

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

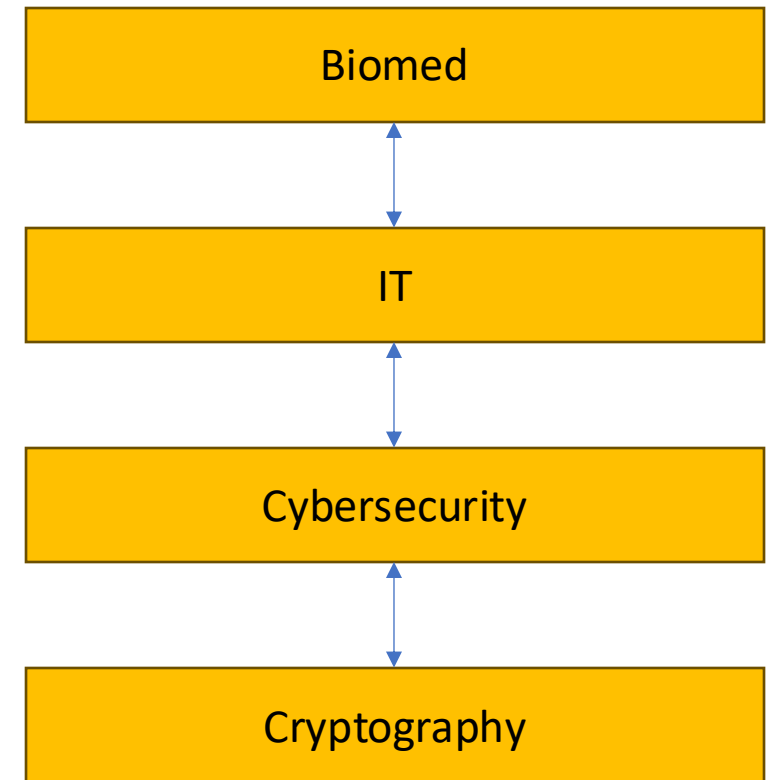
Our Focus

https

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





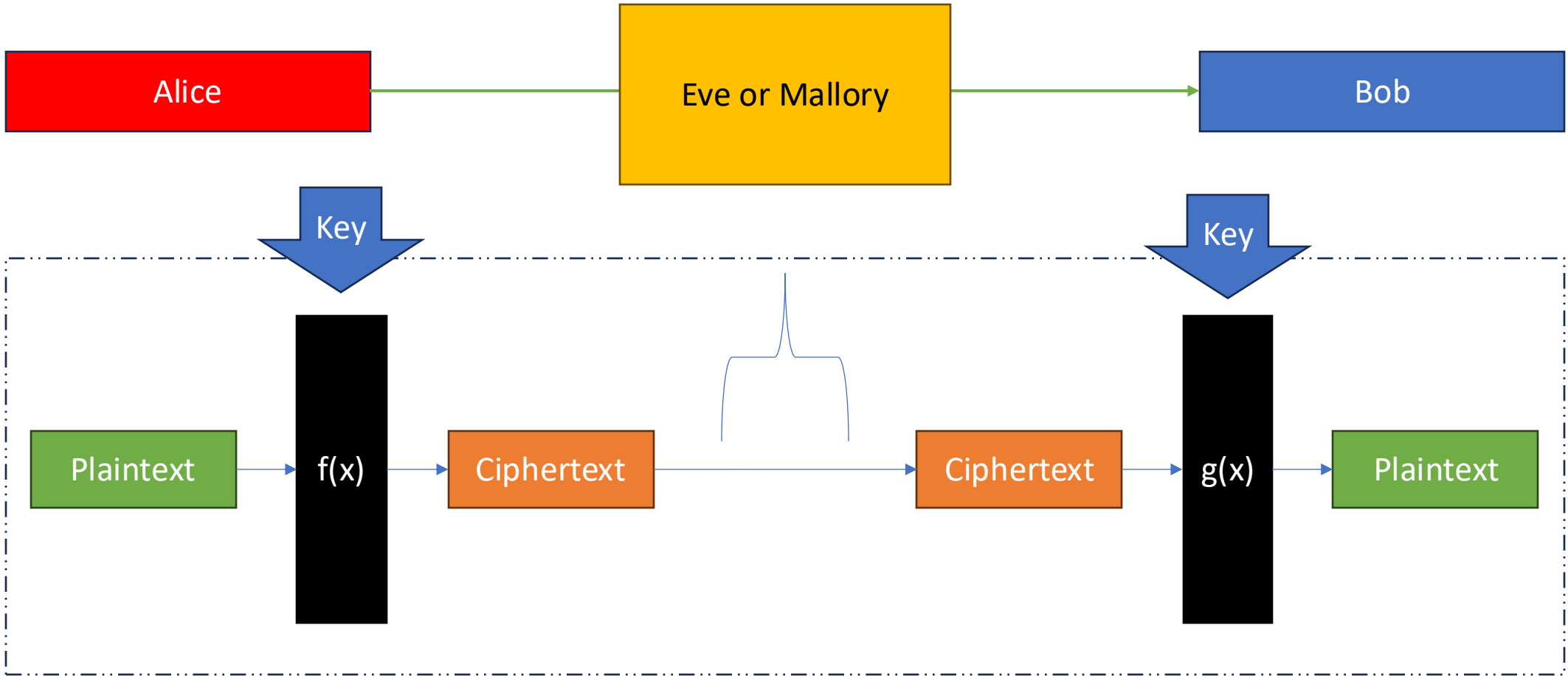
What is Encryption?

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

How can I talk securely to the star destroyer?



