Computer and Network Security

Lecture 05: Confidentiality

COMP-5370/6370
Fall 2024

WARNING

I AM NOT A CRYPTOGRAPHER

WARNING

YOU ARE NOT A CRYPTOGRAPHER

Properties of Secure Channel

A **secure channel** is a mechanism that allows Alice and Bob to communicate with the properties of:

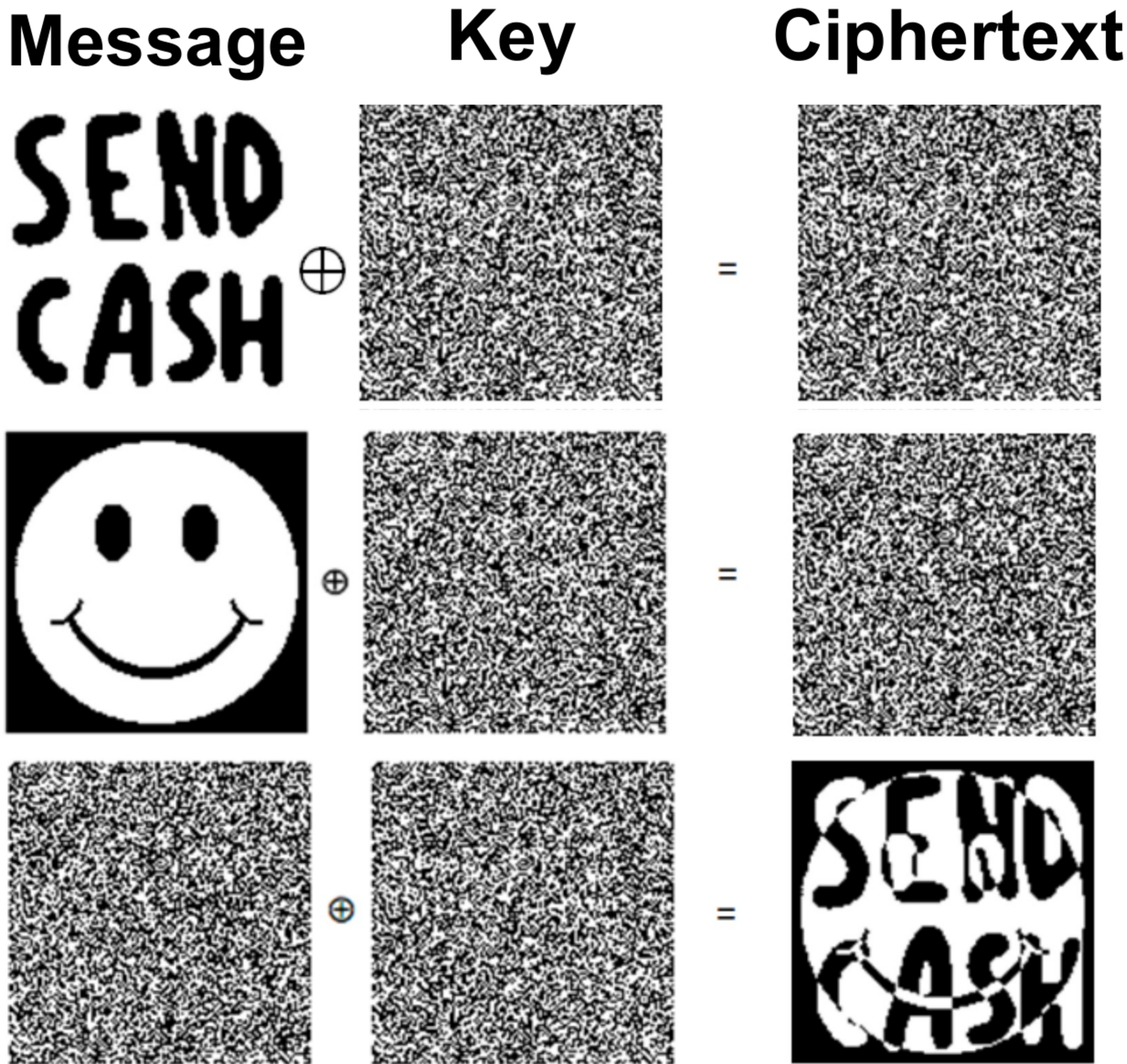
- **Confidentiality**
  - Messages can't be read by a 3rd party (3P)
- **Message Integrity**
  - Messages can't be unknowingly modified by 3P
- **Sender Authenticity**
  - Valid messages creatable **only** by a 1P actor

One-Time Pad

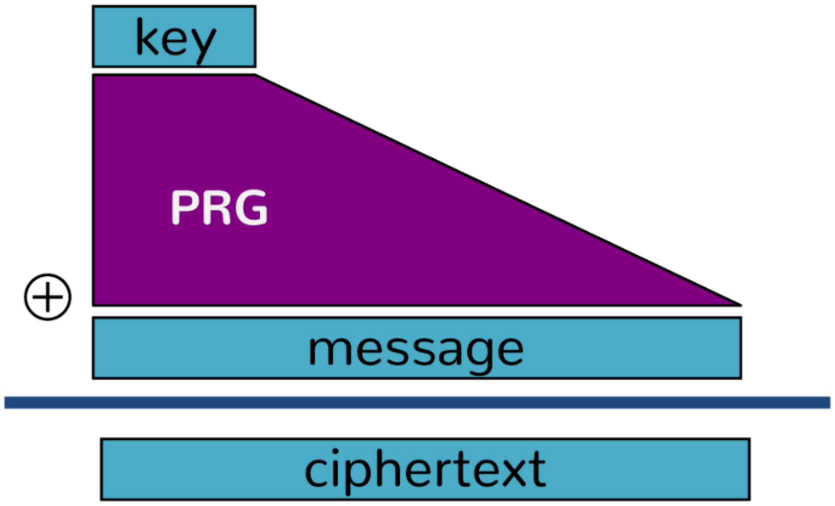
One-Time Pad is the only cryptosystem known to be unbreakable even infinite computational resources.

- `ct[i] = pt[i] XOR key[i]`
- Extremely fast to encrypt and decrypt
- Extremely easy to implement safely

N-Time Pad Leaks Information



Stream Cipher



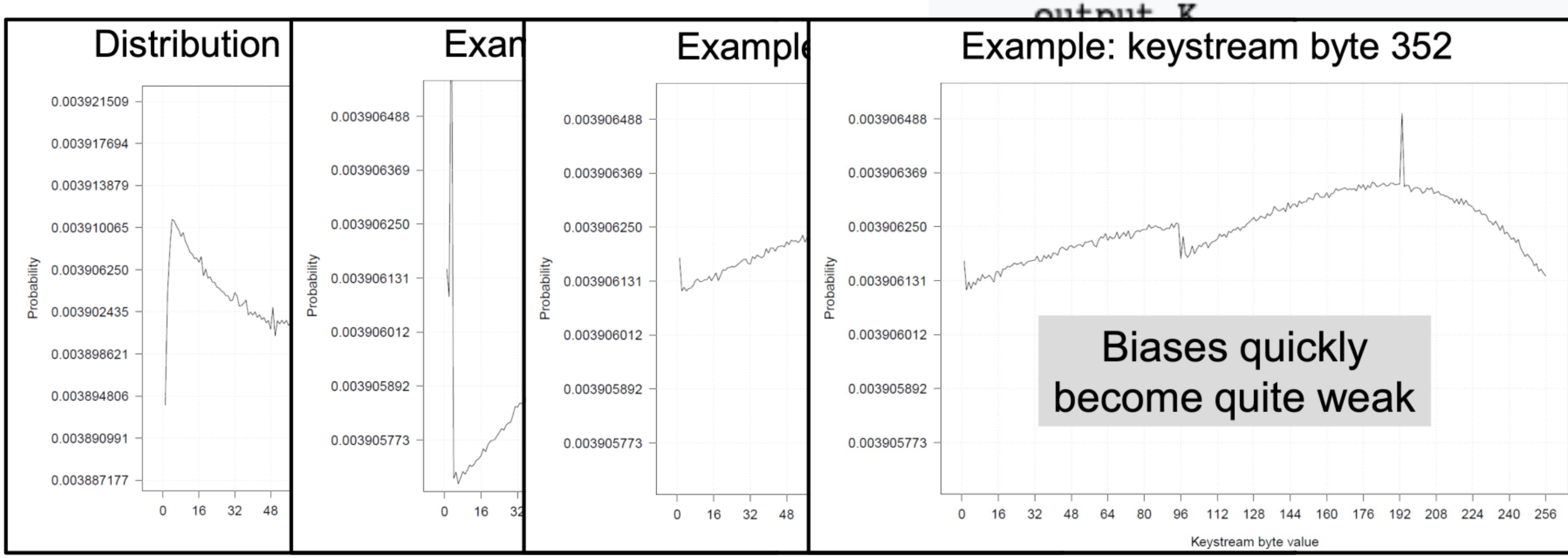
- Shared key known by all participants
- Key is “expanded” to the length of the message
  - PRNG

Infinite-Length One-Time Pad

RC4 Stream Cipher

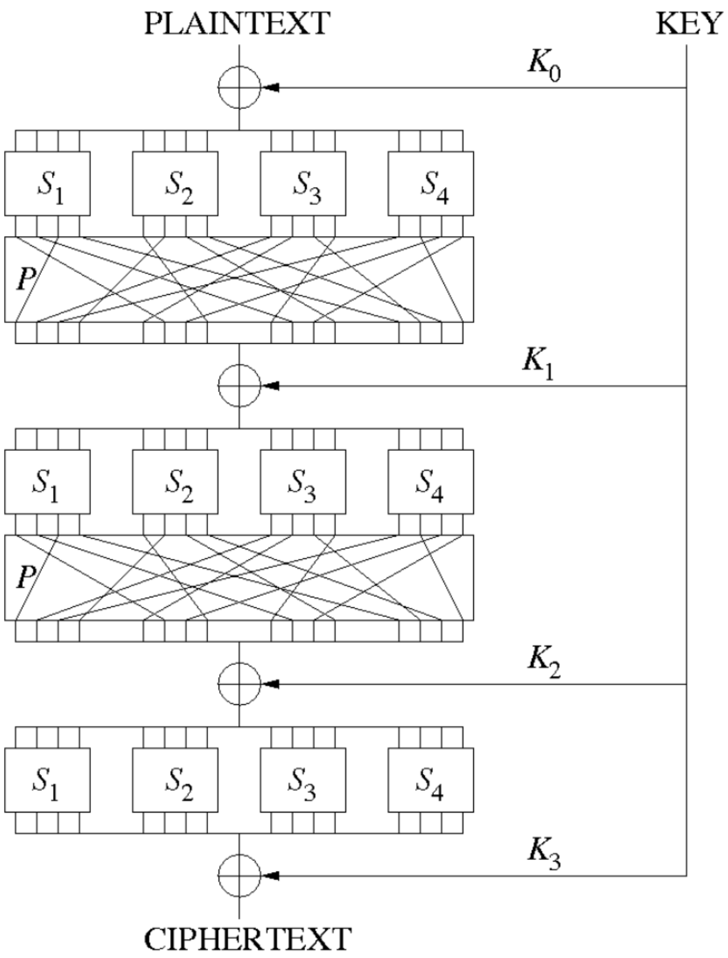
- Was widely used for speed and simplicity
- Should **not** be used

```
i := 0
j := 0
while GeneratingOutput:
    i := (i + 1) mod 256
    j := (j + S[i]) mod 256
    swap values of S[i] and S[j]
    K := S[(S[i] + S[j]) mod 256]
    output K
```



https://www.rc4nomore.com/

Block Cipher



- Fixed-size input
- Fixed-size output
- Substitutions from secret internal state
  - "S-Boxes"
- Multiple "rounds" to increase substitutions

