An Overview of Cryptography

Presented by *Logan Zeien*

About Me

Logan Zeien

Education

- University of Colorado at Colorado Springs
  - Major in B.S in Computer Science with Cybersecurity Emphasis
- CompTIA Security+ CE, CCNA

Experience

- Webmaster | CABMET
- Associate System Engineer – Cybersecurity | Northrop Grumman, Deep-Space Advanced Radar Capability (DARC) Contract
- Publication: *Characterizing Advanced Persistent Threats (APTs) through the Lens of Cyber Attack Flows*, presented at HammerCon 2024.

Roadmap

- What is Cryptography
- Background
- Primitives
  - Symmetric Key Encryption Scheme
  - Asymmetric Key Encryption Scheme
  - Hash Function
  - Message Authentication Codes
  - Digital Signature
- Answer the Question
- Take Aways

What is Cryptography?

- Cryptography is NOT Cartography (study of map making)
- Cryptography is the **science (since early 50s)** of secure communication / secure data
- Rigorously, definitionally, and mathematically proven algorithms

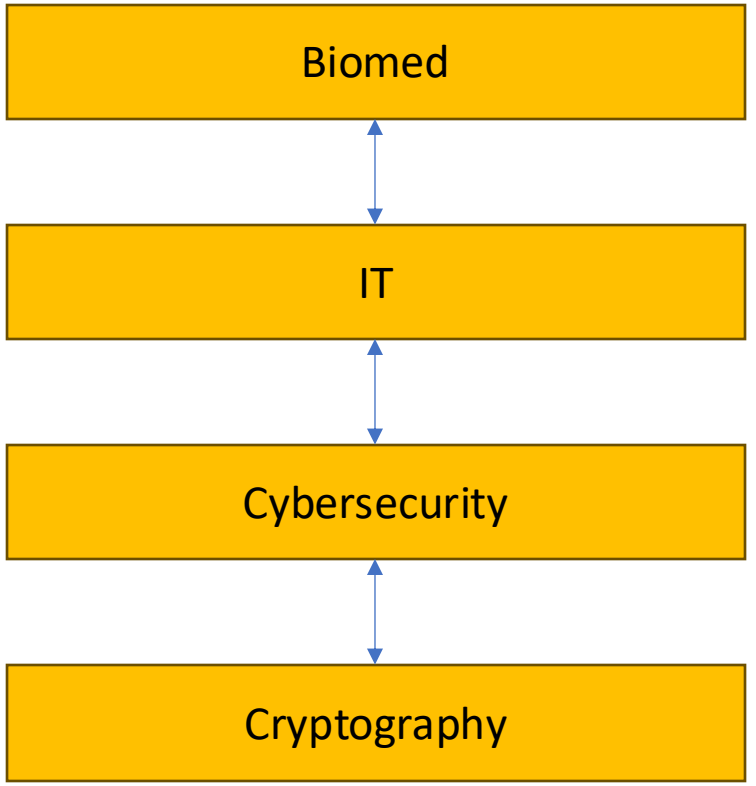
What is secure?

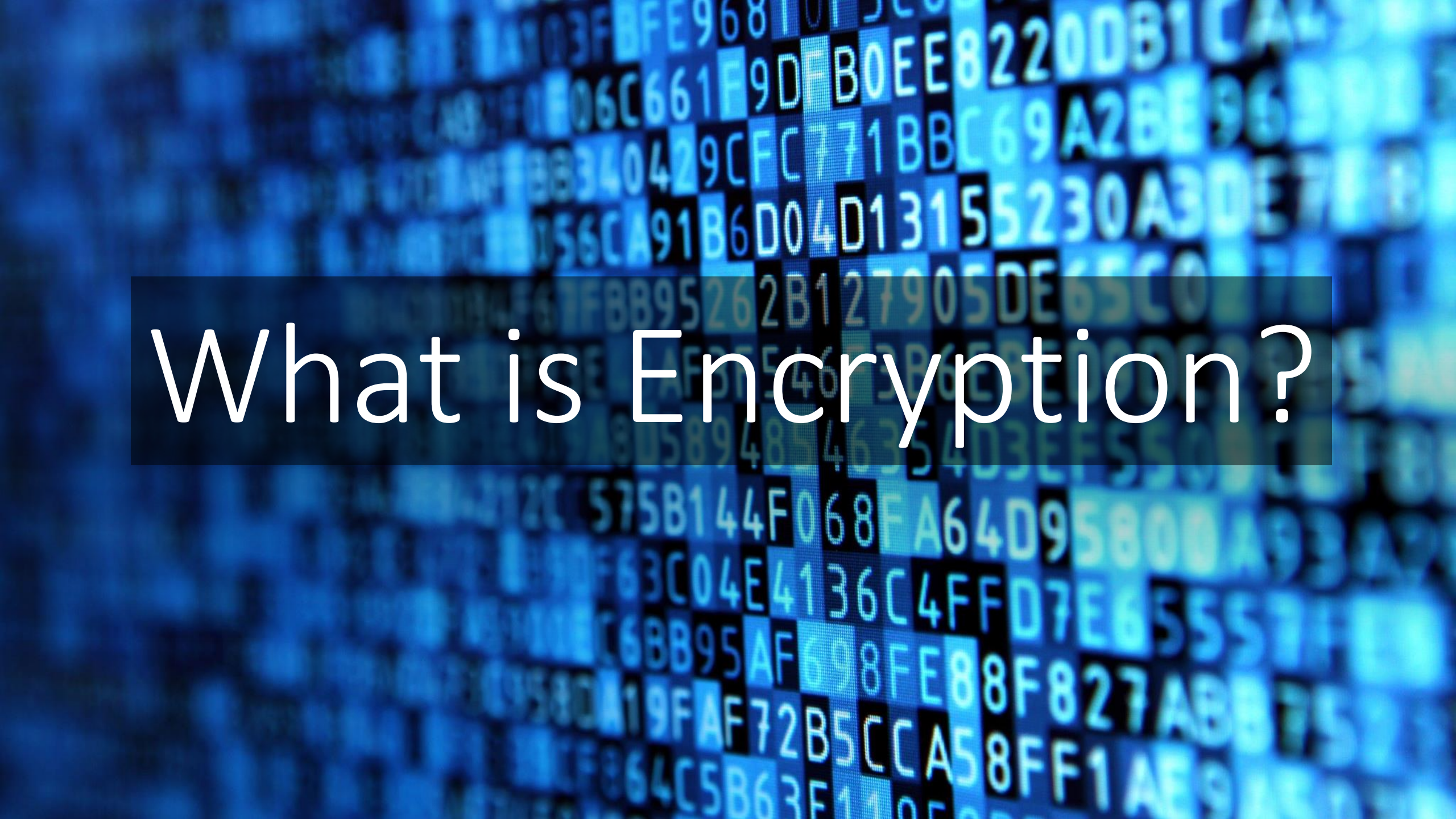
- Confidentiality
- Integrity
- Authenticity
- Non-Repudiability
- Accountability
- Private Computing

Our Focus

Cybersecurity is ever more important!

You use Cryptography everytime you:	
Use a VPN	IPSec
Connect to a website	TLS/SSL
Type a password	Hashes
Sign an email	Digital Signature
Badge In (sometimes)	PKI/Digital Certificates

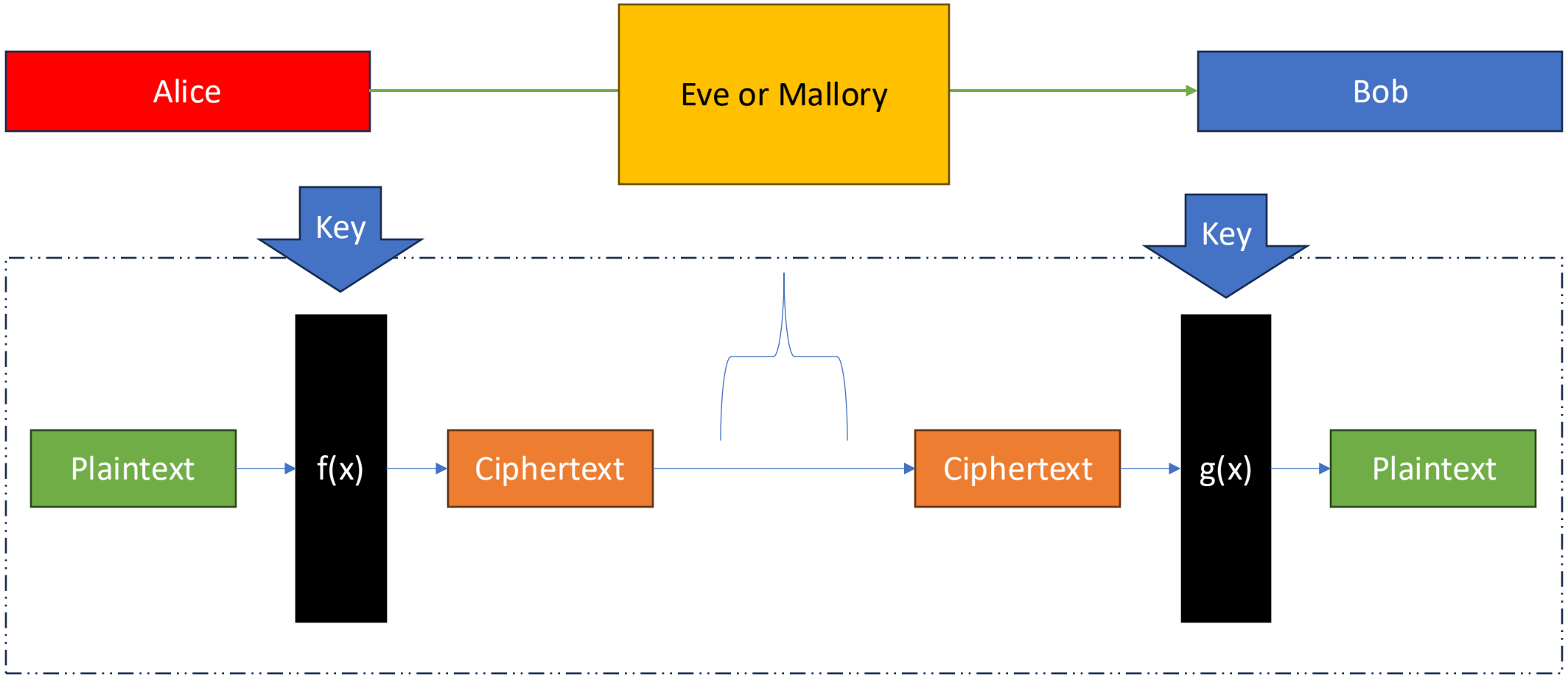



What is Encryption?

Roadmap

- What is Cryptography
- Background
- Primitives
  - Symmetric Key Encryption Scheme
  - Asymmetric Key Encryption Scheme
  - Hash Function
  - Message Authentication Codes
  - Digital Signature
- Answer the Question
- Take Aways

Terminology



Kerckhoff's Principle

Everything about a cryptosystem, except its key, is publicly known.

Why?

