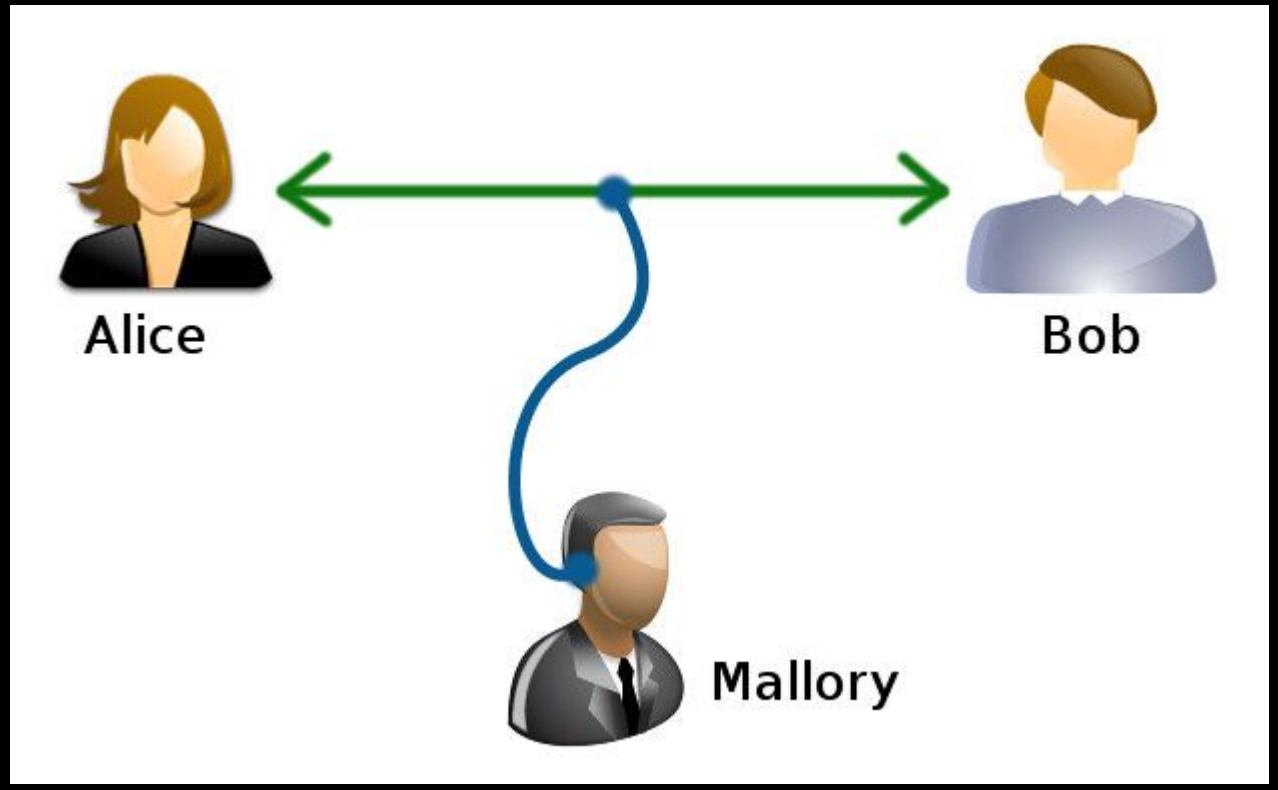
SIGPwny

Embedded FA2025 • 2025-10-06

Secure Protocol Design

Nikhil Date

Obligatory Alice and Bob Slide



Security Properties

- CIA Triad: Confidentiality, Integrity, Availability
- We will mostly be focusing on confidentiality and availability
- Confidentiality: can Mallory read secrets exchanged between Alice and Bob
- Integrity: can Mallory tamper with messages sent between Alice and Bob?
- Can you think of a system you have used where these properties are important?