I have no idea what he said

~~Those were the droids we were looking for~~

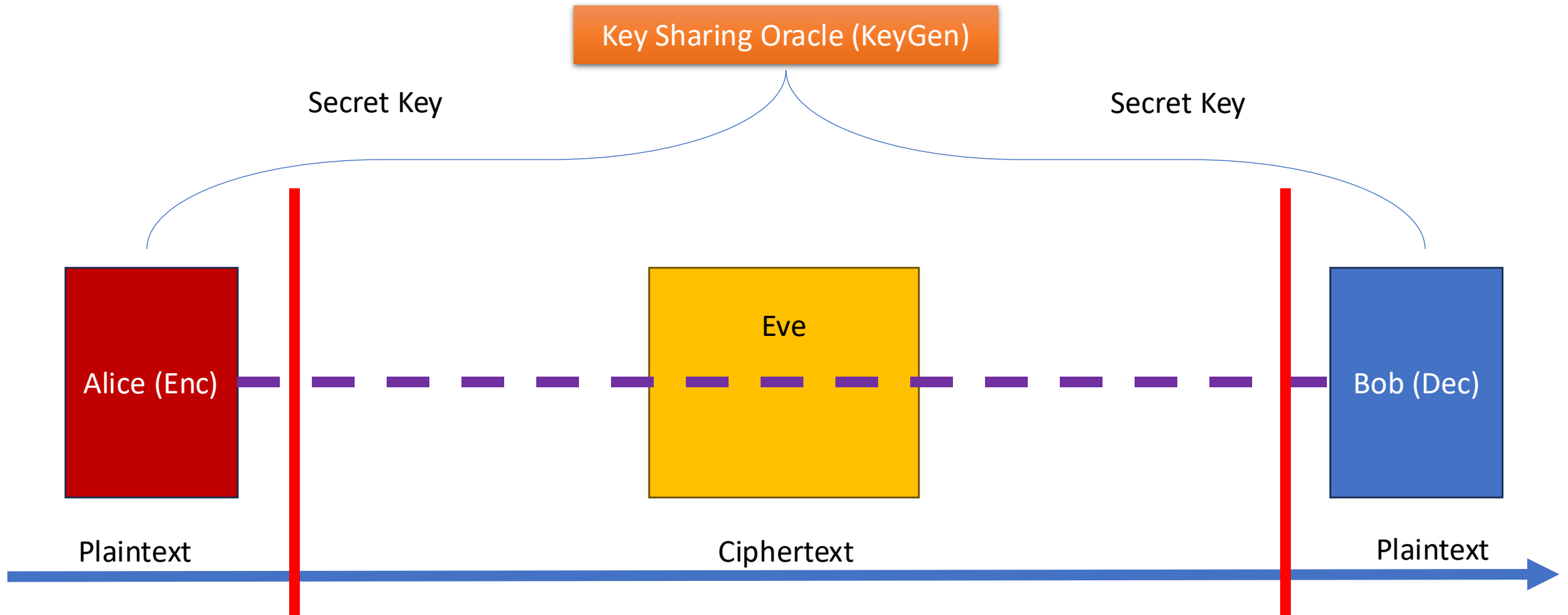
F067e95df2ccba51576349c45a379fa15ab26cbbd1827d87c3bed0...



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



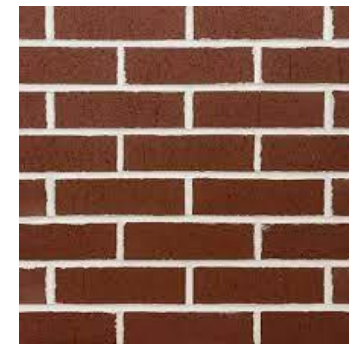
How can Alice securely send Bob a message over a compromised medium with pre-distributed keys?

Block Ciphers

Modes of Operation

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

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:
 $c = \text{Enc}(p, sk)$

Decryption:
 $p = \text{Dec}(c, sk)$

Key
Generation:
 $sk = \text{KeyGen}(n)$

What's n?

SKES

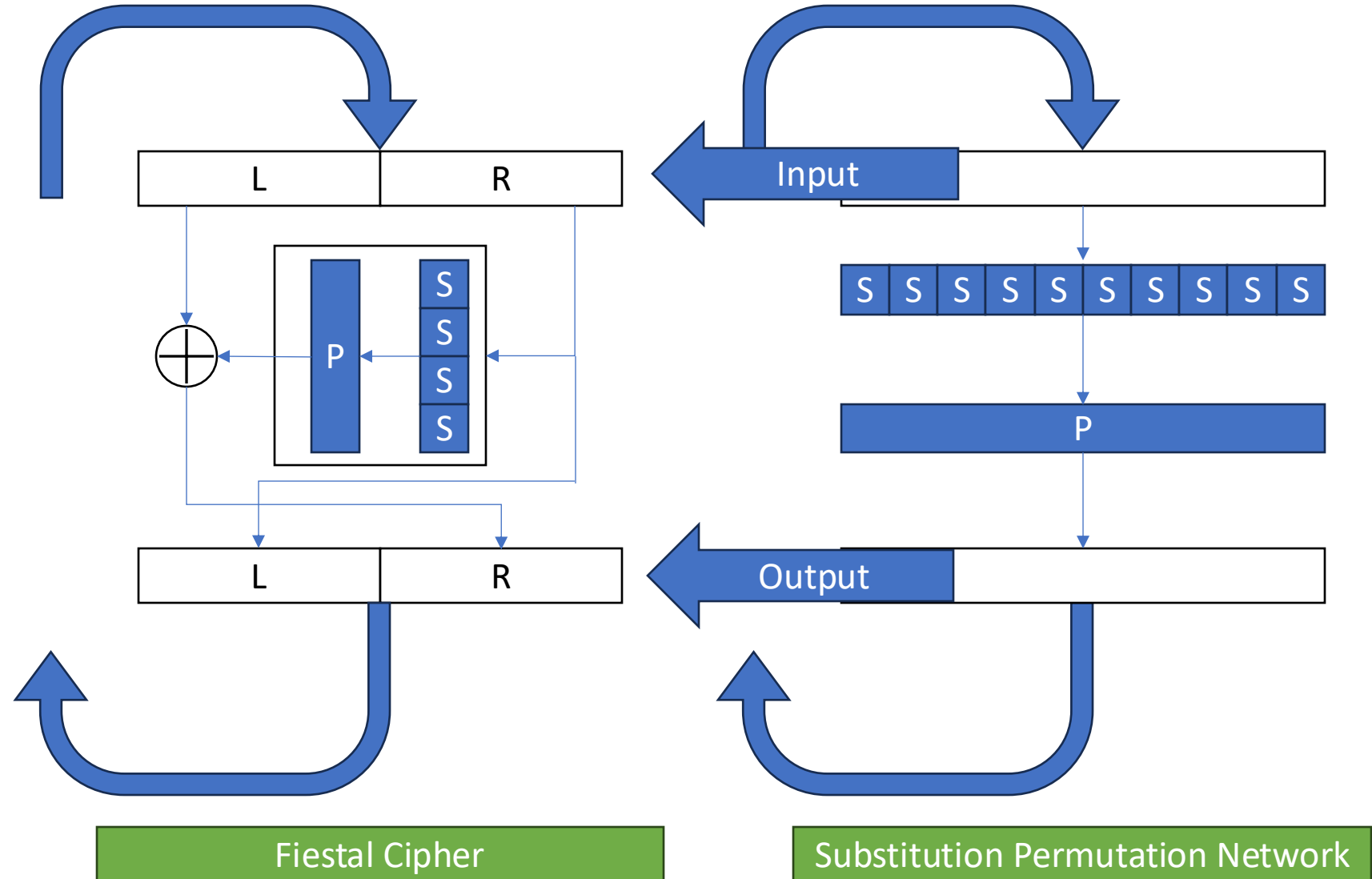
Block Cipher

Block Cipher Architectures

S = Substitution

P = Permutation

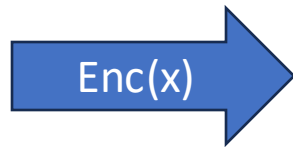
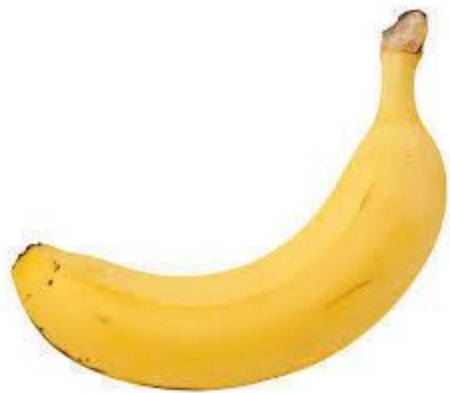
- All provably secure Block Ciphers are created from either the Feistel Cipher or the S/P Network.
- Repeats for 10-30 rounds depending on the cipher
- Examples:
 - Twofish, Serpent, **AES (commonly used)**



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!