What happens if…	then…
A secret algorithm is used?	Semi-trusted users (i.e., YOU) cannot use the algorithm + recoveries of the 'algorithm' can be reversed (think enigma machine)
A different language is used (Navajo Code Talkers)?	Only 1 defective agent required to completely break communications

Onto the Primitives

- What is Cryptography
- Background
- Primitives
  - Symmetric Key Encryption Scheme
  - Asymmetric Key Encryption Scheme
  - Hash Function
  - Message Authentication Codes
  - Digital Signature
- Answer the Question
- Take Aways

IMPORTANT NOTE BEFORE BEGINNING!

- Schemes mentioned are **building blocks** for encryption
- Never design a cryptographic scheme yourself
- Always use proven and regularly implemented schemes (e.g. TLS, IPSec)

Symmetric Key Encryption Scheme (SKES)

Confidentiality, Integrity, Authenticity

Intuition Derivation

I have no idea what he said

How can I talk securely to the star destroyer?

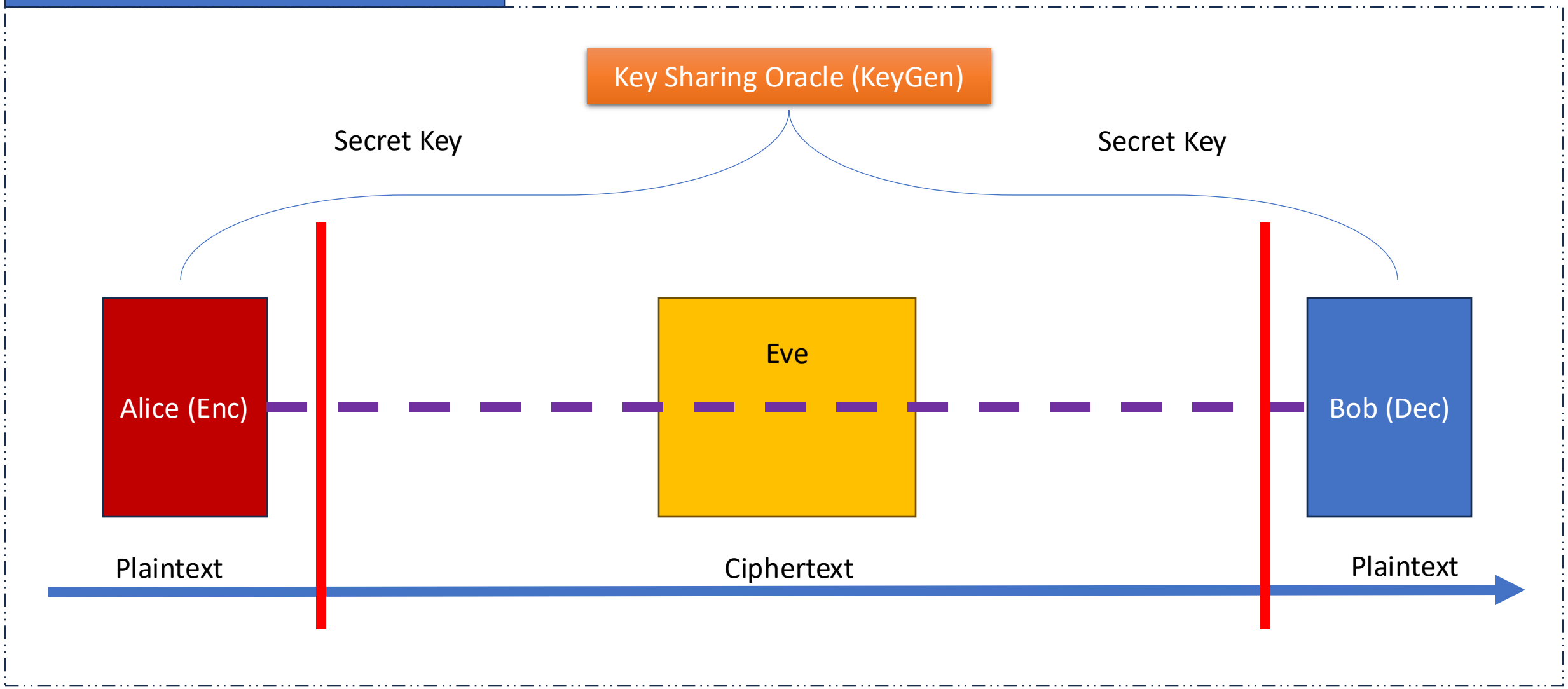
Those were the droids we ~~[illegible]~~

F067e95df2ccba51576349c45a379fa15ab26cbbd1827d87c3bed0…

How can Alice securely send Bob a message over a compromised medium with pre-distributed keys?

How can Alice securely send Bob a message over a compromised medium with pre-distributed keys?

System Model



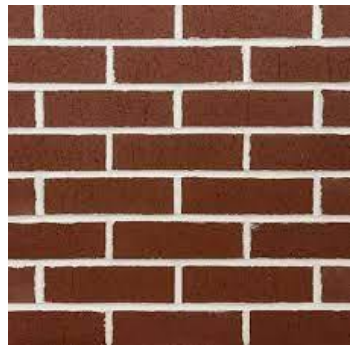
How can Alice securely send Bob a message over a compromised medium with pre-distributed keys?

Block Ciphers

How can Alice securely send Bob a fixed length message over a compromised medium given pre-distributed keys?

Modes of Operation

How can we combine block ciphers to encrypt arbitrary lengths of text?



Block Cipher

How can Alice securely send Bob a fixed length message over a compromised medium given pre-distributed keys? (Block Cipher)

Real World Example

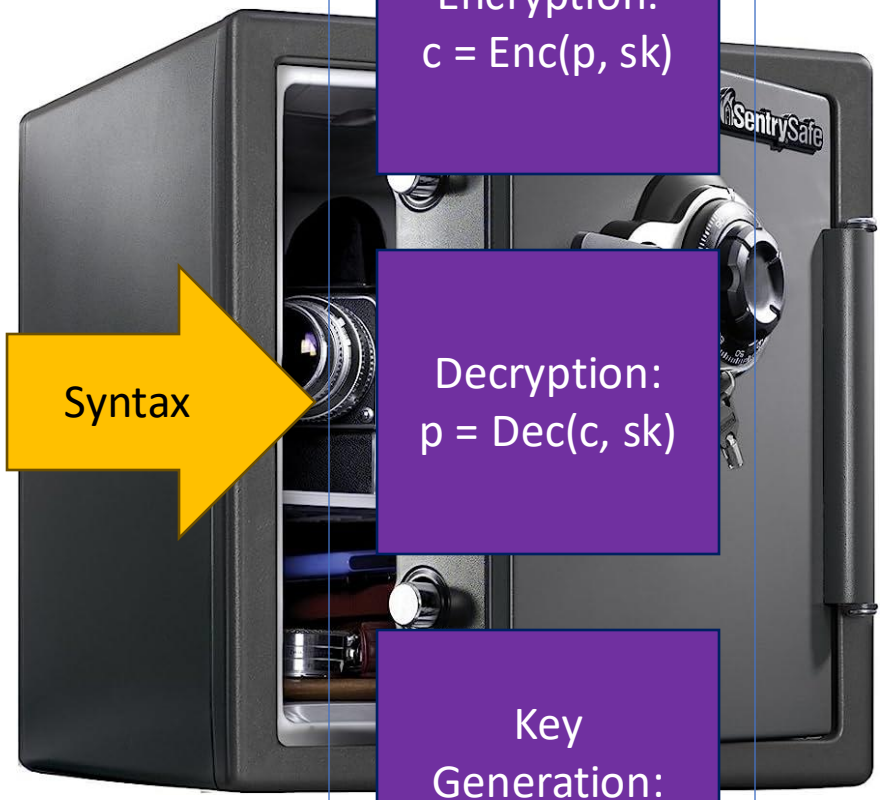
Locking Mechanism

Unlocking Mechanism

Key Making Process

Locked Box Example

Syntax



Encryption:
c = Enc(p, sk)

Decryption:
p = Dec(c, sk)

Key Generation:
sk = Key Gen(n)

What's n?