Threat Models

- Passive attacker (“Eve”)
- Active attacker/man-in-the-middle (“Mallory”)
- Parties themselves are untrusted

Cryptography

- Mathematical and algorithmic techniques to achieve provable security
- Modern methodology of security
  - Define what security means
  - State assumptions
  - Define cryptographic construction
  - Prove that construction satisfies security definition
- Today, we will be looking at three topics within cryptography: symmetric cryptography, asymmetric cryptography, and hash functions

Symmetric Cryptography

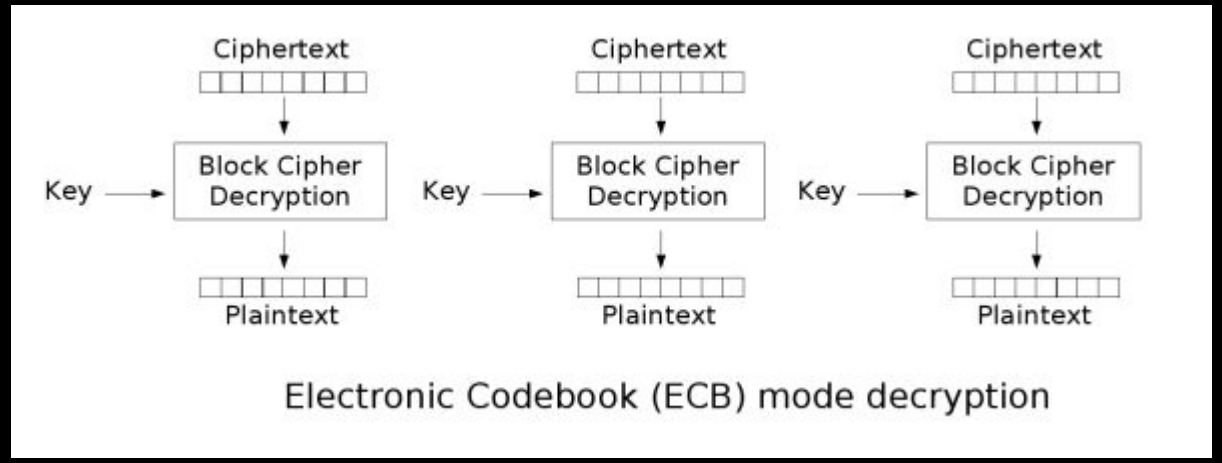
- Alice and Bob share a secret key (we won’t worry about how this happened, perhaps they met in advance)
- Can they achieve confidentiality and integrity given this?
  - Yes!
  - Confidentiality: Symmetric Encryption
  - Integrity: Message Authentication Codes

Symmetric Encryption

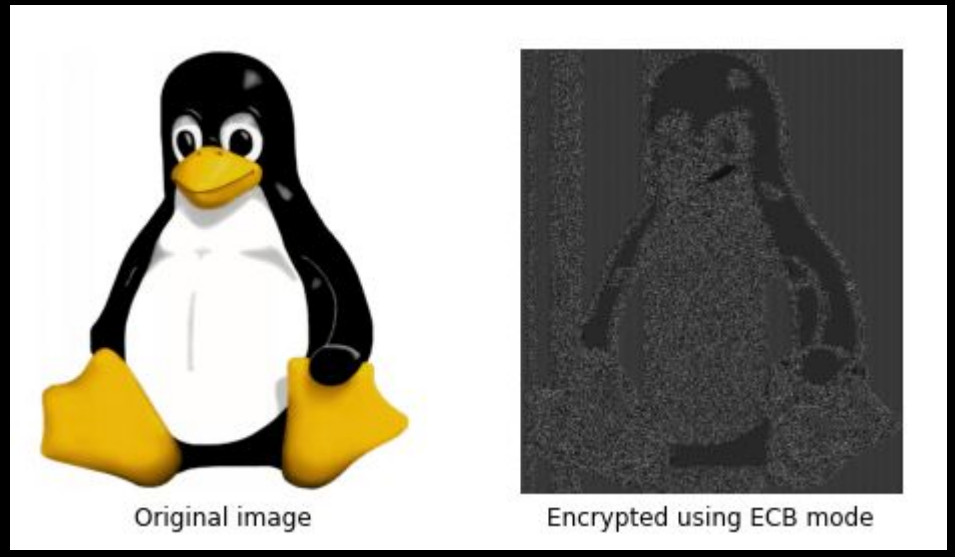
- Same key used to encrypt and decrypt
- Enc(k, m) -> c
- Dec(k, c) -> m
- Security property is “CPA” security
  - I choose a secret key and let you obtain encryptions of any messages of your choice
  - Now I let you pick two messages m_0 and m_1
  - I randomly choose one of these messages to encrypt and give you the ciphertext
  - You have no better way than brute force of guessing which message I encrypted
  - This implies that encryption must be randomized! Why?