Naïve Approach

A diagram illustrating a "Naïve Approach" to encryption. It features a red header box with the text "Naïve Approach". Below the header, there are two images: a top image showing a regular grid of red bricks, and a bottom image showing a corrupted version of the banana image. The banana image is heavily distorted with vertical lines and a large area of multi-colored noise on the right side.

We need this!

A diagram illustrating the desired outcome. It features a green header box with the text "We need this!". Below the header, there are two images: a top image showing a regular grid of red bricks, and a bottom image showing a solid, uniform grey square.

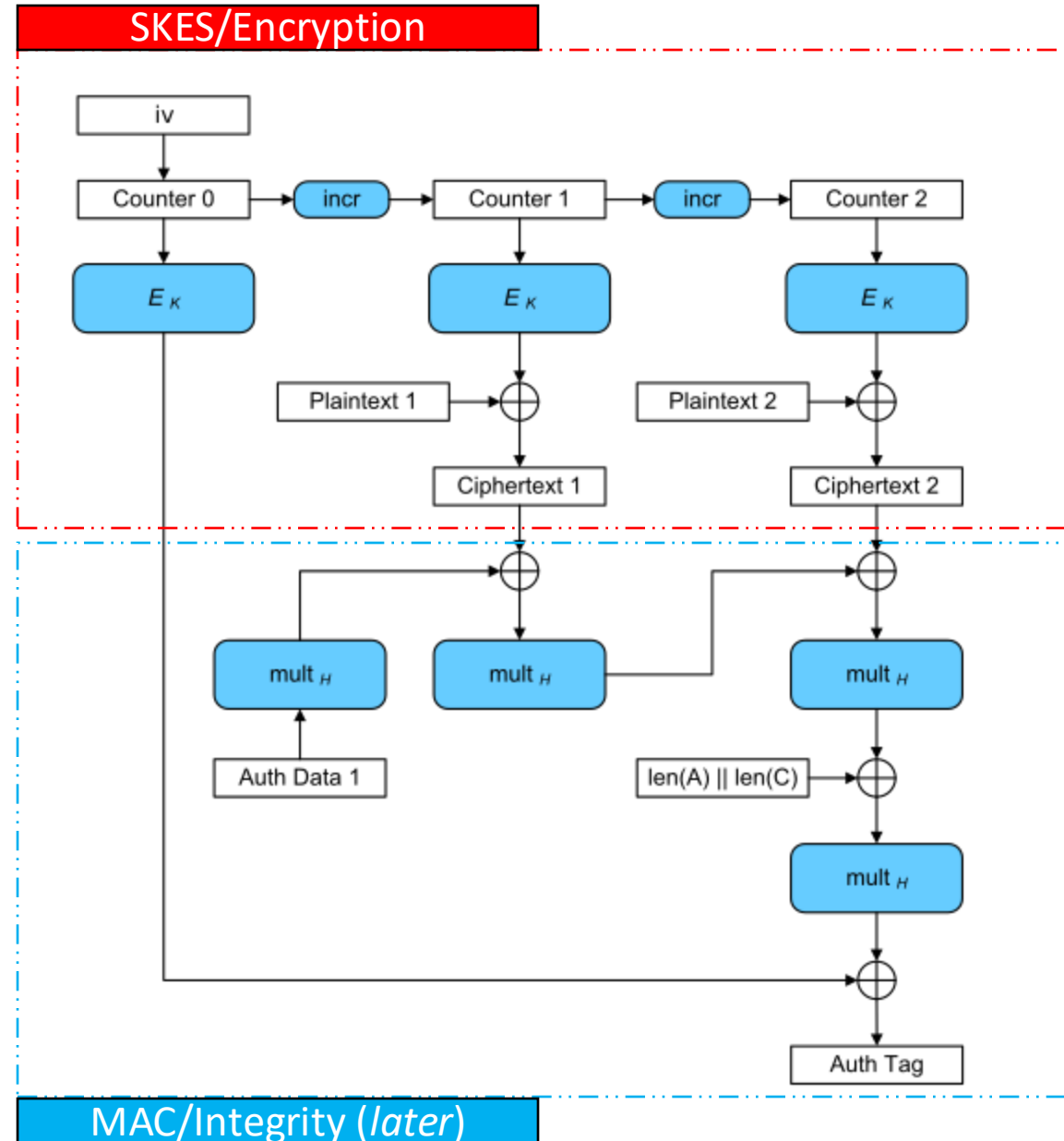
Modes of Operation

- 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

Attacker

Pre-Rounds: Defender randomly creates a key K of size k , and picks $b = 0$ or 1 :
Create Key Randomly, size k : $K \in_R \{0,1\}^k$
Pick b Randomly, 0 or 1: $b \in_R \{0,1\}$

Round 1: Attacker sends two plaintexts of their choosing: $\langle M_1^0, M_1^1 \rangle$
or a ciphertext: $C^{(1)}$

Round 1: Defender sends a ciphertext of the b^{th} plaintext back: $C_1 = ENC_K(M_1^b)$
or a plaintext: $DEC_K(C^{(1)})$

Round $q+w$: Attacker sends two plaintexts of their choosing: $\langle M_q^0, M_q^1 \rangle$
or a ciphertext: $C^{(w)}$

Round $q+w$: Defender sends a ciphertext of the b^{th} plaintext back: $C_q = ENC_K(M_q^b)$
or a plaintext: $DEC_K(C^{(w)})$

Post Rounds: Attacker guesses what b is, or which plaintext message they were sending was getting encrypted.

Post Rounds: Defender says if attacker was correct or not,
assuming $\{C^{(1)}, \dots, C^{(w)}\} \cap \{C_1, \dots, C_q\} = \emptyset$ (attacker didn't decrypt ciphertext received from defender after sending them their plaintexts).