DES – Data Encryption Standard

- 1977 – Standardized by NIST
  - NSA heavily involved in design
- 64-bit block cipher using 56-bit key
- Often implemented in hardware due to unneeded added complexity

- 1990 – Differential cryptanalysis discovered
  - General technique against block ciphers
- 1998 – EFF DES Cracker operational
  - Brute-force attack on key

DES – Data Encryption Standard

Never ever, ever, ever use single-DES

3DES – Triple DES

- 1995 – A “hot patch” for DES via RFC
- Exact same algorithm w/ different key-sched
  - Encrypt → decrypt → encrypt
- Best-case construction is 168-bit key
- Vulnerable to “meet-in-the-middle” attacks
  - Brute-force: 2^{56} space + 2^{112} operations
- 2016 – Practical collision attack (Sweet32)
  - DES is 64-bit block cipher ($2^{36.6}$ blocks needed)
  - “Got lucky” w/ 2^{20} block in 25 minutes vs. TLS

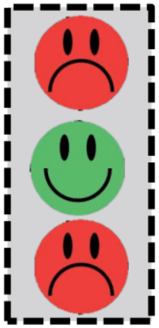
3DES – Triple DES

3DES is a weak cipher and should be immediately deprecated.

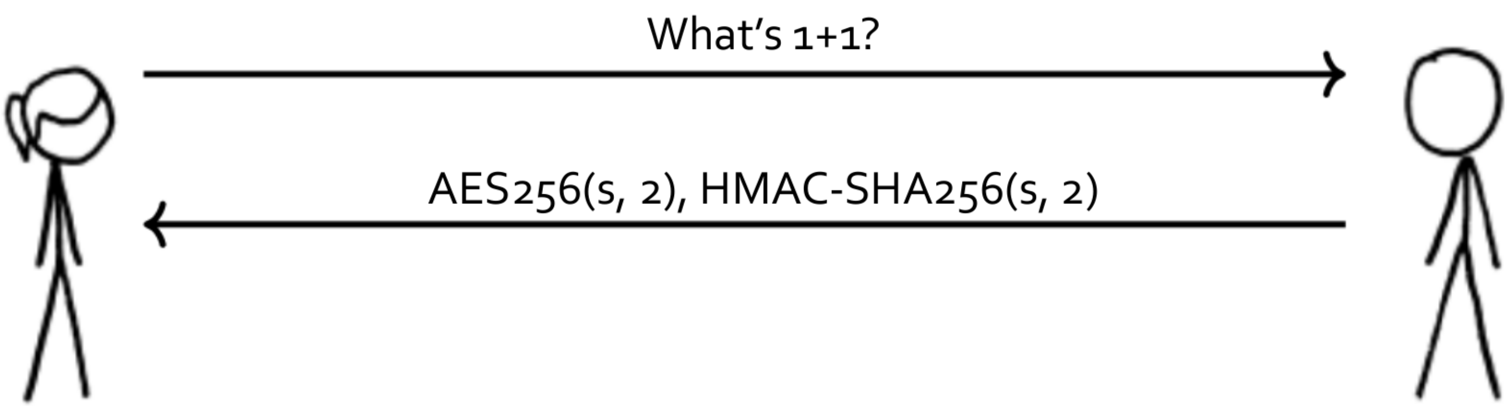
AES – Advanced Encryption Std

- 2001 – Standardized by NIST
- 128-bit block size
- 128/192/256-bit keys
  - Bigger key → same algorithm + more rounds
- Invertible S-boxes
  - Same used for both Encrypt() and Decrypt()
- AES-256 approved for CNSA
  - “Commercial National Security Algorithm Suite”
  - Encrypt TOP SECRET information and broadcast

Building a Secure Channel



Confidentiality
Message Integrity
Sender Authenticity



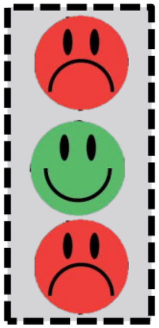
Key Derivation Function (KDF)

A **Key Derivation Function (KDF)** is one which can *safely* turn one shared-secret into multiple shared-secrets deterministically.

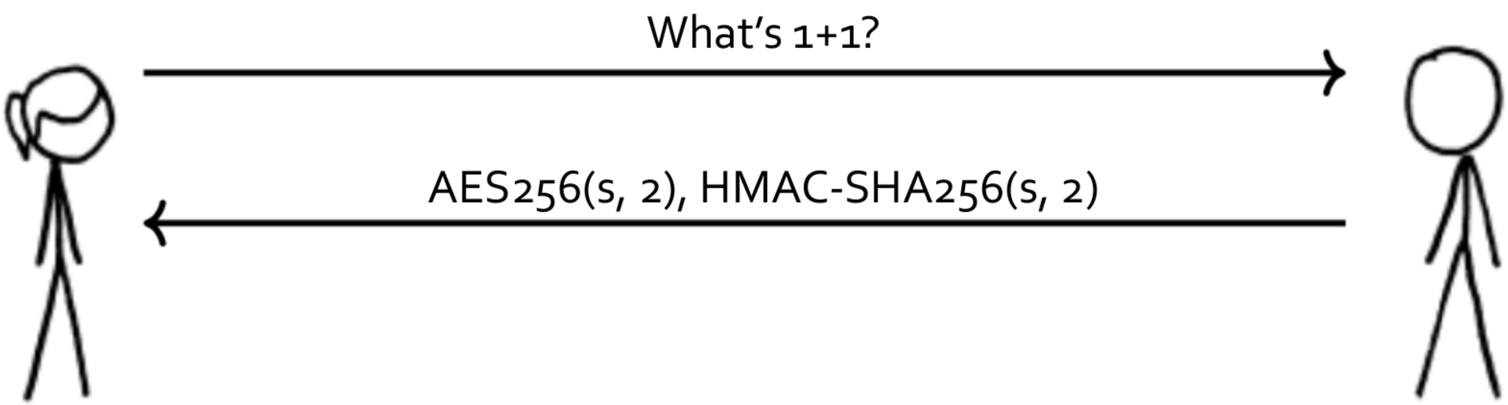
- HKDF is commonly used for protocols

```
PSK ->  HKDF-Extract = Early Secret
             |
             +-----> Derive-Secret(., "ext binder" | "res binder", "")
             |                    = binder_key
             |
             +-----> Derive-Secret(., "c e traffic", ClientHello)
             |                    = client_early_traffic_secret
             |
             +-----> Derive-Secret(., "e exp master", ClientHello)
             |                    = early_exporter_master_secret
             v
       Derive-Secret(., "derived", "")
```

Building a Secure Channel

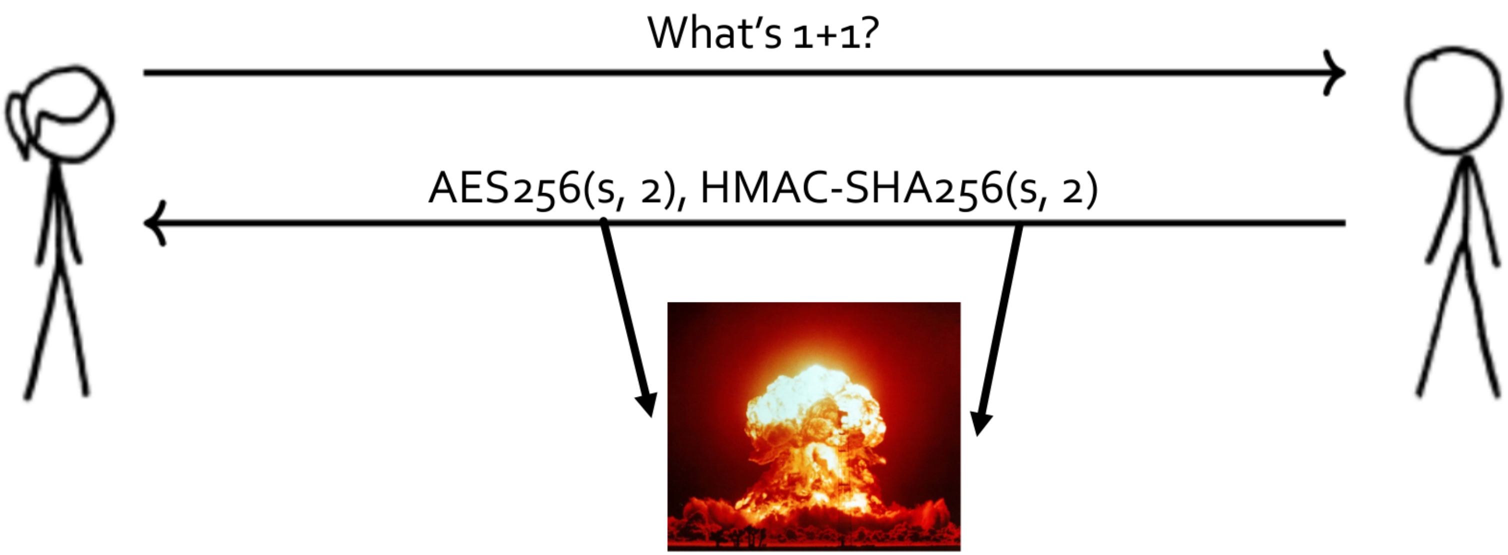


Confidentiality
Message Integrity
Sender Authenticity

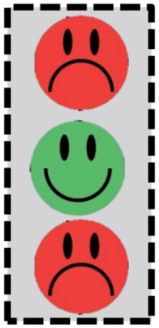


Problem 1

Re-using key material for different algorithms can reveal information about the key material's value.



