Practical Cryptographic Systems

Symmetric Cryptography IV / Public Key Crypto I

Instructor: Matthew Green

Housekeeping

- A1 due tomorrow!
- Reading quiz/assignment coming next Weds!
  - Boneh/Shoup book readings
  - No I will not quiz you about statistics

Numbers stations

https://www.sigidwiki.com/wiki/Category:Numbers_Stations

http://www.numbersoddities.nl/rusmilcw.html

https://en.wikipedia.org/wiki/Lincolnshire_Poacher_(numbers_station)

News

Using Block Ciphers

- ECB is not semantically secure, hence we use a “mode of operation”
  - e.g., CBC, CTR, CFB, OFB (and others)
- These provide:
  - Security for multi-block messages
  - Randomization (through an <u>Initialization Vector</u>)

Using Block Ciphers

- Obvious (bad) idea:
  - Take every consecutive chunk of plaintext
  - Pass it into the block cipher
  - Concatenate all the output blocks
  - This is called “Electronic Codebook Mode” (ECB)

Using Block Ciphers

- Obvious (bad) idea:
  - Take every consecutive chunk of plaintext
  - Pass it into the block cipher
  - Concatenate all the output blocks
  - This is called “Electronic Codebook Mode” (ECB)
- **What’s the problem with this?**

ECB Mode

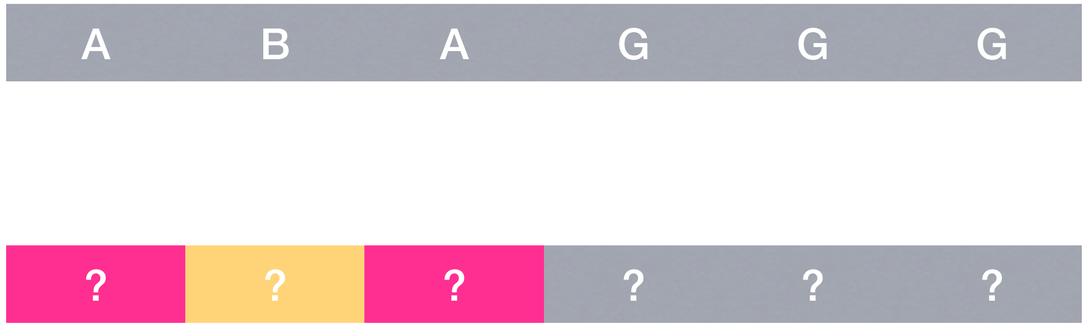
- Problem #1: ECB is <u>deterministic</u>

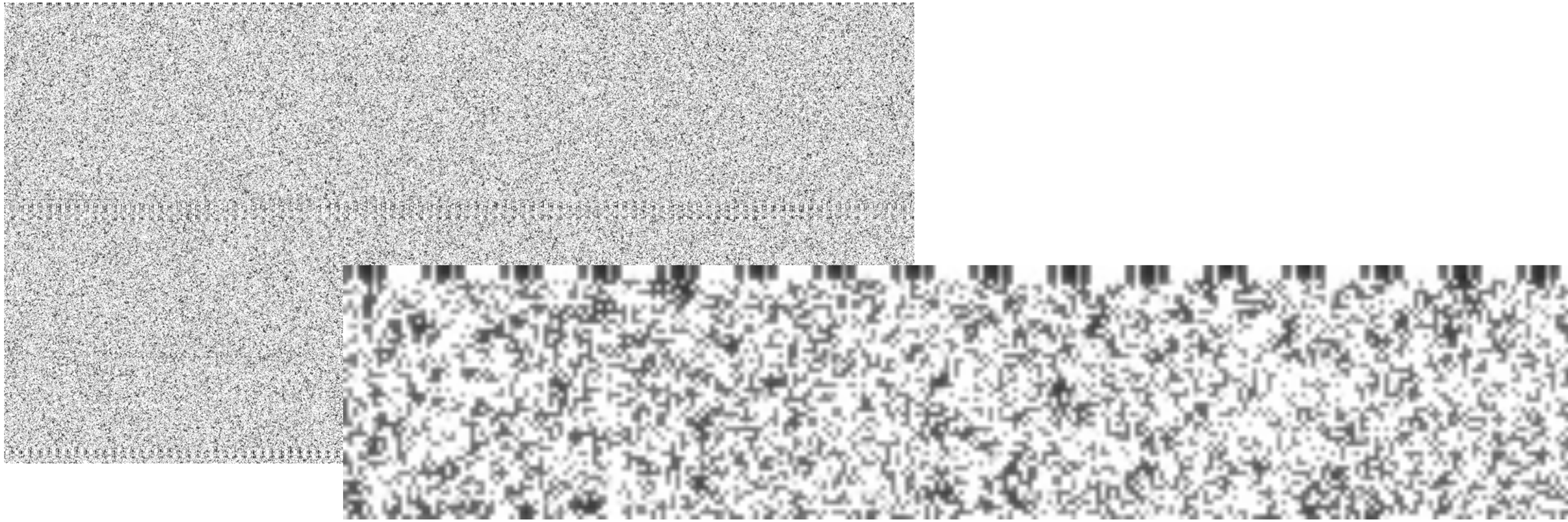
E(Attack Monster) ←

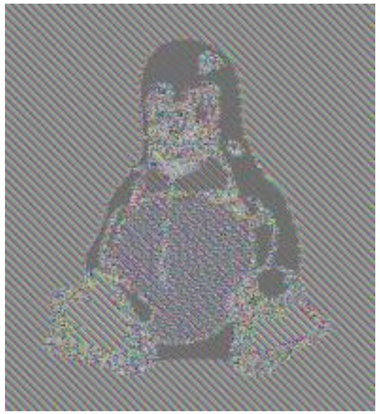
→ E(Monster Attacks)

ECB Mode

- Problem #2: Leakage of plaintext patterns







Semantic security

- Semantic Security (IND-CPA)

$M_1, \ldots, M_n$ →

← $E(k, M_1), \ldots, E(k, M_n)$

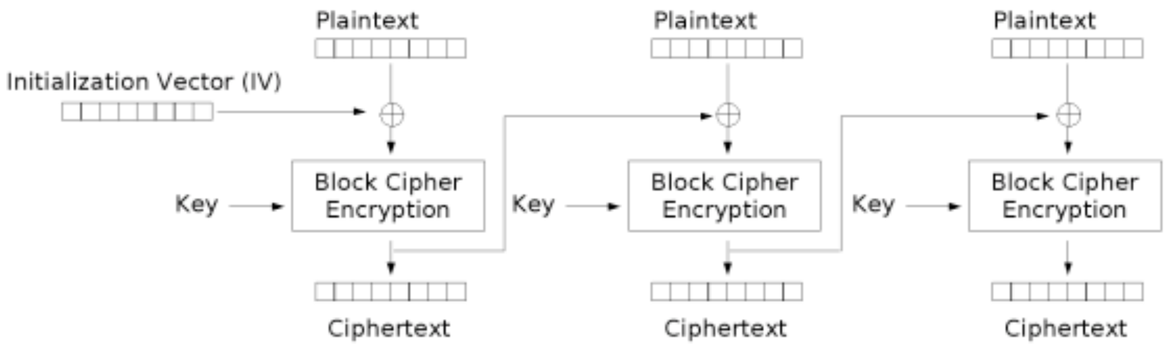
M_0, M_1 →

← $E(M_b)$

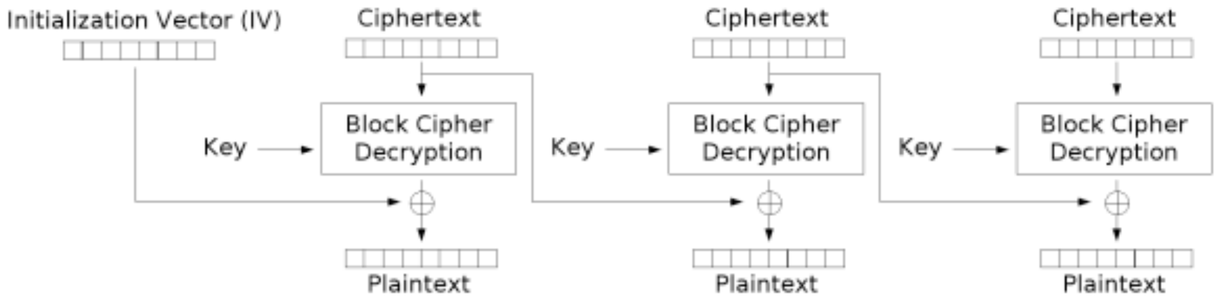
Using Block Ciphers

- ECB is not semantically secure, hence we use a “mode of operation”
  - e.g., CBC, CTR, CFB, OFB (and others)
- These provide:
  - Security for multi-block messages
  - Randomization (through an <u>Initialization Vector</u>)

CBC Mode



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

Security of CBC

- Is CBC a secure encryption scheme?
  - Yes, assuming a secure block cipher (and a <u>passive</u> adversary)
  - Correct (random) IV generation
  - Can <u>prove</u> this under assumption that
    block cipher = Pseudo-Random Permutation (PRP)
- Bellare, Desai, Jokipii & Rogaway (2000)
  - Easy to use wrong...
    - Most important: use a unique & random IV!