Definition: The symmetric encryption scheme $SKES = (\text{KeyGen}, \text{ENC}, \text{DEC})$ is IND-CCA secure if an attacker only gains a negligible advantage ($\Pr \in \{\frac{1}{2} - \text{negl}(k), \frac{1}{2} + \text{negl}(k)\}$) in guessing which two chosen plaintexts $\langle M_q^0, M_q^1 \rangle$ are encrypted after analyzing ciphertext $C_1 = ENC_K(M_q^b)$ and any plaintext $M_w = DEC_K(C^{(w)})$ from a chosen ciphertext $C^{(w)}$, assuming $\{C^{(1)}, \dots, C^{(w)}\} \cap \{C_1, \dots, C_q\} = \emptyset$, where $q+w$ is the round, $b \in_R \{0,1\}$, and $K \in_R \{0,1\}^k$.

Defender

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

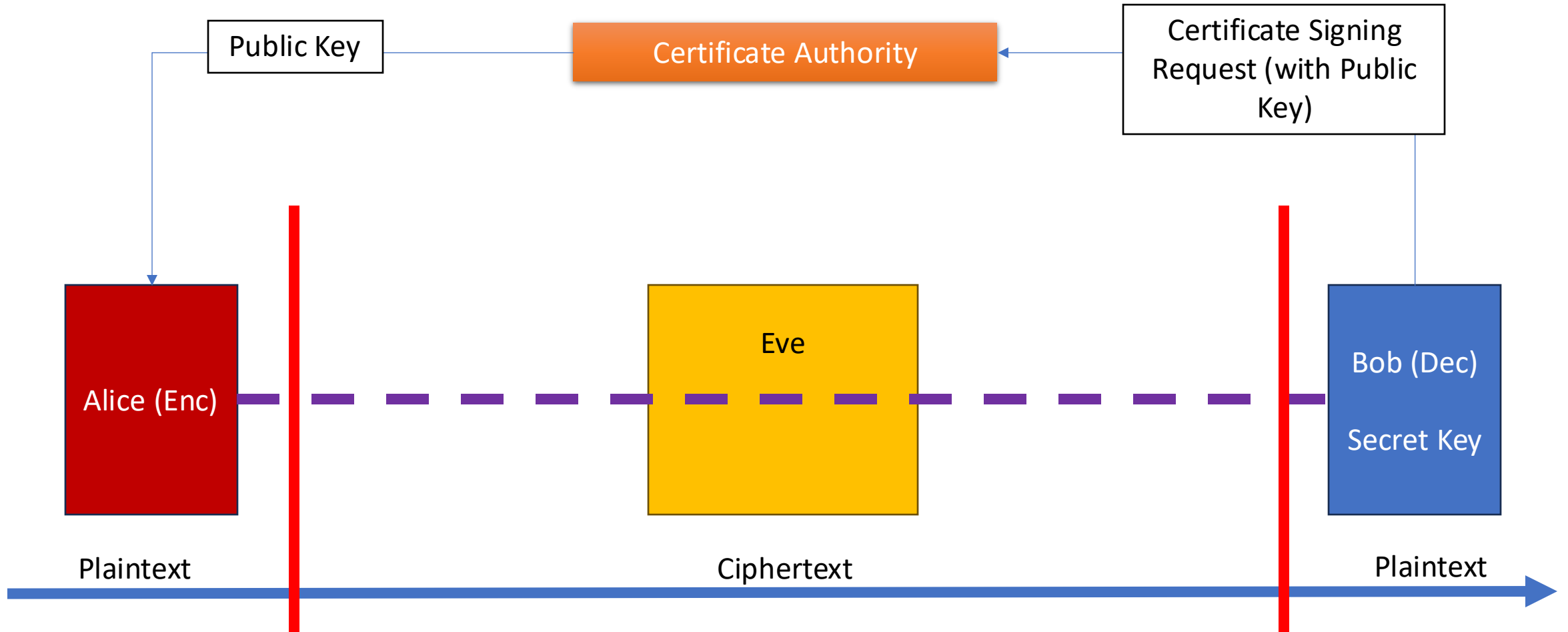
Intuition Derivation



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:
 $c = \text{Enc}(p, pk)$

Decryption:
 $p = \text{Dec}(c, sk)$

Key Generation:
 $sk, pk = \text{KeyGen}(n)$

ASKES

RSA

RSA Function

1. KeyGen:

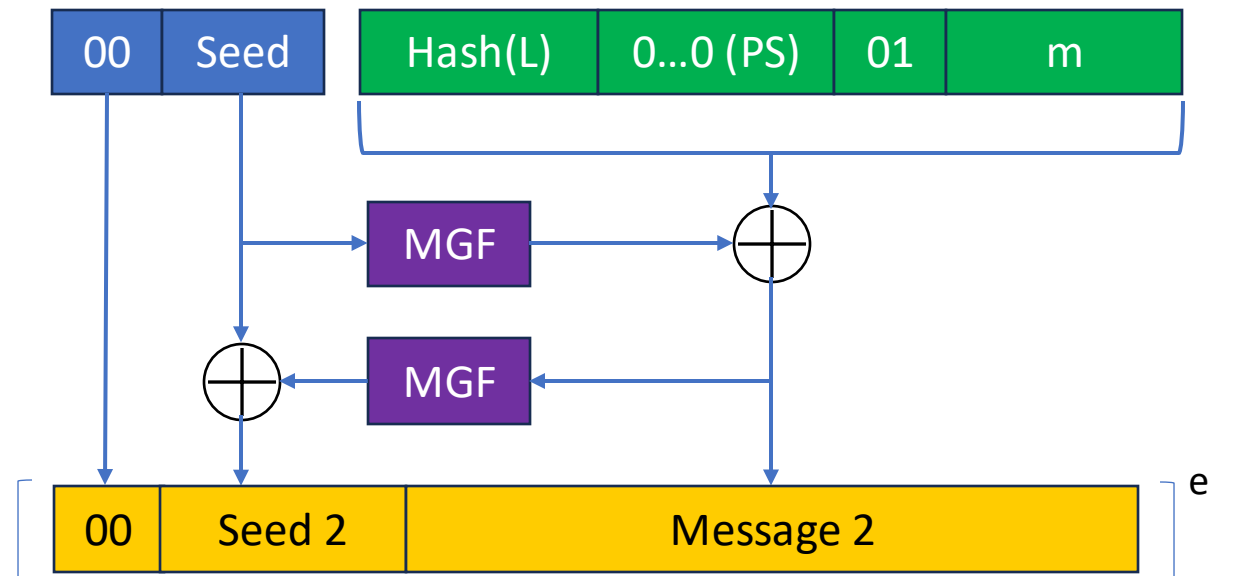
1. Generate large prime numbers p, q
2. $N = p * q$
3. $Z_n = \{a: a \in \{0, 1, \dots, N - 1\}, GCD(a, N) = 1\}$
4. $\phi(N) = (p - 1)(q - 1)$
5. e s.t $GCD(e, \phi(N)) = 1$
6. $d = e^{-1} \text{ mod } \phi(N)$