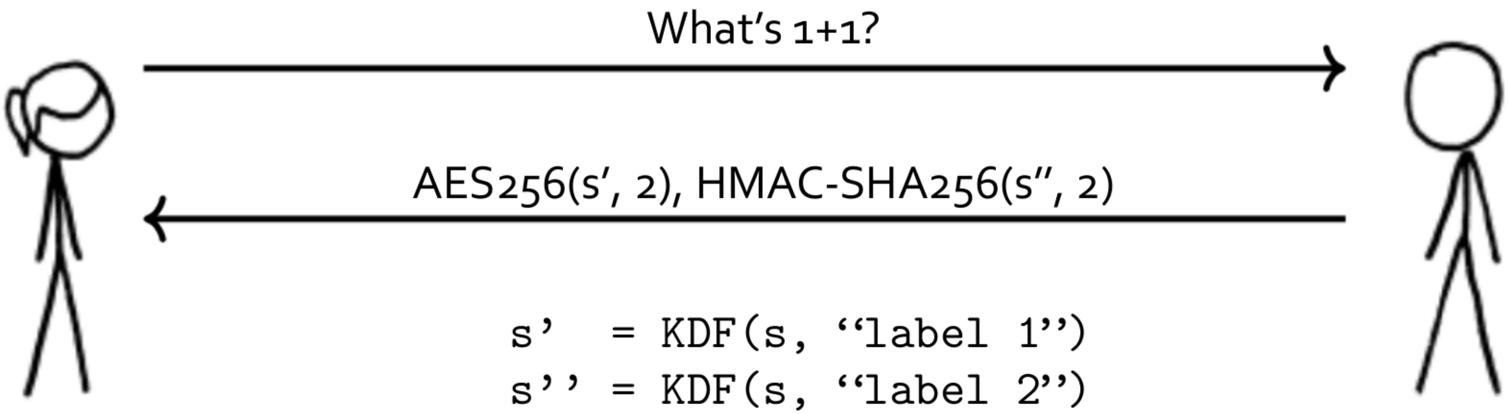
Building a Secure Channel



Confidentiality

Message Integrity

Sender Authenticity



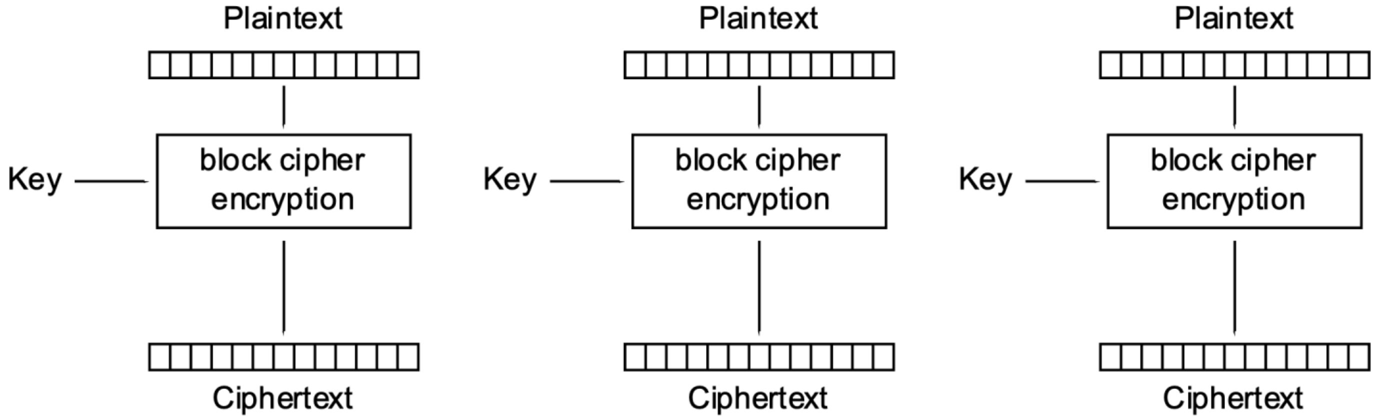
Cipher Mode

A **cipher mode** is a way to use a fixed-size block cipher with arbitrary-sized data.

- Needed for block-ciphers due to small cipher-width (AES256 == 256 bit blocks)
- Choice can heavily impact the performance of the cryptosystem

Electronic Codebook Mode (ECB)

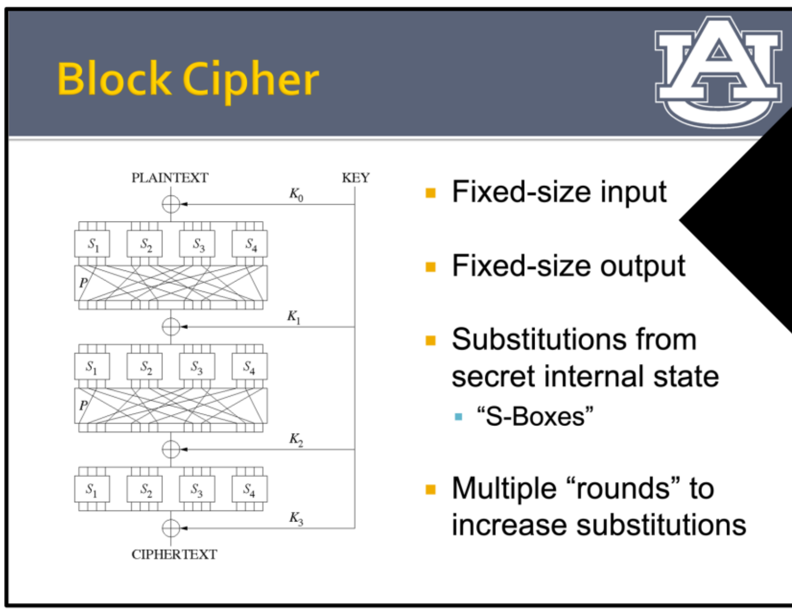
- Pad last block to correct length
- Each block of plaintext fed through cipher independently of all others
- Embarrassingly parallel, random access



Electronic Codebook (ECB) mode encryption

Problem 2

Block ciphers are fixed-length inputs/outputs and messages are … not.



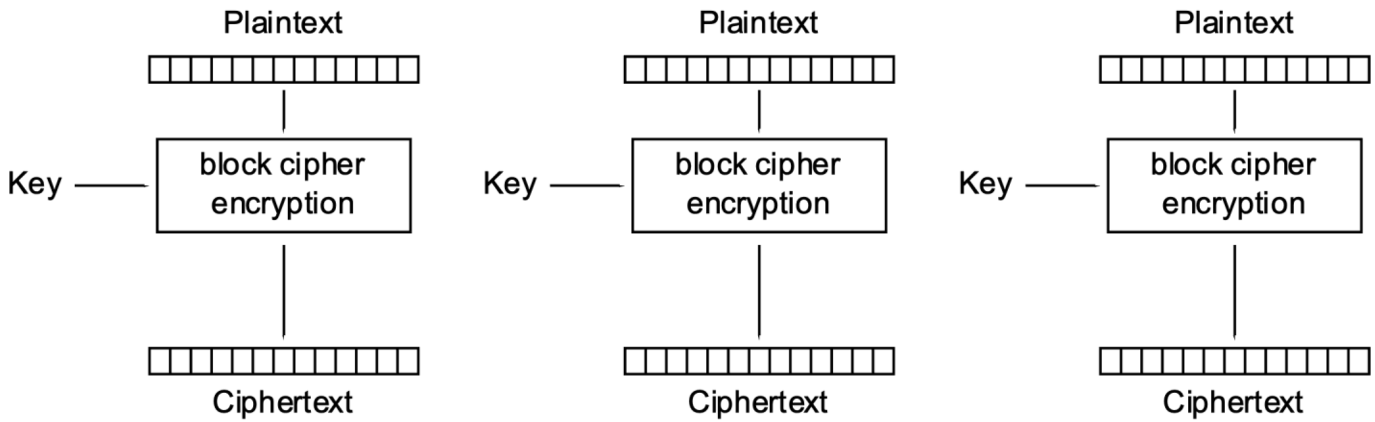
Cipher Mode

A **cipher mode** is a way to use a fixed-size block cipher with arbitrary-sized data.

- Needed due to small/fixed cipher-width (AES256 == 256 bit blocks)
- **Choice can heavily impact the performance of the cryptosystem**

Electronic Codebook Mode (ECB)

- Pad last block to correct length
- Each block of plaintext fed through cipher independently of all others
- Embarrassingly parallel, random access



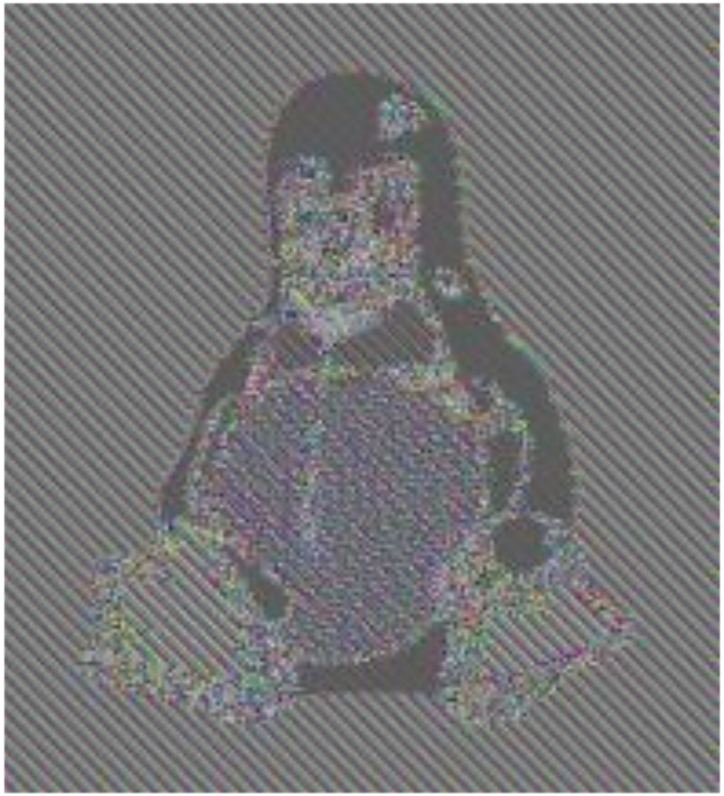
Electronic Codebook (ECB) mode encryption

Electronic Codebook Mode (ECB)

Since the only inputs to the cipher are the plaintext and the key material, identical PT blocks encrypt to identical CT blocks.

AAABBBAAA → UVWXYZUVW
AAA → UVW
BBB → XYZ
AAA → UVW

Electronic Codebook Mode (ECB)



Initialization Vector

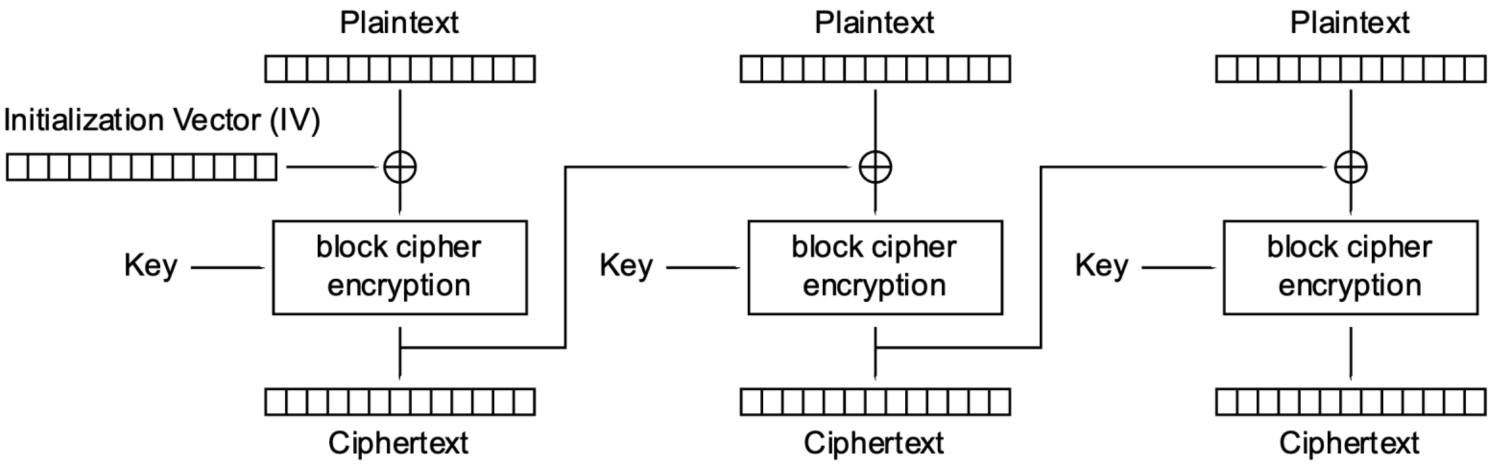
An **Initialization Vector (IV)** is an additional, non-secret input provided to the cipher to remove identical CT leaking data about PT.

