Digital Signatures

SECURE COMMUNICATIONS & CRYPTOGRAPHY

Mark Cummins, Institute of Technology Blanchardstown

A Quick Recap

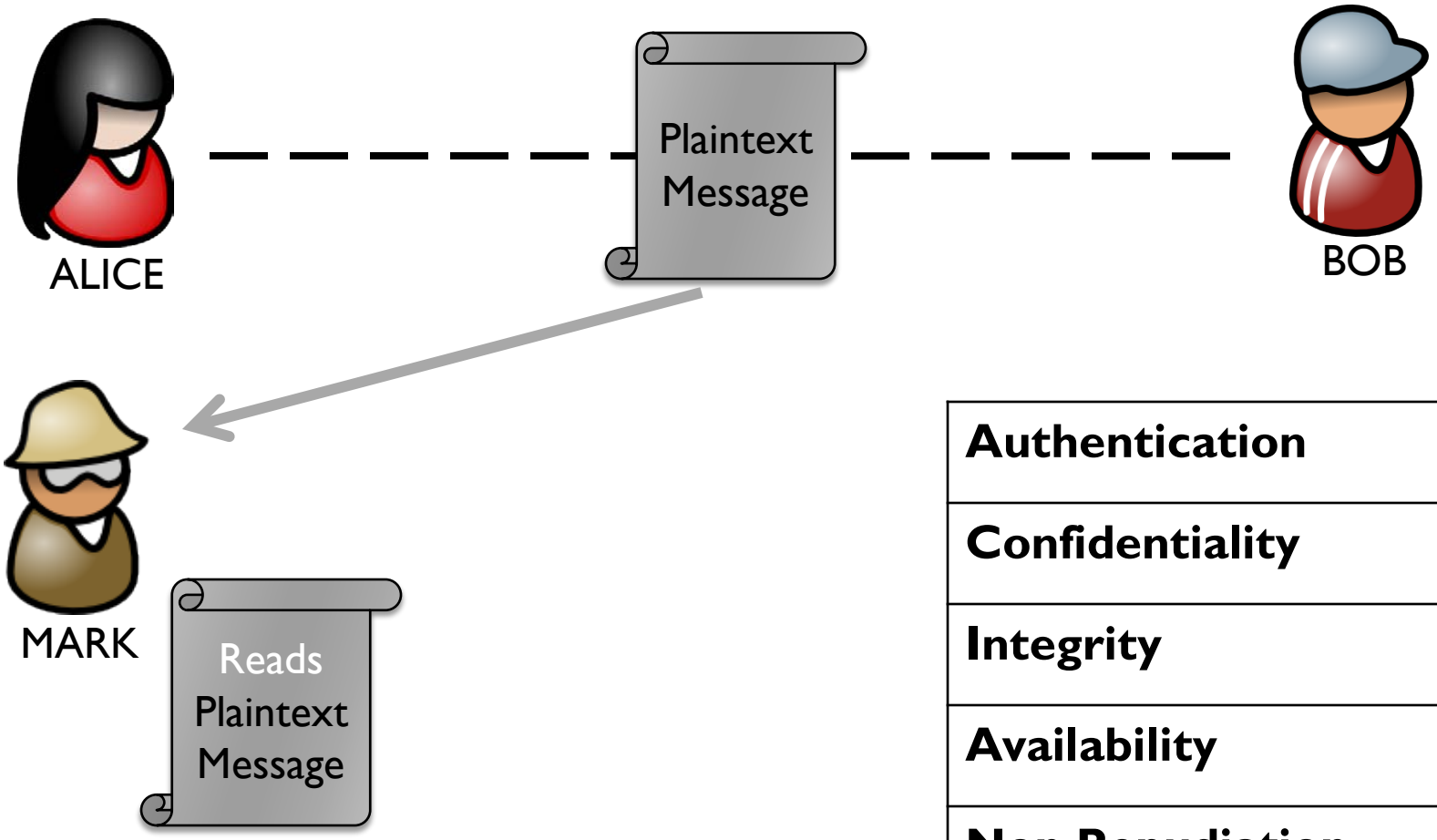
- Alice wants to send a message to Bob
- If they send the message as plaintext, anyone who finds the message can read it, or alter it and resend it.
- So how do we ensure that the message hasn't been intercepted, or tampered?
- How do we know it was even Alice who sent us the message and not someone else pretending to be Alice?