The size of the frame of data to be encrypted or decrypted (i.e. how often a new CBC chain is started) depends on the particular application, and is defined for each in the corresponding format specific books of this specification. Unless otherwise specified, the Initialization Vector used at the beginning of a CBC encryption or decryption chain is a constant, iv_0, which is:

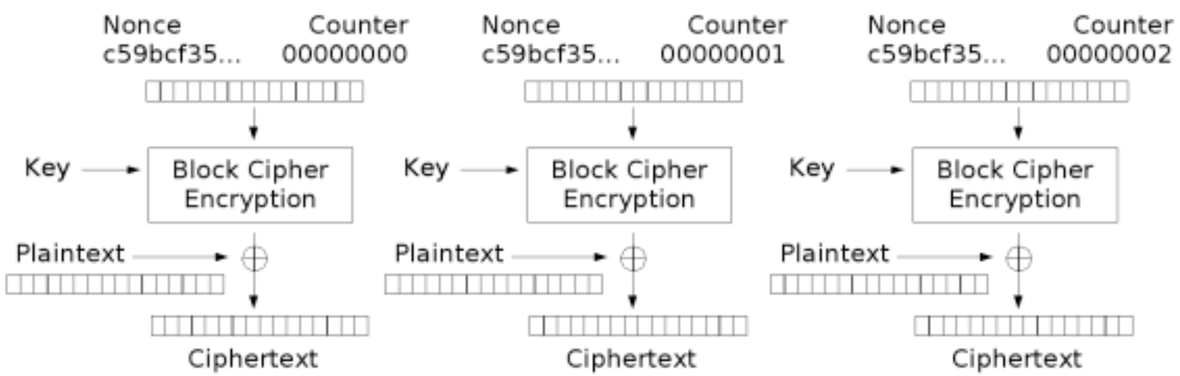
$$0BA0F8DDFEA61FB3D8DF9F566A050F78_{16}$$

Advanced Access Content System (AACS)

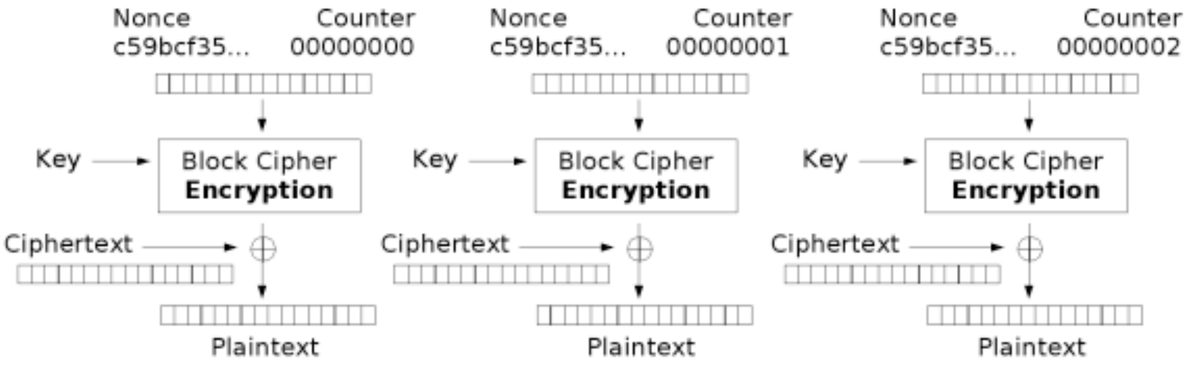
Introduction and
Common Cryptographic Elements

What if the message is not block aligned?

CTR Mode



Counter (CTR) mode encryption



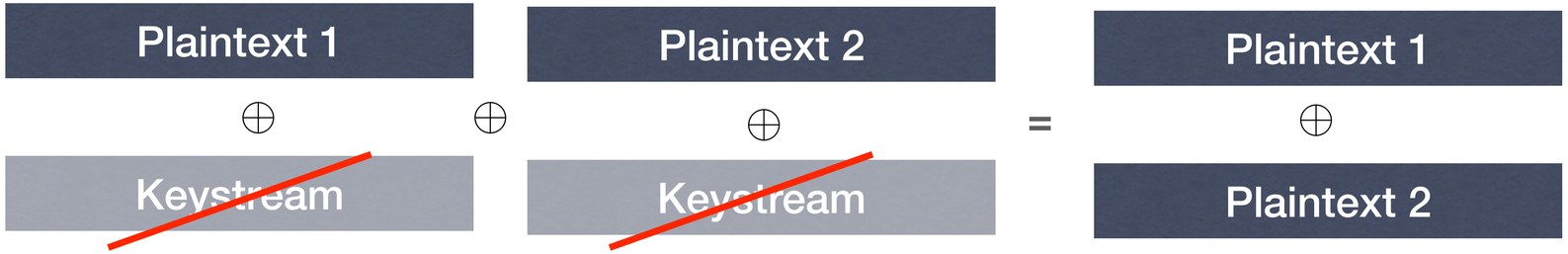
Counter (CTR) mode decryption

CTR (intuition)

- CTR uses the cipher to expand a short cipher key (K) into a long string of <u>pseudorandom output bits</u>
  - This is called a “**keystream**”
  - We then XOR the keystream with a long message (or many messages)
  - This turns a block cipher into a “stream cipher”

Security of CTR

- Yes, assuming secure block cipher (PRP)
- However, counter range must <u>never</u> be re-used



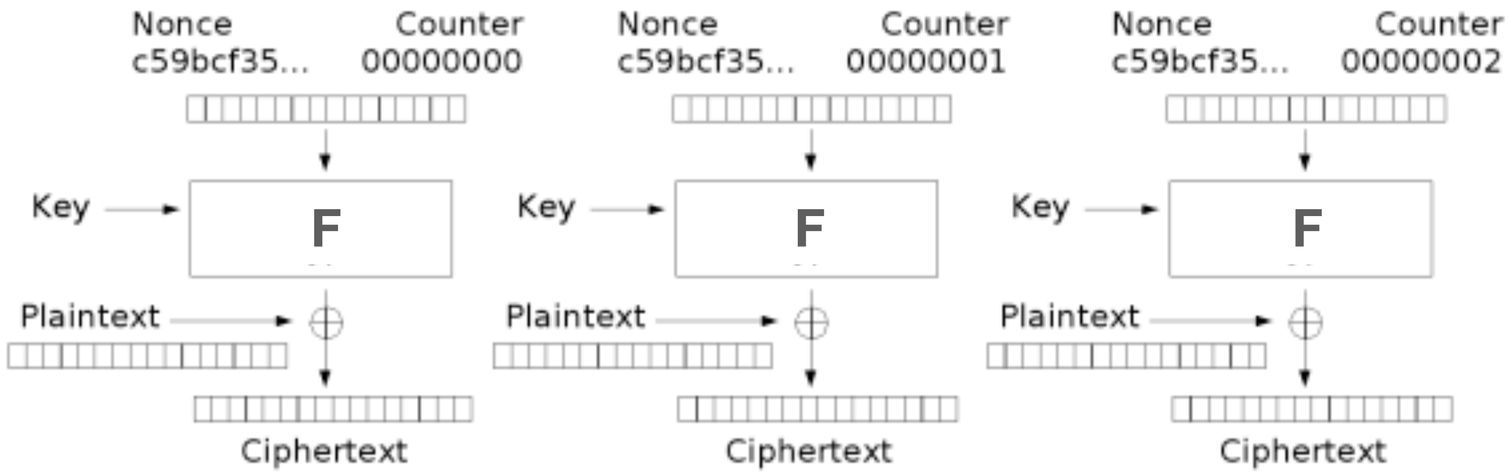
- Similar example: MS Word 2003
  - (they used RC4, but same problem)

CTR with other functions

- We've been assuming a cipher that is <u>invertible</u> (a permutation/block cipher)
- **Observation:** CTR never uses the "Decipher" (invert) algorithm of the cipher
  - So what if we don't use a block cipher at all?