SKES

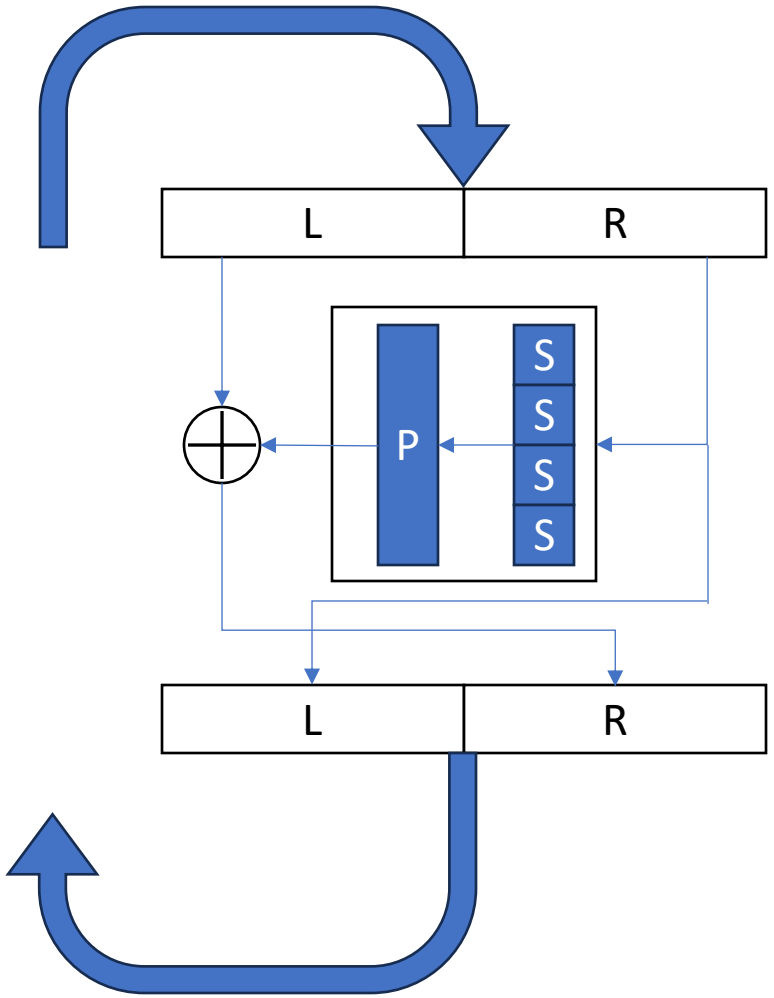
Block Cipher

Block Cipher Architectures

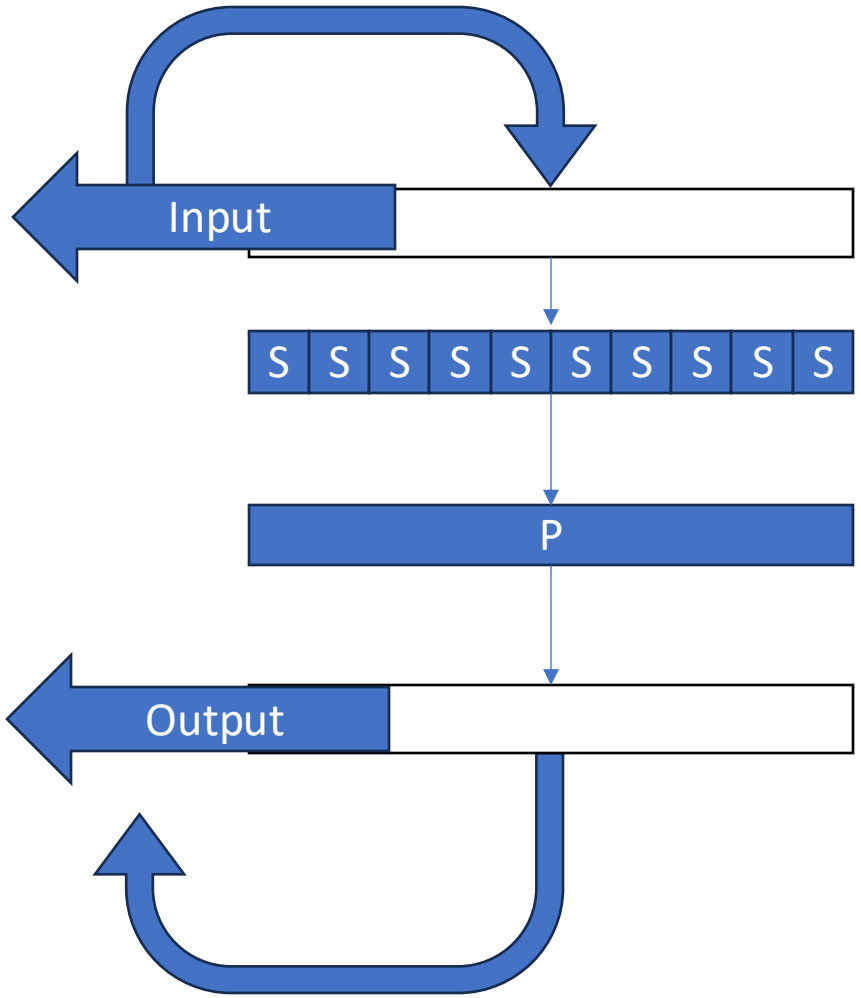
S = Substitution

P = Permutation

- All provably secure Block Ciphers are created from either the Fiestal Cipher or the S/P Network.
- Repeats for 10-30 rounds depending on the cipher
- Examples:
  - Twofish, Serpent, **AES (commonly used)**



Fiestal Cipher



Substitution Permutation Network

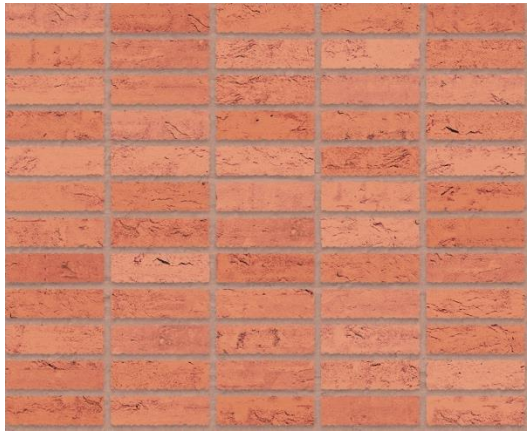
Modes of Operation

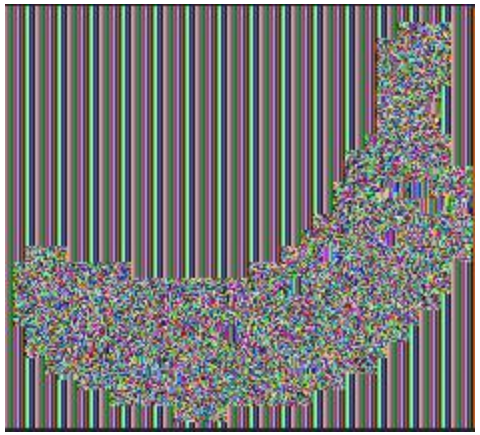
- If Block Ciphers are bricks, Modes of Operation are like stacking them
- **How do you effectively stack bricks?**

Structure of encrypting multiple "bricks" matters greatly!

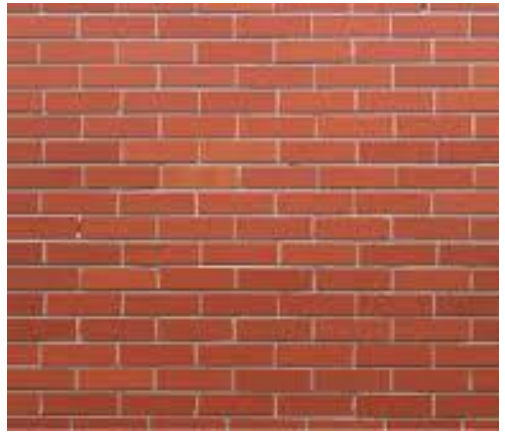
Enc(x)

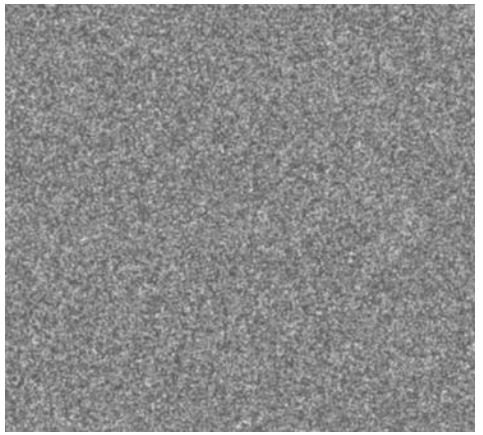
Naïve Approach





We need this!





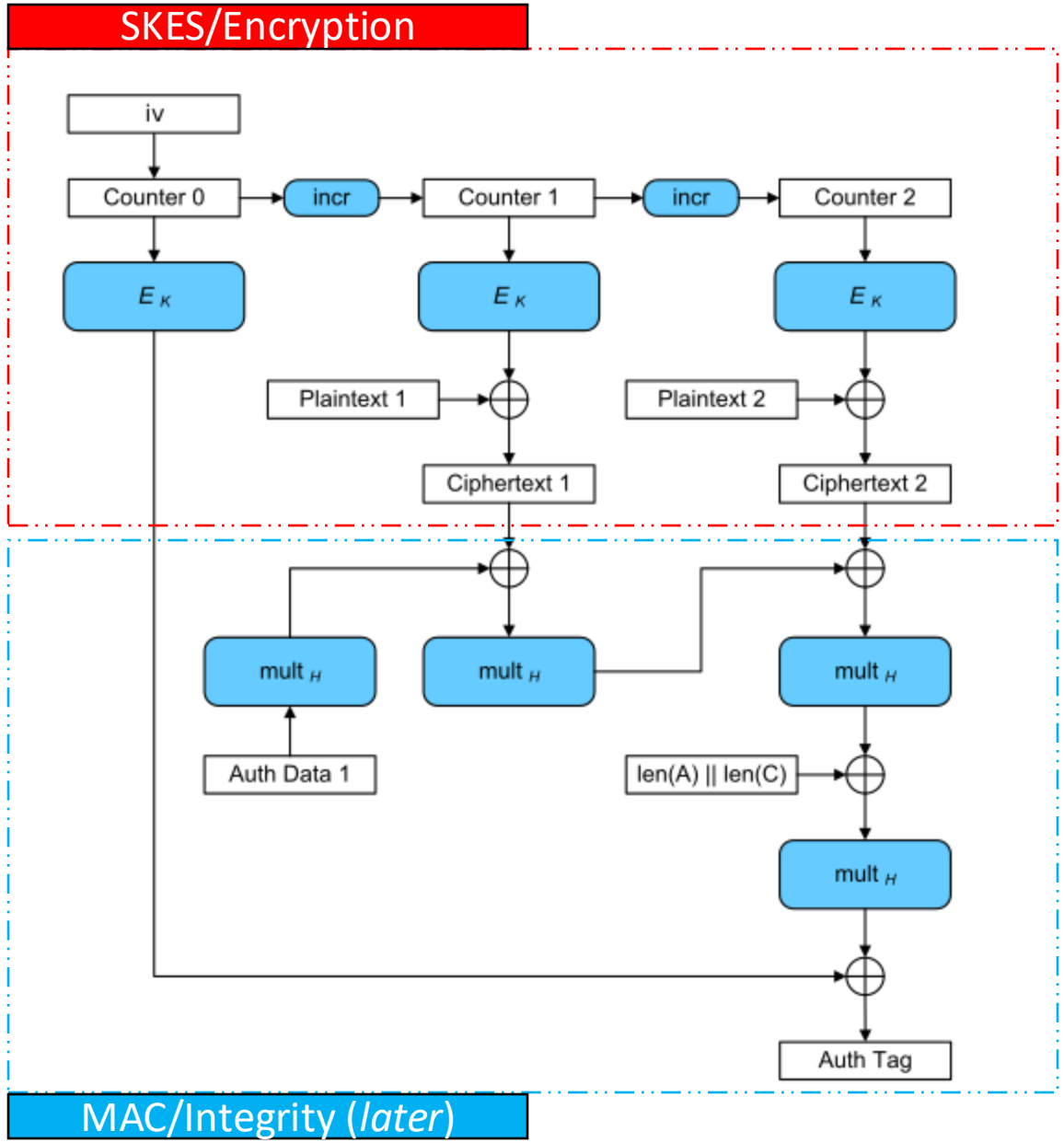
Modes of Operation

- Multiple modes of operation:
  - Electronic Code Book (ECB)
  - Counter Block Chaining (CBC)
  - **Galois Counter Mode (GCM)**
    - GCM provides integrity and authenticity as well with a built in MAC! (*later)*

Modern SKES: AES-GCM

Encryption:
Incorporates an initialization vector (iv) to encrypt counters and XOR the keystream with the plaintext