A Quick Recap

- When designing a secure communication system we need to ensure we meet each of the following:
  - Authentication
  - Confidentiality
  - Integrity
  - Availability
  - Non Repudiation

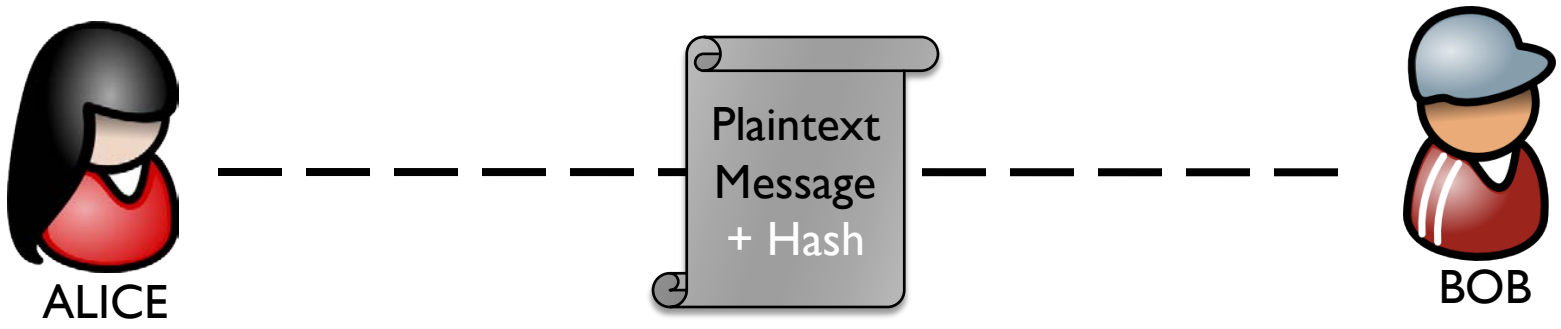
- So lets look at our example of Alice and Bob sending a message to each other.