CTR with other functions

- We've been assuming a cipher that is <u>invertible</u> (a permutation/block cipher)
- **Observation:** CTR never uses the "Decipher" (invert) algorithm of the cipher
  - So what if we don't use a block cipher at all?



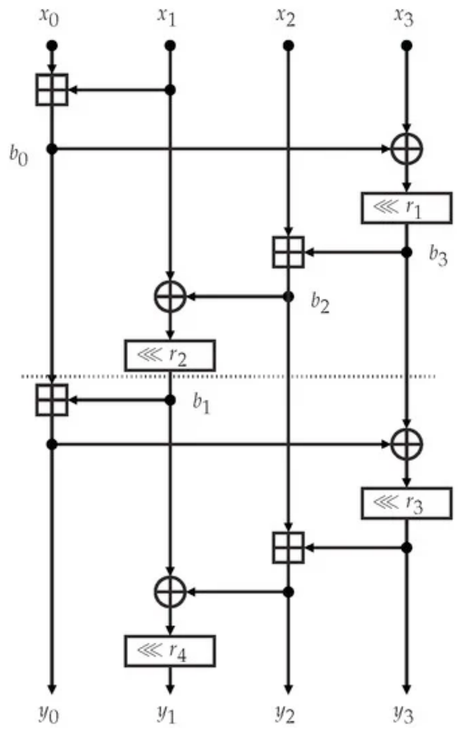
Counter (CTR) mode encryption

ChaCha20

- An example is the ChaCha20 stream cipher
  - Invented by Daniel J. Bernstein (DJB)
- ChaCha20 has one function:

$$F : \{0,1\}^{256} \times \{0,1\}^{512} \rightarrow \{0,1\}^{512}$$

key counter (nonce) keystream



- Note: ChaCha is <u>not</u> a permutation! It is not invertible and may output the same value on two different inputs.
- <u>ChaCha20 is faster in software than AES, unless you have hardware support</u>

ChaCha20

```c
#define ROTL(a,b) (((a) << (b)) | ((a) >> (32 - (b))))
#define QR(a, b, c, d) (            \
    a += b,  d ^= a,  d = ROTL(d,16),   \
    c += d,  b ^= c,  b = ROTL(b,12),   \
    a += b,  d ^= a,  d = ROTL(d, 8),   \
    c += d,  b ^= c,  b = ROTL(b, 7))
#define ROUNDS 20

void chacha_block(uint32_t out[16], uint32_t const in[16])
{
    int i;
    uint32_t x[16];

    for (i = 0; i < 16; ++i)
        x[i] = in[i];
    // 10 loops × 2 rounds/loop = 20 rounds
    for (i = 0; i < ROUNDS; i += 2) {
        // Odd round
        QR(x[0], x[4], x[ 8], x[12]); // column 0
        QR(x[1], x[5], x[ 9], x[13]); // column 1
        QR(x[2], x[6], x[10], x[14]); // column 2
        QR(x[3], x[7], x[11], x[15]); // column 3
        // Even round
        QR(x[0], x[5], x[10], x[15]); // diagonal 1 (main diagonal)
        QR(x[1], x[6], x[11], x[12]); // diagonal 2
        QR(x[2], x[7], x[ 8], x[13]); // diagonal 3
        QR(x[3], x[4], x[ 9], x[14]); // diagonal 4
    }
    for (i = 0; i < 16; ++i)
        out[i] = x[i] + in[i];
}
```

Point of order

- Proofs of security:
  - We don't know how to prove that DES or AES (resp. ChaCha20) are secure block ciphers (resp. Secure pseudorandom function)
  - But if we assume that the block ciphers are secure PRPs then:

•We can prove that CBC & CTR & OFB & CFB etc.
are secure encryption modes against a passive adversary

http://www.cs.ucdavis.edu/~rogaway/papers/sym-enc-abstract.html

Malleability

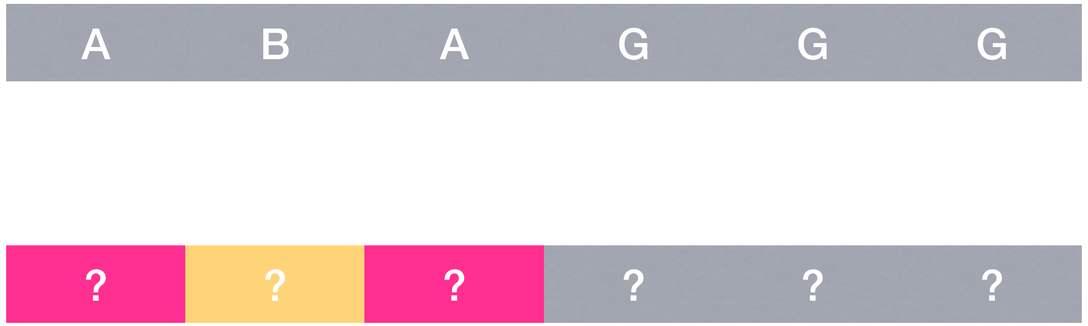
- The ability to modify a ciphertext
  - Such that the plaintext is meaningfully altered
  - CTR Mode (bad)
  - CBC Mode (somewhat bad)

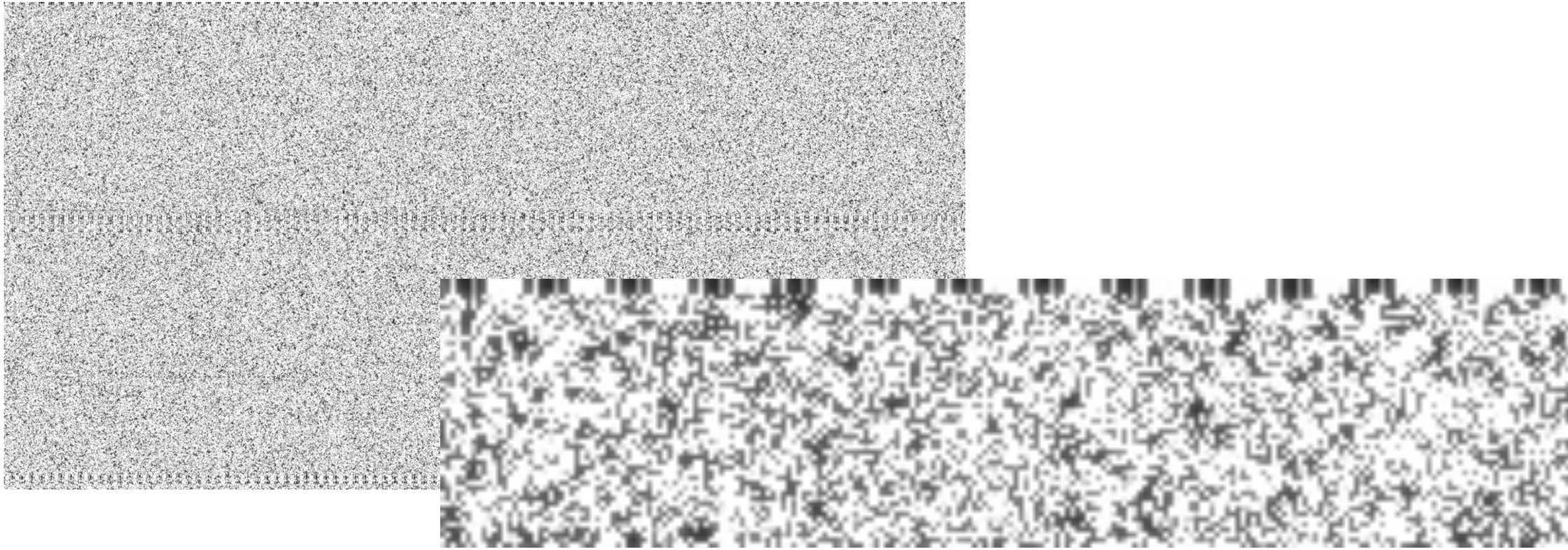
Hash Functions

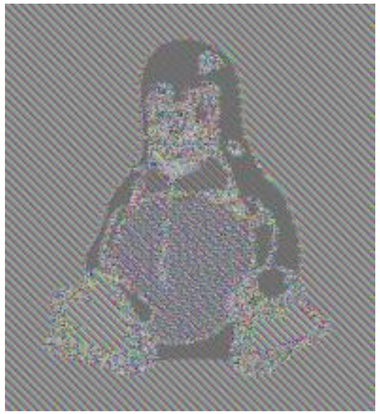
- Convert variable-length string to small “tag”
  - Hash tables
  - Signatures
  - Software checksums
  - MAC functions (HMAC)
  - Encryption (OAEP)

Review

- ECB Mode: Encrypt each block separately
  - Problems?