Decryption:
Put the ciphertext in the place of the plaintext

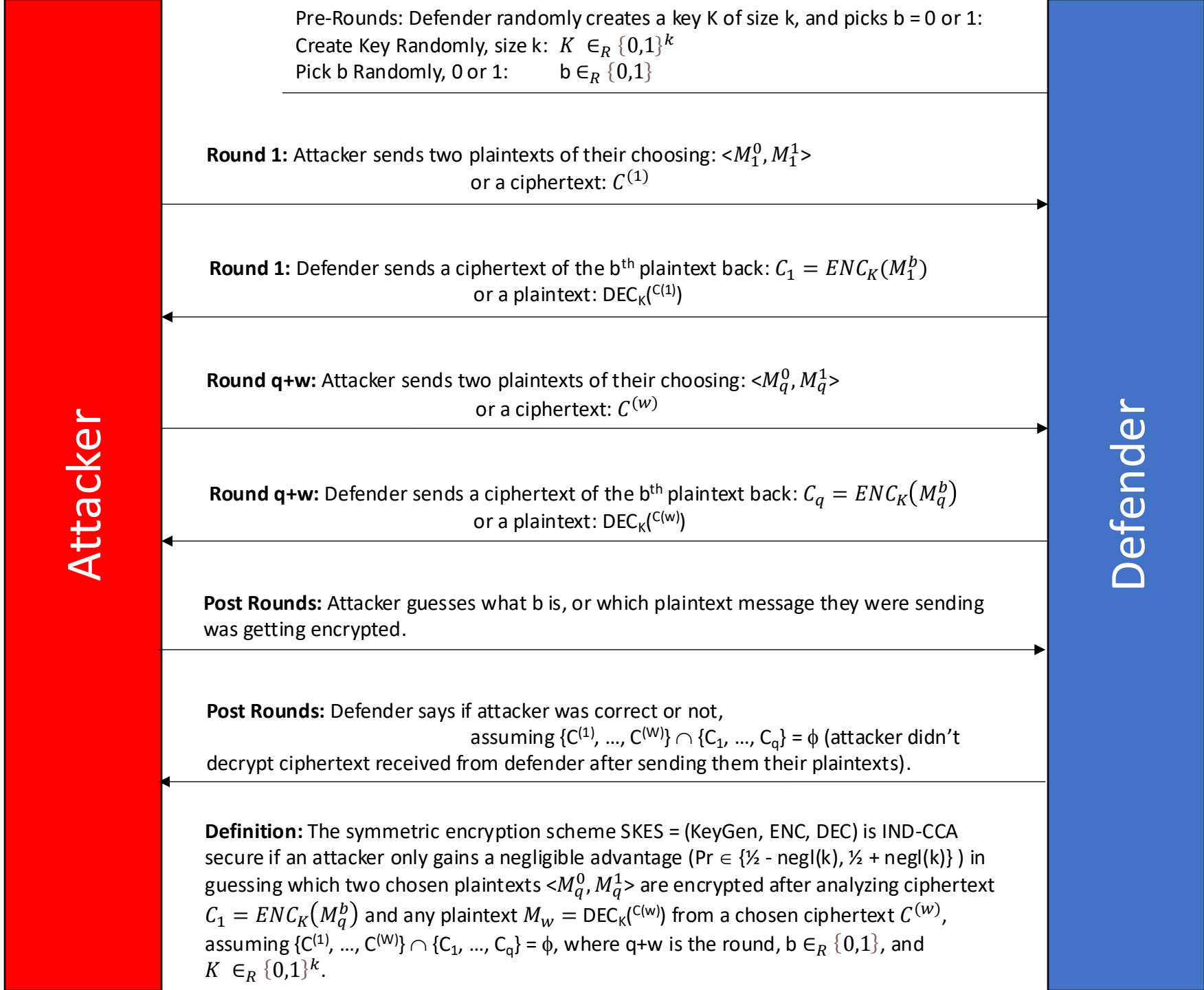


(Picture from Wikipedia: https://en.wikipedia.org/wiki/Galois/Counter_Mode)

Symmetric Key Encryption Scheme (SKES)

Security Definition: IND – CCA for SKES

Confidentiality, Integrity, Authenticity



Onto the Primitives

- What is Cryptography
- Background
- Primitives
  - Symmetric Key Encryption Scheme
  - Asymmetric Key Encryption Scheme
  - Hash Function
  - Message Authentication Codes
  - Digital Signature
- Answer the Question
- Take Aways

Asymmetric Key Encryption Scheme (ASKES)

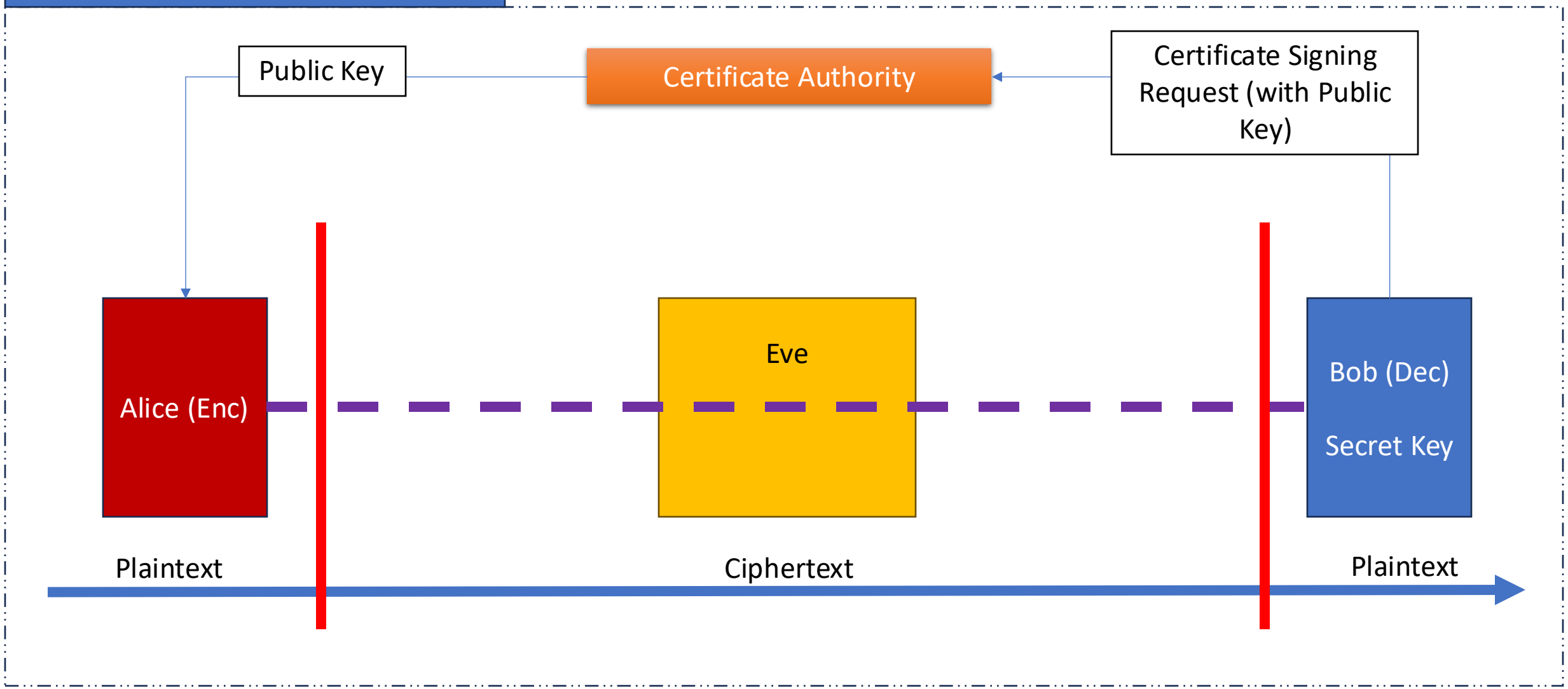
Confidentiality, Integrity, Authenticity

Intuition Derivation

How can Alice securely send Bob a message over a compromised medium with a publicly known key?

How can Alice securely send Bob a message over a compromised medium with a publicly known key?

System Model



How can Alice securely send Bob a message over a compromised medium with a publicly known key?

Real World Example

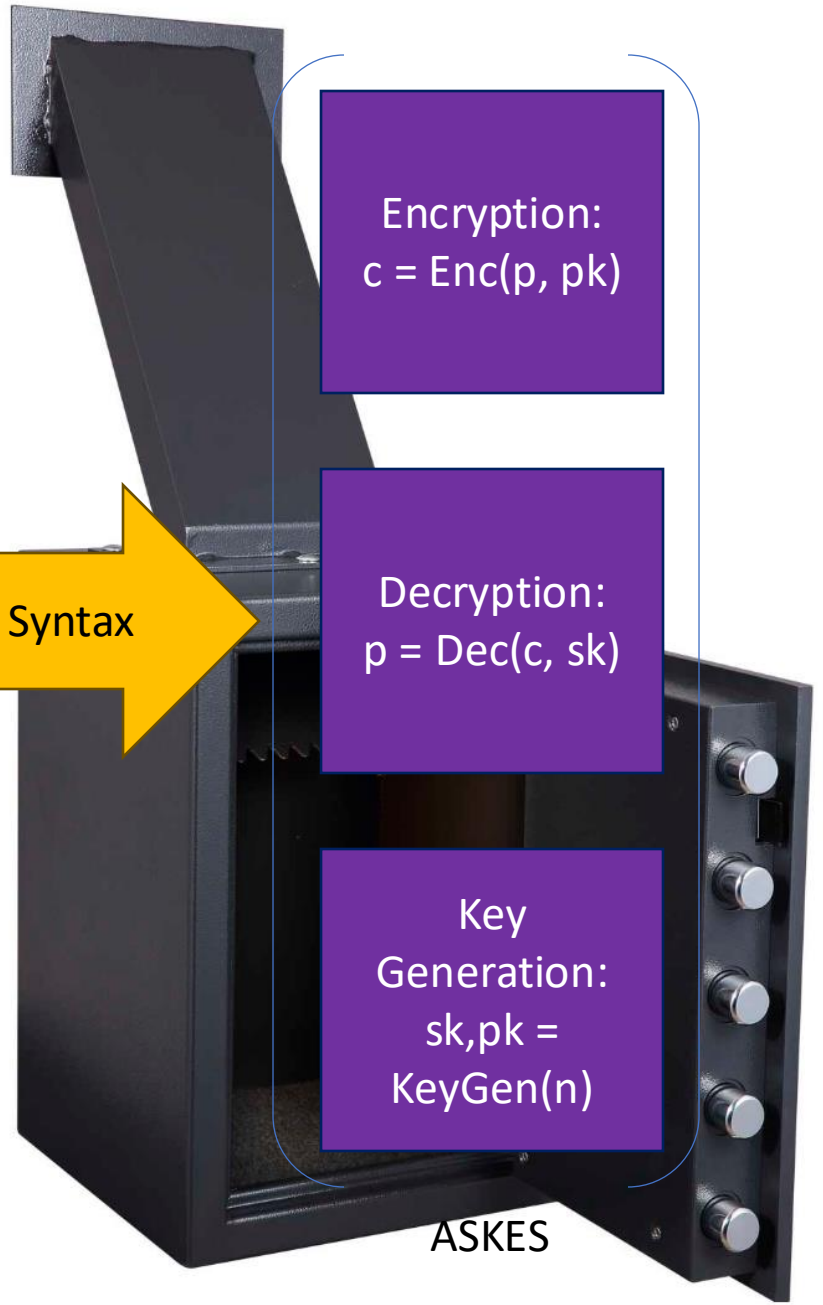
Public Locking Mechanism

Private Unlocking Mechanism

Key Making Process

Locked Box Example

Syntax



Encryption:
c = Enc(p, pk)

Decryption:
p = Dec(c, sk)

Key Generation:
sk,pk = KeyGen(n)