A Quick Recap



Authentication	✗
Confidentiality	✗
Integrity	✗
Availability	✗
Non Repudiation	✗

Add a Hash of the Message

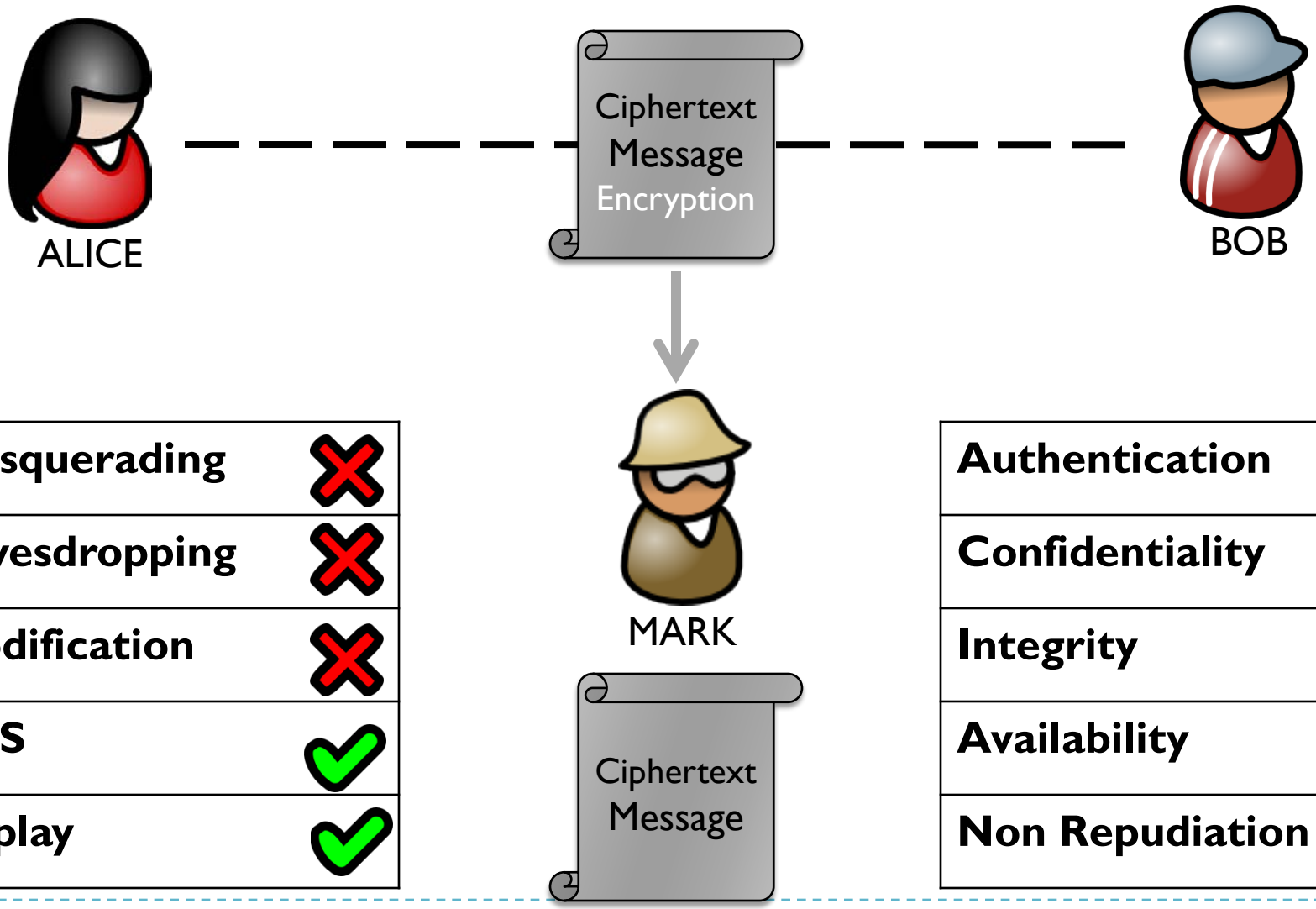


Reads Plaintext Message

Masquerading	✓
Eavesdropping	✓
Modification	✗
DoS	✓
Replay	✓

Authentication	✗
Confidentiality	✗
Integrity	✓
Availability	✗
Non Repudiation	✗

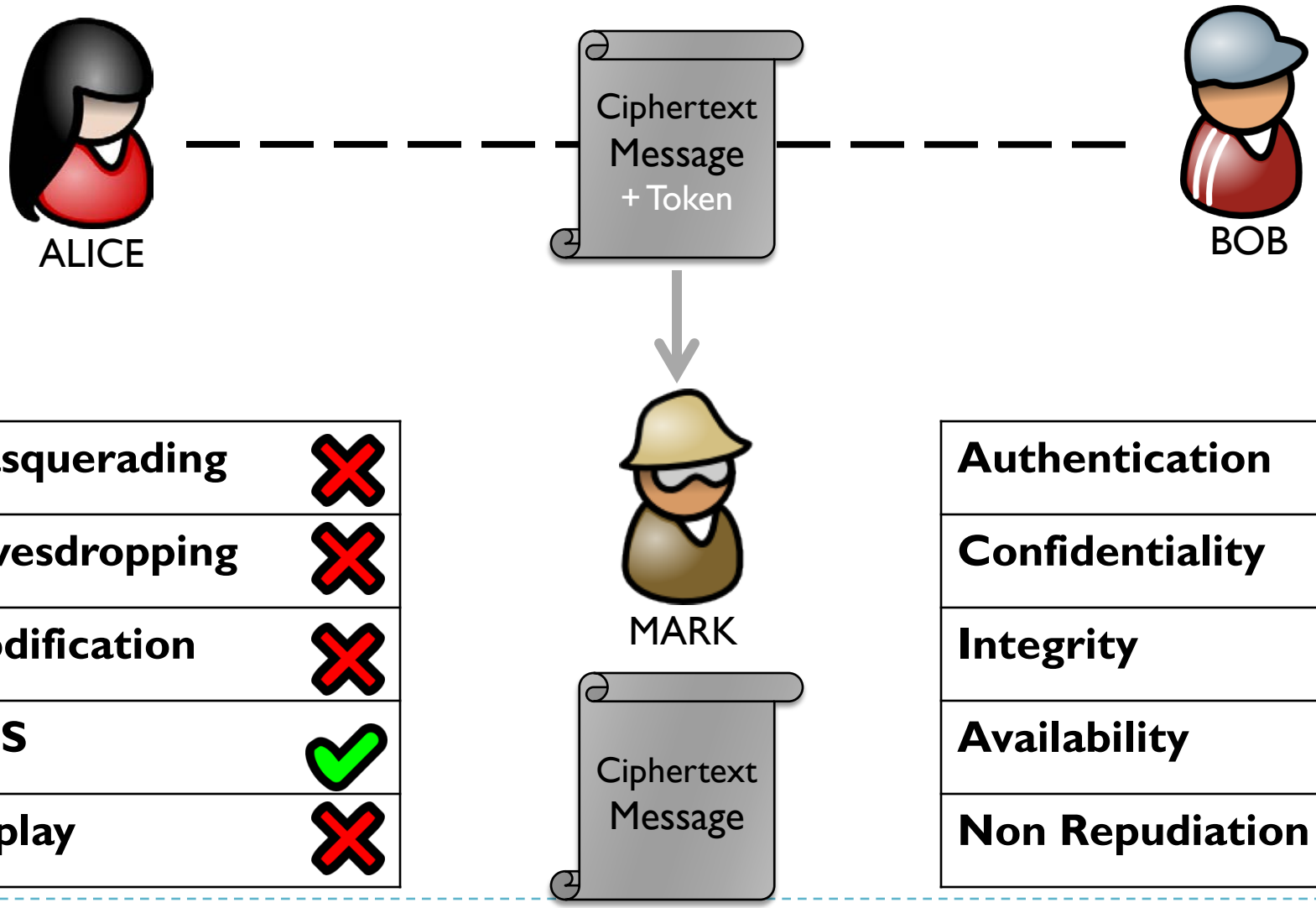