ECB Mode

- ECB is <u>deterministic</u>
  - Leads to problems, e.g.,:

Security of Encryption

- Semantic Security
  - Due to Goldwasser & Micali (1980s)
  - Informally: An encryption scheme is secure if <u>adversary who sees ciphertext</u> “learns as much” as <u>adversary who doesn’t see ciphertext.</u>

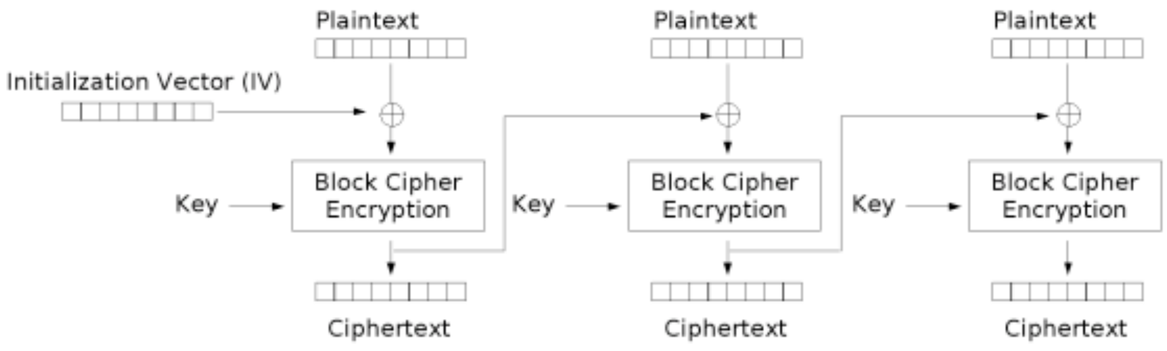
•Even if adversary can request chosen plaintexts

  - How do we state this formally?

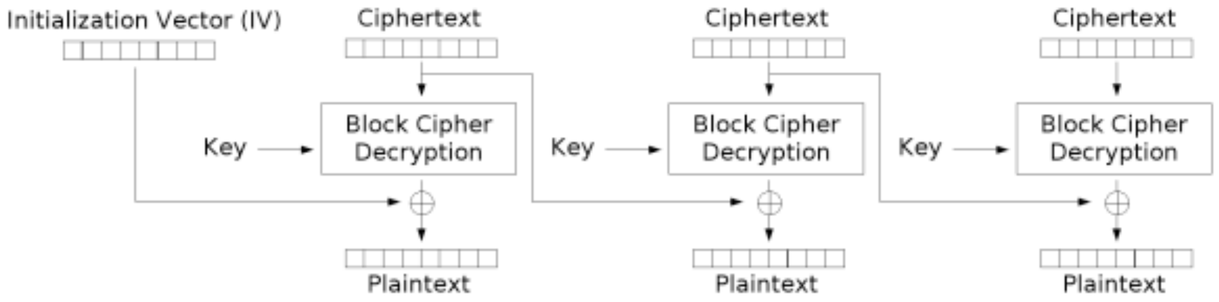
Using Block Ciphers

- ECB is not semantically secure, hence we use a “mode of operation”
  - e.g., CBC, CTR, CFB, OFB (and others)
- These provide:
  - Security for multi-block messages
  - Randomization (through an <u>Initialization Vector</u>)

CBC Mode



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

Security of CBC

- Is CBC a secure encryption scheme?
  - Yes, assuming a secure block cipher
  - Correct (random) IV generation
  - Can <u>prove</u> this under assumption that
    block cipher = Pseudo-Random Permutation (PRP)

•Bellare, Desai, Jokipii & Rogaway (2000)

- Easy to use wrong...
  - Most important: use a unique & random IV!

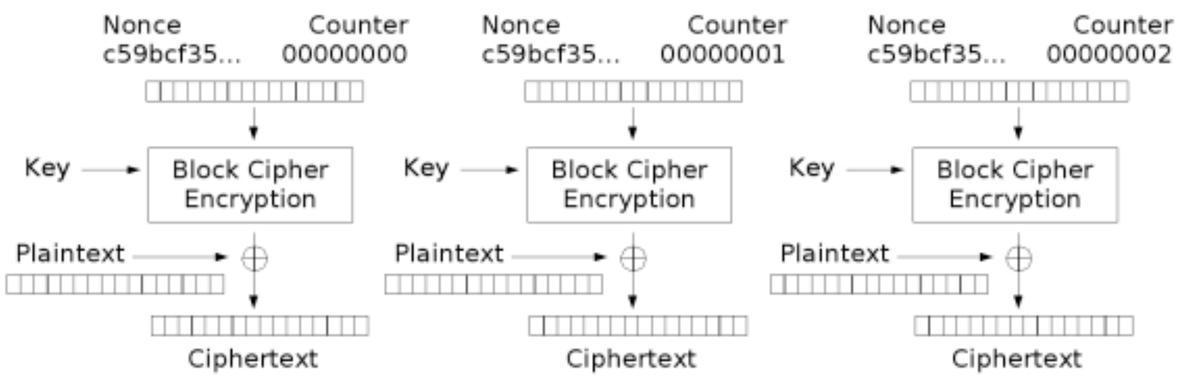
The size of the frame of data to be encrypted or decrypted (i.e. how often a new CBC chain is started) depends on the particular application, and is defined for each in the corresponding format specific books of this specification. Unless otherwise specified, the Initialization Vector used at the beginning of a CBC encryption or decryption chain is a constant, iv_0, which is:

$$0BA0F8DDFEA61FB3D8DF9F566A050F78_{16}$$

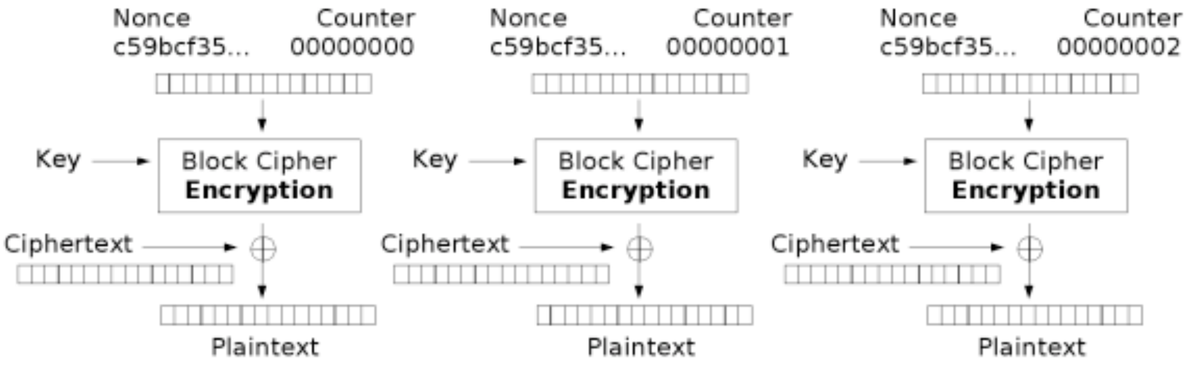
Advanced Access Content System (AACS)

