
If both digests match, the integrity of the plaintext message is verified.


proposed password (cleartext)
"world"
wrong password!
password store
hash function
hashes don't match!
1OUI3t9gn$0o1HTDJcZDVNxcLcoFbai.
1r6T8SUB9$Qxe41FJyF/3gkPIuvKOQ90
saved password hash
Do hashes match exactly?
no
yes
Access Denied
Access Granted

Message Digest

- Message digest is called a **finger print** or the **summary of the message**
- This is used to **verify the integrity** of the data
- i.e it has **not been tampered** before it reaches the receiver
- This requires a calculation of **LRC** – Longitudinal Redundancy Check

LRC

Original data

11100100	11011101	00111001	00101001

11100100
11011101
00111001
00101001
00101001

Original data arranged as rows of a list

Even parity : 0 Column 1
Odd Parity : 1 Column 3

11100100	11011101	00111001	00101001	00101001

Original data and LRC

How to calculate LRC?

1. If we want to send 32 bits , **arrange them horizontal row** (4 horizontal rows)
2. Check **how many one bits** occur
3. If the number of 1s is odd then call it as **odd parity**
4. Otherwise **even parity**
5. This LRC is originally fingerprint of the original message
6. Data along with LRC is sent to the receiver
7. Receiver separates the data block and LRC block
8. Performs LRC on data block alone and compares with the one sent
9. If they match receiver has reasonable confidence.

Idea of message digest

- Similar principle as LRC
- Suppose we have a number 4000, we divide it by 4 and get 1000;
- Thus **4 is the fingerprint of 4000**
- **Always 4000/4 = 1000**
- If you change 4000 or 4 then the result will not be same
- If just given the number 4 **we cannot track 4 is by 4 x 1000**
- Similarly fingerprint of a message **does not say anything about the message**
- There are **infinite possibilities** for possible equations

Example

Original number is 7391743

Operation	**Result**
Multiple 7 by 3	21
Discard the first digit	1
Multiply 1 by 9	9
Multiply 9 by 1	9
Multiply 9 by 7	63
Discard the first digit	3
Multiply 3 by 4	12
Discard the first digit	2
Multiply 2 by 3	6

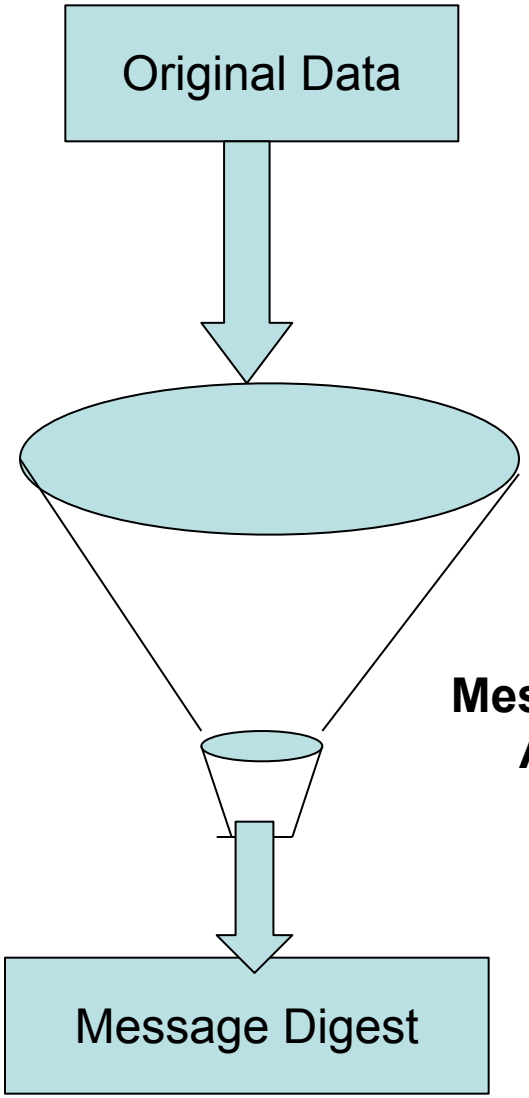
MESSGAE DIGEST IS 6

Message digests

- So far we are considering very simple case of message digest
- Actually **message digest are not so small** and straight forward to complete
- Usually consists of **128 or more bits**
- Therefore the chance of two digests being the same is between **0 to 2^{128}**
- The message digest is chosen so long with the purpose that it **minimizes two digests being the same.**