- Must be known to Alice and Bob but **is not** required to be secret
- Often called a "nonce"

$\mathtt{n}_{\mathtt{once}} \rightarrow \mathtt{nonce}$

Cipher Block Chaining (CBC)

- IV is the previous block's CT
- Pad last block in a deterministic way
  - AES-128 24-byte message = 8x 0x08 padding
  - AES-128 30-byte message = 2x 0x02 padding



Cipher Block Chaining (CBC) mode encryption

CBC Padding Oracle

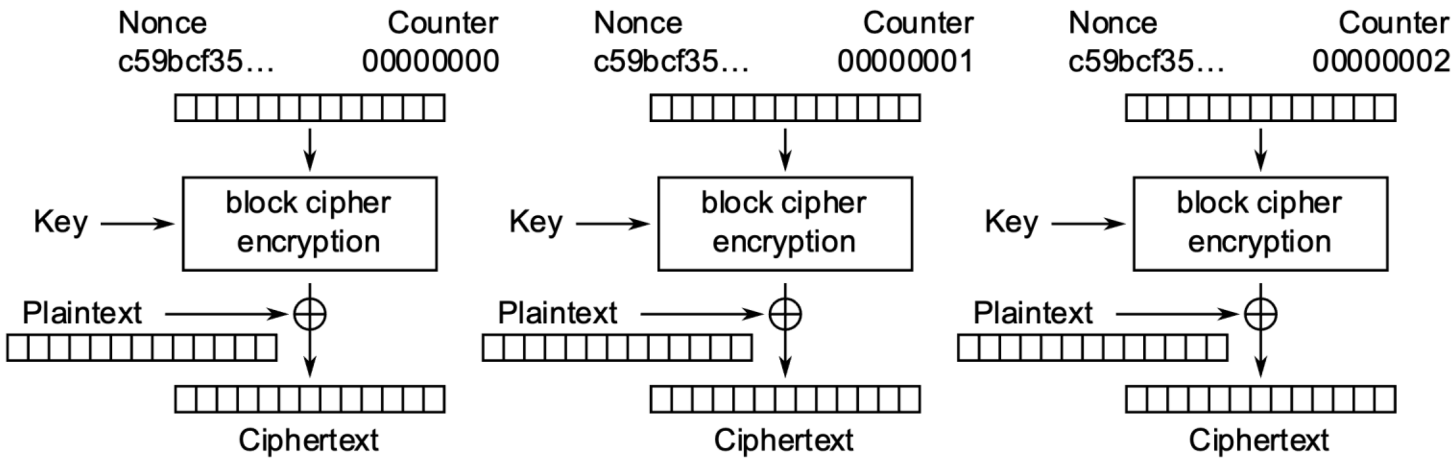
CBC mode usually vulnerable to **padding oracle attacks** due to the difficulty of handling the padded block.

- Extremely easy to leak internal cipher state
- Writing safe software is hard
- Writing safe security-related software is *really, really hard*
- Writing safe crypto-software is one of the reasons **we don't roll our own crypto**

CBC Padding Oracle

Counter Mode (CTR)

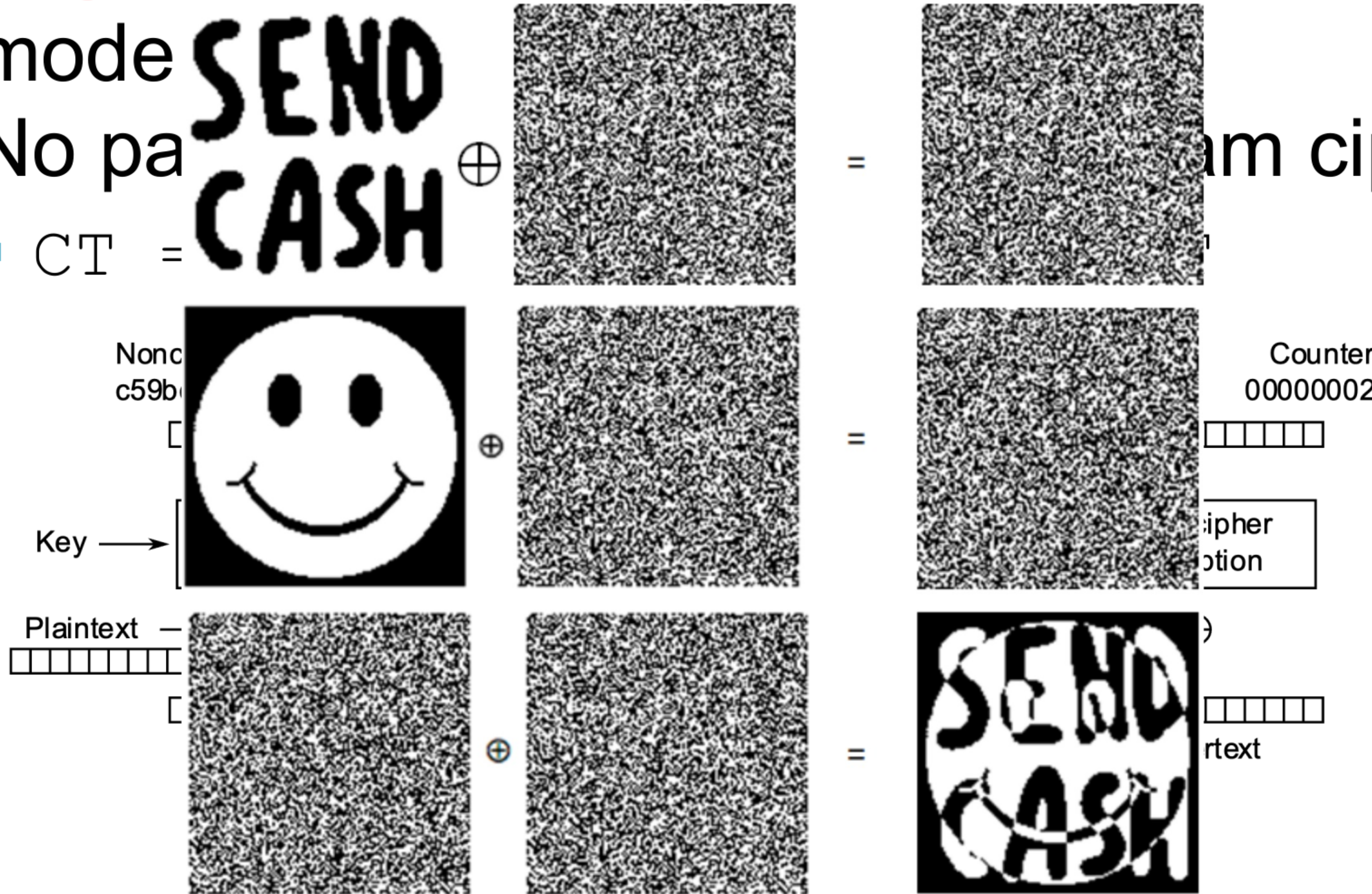
- Key-unique nonce || counter to avoid ECB mode inter-block leakage
- No padding because used as stream cipher
  - `CT = Encrypt(key, IV) XOR PT`



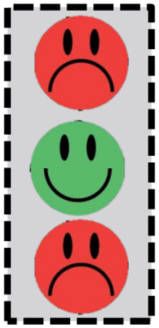
Counter (CTR) mode encryption

Counter Mode (CTR)

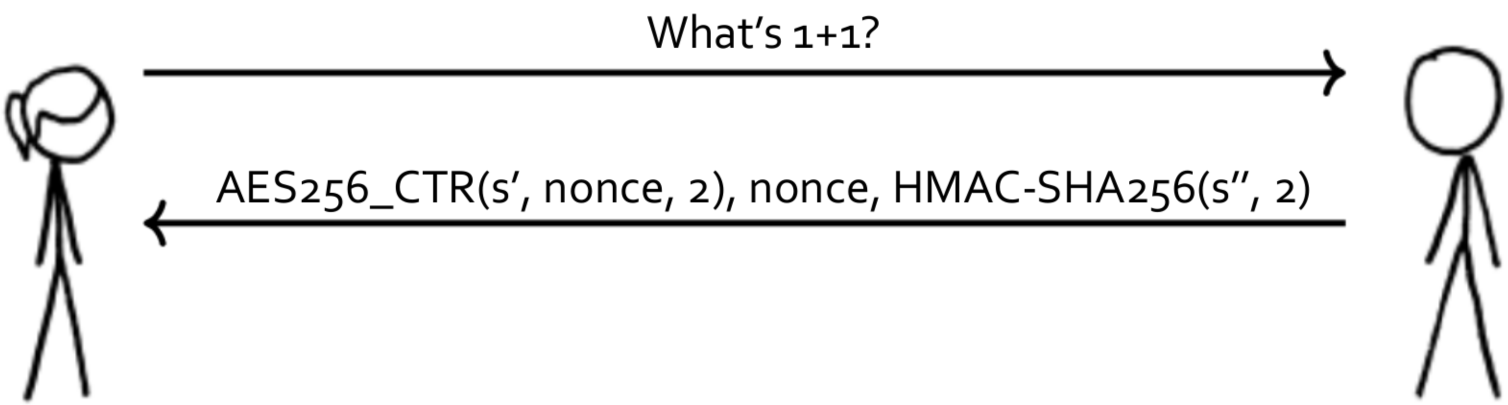
- **Key-unique** nonce || counter to avoid ECB mode
- No pa am cipher
  - CT =