2. Enc: $\forall m \in Z_n, m^e \text{ mod } N$

3. Dec: $\forall c \in Z_n, c^d \text{ mod } N$

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

Attacker

Round 0: Defender randomly creates keys (pk, sk) of size k and picks binary bit b :
Create Keys: $(pk, sk) \leftarrow \text{KeyGen}(1^k)$, $|pk|, |sk| \geq k$
Pick b : $b \in_R \{0,1\}$

Round 0: Defender sends public key: $\langle pk \rangle$

Round 1: Attacker adaptively sends a ciphertext of their choosing: $\langle C_1 \rangle$

Round 1: Defender sends a plaintext back: $\text{Dec}_{sk}(C_1)$

Round n: Attacker sends two plaintext of their choosing: $\langle m^0, m^1 \rangle$, $m \in \text{MessageSpace}$

Round n: Defender sends a ciphertext of the b^{th} plaintext back: $C = \text{Enc}_{pk}(m^b)$

Round w: Attacker adaptively sends a ciphertext of their choosing: $\langle C_w \rangle$

Round w: Defender sends a plaintext back: $\text{Dec}_{sk}(C_w)$

Post Rounds: Attacker guesses what b is, or which plaintext message they sent got encrypted: b'

Post Rounds: Defender says if attacker was correct or not: $b' == b$?
Game returns Boolean evaluation of $b' == b$, where 1=True=win, and 0=False=loss, assuming $C \notin \{C^{(1)}, \dots, C^{(w)}\}$

Definition: The asymmetric encryption scheme ASKES = (KeyGen, Enc, Dec) is IND-CCA secure if \nexists attacker who gains a non-negligible advantage in 'winning' game G , $\Pr[G = 1] > \frac{1}{2} + \text{negl}(k)$ after adaptively analyzing polynomially bounded q <plaintext, ciphertext> pairs $\langle m_1, c_1 \rangle$, $\langle m_2, c_2 \rangle$, ..., $\langle m_q, c_q \rangle$, where $c_q = \text{Enc}_{pk}(m_q)$ (encryption oracle) or $m_w = \text{Dec}_{sk}(c_w)$ (decryption oracle) $1 \leq w \leq q$, after given the public key pk by the defender, $(pk, sk) \leftarrow \text{KeyGen}(1^k)$, under the assumption that $C \notin \{C^{(1)}, \dots, C^{(w)}\}$ and $1 \leq n \leq w$. The attacker may, at any time, self-query the encryption oracle.

Defender

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

Input
CABMET

Input
CABMET2

Hash Function
 $y = H(x)$

Output y
3787a83a4aa169e5b18e38f3a8af331
fe8b4fd15e4ba78cad17a53e6230286
a2

Output y
a3551b82d52d2647ab35b7b70c669d
469c671c81f9fc51398fba723641734a
04

Why is the output completely different?