ASKES

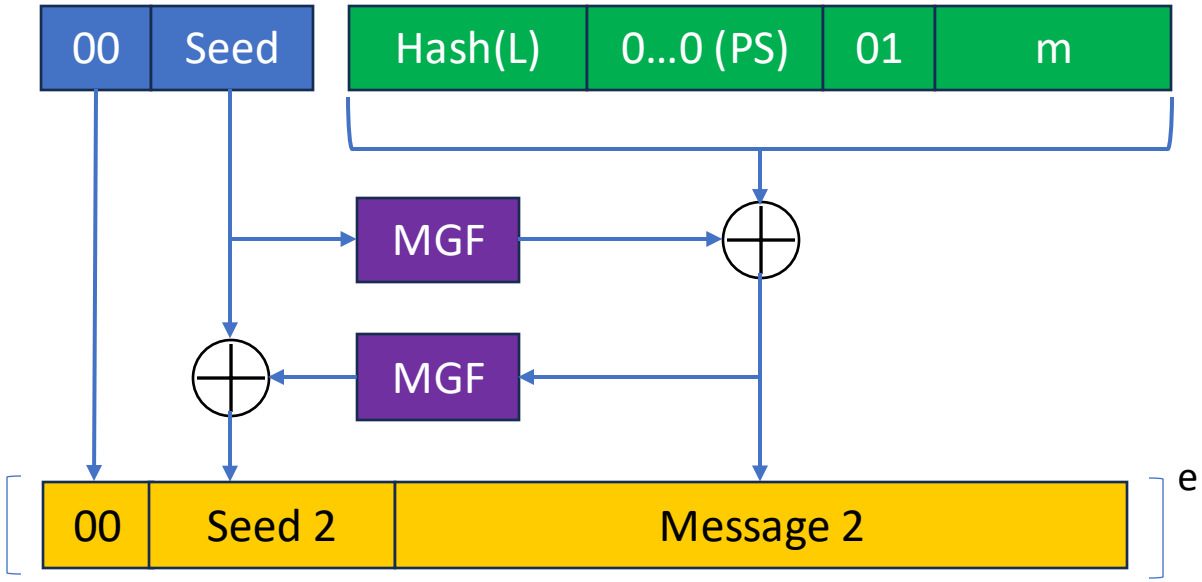
RSA

RSA Function

1. **KeyGen:**
   1. Generate large prime numbers p,q
   2. N = p*q
   3. $Z_n = \{a: a \in \{0, 1, \dots, N-1\}, GCD(a, N) = 1\}$
   4. $\phi(N) = (p-1)(q-1)$
   5. **e** s.t $GCD(e, \phi(N)) = 1$
   6. $\boldsymbol{d} = e^{-1} \, mod \, \phi(N)$
2. **Enc**: $\forall \, m \in Z_n, m^e \, mod \, N$
3. **Dec**: $\forall \, c \in Z_n, c^d \, mod \, N$

Complicated? This is not even secure! We need OAEP (Optimal Asymmetric Encryption Padding).

RSA Cryptosystem & OAEP

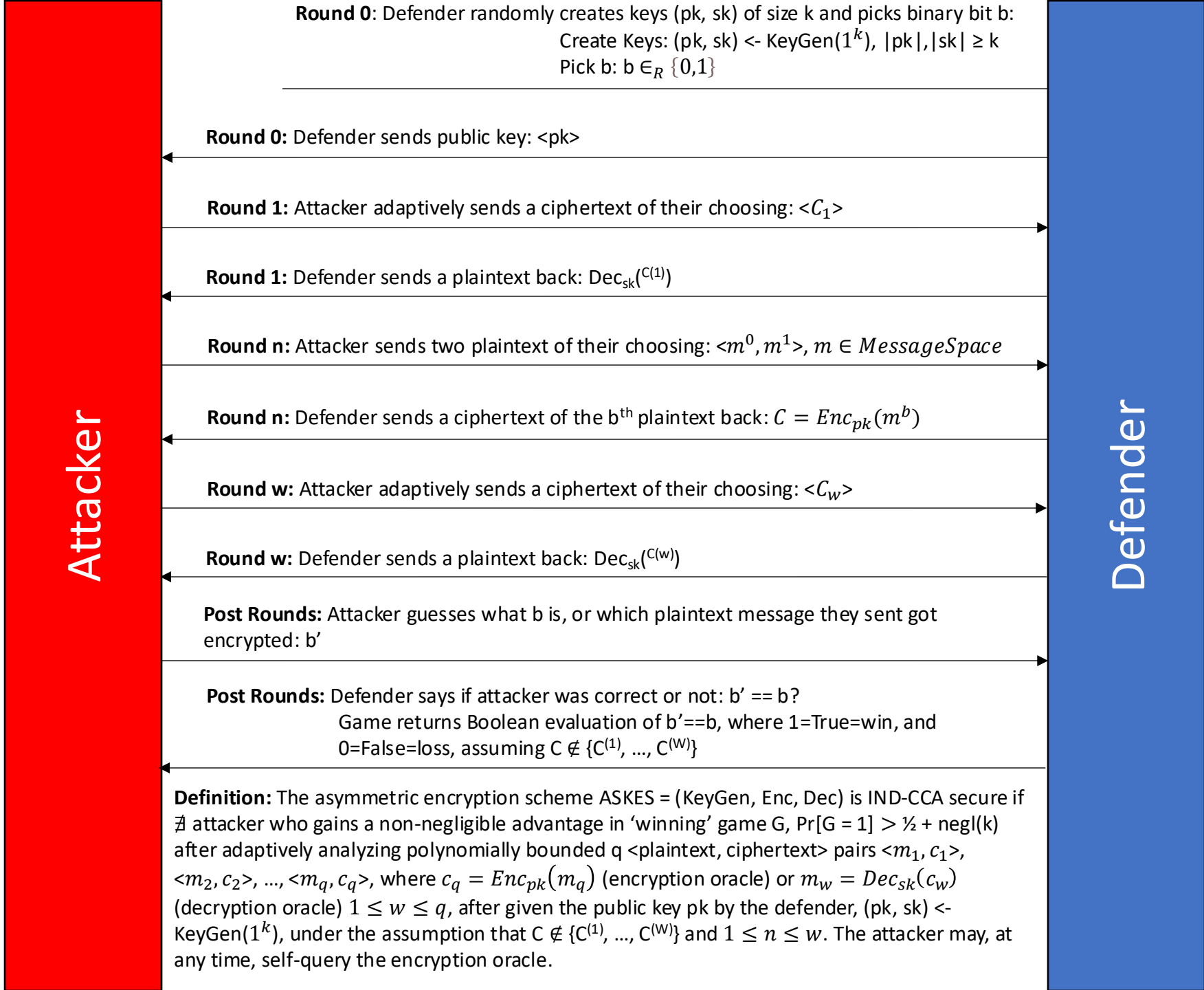


(Picture derived from https://en.wikipedia.org/wiki/Optimal_asymmetric_encryption_padding)

Asymmetric Key Encryption Scheme (ASKES)

Security Definition: IND-CCA for ASKES

Confidentiality, Integrity, Authenticity



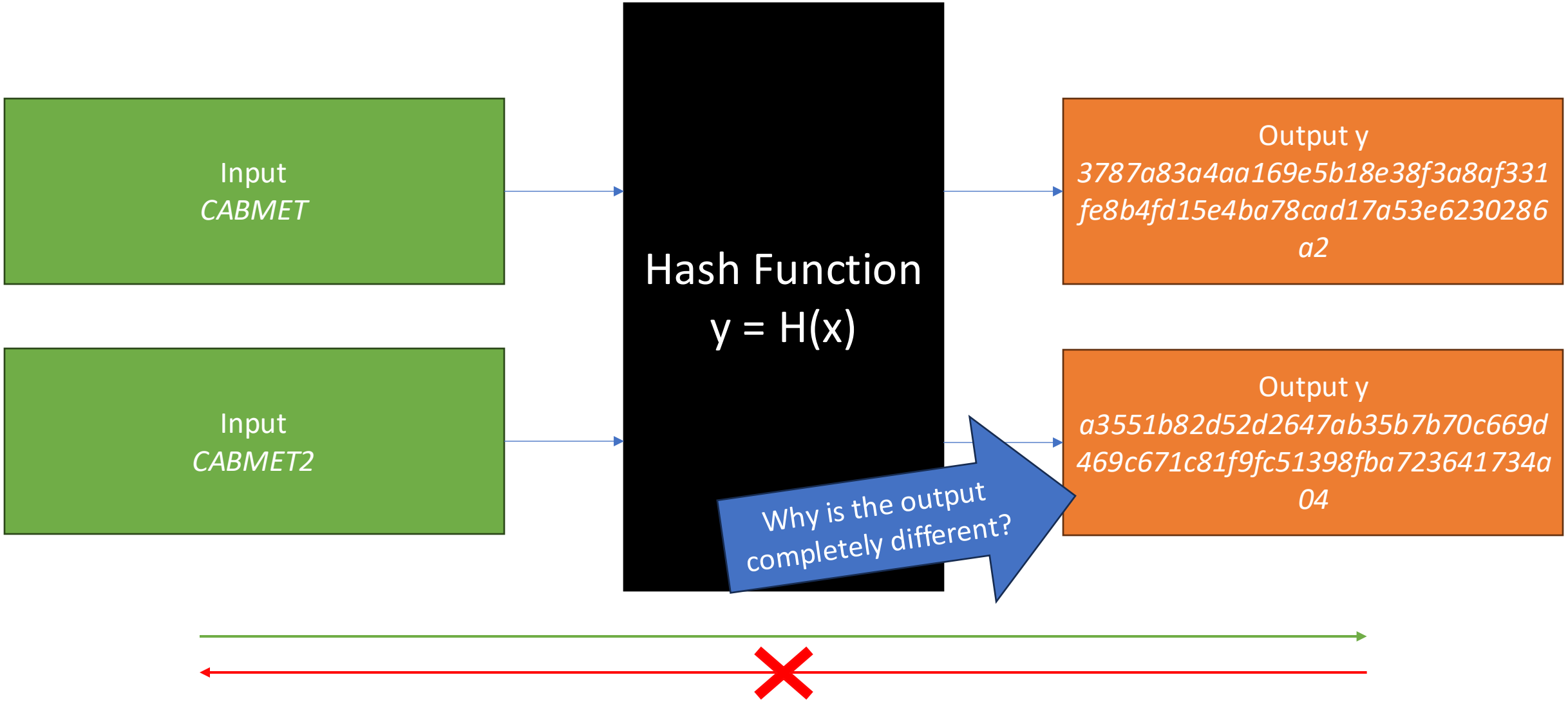
Quick Checkup

- What is Cryptography
- Background
- Primitives
  - Symmetric Key Encryption Scheme
  - Asymmetric Key Encryption Scheme
  - Hash Function
  - Message Authentication Codes
  - Digital Signature
- Answer the Question
- Take Aways

Hashes

- Checksums, verifying validity of files, blockchain, messages
- Protecting passwords
- Securely masking a value

System Model



$\nexists\ PPT$ to find x given y = H(x): Preimage Resistance

Applications – Password Storage

Naïve Approach

Username	Password
	Password123
	Password123
	123456

Why is this insecure?