Encrypt the Message



Masquerading	✗
Eavesdropping	✗
Modification	✗
DoS	✓
Replay	✓

Authentication	✓
Confidentiality	✓
Integrity	✓
Availability	✗
Non Repudiation	✗

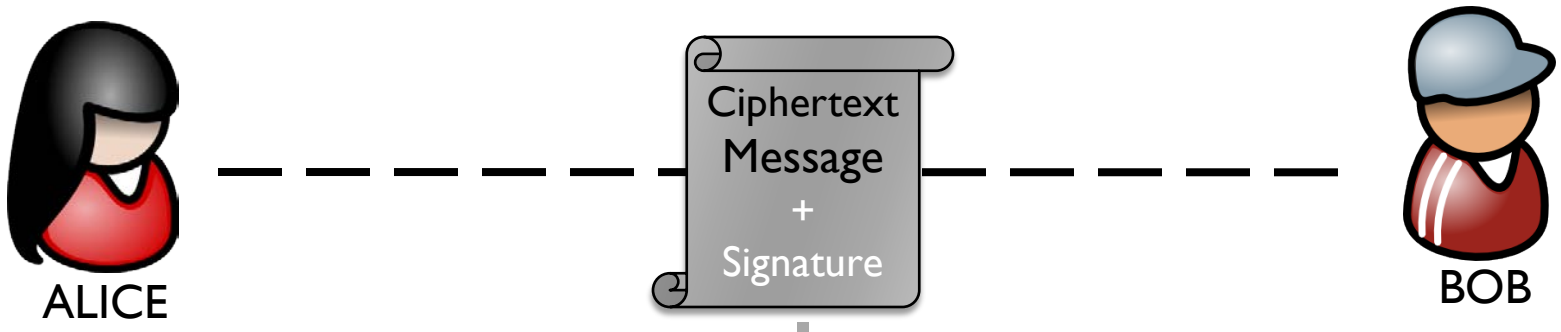
Encrypt the Message & add a Token



Masquerading	✗
Eavesdropping	✗
Modification	✗
DoS	✓
Replay	✗

Authentication	✓
Confidentiality	✓
Integrity	✓
Availability	✗
Non Repudiation	✗