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

Applications – Password Storage

Naïve Approach

Username	Password
Logan	Password123
Cabmet	Password123
Administrator	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	fuijosh
Administrator	c4b95cb0f8c5b	Y#*()H

```
root@test:~# cat /etc/shadow
root:$6$j1WDO017$I6vMtIJIa7OctYmcXEGz9.jt//c1tqW3hbP4MKN7Eh098Q4mcbZHxvHt0gcrMZFbI/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



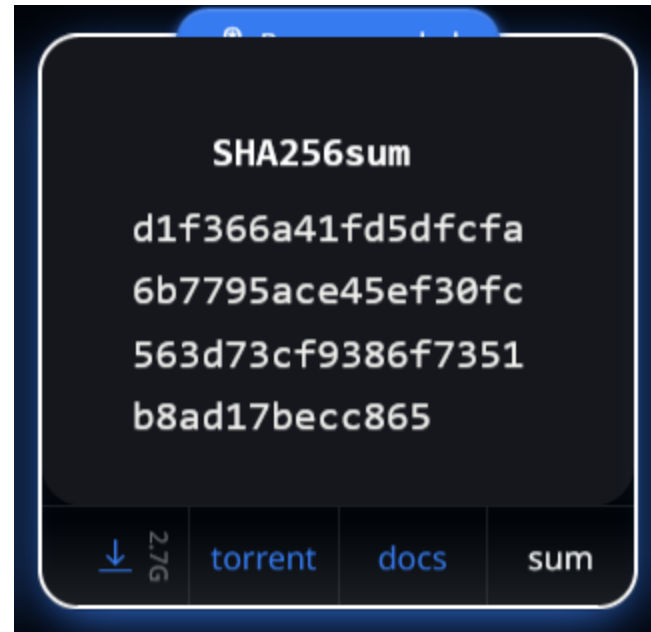
 kali-linux-2023.2-virtualbox-amd64.7z

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

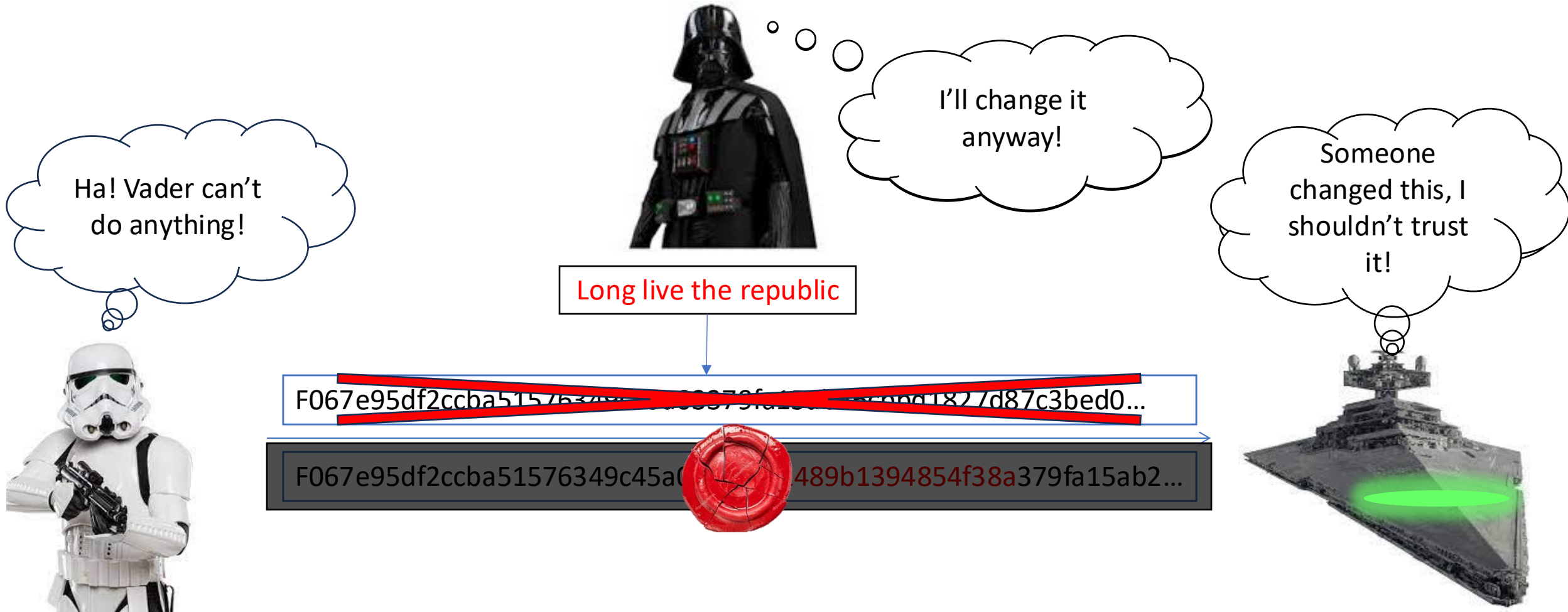
- What is Cryptography
- Background
- Primitives
 - Symmetric Key Encryption Scheme
 - Asymmetric Key Encryption Scheme
 - Hash Function
 - Message Authentication Codes
 - Digital Signature
- Answer the Question
- Take Aways



Message Authentication Scheme / Code

Confidentiality, Integrity, Authenticity

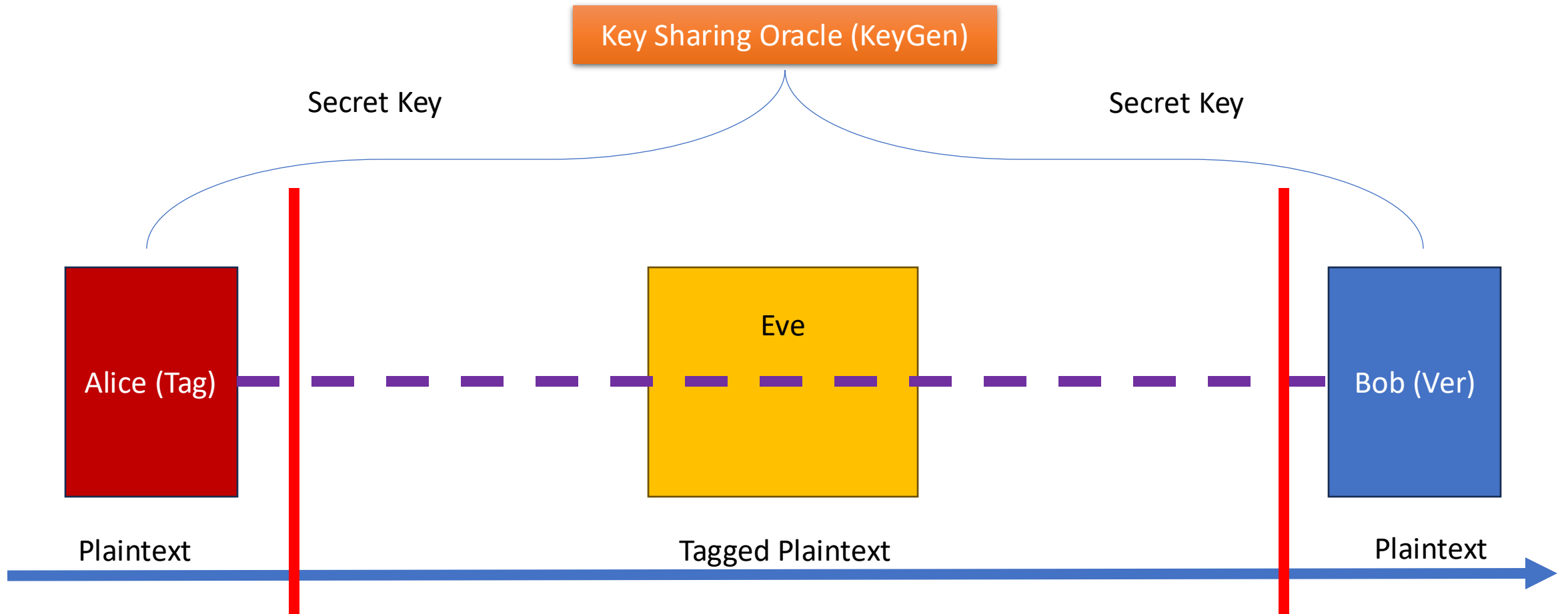
Intuition Derivation



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:
 $\delta = \text{Tag}(H(p), sk)$

Verification:
 $\text{valid} = \text{Ver}(\delta, H(p), sk)$

Key Generation:
 $sk = \text{KeyGen}(n)$

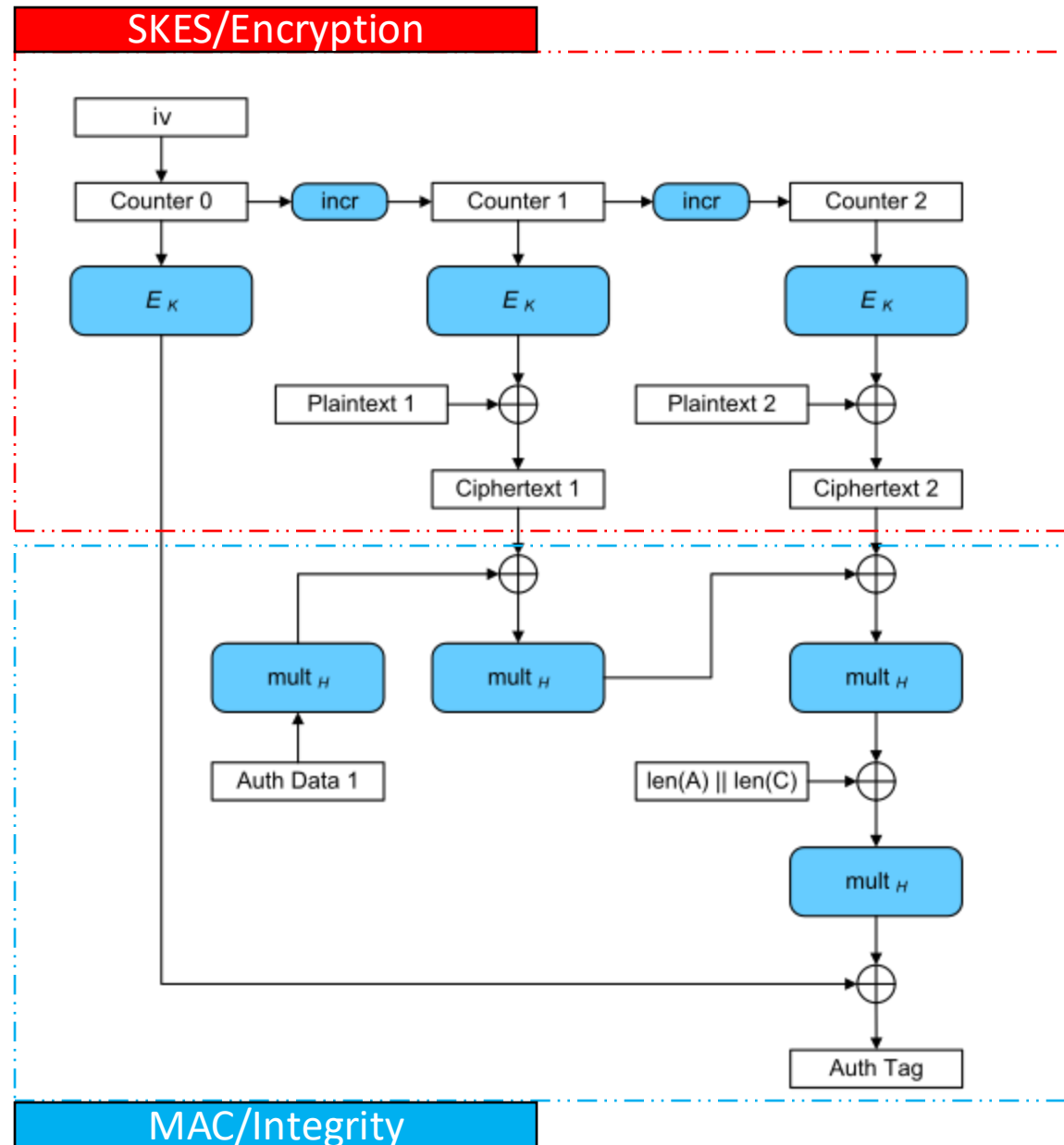