Insecure Approach

Username	Encrypted Password
Logan	Ql3kjD67as9PiS2wT3
Cabmet	Ql3kjD67as9PiS2wT3
Administrator	Lk7GnO1pJt6YhI9dW2
Key: 6b8dca09e851a9870504	

Secure

Username	H(pwd,salt)	Salt
Logan	d7c4e6a038f67	salt
Cabmet	5baa75029050	fuijosho
Administrator	c4b95cb0f8c5b	Y#*()H

```
root@test:~# cat /etc/shadow
root:$6$j1WDO017$I6vMtiJIa7OctYmcXEGz9.jt//c1tqW3hbP4MKN7EHo98Q4mcbZHzxvHt0gcrMZFbI/B
RJoyhLhEgA0mdJPsG1:19571:0:99999:7:::
daemon:*:19523:0:99999:7:::
bin:*:19523:0:99999:7:::
sys:*:19523:0:99999:7:::
sync:*:19523:0:99999:7:::
games:*:19523:0:99999:7:::
man:*:19523:0:99999:7:::
lp:*:19523:0:99999:7:::
mail:*:19523:0:99999:7:::
news:*:19523:0:99999:7:::
uucp:*:19523:0:99999:7:::
proxy:*:19523:0:99999:7:::
```

Applications - Checksums

1. Download File

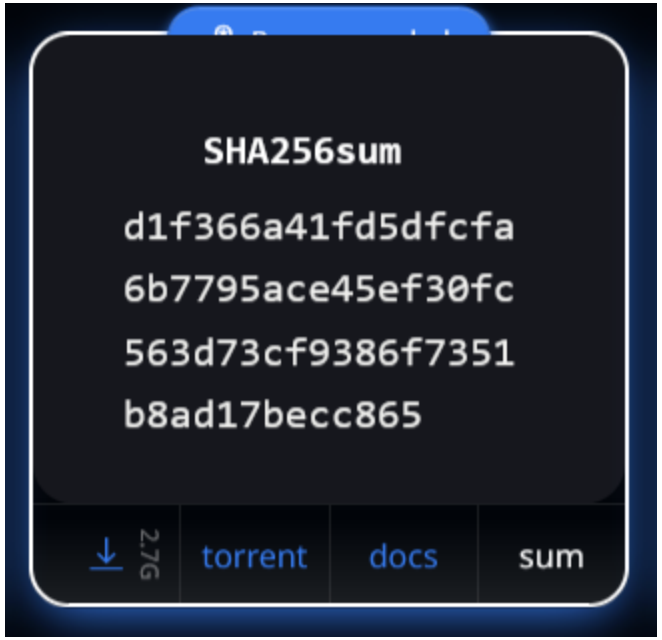
kali-linux-2023.2-virtualbox-amd64.7z

2. Hash the Downloaded File

```
PS C:\Users\radia\Downloads> Get-FileHash .\kali-linux-2023.2-virtualbox-amd64.7z

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          D1F366A41FD5DFCFA6B7795ACE45EF30FC563D73CF9386F7351B8AD17BECC865       C:\Users\radia\Downloads\kali...
```

3. Verify it Matches



Can’t the attacker change the hash?

Security Definitions:

- Preimage Resistance
- 2nd Preimage Resistance
- Collision Resistance Resistance

Preimage: $\nexists\ PPT$ to find x given y = H(x)

2nd Preimage: Given m_1, $\nexists\ PPT$ to find m_2 such that $H(m_1) = H(m_2), m_1 \neq m_2$

Collision: $\nexists\ PPT$ to find m_1, m_2 such that $H(m_1) = H(m_2), m_1 \neq m_2$

Onto Integrity

- What is Cryptography
- Background
- Primitives
  - Symmetric Key Encryption Scheme
  - Asymmetric Key Encryption Scheme
  - Hash Function
  - Message Authentication Codes
  - Digital Signature
- Answer the Question
- Take Aways


Message Authentication
Scheme / Code
Confidentiality, Integrity, Authenticity

Intuition Derivation

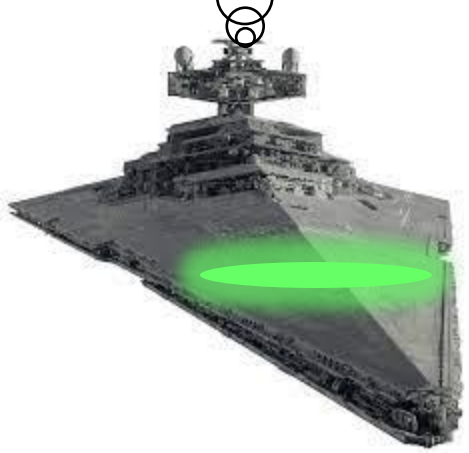
I'll change it anyway!

Ha! Vader can't do anything!

Long live the republic

F067e95df2ccba51576349c45a0[illegible]489b1394854f38a379fa15ab2…

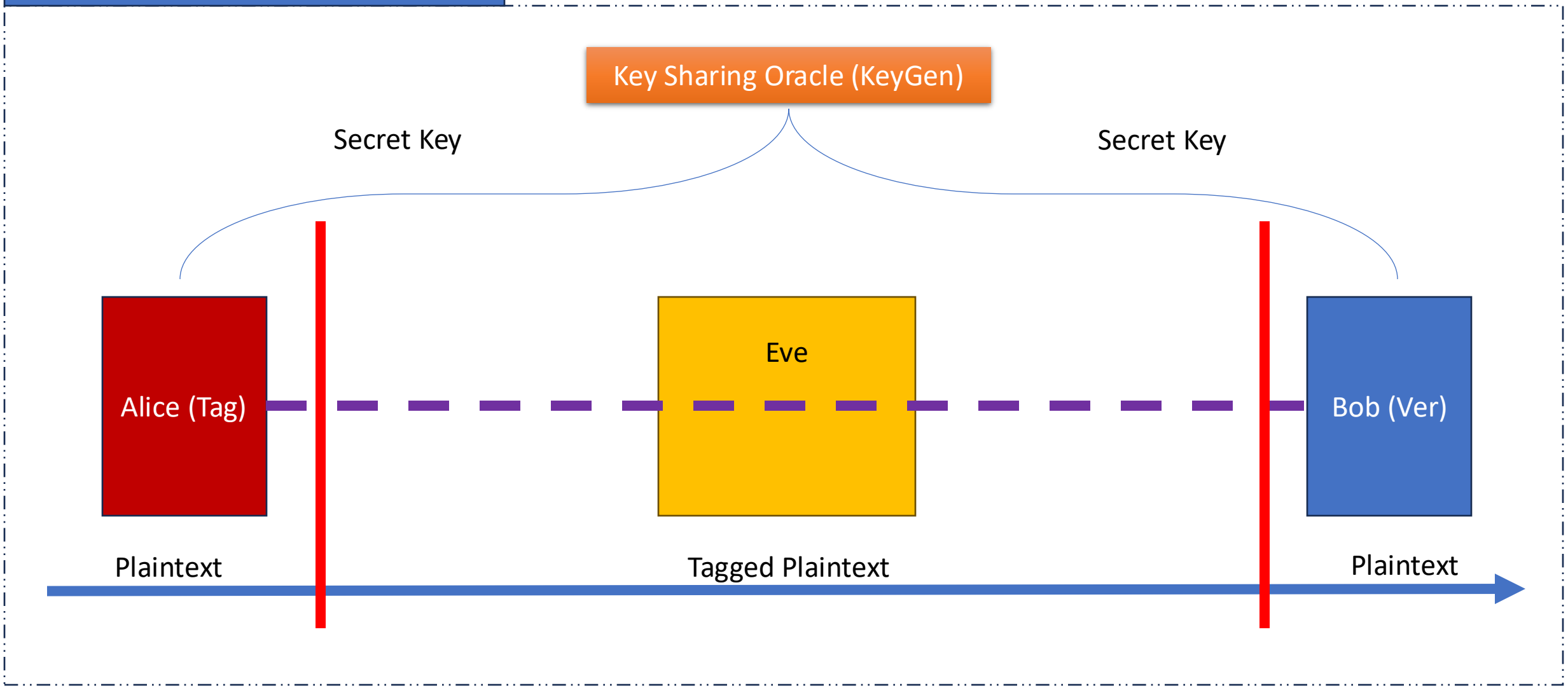
Someone changed this, I shouldn't trust it!



How can Bob authentically guarantee a message sent over a compromised medium with pre-distributed keys?

How can Bob authentically guarantee a message sent over a compromised medium with pre-distributed keys?

System Model



How can Bob authentically guarantee a message sent over a compromised medium with pre-distributed keys?

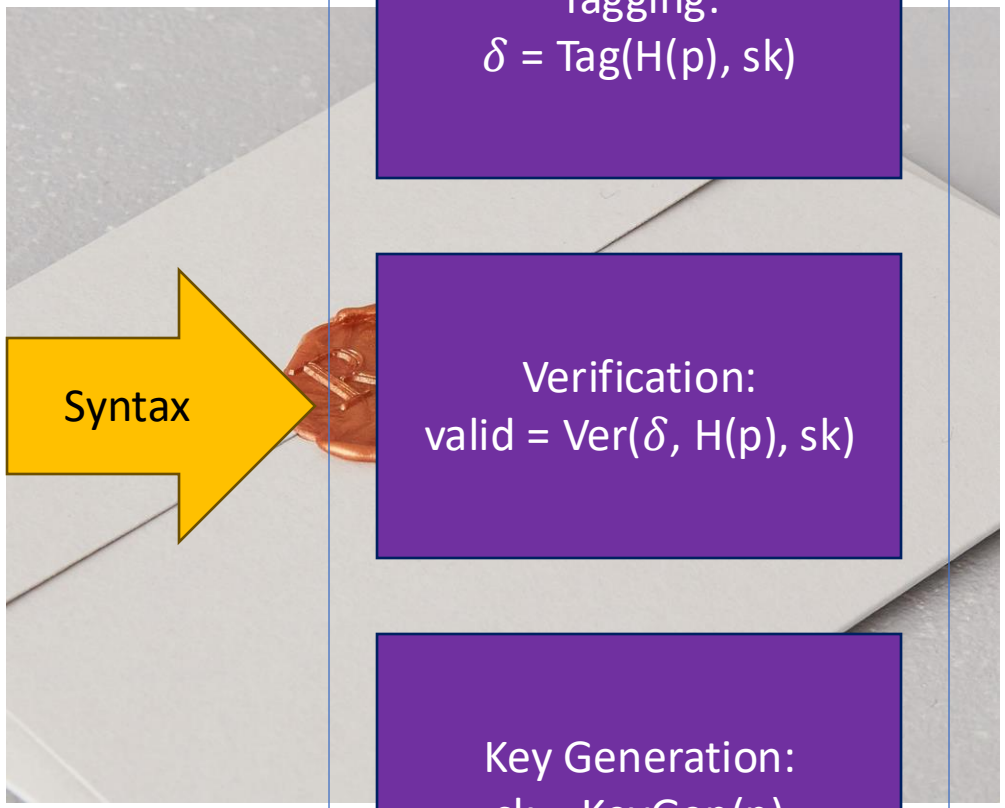
Real World Example

Sealing Mechanism

Verification Mechanism

Key Distribution

Envelope Example



Syntax

Tagging:
δ = Tag(H(p), sk)

Verification:
valid = Ver(δ, H(p), sk)

Key Generation:
sk = KeyGen(n)