Introduction and
Common Cryptographic Elements

CTR Mode



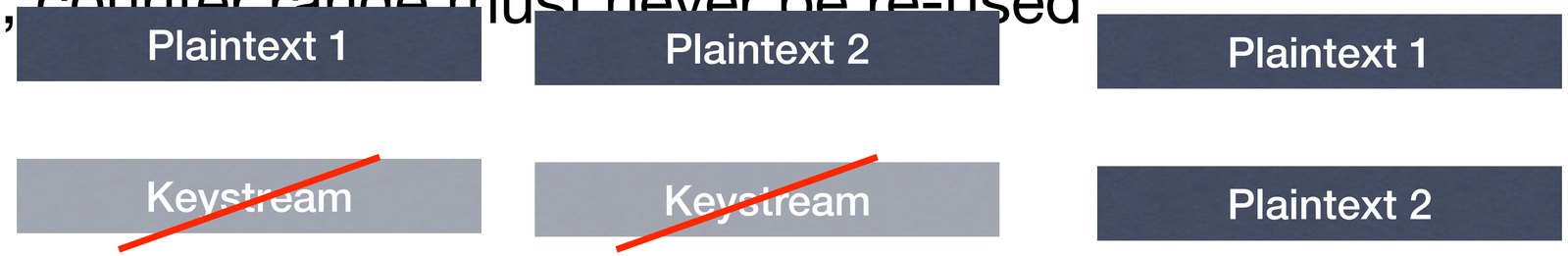
Counter (CTR) mode encryption



Counter (CTR) mode decryption

Security of CTR

- Yes, assuming secure block cipher (PRP)
- However, counter range must never be re-used



- Similar example: MS Word 2003
  - (they used RC4, but same problem)

Point of order

- Proofs of security:
  - We <u>don't</u> know how to prove that DES or AES are secure block ciphers
  - But if we assume that the block ciphers <u>are</u> secure PRPs then:

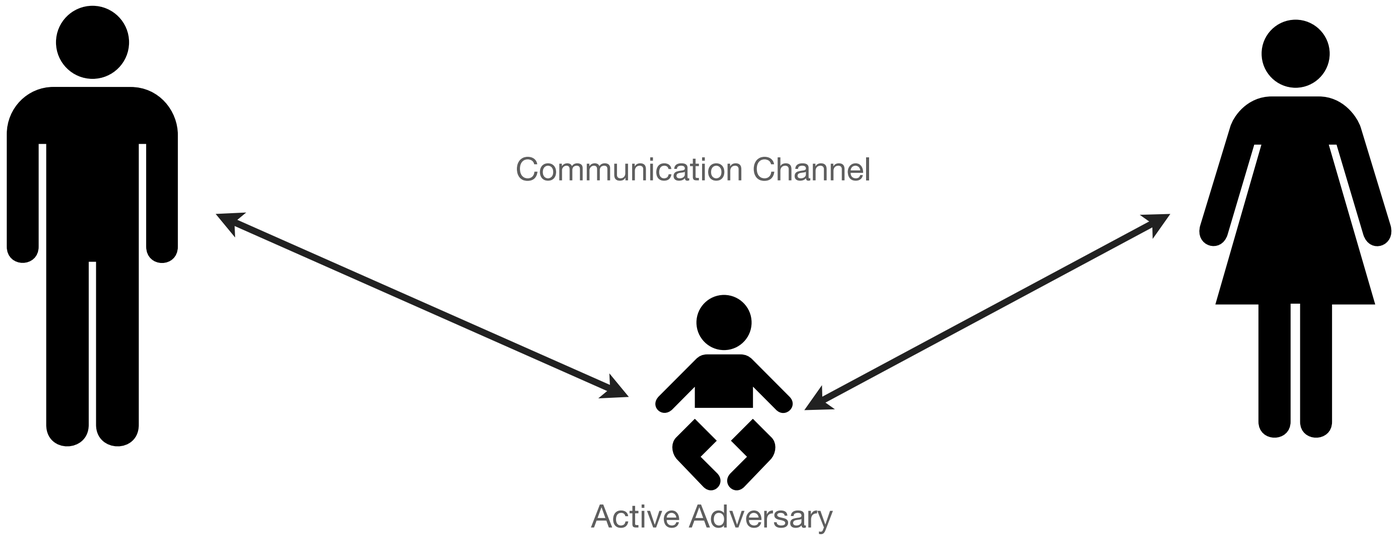
•We can prove that CBC & CTR & OFB & CFB etc. are secure encryption modes.

http://www.cs.ucdavis.edu/~rogaway/papers/sym-enc-abstract.html

Malleability

- The ability to modify a ciphertext
  - Such that the plaintext is meaningfully altered
  - CTR Mode (bad)
  - CBC Mode (somewhat bad)

Authenticated Encryption



MACs

- Symmetric-key primitive
  - Given a key and a message, compute a “tag”
  - Tag can be verified using the same key
  - Any changes to the message detectable
- To prevent malleability:
  - Encrypt <u>then</u> MAC
  - Under separate keys

MACs

- Definitions of Security
  - Existential Unforgeability under Chosen Message Attack (EU-CMA)
- Examples:
  - HMAC (based on hash functions)
  - CMAC/CBC-MAC (block ciphers)

Authenticated Encryption

- Two ways to get there:
  - <u>Generic composition</u>
    Encrypt (e.g., CBC mode) then MAC

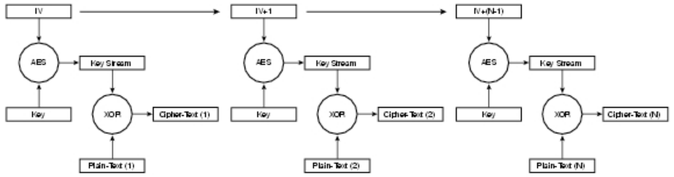
•two different keys, multiple primitives

  - <u>Authenticated</u> mode of operation

•Integrates both encryption & authentication

•Single key, typically uses only one primitive (e.g., block cipher)

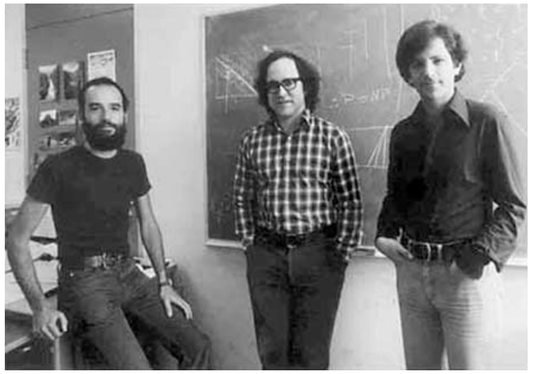
•Ex: CCM, OCB, GCM modes



Hash Functions

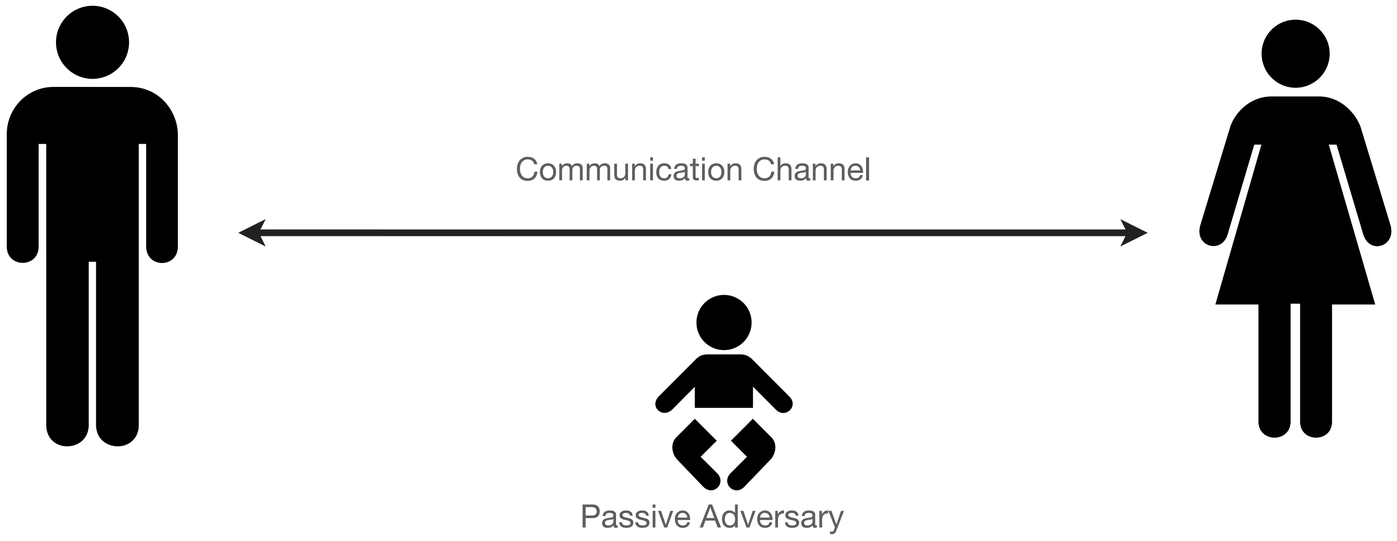
Asymmetric Crypto

- So far we've discussed <u>symmetric</u> crypto
  - Requires both parties to share a key
  - Key distribution is a hard problem!