SIG

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

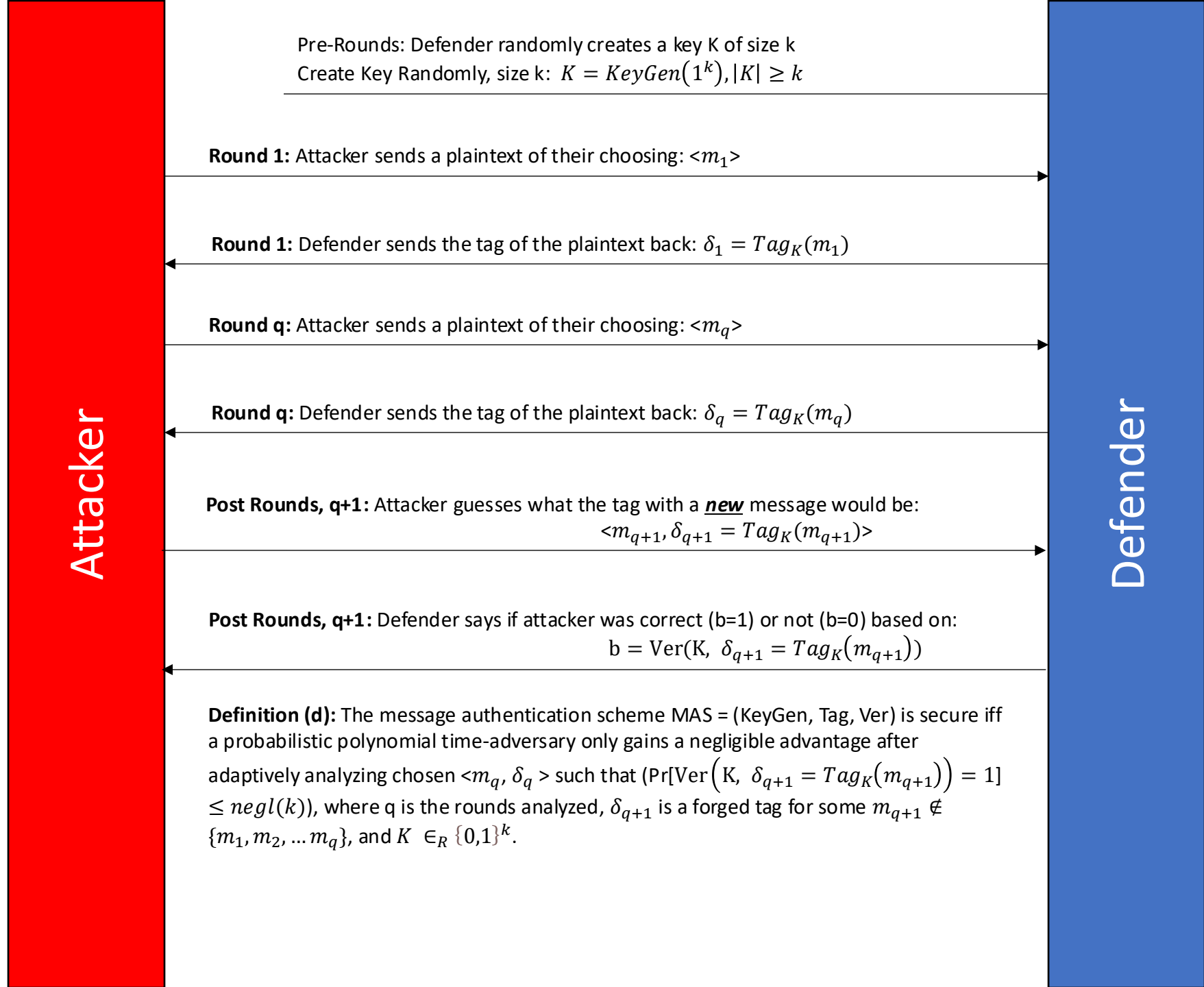


(Picture from Wikipedia: https://en.wikipedia.org/wiki/Galois/Counter_Mode)

Message Authentication Codes/Scheme (MAC/MAS)

Security Definition:
Unforgeable Under
Adaptive Chosen
Message Attack

Integrity, Authenticity,
Non-Repudiability



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

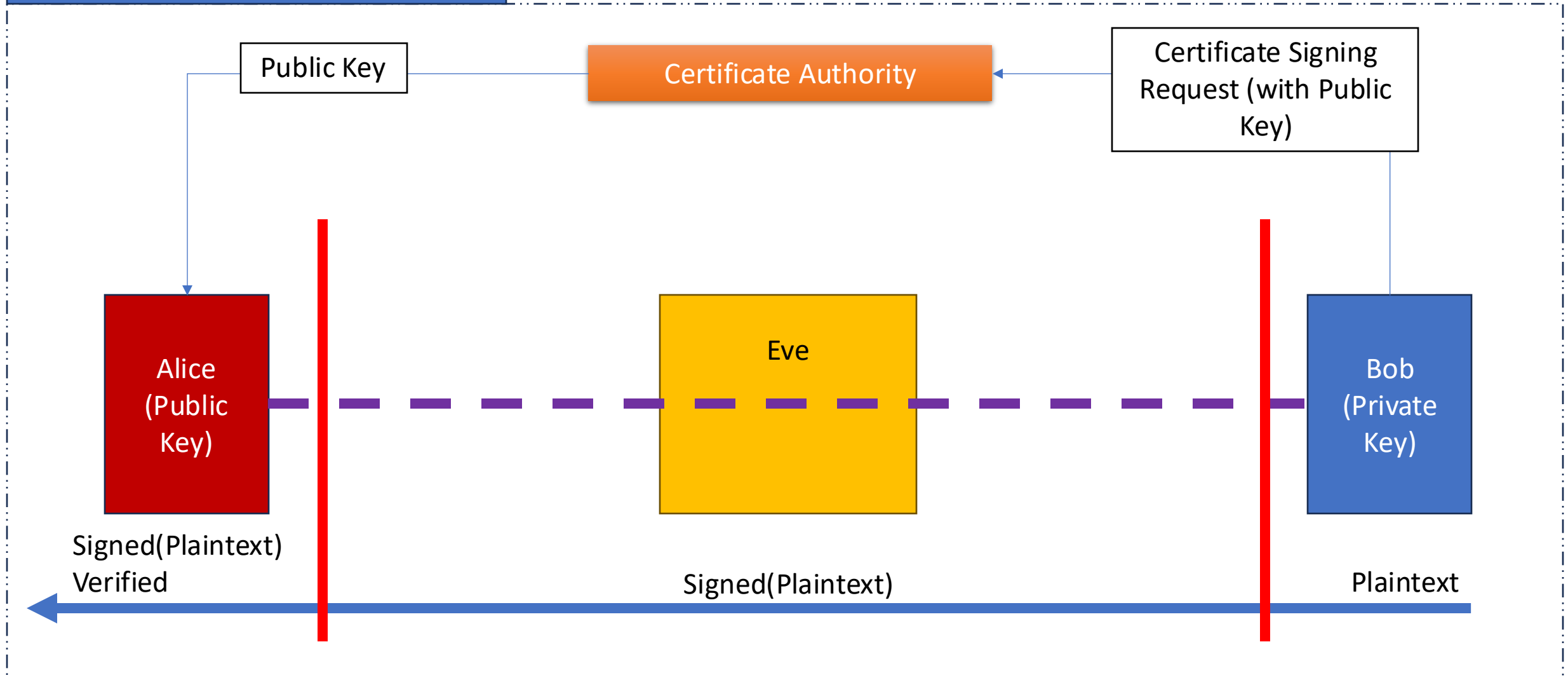
Intuition Derivation



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:
 $\delta = \text{Sign}(H(m), sk)$

Key Generation:
 $pk, sk = \text{KeyGen}(n)$

SIG

RSA, Again?

RSA Function

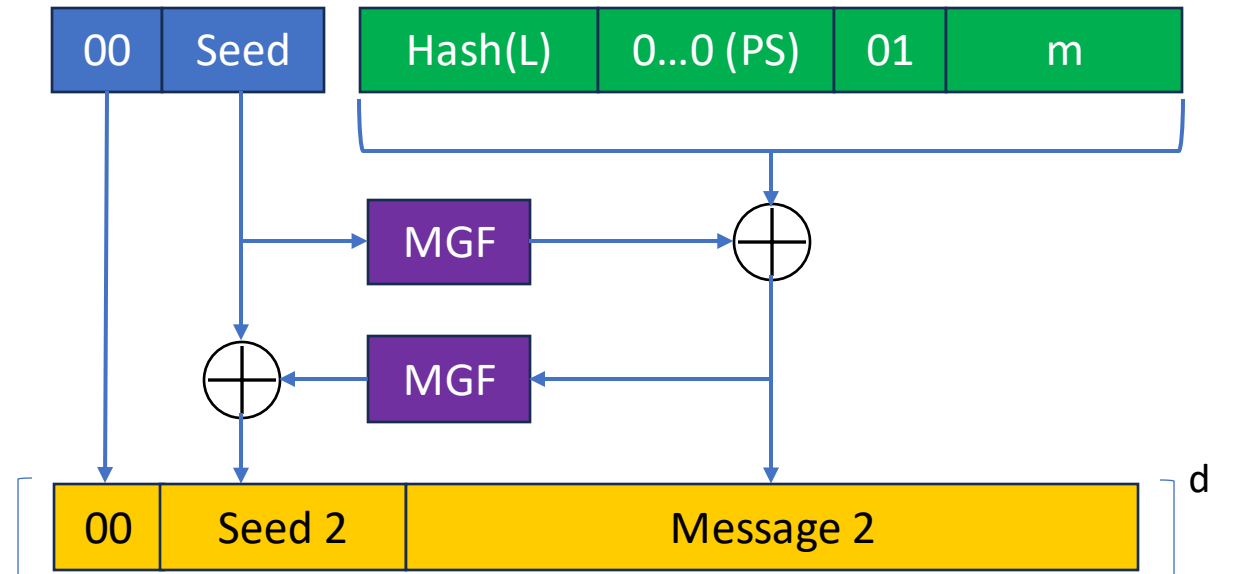
1. KeyGen:

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. $d = e^{-1} \text{ mod } \phi(N)$

2. Sign: $\forall m \in Z_n, m^d \text{ mod } N$

3. Ver: $\forall m \in Z_n, m^e \text{ mod } N$

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

Attacker

Round 0: Defender randomly creates keys (pk, sk) of size k :
 $(pk, sk) \leftarrow \text{KeyGen}(1^k), |pk|, |sk| \geq k$

Round 0: Defender sends public key: $\langle pk \rangle$

Round 1: Attacker sends a message of their choosing: $\langle m_1 \rangle$