Requirements of message digest

- Given a message it should be very **easy to find** its corresponding **message digest**.
- Also for a given message the message digest must always be the same

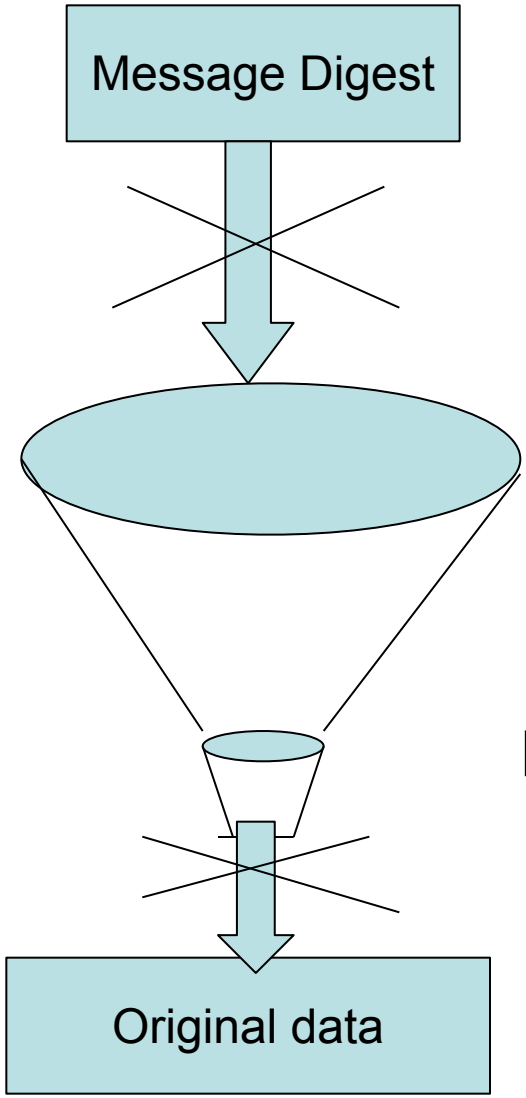


Message Digest Algorithm

SHOULD BE POSSIBLE AND THE RESULT SHOULD ALWAYS BE THE SAME

Requirements of message digest

- Given the message digest it should be **very difficult to find the original message** for which the digest was created