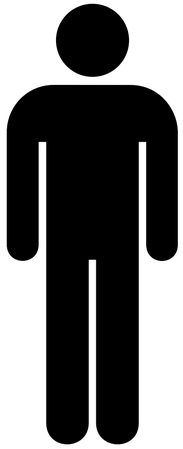
Key Agreement

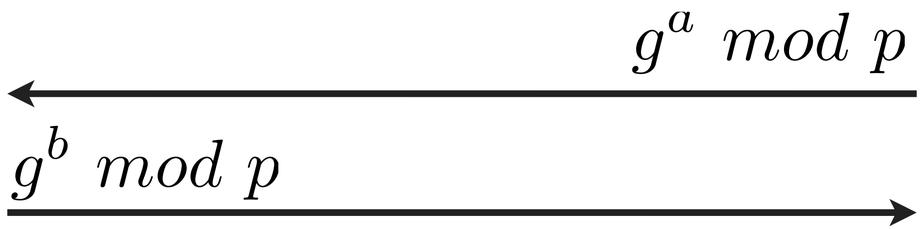
- Establish a shared key in the presence of a passive adversary

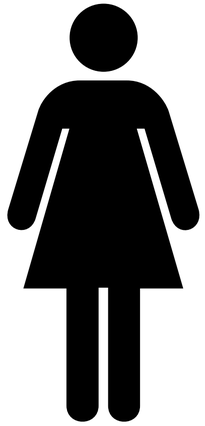


D-H Protocol

$b \in \mathbb{Z}_q$ $\qquad$ $p, q : p = 2q + 1$ $\qquad$ $a \in \mathbb{Z}_q$



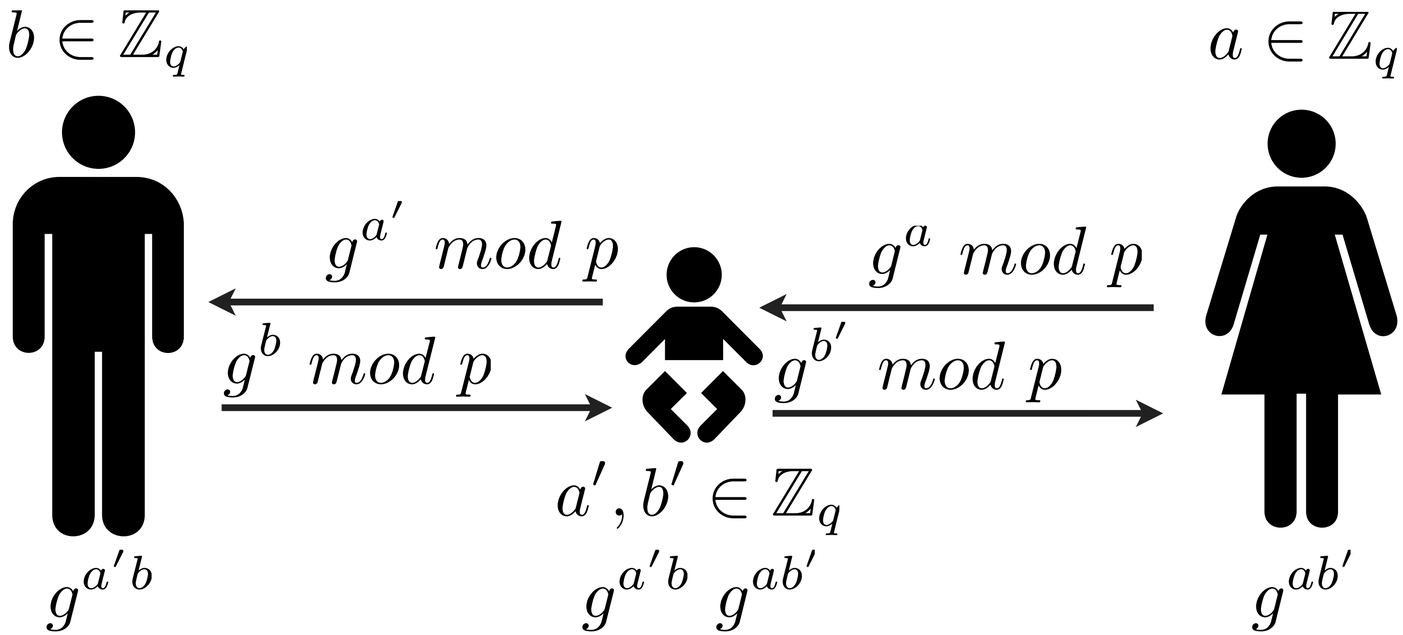




g^{ab} $\qquad$ g^{ab}

Man in the Middle

- Assume an active adversary:



Man in the Middle

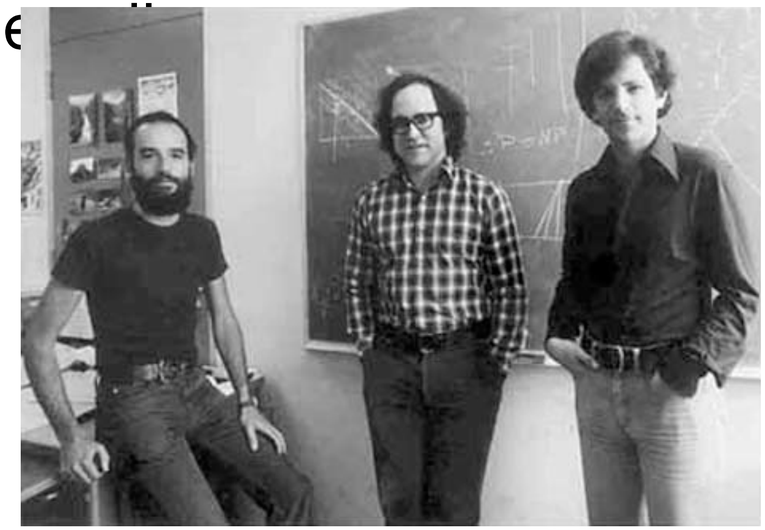
- Caused by lack of <u>authentication</u>
  - D-H lets us establish a shared key with anyone... but that’s the problem...
- Solution: Authenticate the remote party