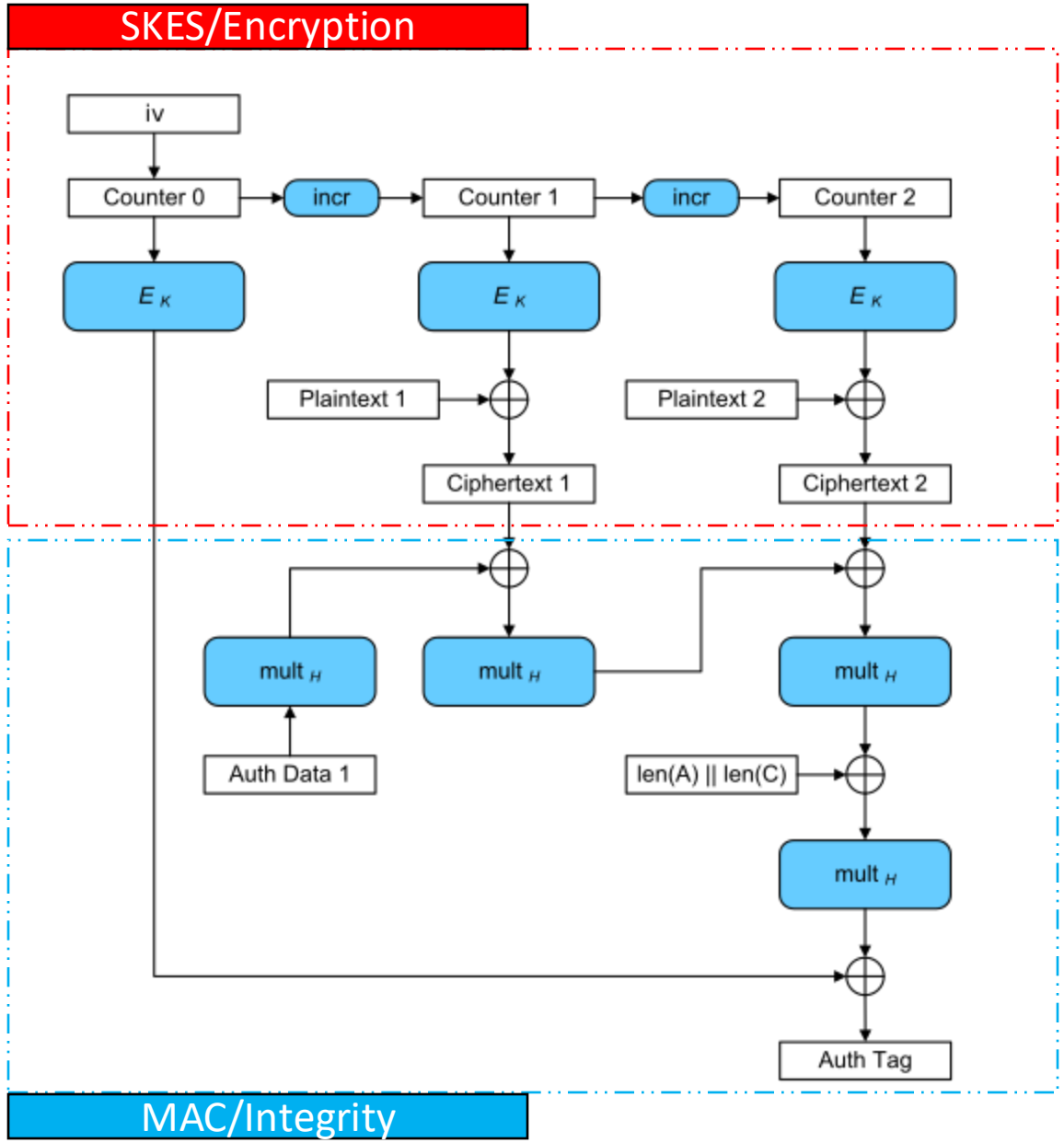
SIG

Wrapping Around – AES_GCM

Modern MAC: AEAD – Included in AES-GCM

Tagging:
Incorporates Hashes, Multiplication, and Key to create a “tag”

Verification:
Recompute the tag and compare



(Picture from Wikipedia: https://en.wikipedia.org/wiki/Galois/Counter_Mode)

Message Authentication Codes/Scheme (MAC/MAS)

Security Definition: Unforgeable Under Adaptive Chosen Message Attack

Integrity, Authenticity, ~~Non-Repudiability~~

Why?

Attacker

Defender

Pre-Rounds: Defender randomly creates a key K of size k
Create Key Randomly, size k: $K = KeyGen(1^k), |K| \geq k$

Round 1: Attacker sends a plaintext of their choosing: <m_1>

Round 1: Defender sends the tag of the plaintext back: $\delta_1 = Tag_K(m_1)$

Round q: Attacker sends a plaintext of their choosing: <m_q>

Round q: Defender sends the tag of the plaintext back: $\delta_q = Tag_K(m_q)$

Post Rounds, q+1: Attacker guesses what the tag with a ***new*** message would be:
<$m_{q+1}, \delta_{q+1} = Tag_K(m_{q+1})$>

Post Rounds, q+1: Defender says if attacker was correct (b=1) or not (b=0) based on:
$b = \text{Ver}(K, \ \delta_{q+1} = Tag_K(m_{q+1}))$

Definition (d): The message authentication scheme MAS = (KeyGen, Tag, Ver) is secure iff a probabilistic polynomial time-adversary only gains a negligible advantage after adaptively analyzing chosen <m_q, δ_q > such that ($Pr[\text{Ver}(K, \ \delta_{q+1} = Tag_K(m_{q+1})) = 1] \leq negl(k)$), where q is the rounds analyzed, δ_{q+1} is a forged tag for some $m_{q+1} \notin \{m_1, m_2, \ldots m_q\}$, and $K \in_R \{0,1\}^k$.

Final One

- What is Cryptography
- Background
- Primitives
  - Symmetric Key Encryption Scheme
  - Asymmetric Key Encryption Scheme
  - Hash Function
  - Message Authentication Codes
  - Digital Signature
- Answer the Question
- Take Aways

Digital Signatures

Confidentiality, Integrity, Authenticity, Non-repudiability

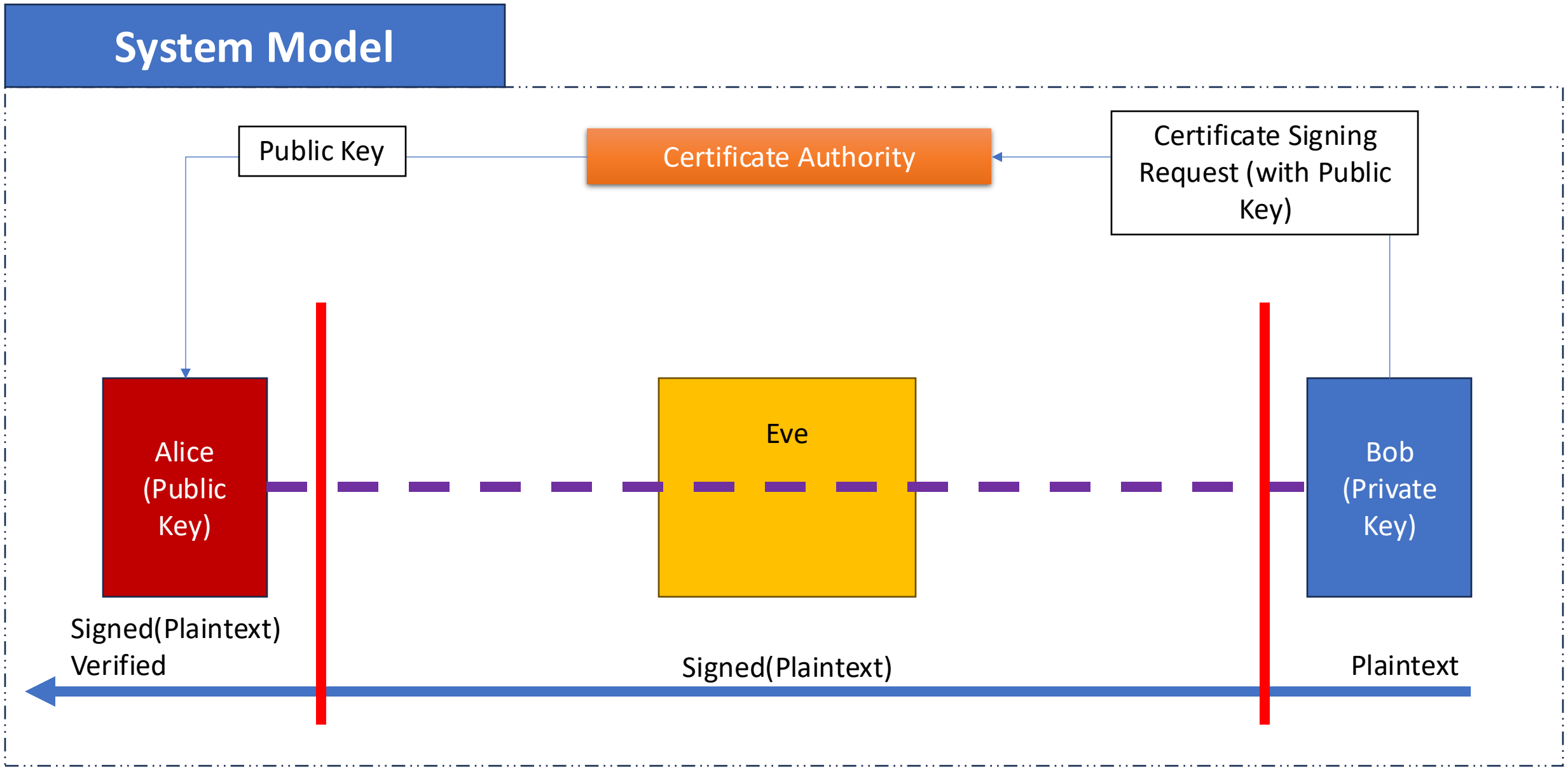
Intuition Derivation

We need new orders!

How can Bob authentically guarantee a message sent over a compromised medium with publicly known keys?

How can Bob authentically guarantee a message sent over a compromised medium with publicly known keys?

System Model



Digital Signature Scheme (SIG)

How can Bob "sign" a message, proving that its his?