Round 1: Defender sends a signature/tag of the message back: $\langle \delta_1 = \text{Sign}_{sk}(m_1) \rangle$

Round q: Attacker sends a message of their choosing: $\langle m_q \rangle$

Round q: Defender sends a signature/tag of the message back: $\langle \delta_q = \text{Sign}_{sk}(m_q) \rangle$

Post Rounds q+1: Attacker attempts to forge a signature/tag: $\langle m_{q+1}, \delta \rangle$

Post Rounds q+1: Defender says if attacker was correct ($b=1$) or not ($b=0$) and returns b :
 $\langle b = \text{Ver}_{pk}(m_{q+1}, \delta), b \in \{0,1\} \rangle$ subject to $m_{q+1} \notin \{m_1, m_2, \dots, m_q\}$

Definition: The digital signature scheme $\text{SIG} = (\text{KeyGen}, \text{Sign}, \text{Ver})$ is secure, or existentially unforgeable under an adaptive chosen message attack, if \nexists attacker who gains a non-negligible advantage in 'winning' game G , $\Pr[G = 1] > \text{negl}(k)$, subject to $m_{q+1} \notin \{m_1, m_2, \dots, m_q\}$, after adaptively analyzing polynomially bounded q rounds of message signature/tag pairs, $\langle m_1, \delta_1 \rangle$, $\langle m_2, \delta_2 \rangle$, ..., $\langle m_q, \delta_q \rangle$, given the public key pk .

Defender

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