Block/Stream Ciphers

- In practice, we typically use block ciphers or stream ciphers to achieve symmetric encryption of messages of arbitrary length
  - AES (with different modes)
  - ChaCha20
- Block ciphers need “modes” to encrypt long messages
  - Some modes are not secure (like ECB)!



Electronic Codebook (ECB) mode decryption


Original image

Encrypted using ECB mode

Message Authentication Codes

- tag(k, m) -> t
- check(k, m, t) -> {good, bad}
- We can use secret key to assign a “tag” to a message
- Other party can check the tag if they have the key
- If message is tampered with, tag won’t match
- Security property: “unforgeability”
  - I let you obtain tags for as many messages as you want of your choice
  - It’s infeasible for you to forge a tag for a message you didn’t already ask me for a tag

Authenticated Encryption

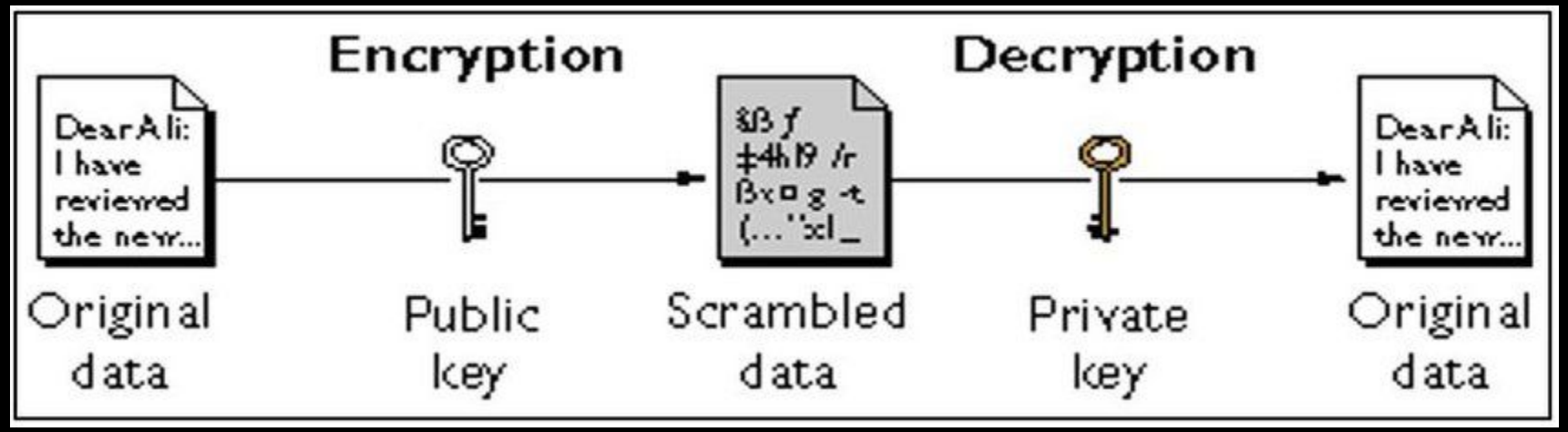
- If we “correctly” combine symmetric encryption and MAC, we get “authenticated encryption”
- This basically means that the adversary can do nothing
  - Can’t read messages
  - Can’t tamper with messages
- “Secure Channel”
- This is usually what we want to use when we’re dealing with symmetric cryptography
- There also exist specialized “authenticated ciphers” that combine confidentiality + integrity like AES-GCM, Ascon, etc.

Asymmetric Cryptography

- We assume initially that Alice and Bob have a shared secret
  - How did they get this secret?
  - What if they have no way to meet in advance?
  - What if we want multiple people to securely send messages to Alice?
- We can use asymmetric cryptography
  - Each party holds a public key pk and a secret key sk

Asymmetric Encryption

- Enc(pk, m) -> c (anybody can encrypt)
- Dec(sk, c) -> m (only holder of secret key can decrypt)
- Security definition is similar to symmetric encryption
- Can you think of a system you use where these are needed?