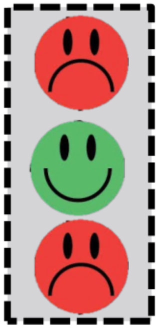
Building a Secure Channel



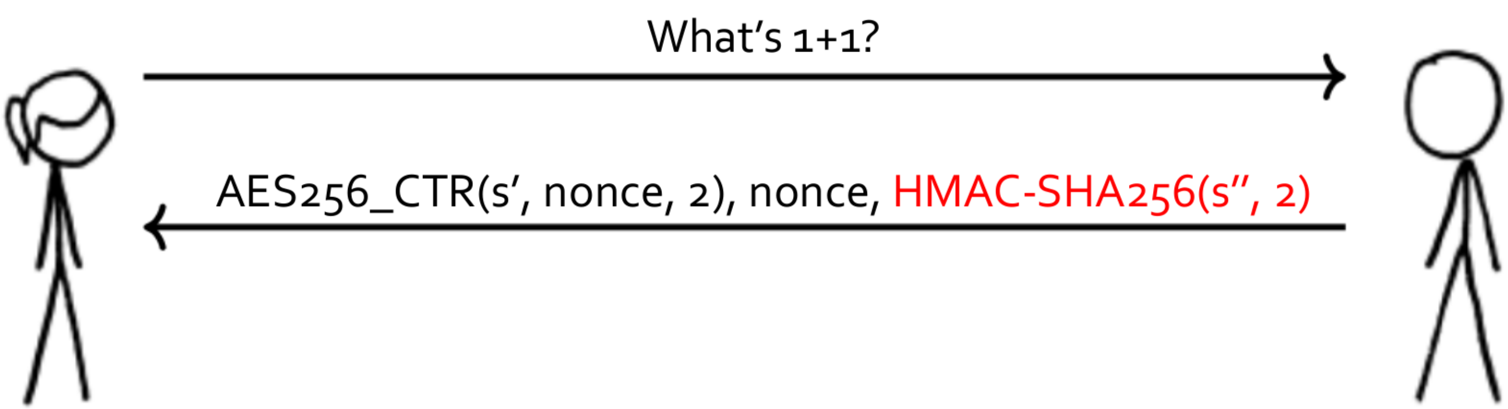
Confidentiality
Message Integrity
Sender Authenticity



Building a Secure Channel



Confidentiality
Message Integrity
Sender Authenticity

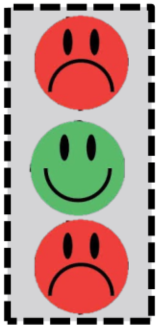


Cryptographic Doom Principle

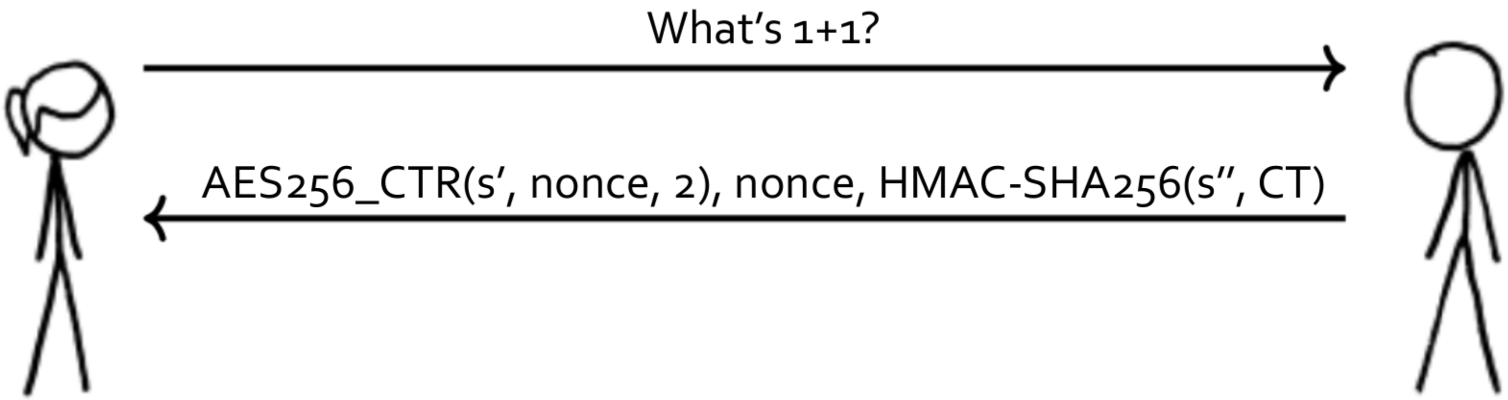
If you have to perform **any** cryptographic operation before verifying the MAC on a message you've received, it will **somehow** inevitably lead to doom.

-Moxie Marlinspike

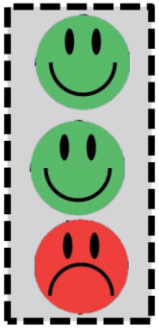
Building a Secure Channel



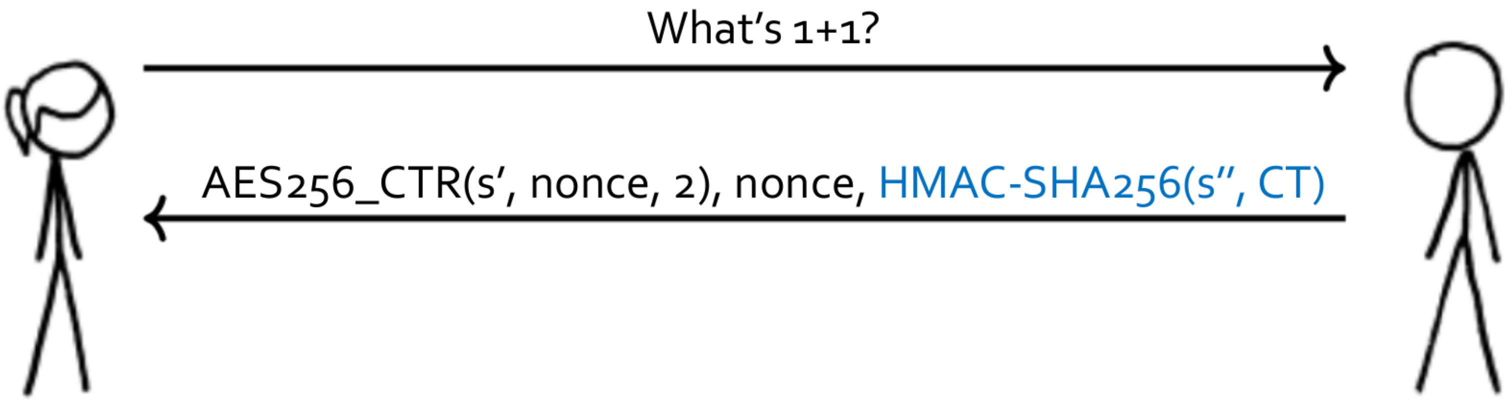
Confidentiality
Message Integrity
Sender Authenticity



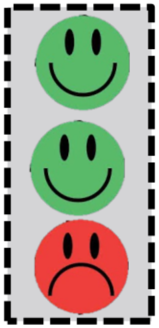
Building a Secure Channel



Confidentiality
Message Integrity
Sender Authenticity



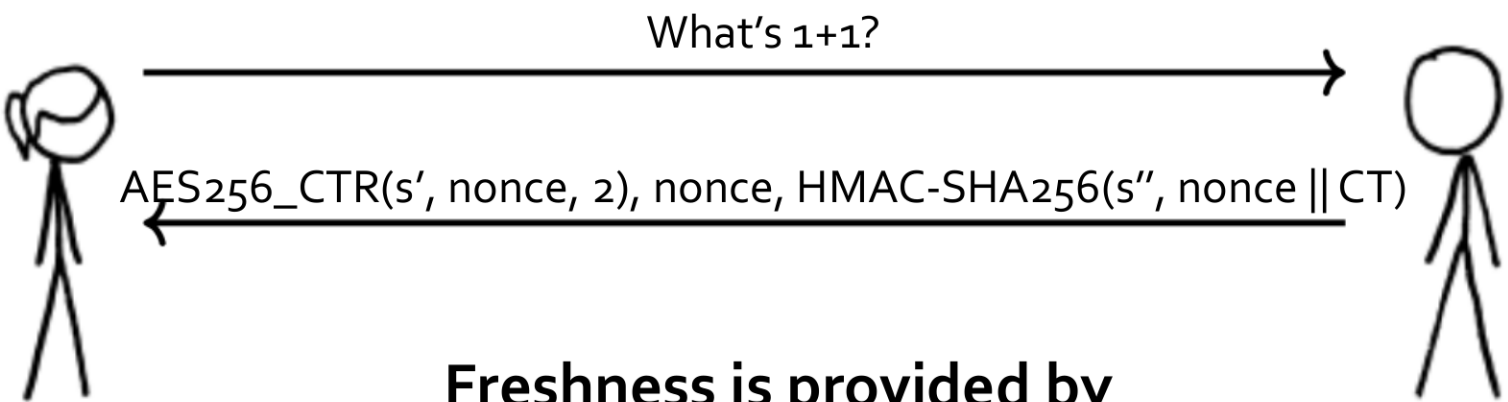
Building a Secure Channel



Confidentiality

Message Integrity

Sender Authenticity



Freshness is provided by CTR-mode counter.

AEAD Cipher Modes

Authenticated Encryption with Associated Data (AEAD) cipher modes provide confidentiality and message integrity simultaneously.

- Provides **confidentiality**
- Provides **message integrity**
- Does **not** provide **sender authenticity**
- Commonly use `seal()` and `unseal()` instead of `encrypt()` and `decrypt()`

The “AD” in AEAD

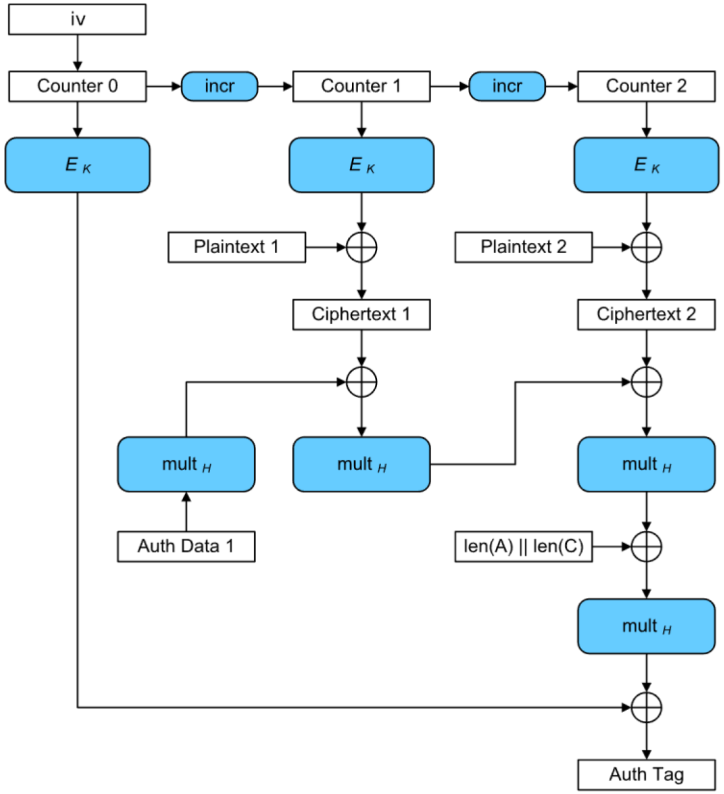
AEAD cipher modes allow some data (the “Associated Data”) to be *authenticated but not encrypted*.