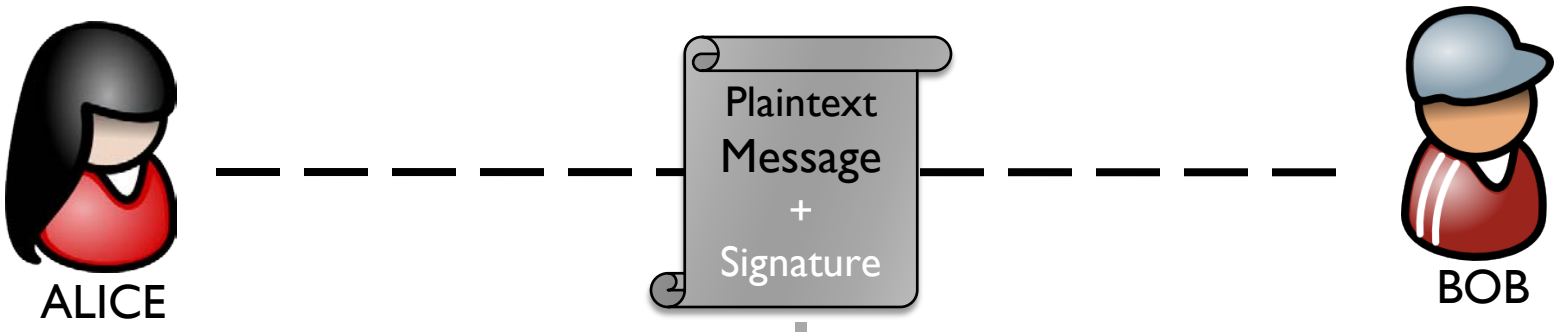
Encryption and Signature

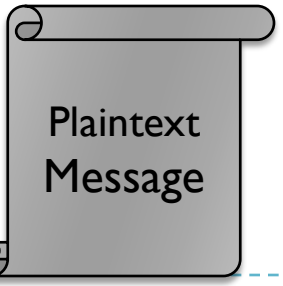


Masquerading	✗
Eavesdropping	✗
Modification	✗
DoS	✓
Replay	✓

Authentication	✓
Confidentiality	✓
Integrity	✓
Availability	✗
Non Repudiation	✓

Just Signature





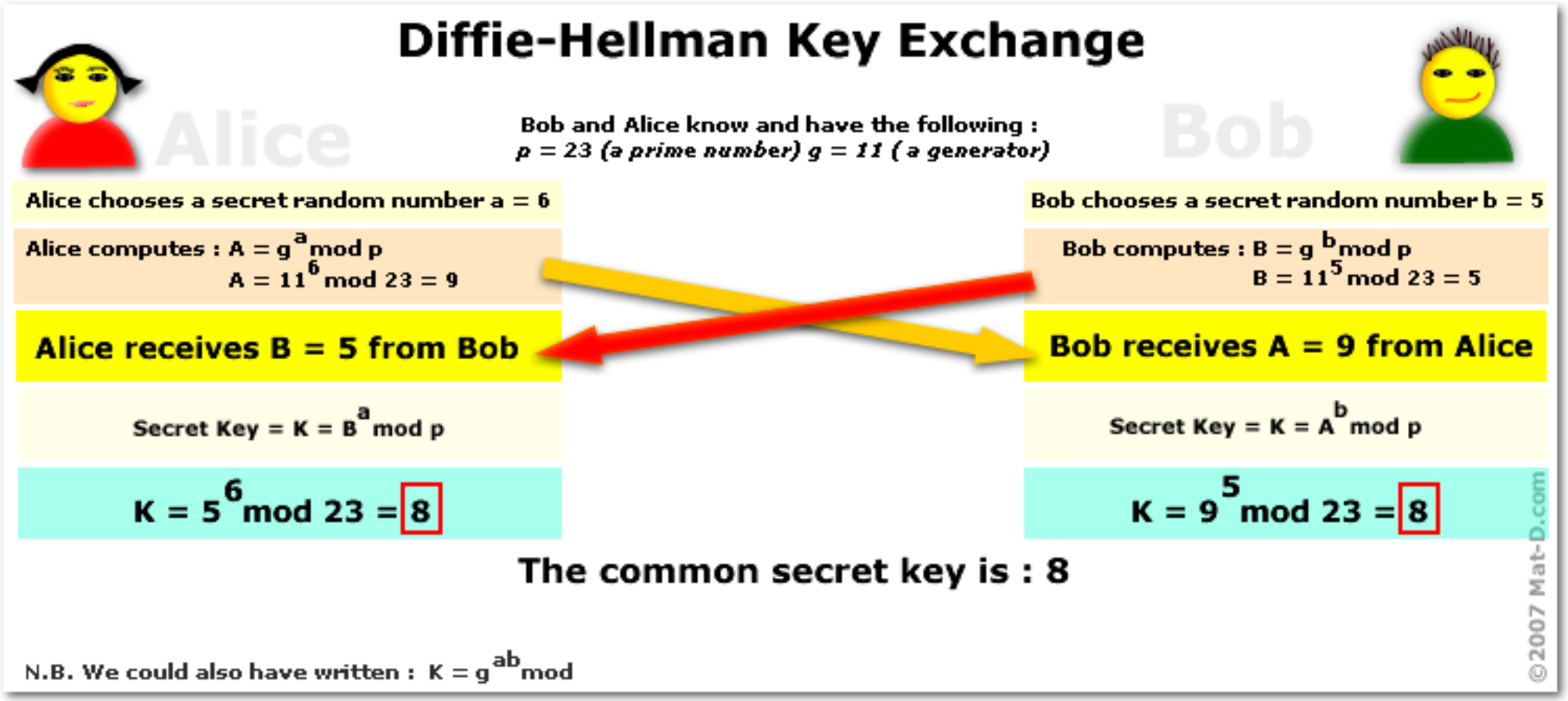
Masquerading	✗
Eavesdropping	✓
Modification	✗
DoS	✓
Replay	✓