Digital Signatures

- Asymmetric equivalent of message authentication codes
- Sign(sk, m) -> sigma
- Verify(pk, m, sigma) -> {0, 1}
- Only the holder of the secret key can sign messages, but anyone can verify
- Can you think of a system you use where they are needed?
- Can this be combined with asymmetric encryption?
  - How?

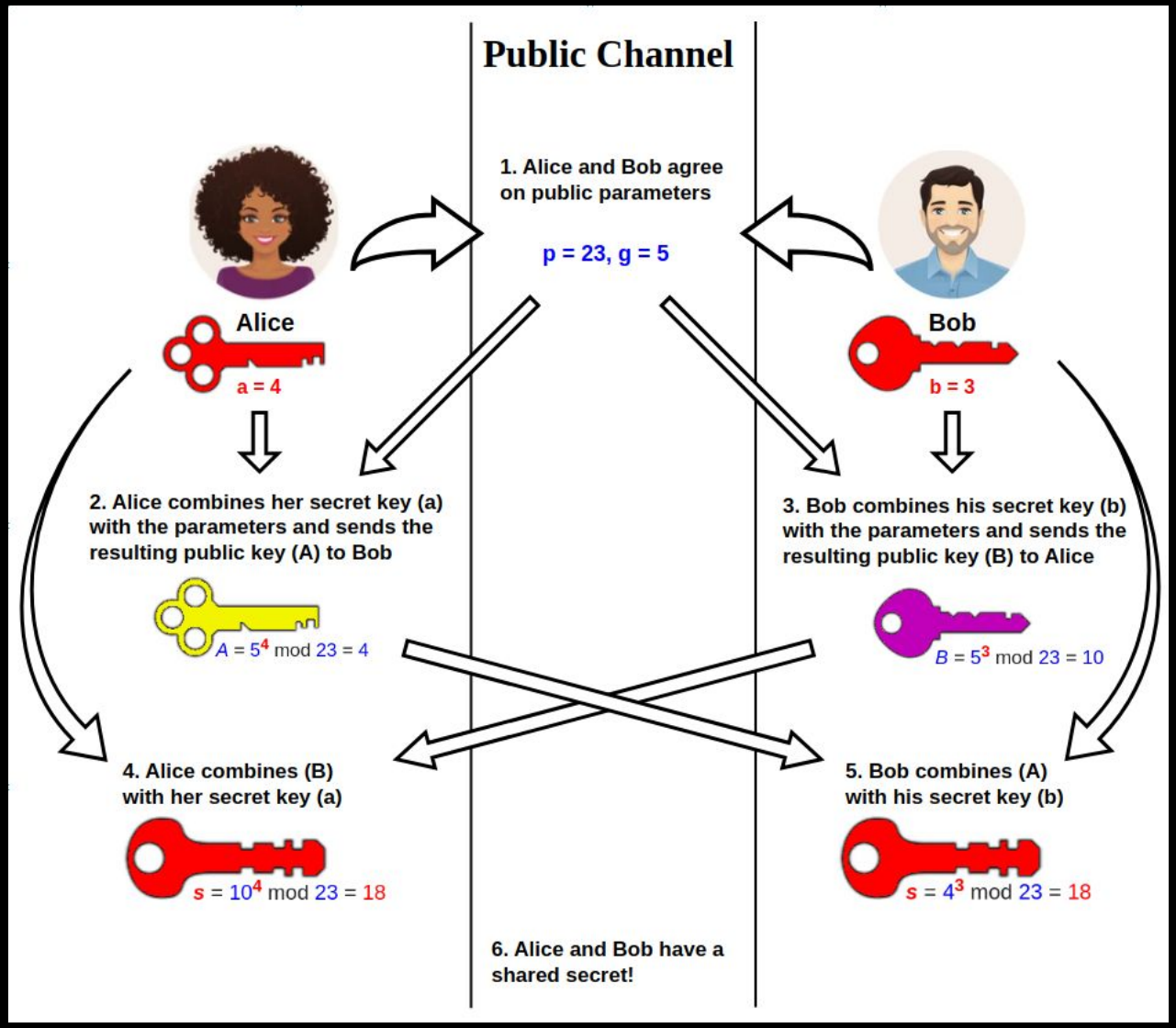
Asymmetric Crpytography Algorithms

- RSA
- Elliptic Curves
  - Generally best practice to use this
- Usually require longer key lengths and are slower