- `CT ← Seal(key, nonce, PT, AD)`

- To recover & validated PT, must have CT, key, nonce, **and** AD

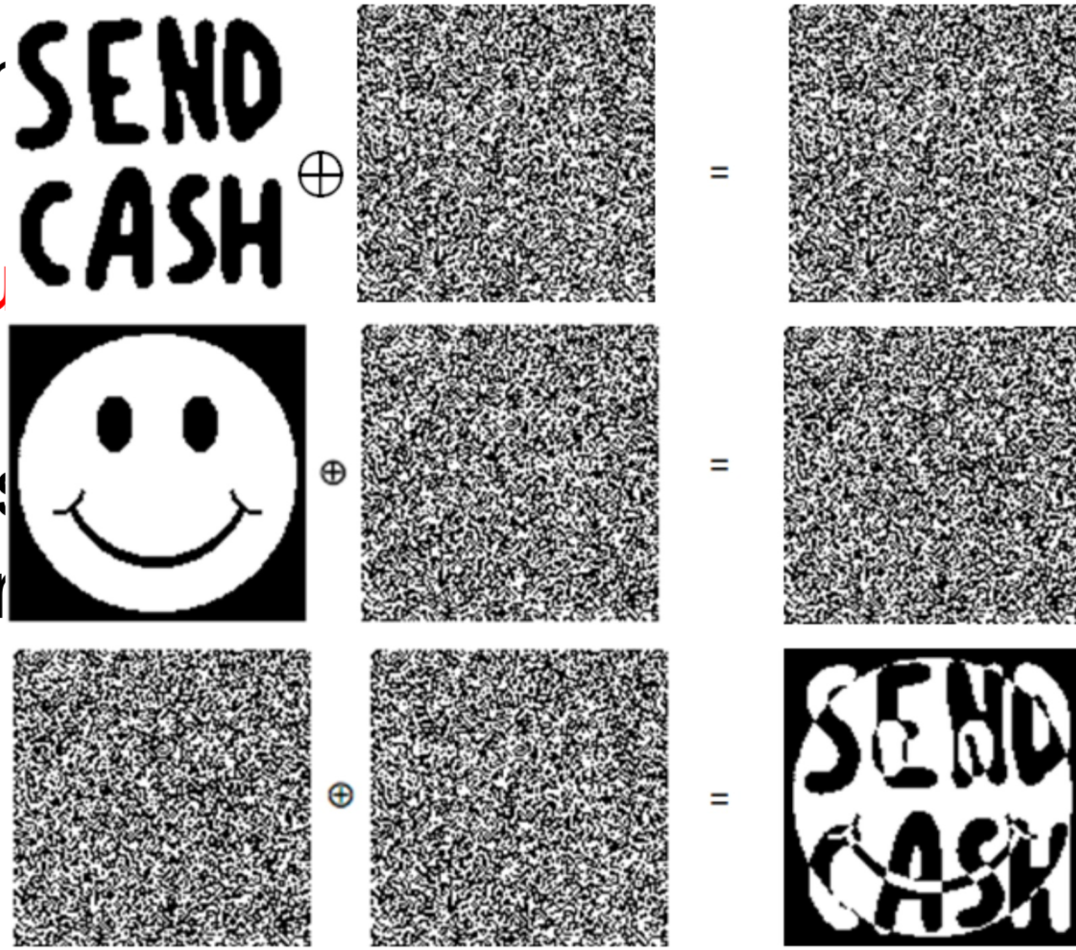
Galois/Counter Mode (GCM)

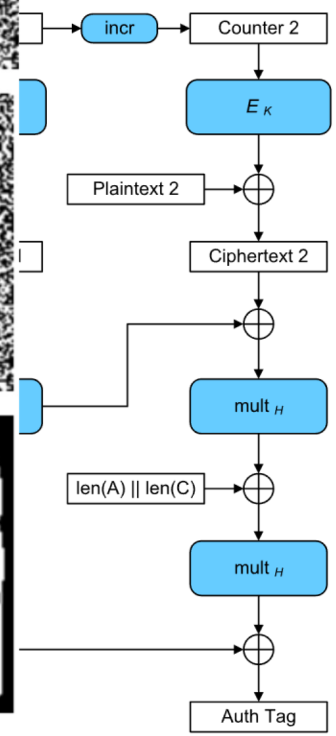
- CTR mode with built-in integrity checking
- Key-unique IV
- Makes protocols much easier to implement



Galois/Counter Mode (GCM)

- CTR r ... ecking
- **Key-u**
- Makes ... easier





AES-GCM-SIV

- Nonce misuse-resistant version of GCM
- Still provides confidentiality and message integrity in single abstraction
- Low-/Early-adoption (very recent)

AES-GCM-SIV: Specification and Analysis

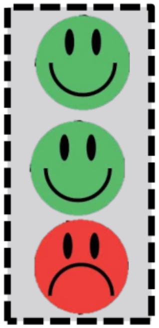
Shay Gueron[1], Adam Langley[2], and Yehuda Lindell[3]*

[1] University of Haifa, Israel and Amazon Web Services
[2] Google, Inc.
[3] Bar-Ilan University, Israel

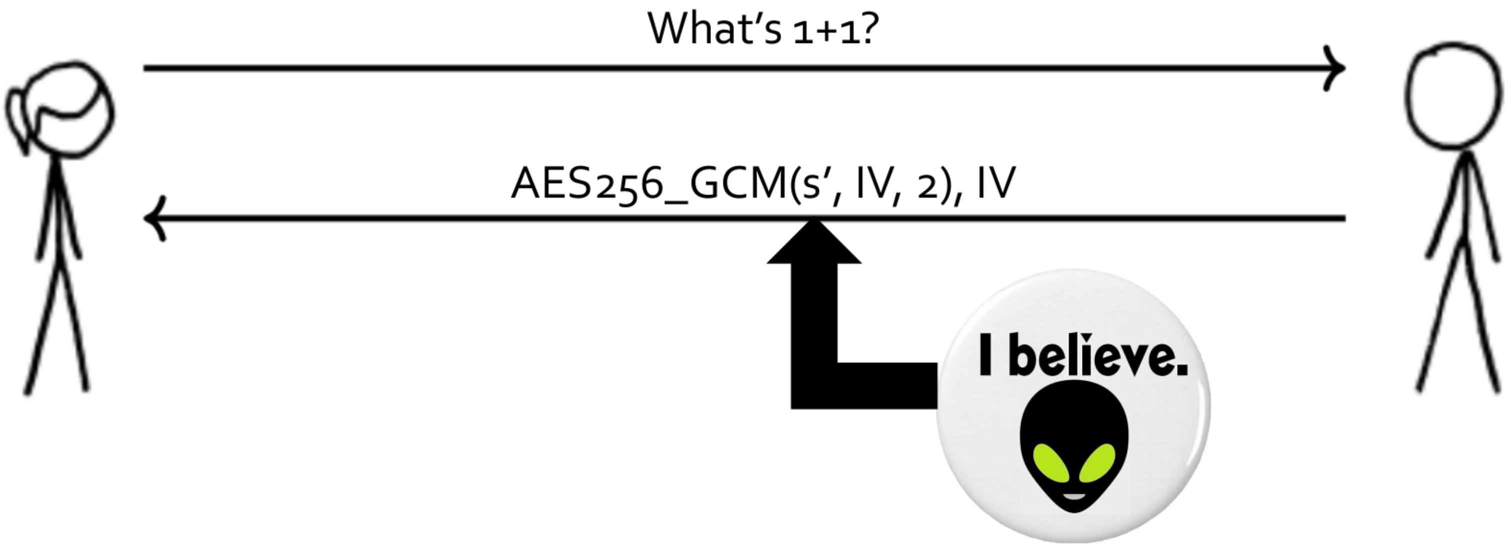
December 14, 2018

Abstract. In this paper, we describe and analyze the security of the AES-GCM-SIV mode of operation, as defined in the CFRG specification [10]. This mode differs from the original GCM-SIV mode that was designed in [11] in two main aspects. First, the CTR encryption uses a 127-bit pseudo-random counter instead of a 95-bit pseudo-random value concatenated with a 32-bit counter. This construction leads to improved security bounds when encrypting short messages. In addition, a new key derivation function is used for deriving a fresh set of keys for each nonce. This addition allows for encrypting up to 2^{50} messages with the same key, compared to the significant limitation of only 2^{32} messages that were allowed with GCM-SIV (which inherited this same limit from AES-GCM). As a result, the new construction is well suited for real world applications that need a nonce-misuse resistant Authenticated Encryption scheme. We explain the limitations of GCM-SIV, which motivate the new construction, prove the security properties of AES-GCM-SIV, and show how these properties support real usages. Implementations are publicly available in [8]. We remark that AES-GCM-SIV is already integrated into Google's BoringSSL library [1] and is deployed for ticket encryption in QUIC [17].

Building a Secure Channel



Confidentiality
Message Integrity
Sender Authenticity



Computer and Network Security

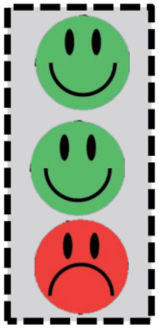
Lecture 05:
KEX & Asymmetric Operations

COMP-5370/6370
Fall 2024

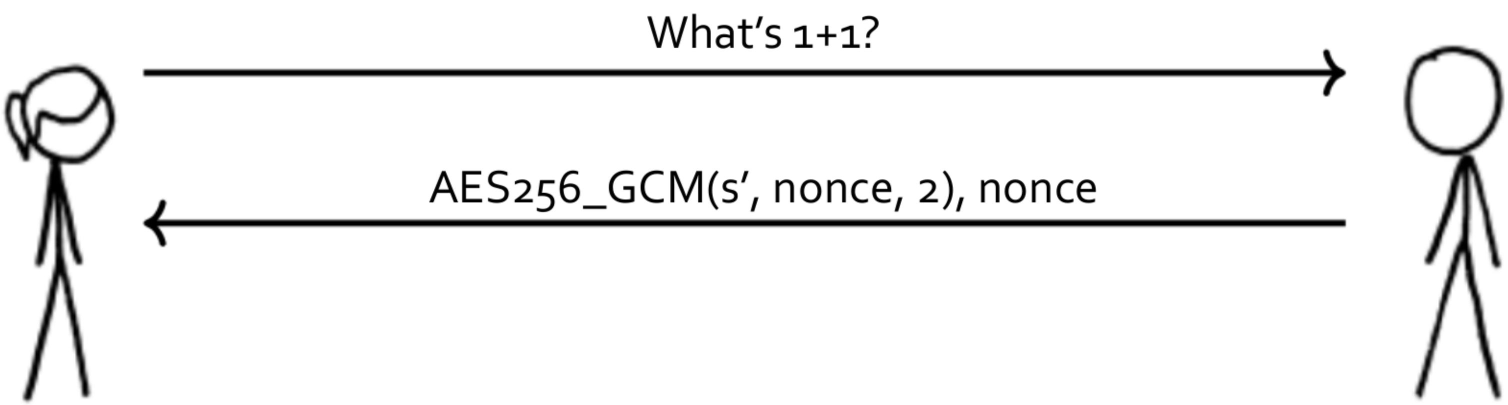
Key Distribution Problem

Key Distribution Problem is the generic name used to reference real-world challenges from a nominally simple need.