Real World Example

Signing Mechanism

Verification Mechanism

Signature or Reference Creation

Signing Example

Sign:
δ = Sign(H(n), sk)

Key Generation:
pk, sk = KeyGen(n)

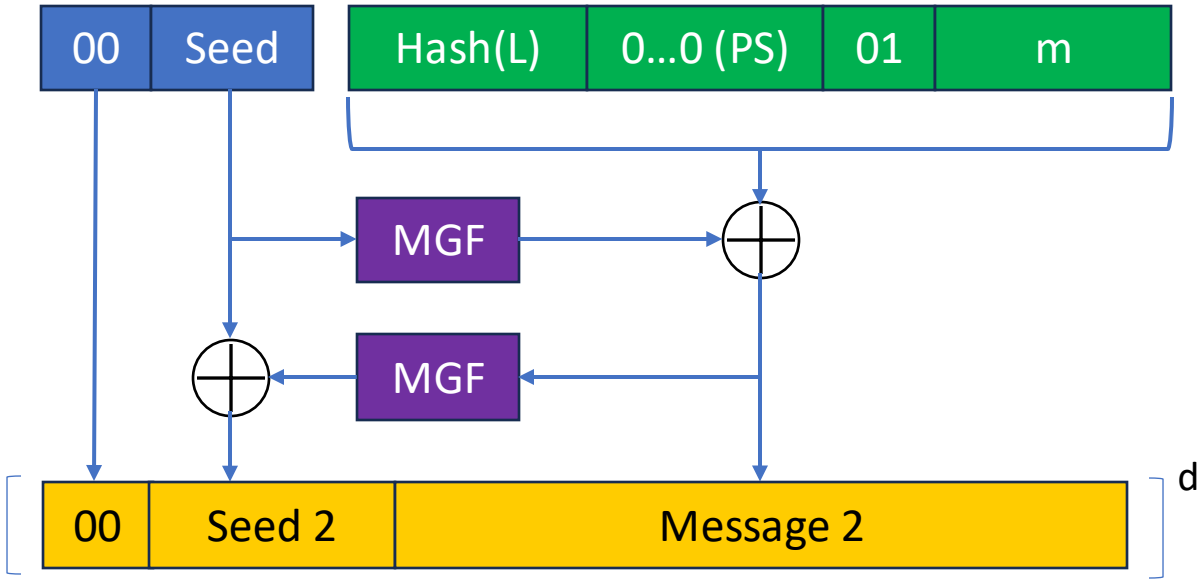
SIG

RSA, Again?

RSA Function

1. **KeyGen:**
   1. Generate large prime numbers p,q
   2. N = p*q
   3. $Z_n = \{a: a \in \{0, 1, \dots, N-1\}, GCD(a, N) = 1\}$
   4. $\phi(N) = (p-1)(q-1)$
   5. **e** s.t $GCD(e, \phi(N)) = 1$
   6. $\boldsymbol{d} = e^{-1} \, mod \, \phi(N)$
2. **Sign**: $\forall \, m \in Z_n, m^d \, mod \, N$
3. **Ver**: $\forall \, m \in Z_n, m^e \, mod \, N$

RSA Cryptosystem & OAEP



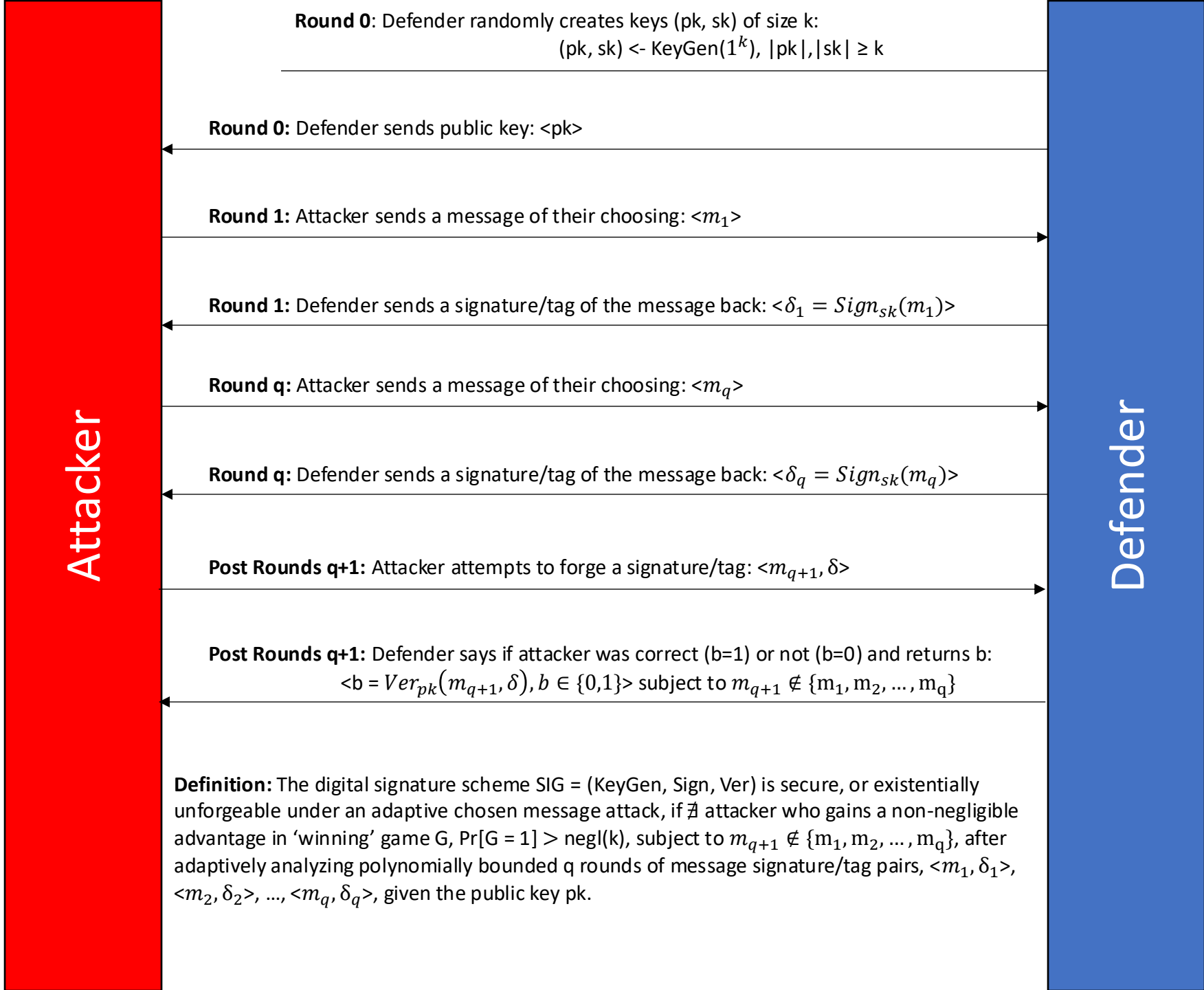
(Picture derived from https://en.wikipedia.org/wiki/Optimal_asymmetric_encryption_padding)

Key Difference: Swap d and e so that: d “encrypts” | e “decrypts” – swap the keys!

Digital Signature Scheme (SIG)

Security Definition: Unforgeable Under Adaptive Chosen Message Attack

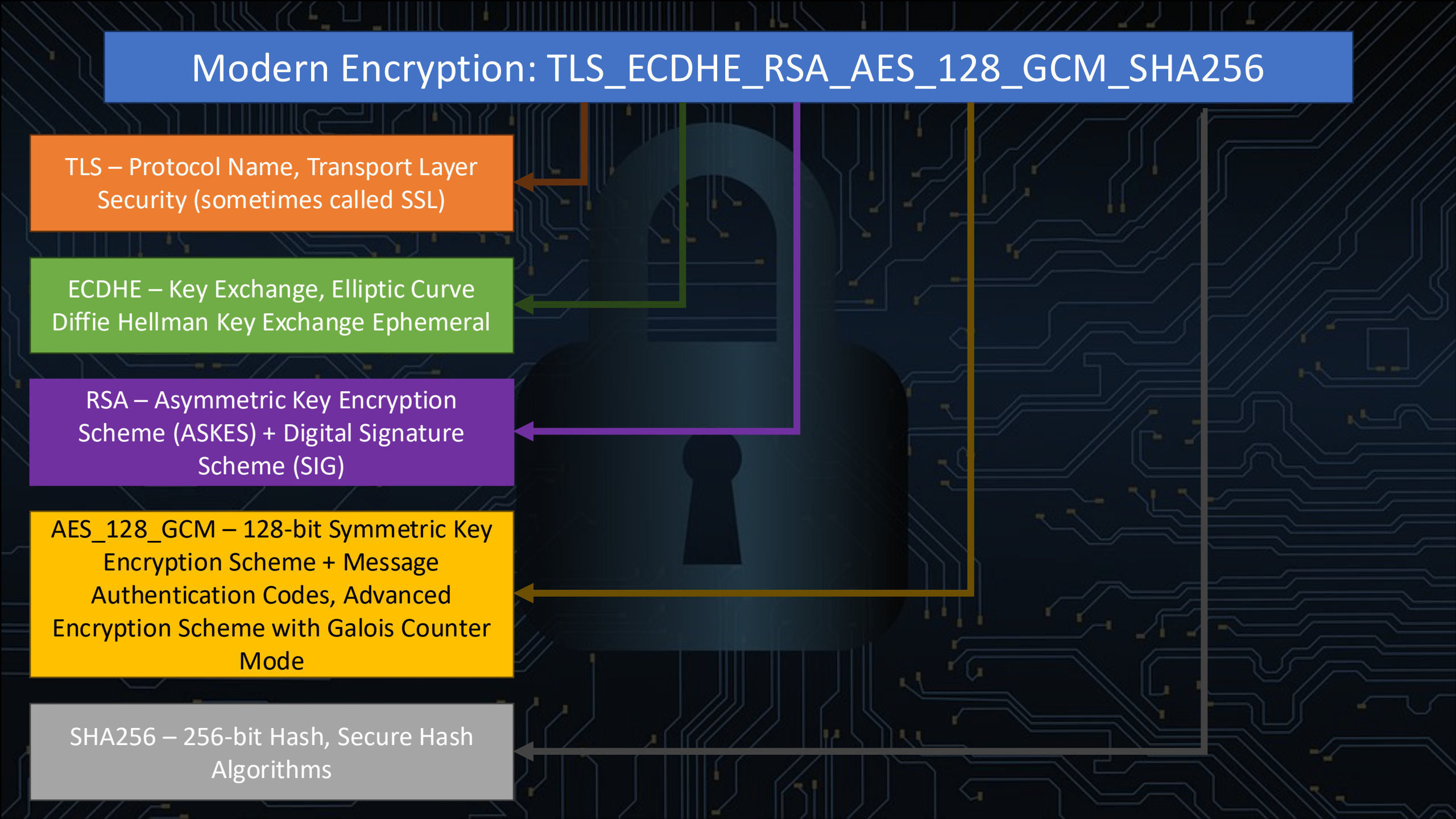
Integrity, Authenticity, Non-Repudiability



What is Encryption?

- What is Cryptography
- Background
- Primitives
  - Symmetric Key Encryption Scheme
  - Asymmetric Key Encryption Scheme
  - Hash Function
  - Digital Signature
  - Message Authentication Codes
- Answer the Question
- Take Aways

Modern Encryption: TLS_ECDHE_RSA_AES_128_GCM_SHA256



TLS – Protocol Name, Transport Layer Security (sometimes called SSL)

ECDHE – Key Exchange, Elliptic Curve Diffie Hellman Key Exchange Ephemeral

RSA – Asymmetric Key Encryption Scheme (ASKES) + Digital Signature Scheme (SIG)

AES_128_GCM – 128-bit Symmetric Key Encryption Scheme + Message Authentication Codes, Advanced Encryption Scheme with Galois Counter Mode

SHA256 – 256-bit Hash, Secure Hash Algorithms

Takeaways:

We have the technical ability to provide confidentiality, integrity, authenticity, non-repudiability to all messages

Hackers can exploit implementation vulnerabilities or deprecated schemes, but not abstract model (yet)

Cryptographers must prove all cryptographic schemes

You use proven encryption daily | DO NOT design your own scheme

Use the latest proven schemes for maximum security