Key Exchange

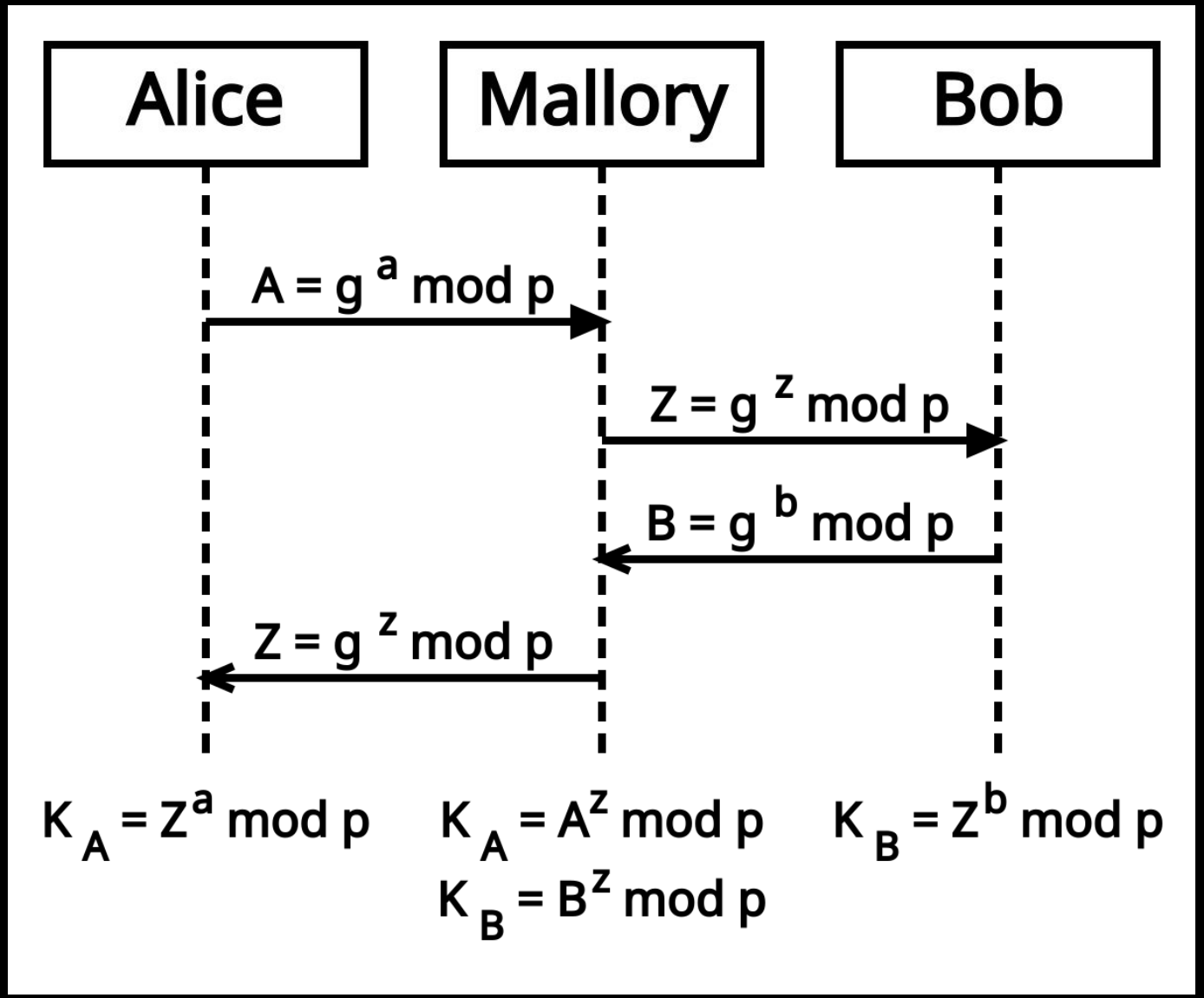
- At a high level, use asymmetric cryptography to establish a shared symmetric key
- One option
  - Encrypt symmetric key and send to other party
- Another option is Diffie-Hellman key exchange