Preventing MITM

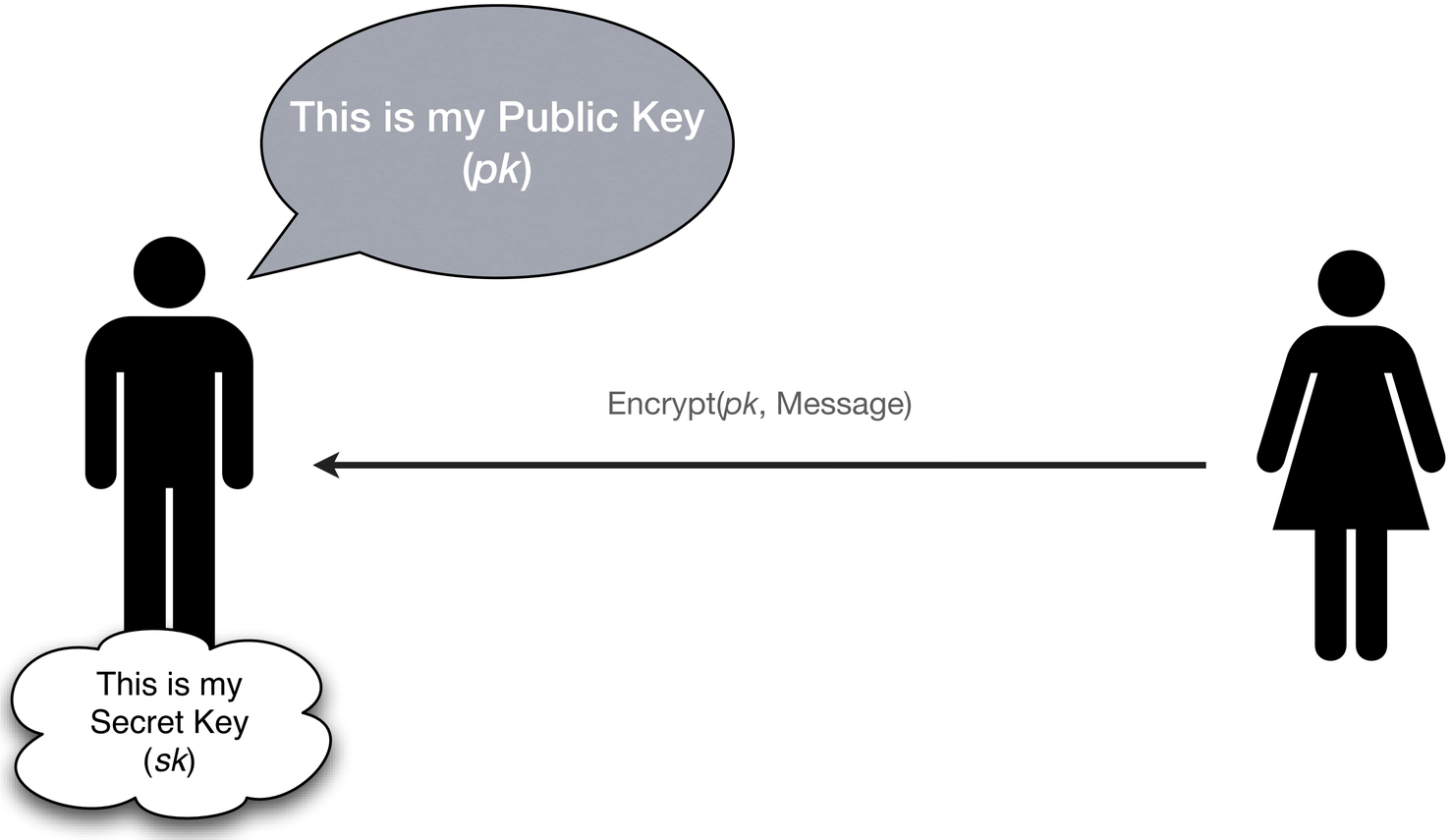
- Verify key via separate channel
- Password-based authentication
- Authentication via PKI

Public Key Encryption

- What if our recipient is <u>offline?</u>
  - Key agreement protocols are interactive
  - e.g., want to send an e



Public Key Encryption



RSA Cryptosystem

“Textbook RSA”

- In practice, we don’t use Textbook RSA
  - Fully deterministic (not semantically secure)
  - <u>Malleable</u>
  - Might be <u>partially</u> invertible

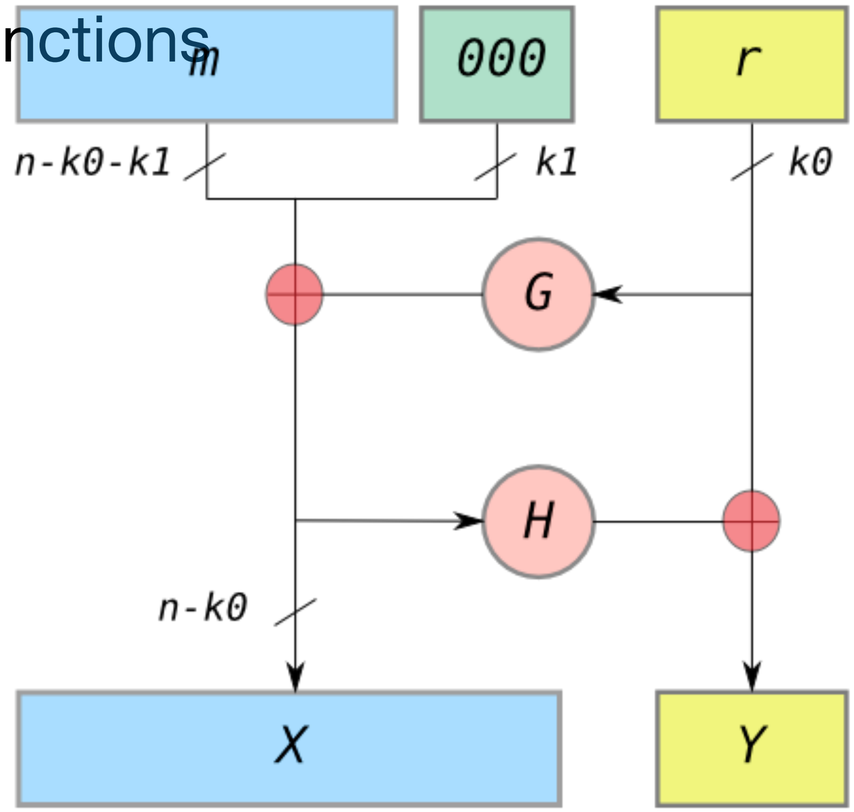
•Coppersmith’s attack: recover part of plaintext (when *m* and *e* are small)

RSA Padding

- Early solution (RSA PKCS #1 v1.5):
  - Add “padding” to the message before encryption
  - Includes randomness
  - Defined structure to mitigate malleability
  - PKCS #1 v1.5 badly broken (Bleichenbacher)

RSA Padding

- Better solution (RSA-OAEP):
  - G and H are hash functions



Efficiency

	Cycles/Byte
AES (128 bit key)	18
DES (56 bit key)	51
RSA (1024 bit key) <u>Encryption</u>	1,016
RSA (1024 bit key) <u>Decryption</u>	21,719

Hybrid Encryption

- Mixed Approach
  - Use PK encryption to encrypt a symmetric key
  - Use (fast) symmetric encryption on data

Key Strength

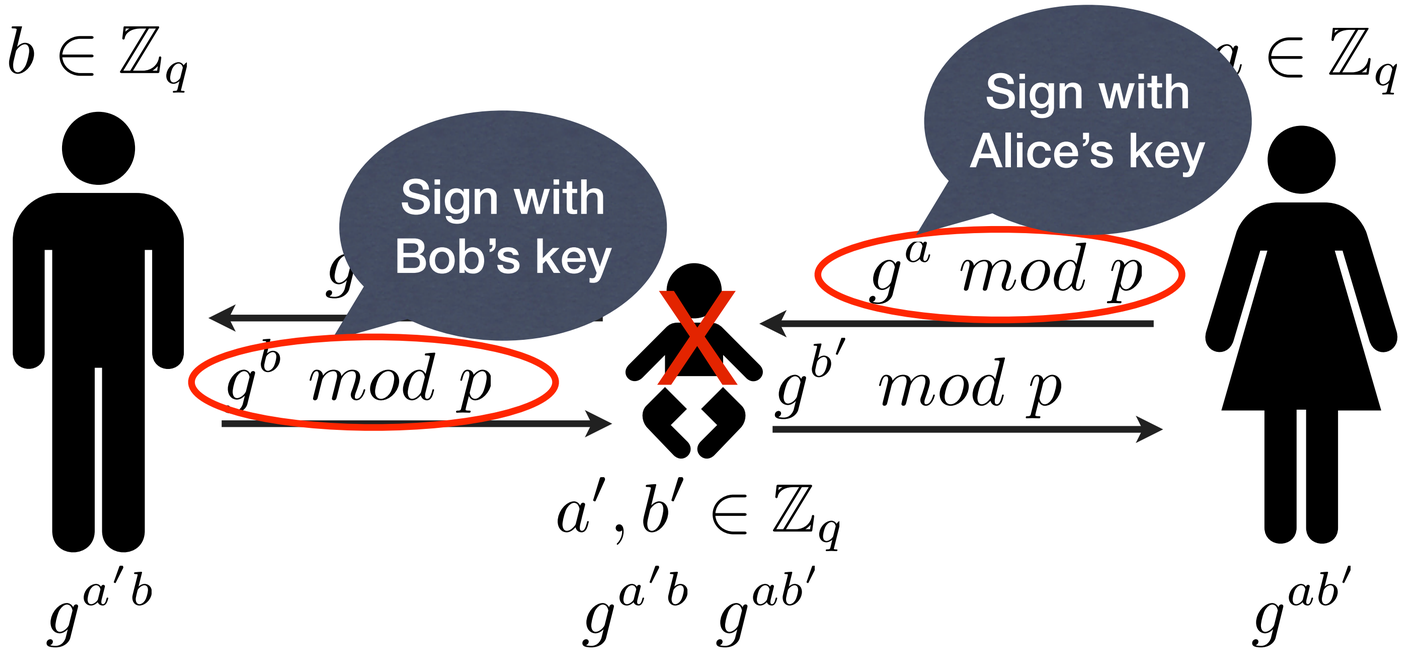
Level	Protection	Symmetric	Asymmetric	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash
1	Attacks in "real-time" by individuals *Only acceptable for authentication tag size*	32	-	-	-	-	-
2	Very short-term protection against small organizations *Should not be used for confidentiality in new systems*	64	816	128	816	128	128
3	Short-term protection against medium organizations, medium-term protection against small organizations	72	1008	144	1008	144	144
4	Very short-term protection against agencies, long-term protection against small organizations *Smallest general-purpose level,* *Use of 2-key 3DES restricted to 2^{40} plaintext/ciphertexts,* *protection from 2009 to 2011*	80	1248	160	1248	160	160
5	Legacy standard level *Use of 2-key 3DES restricted to 10^6 plaintext/ciphertexts,* *protection from 2009 to 2018*	96	1776	192	1776	192	192
6	Medium-term protection *Use of 3-key 3DES,* *protection from 2009 to 2028*	112	2432	224	2432	224	224
7	Long-term protection *Generic application-independent recommendation,* *protection from 2009 to 2038*	128	3248	256	3248	256	256
8	"Foreseeable future" *Good protection against quantum computers*	256	15424	512	15424	512	512