Authentication	✓
Confidentiality	✗
Integrity	✓
Availability	✗
Non Repudiation	✓

A Quick Recap

- So Hashing, encryption and digital signatures are all tools.
- They each offer us different things and can be used in various combinations, to strengthen our security.
- Each of the tools may have its own weaknesses also
  - MD5 hashing is vulnerable to collision attacks
  - Diffie-Hellman key exchange is vulnerable to MITM attack

Another look at Diffie-Hellman

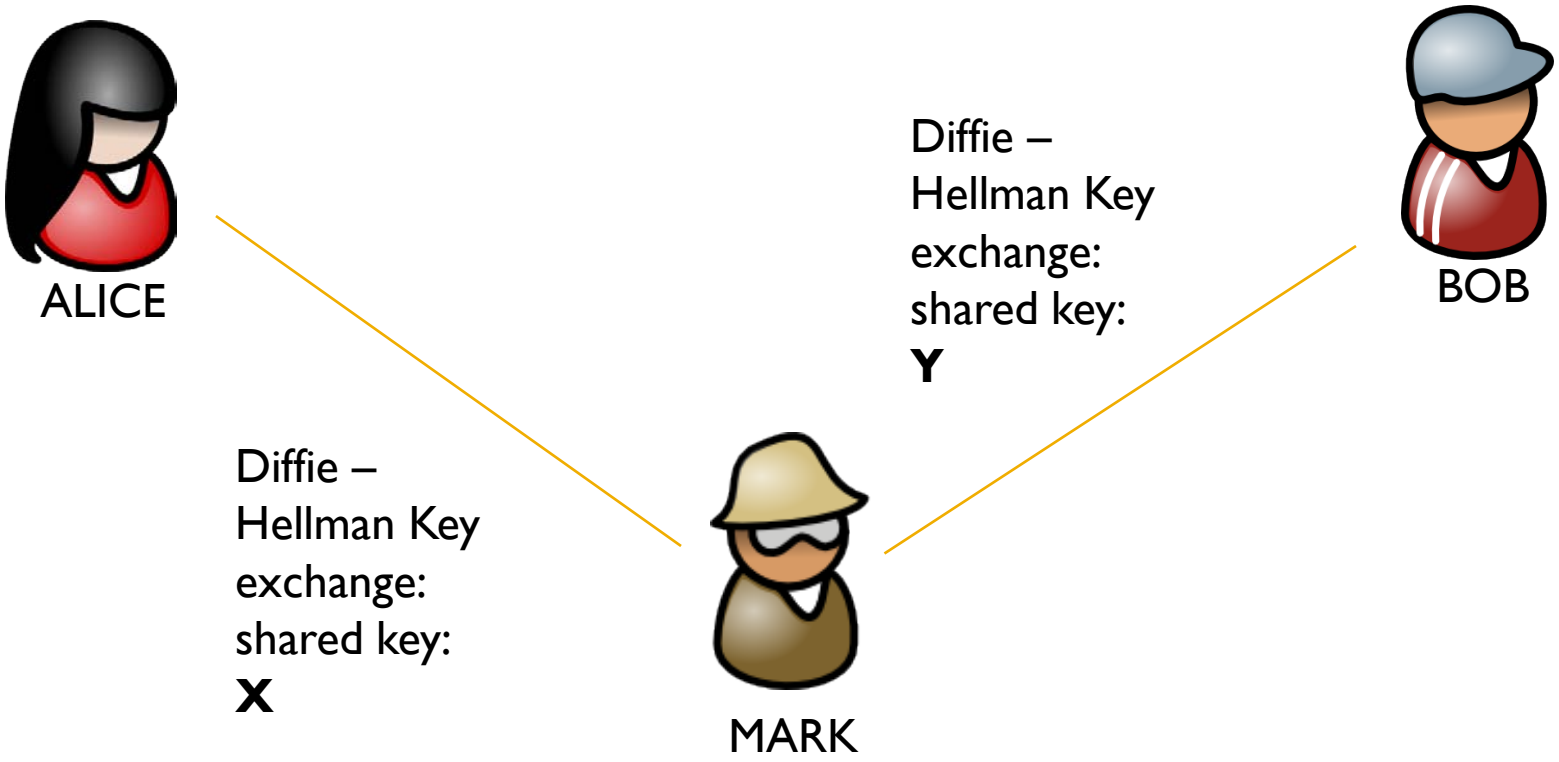


Another look at Diffie-Hellman

- But what if I got in the middle and pretended to be Alice to Bob and pretended to be Bob to Alice?