Diffie-Hellman



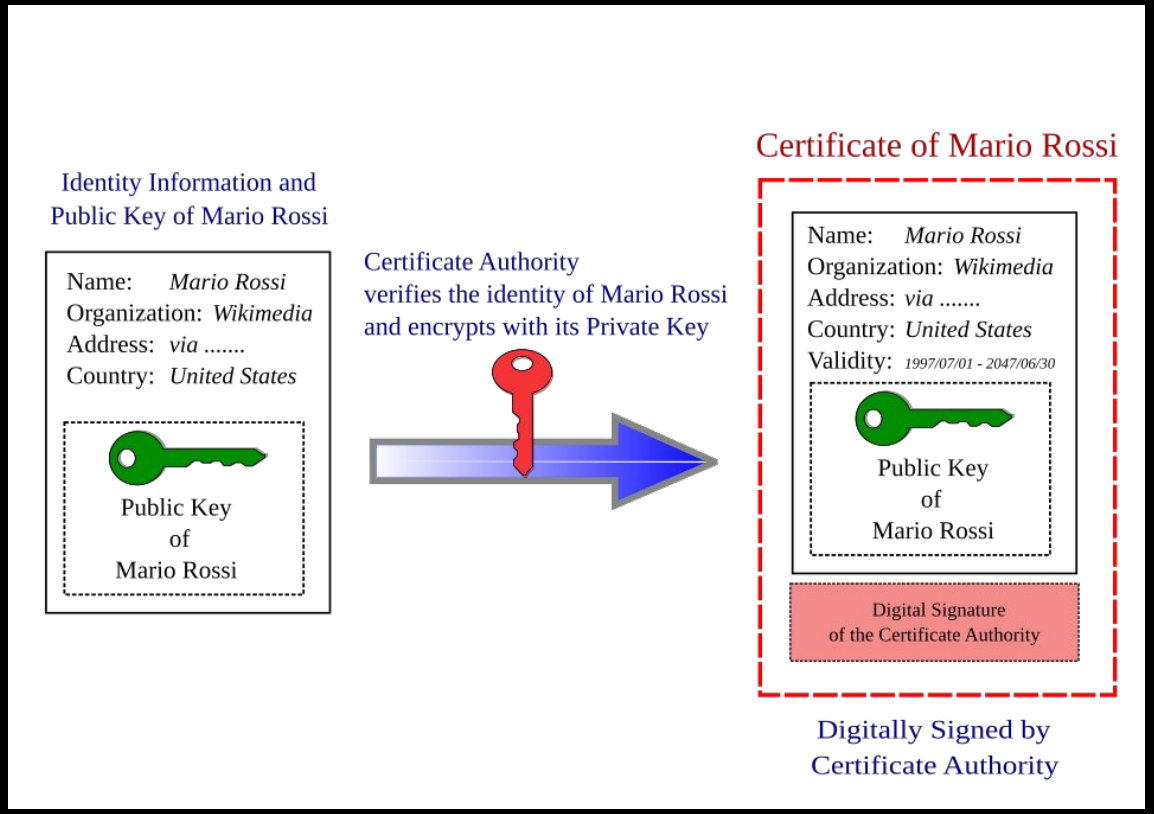
Man-in-the-middle attacks

-



Certificates

- Signed public key



Cryptographic Hash Functions

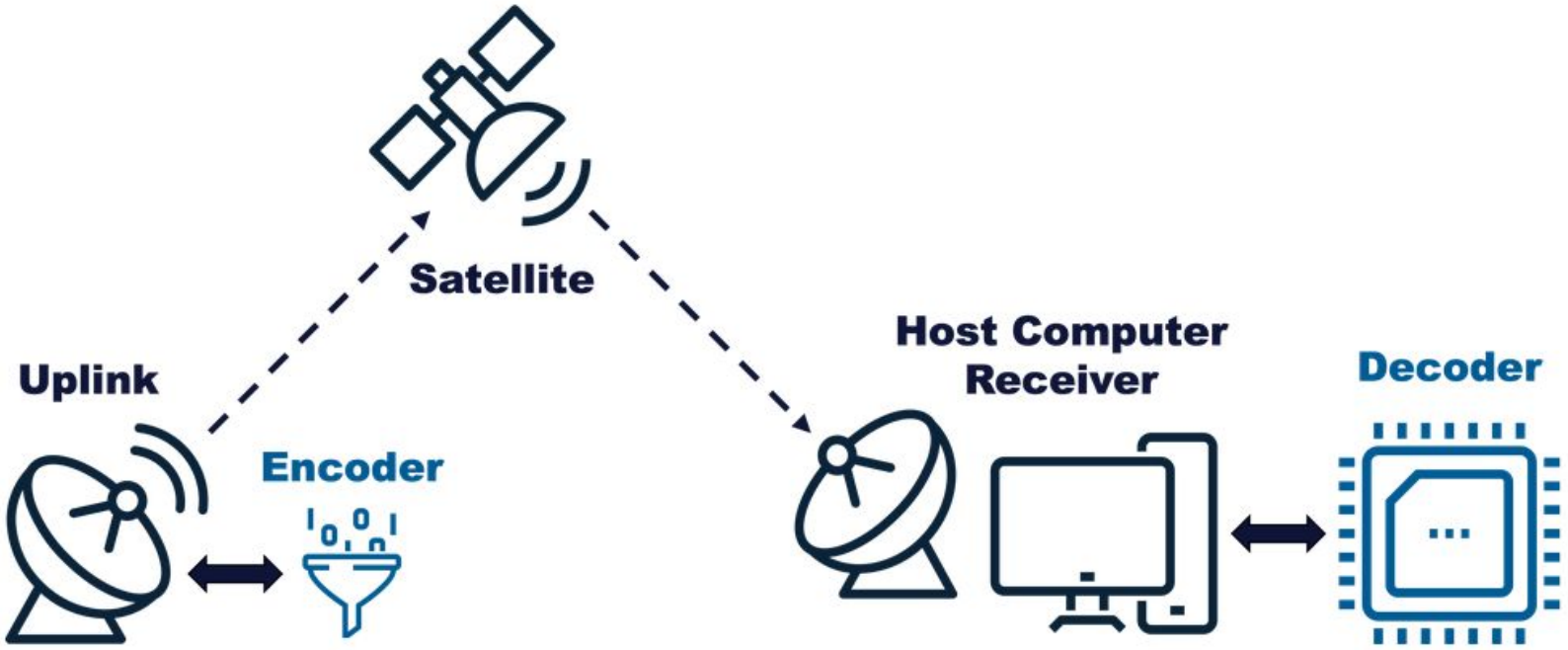
- H(m) = x
- Important: no secret keys!
- Can roughly think of it as a “pseudorandom” function
- Properties
  - Preimage resistance: given H(m), can’t figure out m
  - Collision resistance: can’t find m_1, m_2 such that H(m_1) = H(m_2)
- Uses
  - “Commitment scheme”
    - I pick a secret x
    - commitment = H(x || r)
    - later reveal secret (“open”) by sending x and r
    - Hiding: commitment doesn’t reveal x
    - Binding: can’t claim to have chosen a different secret

More Complex Attacks

- Replay attacks
  - Might be able to replay encrypted + authenticated message
- What if keys are compromised?
  - Is everything lost?
  - Can we finely scope keys to limit damage?
  - Forward Secrecy: past communications are safe even if keys are compromised
    - Ephemeral Diffie-Hellman uses “long-term secret” to generate fresh public keys for each session

eCTF 2025 Practice

Your team's design will consist of an Uplink and Encoder streaming data to a Satellite, a Host Computer, and a hardware Decoder. The Decoder firmware you design will securely decode TV frame data streamed over a satellite's unidirectional data stream. The image below shows a high-level overview of the system architecture.



High-Level Satellite TV System

Next Meetings

`2025-10-13` • **Next Monday**

- Embedded PWN and Software Security

Meeting content can be found at sigpwny.com/meetings.

SIGPwny