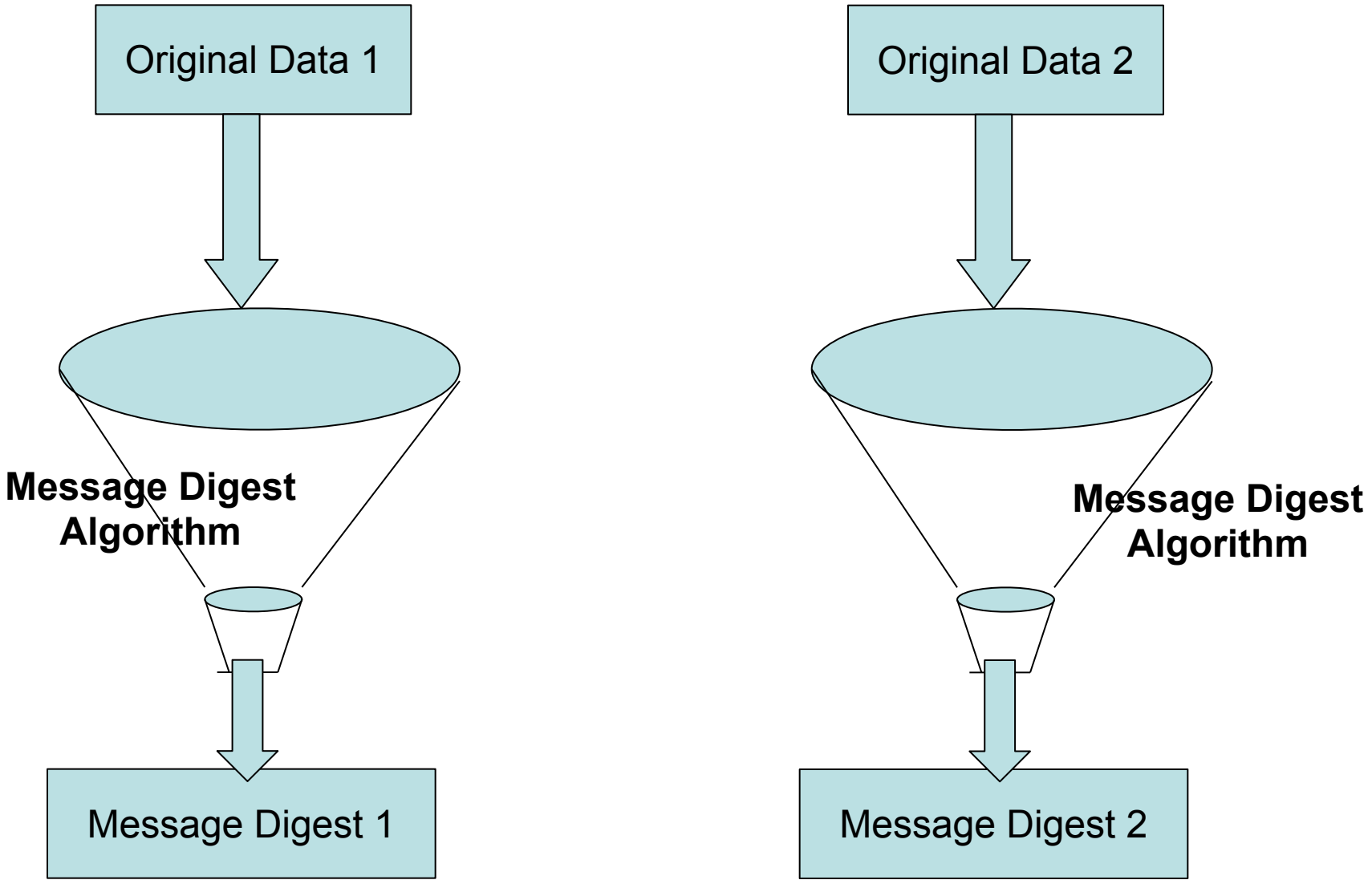


SHOULD NOT BE POSSIBLE

Reverse Message Digest Algorithm

Requirements of message digest

- Given two messages if we calculate their message digests the **two digests must be different**



These two message digests must be different

Message Digest Example

MESSAGE	Please pay the newspaper bill today
MESSAGE DIGEST	306706092A864886F70D010705A05A30580 …… ----------646
MESSAGE	Please pay the newspaper bill tomorrow
MESSAGE DIGEST	306A06092A864886F70D010705A05D305B020 …… ----------C6C20746

What is a hash function?

- A **hash function** is any **function** that can be used to map data of arbitrary size to data of fixed size.

- The values returned by a **hash function** are called **hash** values, **hash** codes, **hash** sums, or simply **hashes**.

- Since the hash is short, using the hash instead of the original input is much faster.

- **Cryptographic** hash functions take an input of arbitrary length and produces a **message digest** that is of a fixed, short length (e.g. 128 or 160 bits).

- The **digest** is sometimes also called the "hash" or "fingerprint" of the input

What is one way encryption?

- One-way encryption, or a **one-way hash function**, is designed in such a way that it is hard to reverse the process, that is, to find a string that hashes to a given value (hence the name one-way.)
- A good hash function also makes it hard to find two strings that would produce the **same hash value**.

Requirements for a Cryptographic Hash Function H

Requirement	Description
Variable input size	H can be applied to a block of data of any size.
Fixed output size	H produces a fixed-length output.
Efficiency	$H(x)$ is relatively easy to compute for any given x, making both hardware and software implementations practical.
Preimage resistant (one-way property)	For any given hash value h, it is computationally infeasible to find y such that $H(y) = h$.
Second preimage resistant (weak collision resistant)	For any given block x, it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$.
Collision resistant (strong collision resistant)	It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.
Pseudorandomness	Output of H meets standard tests for pseudorandomness

Requirements for One-way Hash Function

- H can be applied to a block of data of any size.
- H produces a fixed length output
- H(x) is relatively easy to compute, making hw/sw implementations practical
- For any given code m it is computationally infeasible to find x such that H(x)=m
- For any given block x, it is computationally infeasible to find y not= x with H(x)=H(y)

What is a md5 hash?

- The **MD5** message-digest algorithm is a widely used cryptographic **hash** function producing a 128-bit (16-byte) **hash** value, typically expressed in text format as a 32 digit hexadecimal number. **MD5** has been utilized in a wide variety of cryptographic applications, and is also commonly used to verify data integrity.

Next Class

- MD5 Algorithm for Message Digest