Building a Secure Channel



- Confidentiality
- Message Integrity
- Sender Authenticity

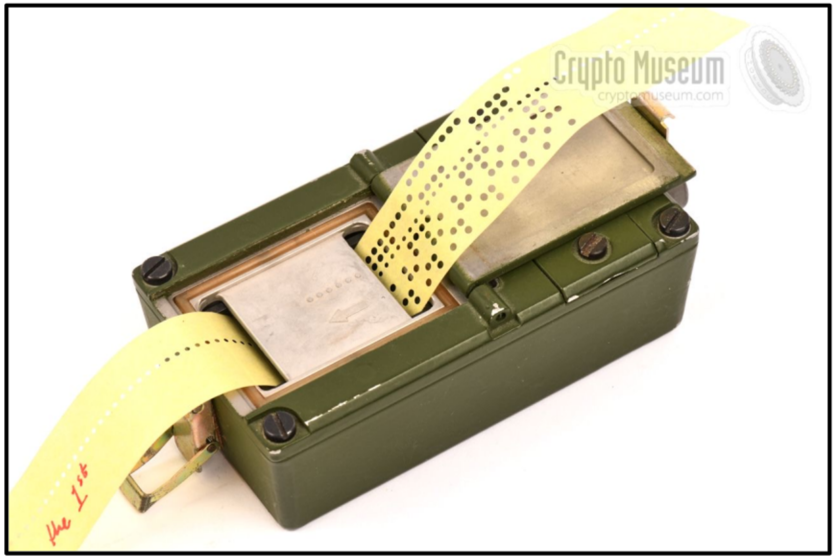


Symmetric Keys

A **symmetric key** is key that is identical for all parties involved.

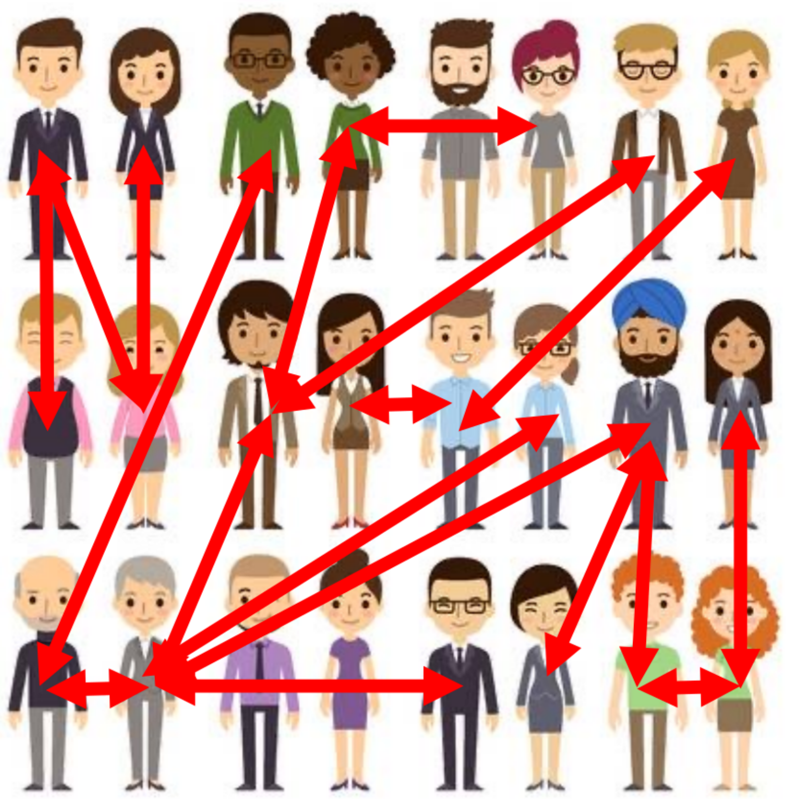
EXAMPLE:

- AES cipher key
- HMAC key
- Any “shared secret”



Key Distribution

Key Distribution

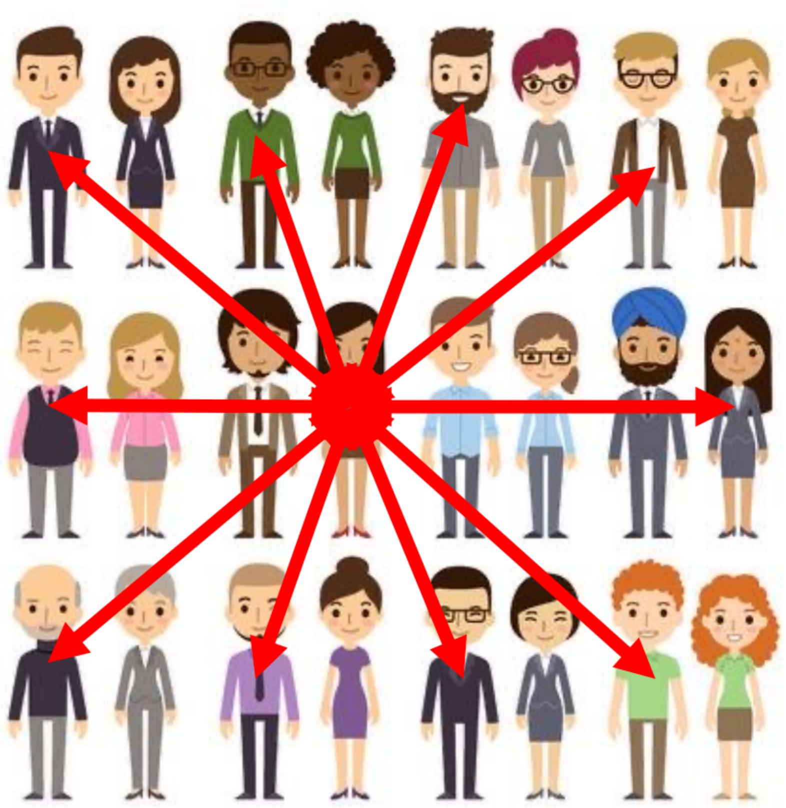


- Ad hoc independent
  - People are bad at predicting and planning

Key Distribution

- Ad hoc independent
  - People are bad at predicting and planning
- Transitive trust
  - Who do you trust?

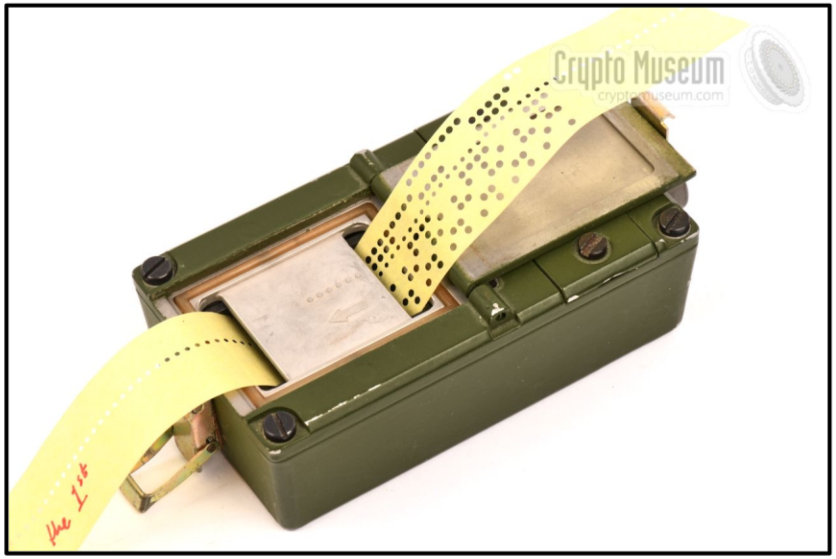
Key Distribution



- Ad hoc independent
  - People are bad at predicting and planning
- Transitive trust
  - Who do you trust?
- Centralized issuance
  - Single point of trust
  - Single point of failure

Symmetric Keys

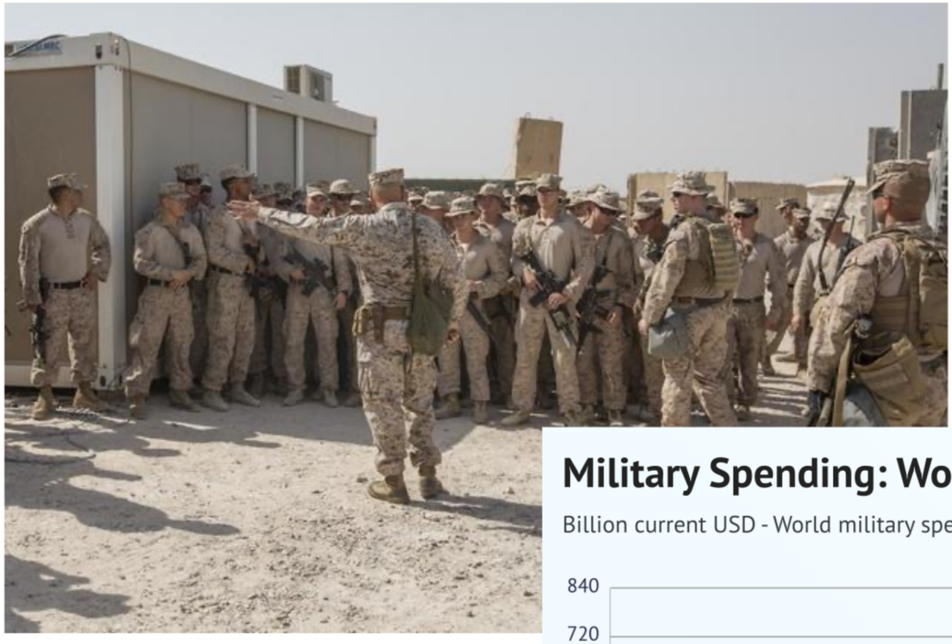
A **symmetric key** is key that is identical for all parties involved.

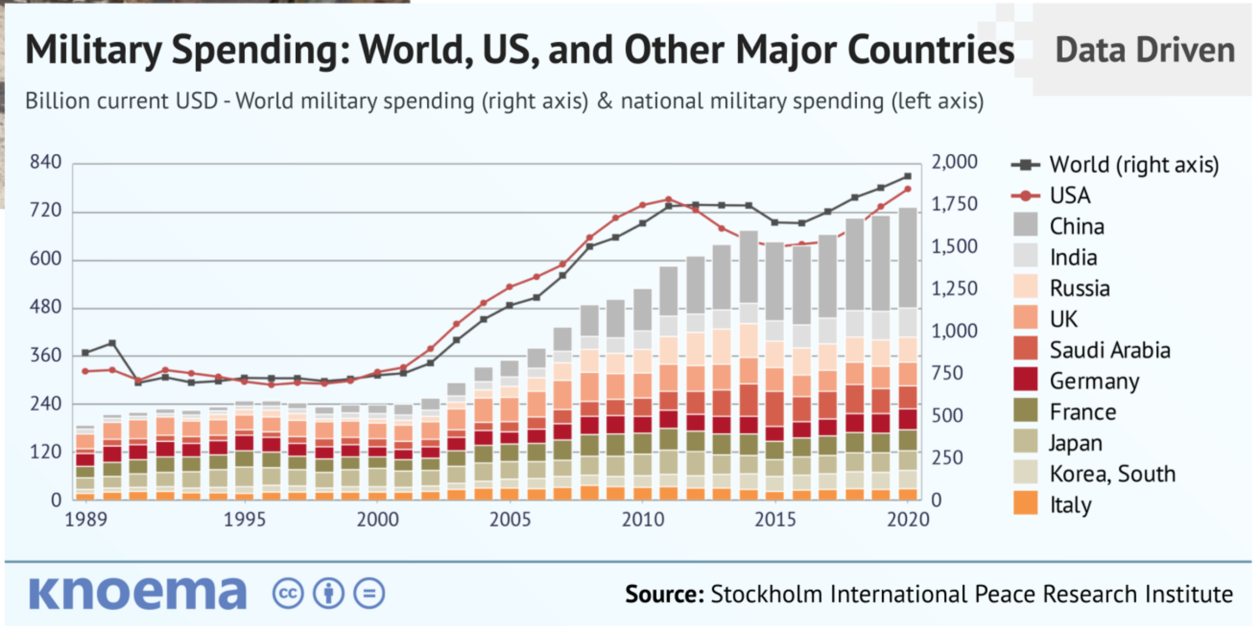


EXAMPLE:

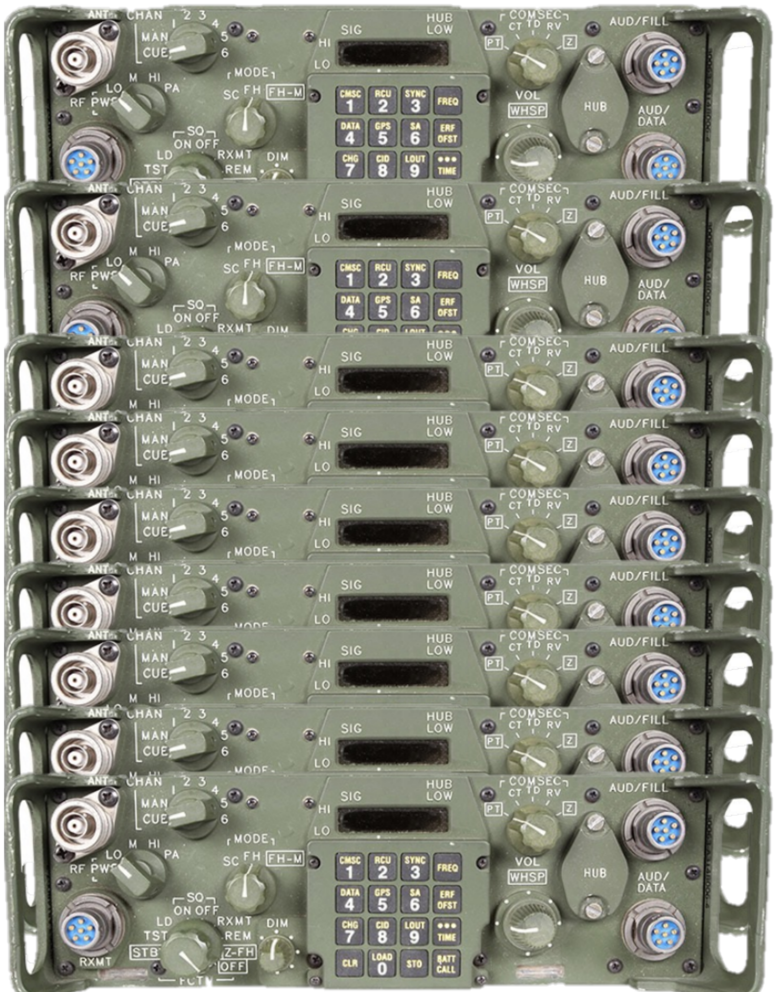
- AES cipher key
- HMAC key
- Any "shared secret"

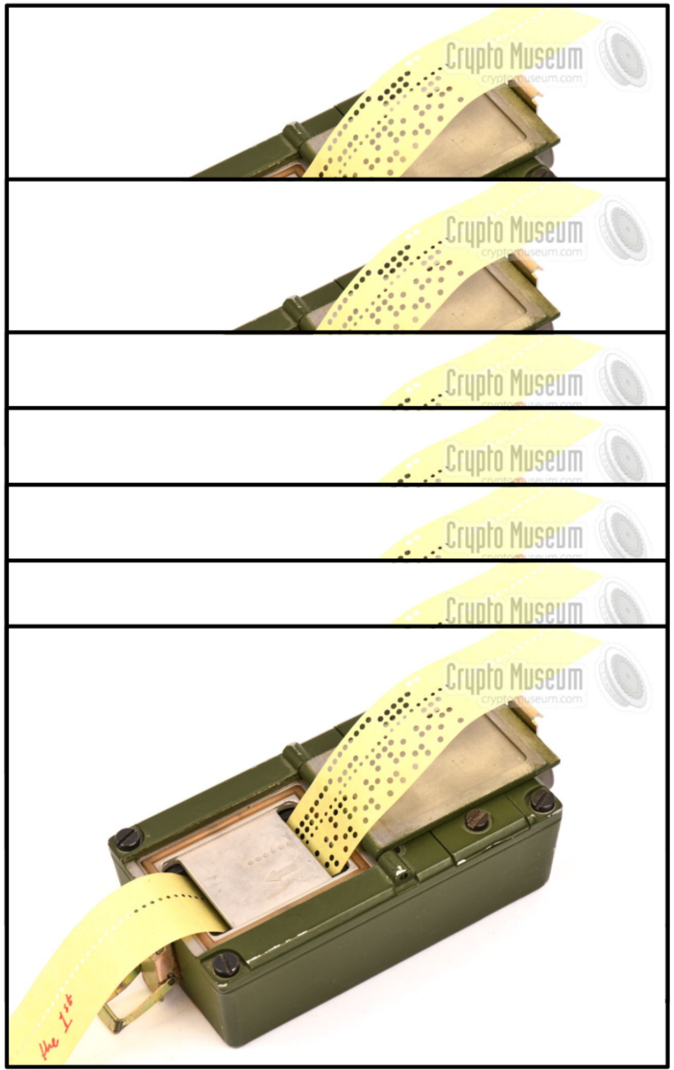
Real-World Key Distribution





Real-World Key Distribution





Real-World Key Distribution

AN/CYZ-10

The AN/CYZ-10 is the full keyboard version and the AN/CYZ-10A is the limited keyboard version of the DTD.

https://slideplayer.com/slide/6637664/

Real-World Key Distribution

Real-World Key Distribution

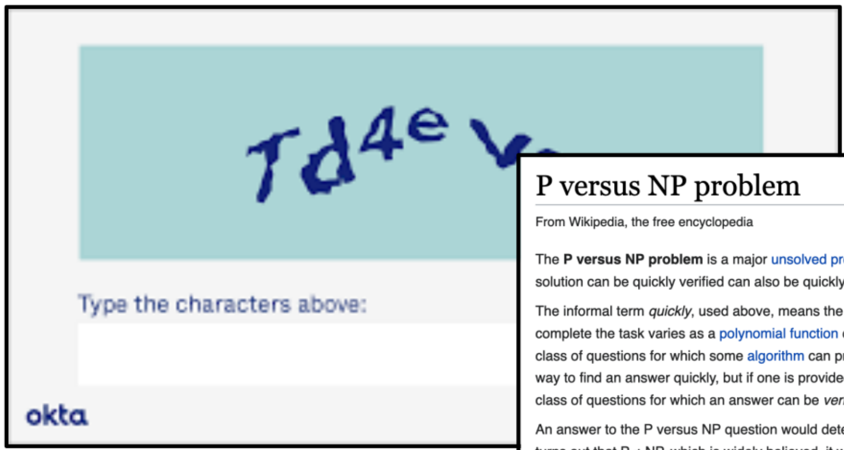
Key Distribution Problem

Key Distribution Problem is the generic name used to reference real-world challenges to values being shared by the actors manually or *out of band*.

- Is well-known and widely maligned
- Directly applicable to shared secrets
- Also applicable to non-secret provenance

Security Analysis Revisited

- Attackers should be fundamentally limited in what they can-do not what they should-do
  - Things that are “computationally infeasible” or “fundamental unknown”



P versus NP problem

From Wikipedia, the free encyclopedia

The **P versus NP problem** is a major unsolved problem in theoretical computer science. In informal terms, it asks whether every problem whose solution can be quickly verified can also be quickly solved.

The informal term *quickly*, used above, means the existence of an algorithm solving the task that runs in polynomial time, such that the time to complete the task varies as a polynomial function on the size of the input to the algorithm (as opposed to, say, exponential time). The general class of questions for which some algorithm can provide an answer in polynomial time is "**P**" or "**class P**". For some questions, there is no known way to find an answer quickly, but if one is provided with information showing what the answer is, it is possible to verify the answer quickly. The class of questions for which an answer can be *verified* in polynomial time is **NP**, which stands for "nondeterministic polynomial time".[Note 1]

An answer to the P versus NP question would determine whether problems that can be verified in polynomial time can also be solved in polynomial time. If it turns out that P ≠ NP, which is widely believed, it would mean that there are problems in NP that are harder to compute than to verify: they could not be solved in polynomial time, but the answer could be verified in polynomial time.

The problem has been called the most important open problem in computer science.[1] Aside from being an important problem in computational theory, a proof either way would have profound implications for mathematics, cryptography, algorithm research, artificial intelligence, game theory, multimedia processing, philosophy, economics and many other fields.[2]

It is one of the seven Millennium Prize Problems selected by the Clay Mathematics Institute, each of which carries a US$1,000,000 prize for the first correct solution.

Unsolved problem in computer science:

? *If the solution to a problem is easy to check for correctness, must the problem be easy to solve?*

(more unsolved problems in computer science)

Millennium Prize Problems

Birch and Swinnerton-Dyer conjecture
Hodge conjecture
Navier–Stokes existence and smoothness
P versus NP problem
Poincaré conjecture (solved)
Riemann hypothesis
Yang–Mills existence and mass gap

v · t · e

Public Key Cryptography

Public key cryptography is a family of cryptosystems that leverage key pairs to perform asymmetric cryptographic operations.

Public Key Cryptography

Public key cryptography is a family of cryptosystems that leverage key pairs to perform asymmetric cryptographic operations.

Not a single shared secret between all parties

Public key & Private key

- Public key == pub-key == pk
- Private key == priv-key == sk ("secret key")

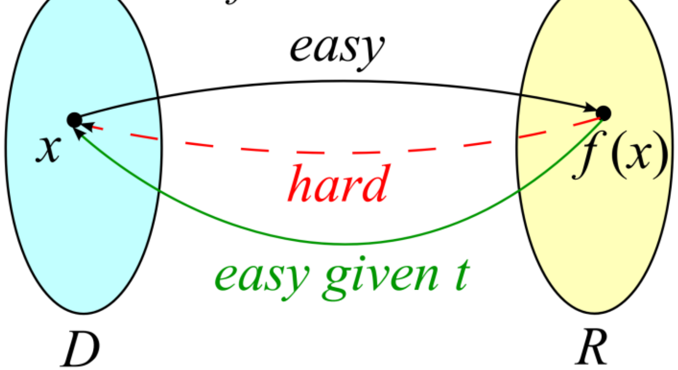
Trapdoor Function

A **trapdoor function** is one which can convert between two states but:

- Is computationally easy D → R
- Is computationally hard D ← R
- Is computationally easy D ← R given a secret

$$(f, t) = \mathbf{Gen}(1^n)$$

$$f : D \rightarrow R$$



Computer and Network Security

Lecture 05:
KEX & Asymmetric Operations

COMP-5370/6370
Fall 2024