- This is called a Man-in-the-middle attack

Another look at Diffie Hellman



Another look at Diffie-Hellman

- So Diffie – Hellman was a key exchange algorithm,
- It creates a shared secret key, similar to symmetric encryption.
- It gets around the key exchange problem.
- It is rarely used today because of the possibility for MITM attack

Why is RSA different?

- The RSA algorithm involves three steps:
  - key generation,
  - encryption
  - decryption.

- RSA creates a key pair (public and private) for a single user.

- There is a relationship between these keys and the RSA algorithm, that allows one key to encrypt and the other decrypt

Digital Signatures

- RSA can be equally used for signing and for encryption.

- This is a very specific property of RSA, and can be blamed for most of the misconceptions about digital signatures and public key cryptography.

So what is a Digital Signature?

- When we created a hash of a document, the purpose was to show that the original document had not been tampered with.

- Websites that allow there software to be downloaded from other mirror sites sometime include a hash of the original software, so users can be sure that the version they download from a mirror site (over which the developers have no control) hasn't been tampered with.

So what is a Digital Signature?

- But what if I send you an email with a document etc. If I include a hash, anyone who intercepts the message could change the Email and document and simply create a new hash.

- How do we get around this problem?

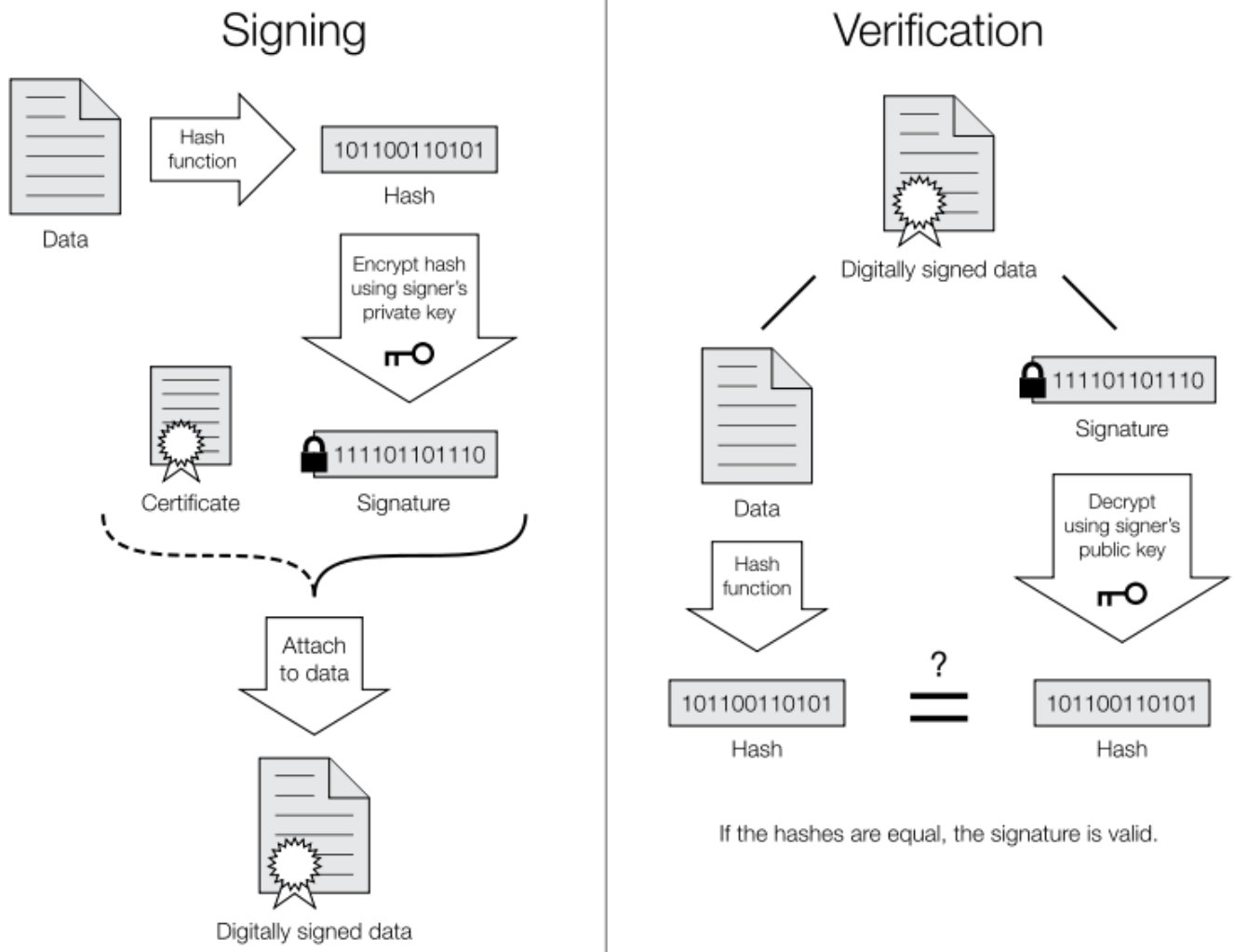
So what is a Digital Signature?

- The key is to use signatures.
- When I create a hash value of a document, I encrypt the hash with my private key
- Any one can decrypt the encrypted hash using my freely available public key.
- Problem solved?

Problem Solved?

- The only problem now is my public key.
- Someone else could issue a public key and say that its my key. They could then send messages pretending to be me.
- So we are back where we started.
- The solution is for me to register my public key in a large public database, who they issue me a certificate.
  - Mark Cummins owns key 343434343

Digital Signatures



Same But Different

- So what is the difference between a Hash, A MAC and Signature?
- A hash use a standard algorithm (MD5 or SHA for example) to create a hash. Anyone can recreate it.
- MAC's (message authentication codes) are sometimes called keyed hash functions. They create a hash based on the document and a shared secret key (like symmetric encryption, a shared secret key).
- However MAC's do not constitute evidence that a third party could use to decide whether A or B sent a particular message.
- A signature uses a keyed hash function similar to a MAC but the key is unique to a user. So it can be uses by a third party as evidence

Digital Signature Algorithms

- There are several algorithms that can be used for digital signatures. However the two most common are
  - RSA with MD5
  - DSA with SHA1
- A digital signature algorithm consists of a key generation algorithm, a signature algorithm and a verification algorithm.

Digital Signature Algorithms

- Digital signature algorithms support non-repudiation. And are the main legal component of digital commerce.

- Digital signature schemes need not be invertible, where as an encryption scheme needs to be invertible. So you should draw a clear distinction between digital signatures and public key encryption schemes.

El Gamal and DSA

- The El Gamal signature scheme was published in 1985
- There have been many other (stronger) schemes based on the original algorithm.
- In 2000, the NSA derived a version that they used as the Digital signature Algorithm (This is basically a tweaked version of El Gamal)
- The ECDSA is similar to DSA but works with points of an elliptic curve instead of integers modulo some prime.

Public Key Infrastuctures

- Up till now we have seen how different algorithms rely on the use of public/private key pairs. But how do we know that the keys we are using belong to the right party?
- This is a critical problem in public key cryptography
- In 1976 Diffie and Hellman envisaged a public directory where everyone could look up the public key of users, just like a phone book.
- This was implemented as a student project in 1978 by Kohnfelder, and he coined the term certificates for each of the records (signed version of name and key)

Certificate Authorities

- The binding between a user and their key (and whatever other information) is established by the party who issues the certificate.
- This party is known as the issuer or certificate authorities (CA)
- The CA has to protect it own private key, as they may have to verify users certificates.

Certificate Authorities

- At the lowest level a CA may just check that the person name is unique
- At he highest levels the person would have to appear in person with legal ID
- Public Key Infrastructure (PKI) is the somewhat imprecise term used to describe the system for issuing and managing certificates.

X.509/PKIX certificates

- Today X.509 (version 3) is the most widely used PKI standard.
- It was originally part of the X.500 directory
- X.500 was intended as a global, distributed database of named entities and the X.509 certificates would hold their public keys
- PKIX is the internet X.509 PKI

Certificate Chains

- To check a certificate is valid, another verification key is needed.
- This creates a certificate chain
- Ultimately you need a root verification key whose authenticity cannot be guaranteed by a certificate.
- Typically a set of root verification keys are stored in are stored in browsers/ mail programs.

Certificate Chains

- When you get a message like "you do not trust this certificate" then there is no chain root recorded in your system.
- In a system that requires certificates for signature verification you can use self-signed certificates to store root keys
- Certificates also have expiry dates, and how we handle expired certs is a policy decision.

Revocation

- A certificate may have to be revoked if a corresponding private key has been compromised.
- Certificate revocation lists (CRLs) are distributed at regular intervals or on demand.

Electronic Signatures

- There have been numerous efforts to integrate electronic signatures into legal systems.
- The EU electronic signature directive (1999/93/EC) states that electronic signatures have to be implemented as digital signatures.
- They are some extra requirements on the strengths of the certificates, and on the CA.