Digital Signatures

- Similar to MACs, with public keys
  - Secret key used to sign data
  - Public key can verify signature
  - Advantages over MACs?

Preventing MitM

- Assume an active adversary:



PKI & Certificates

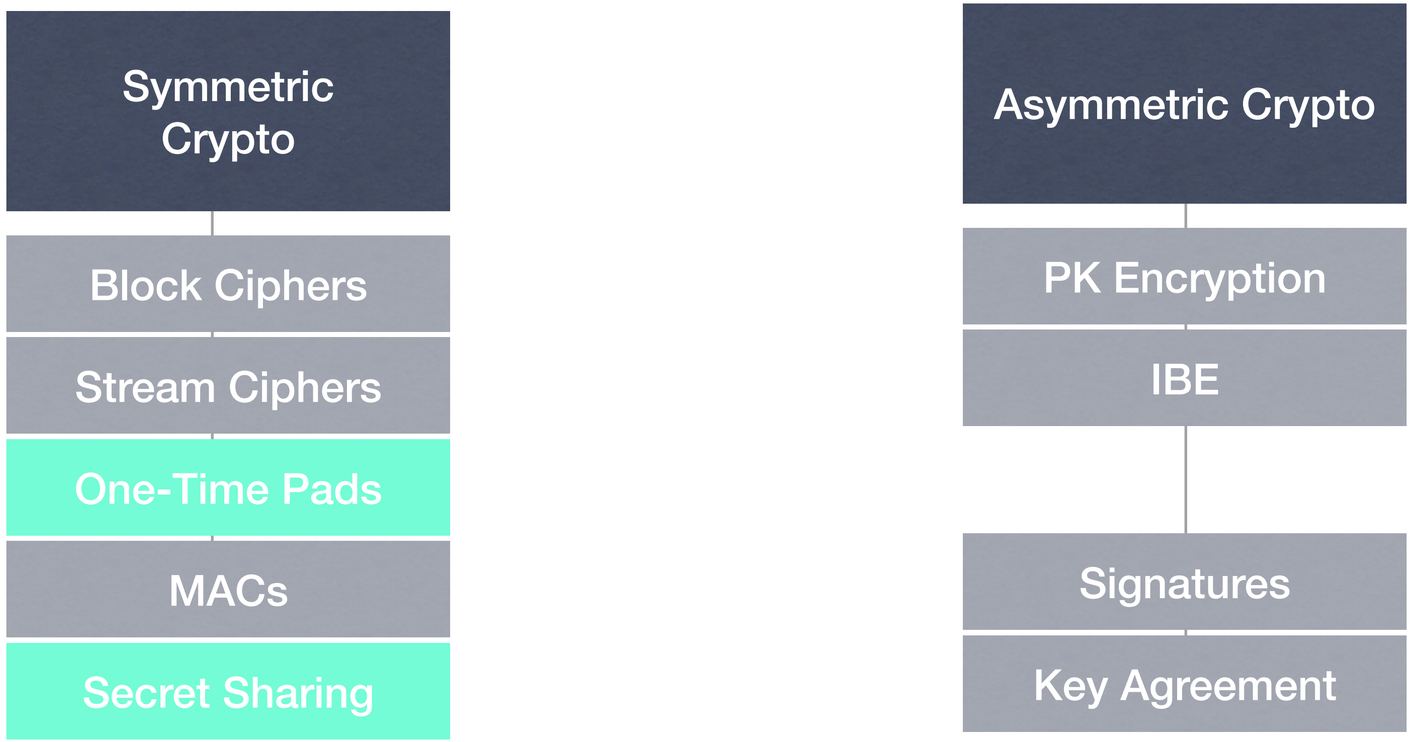
- How do I know to trust your public key?
  - Put it into a file with some other info, and get someone else to sign it!

VeriSign®

Next Time

- Protocols & Implementation
- Reading!
- A2 coming up this week

Basic Primitives

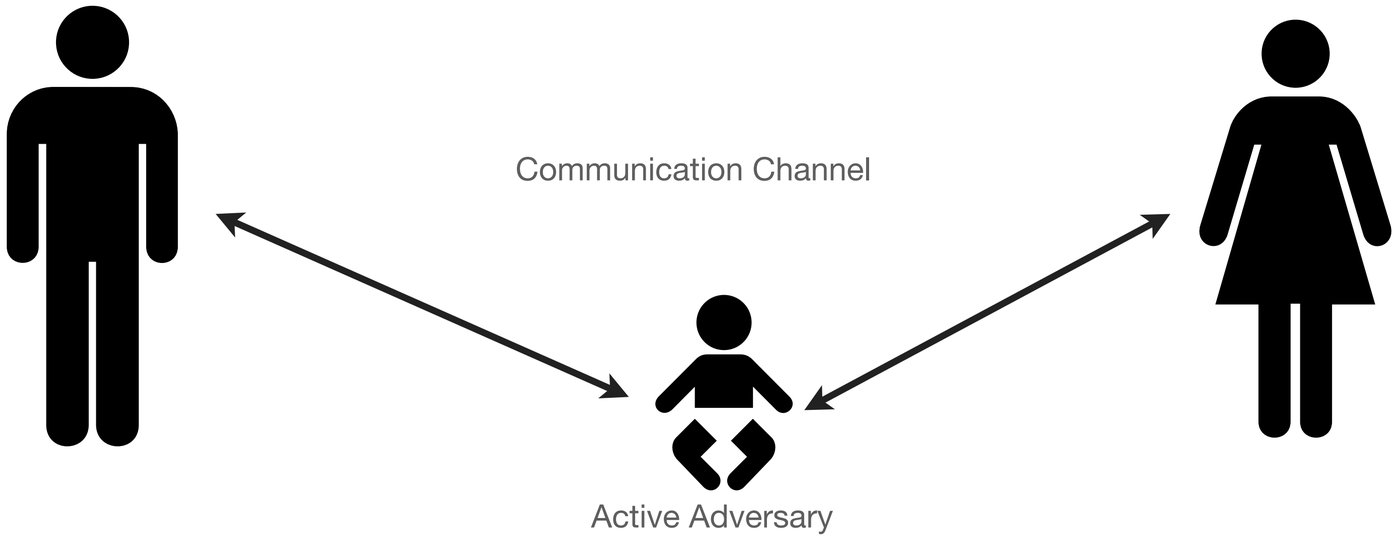


Information Theoretic Complexity The

Next Time

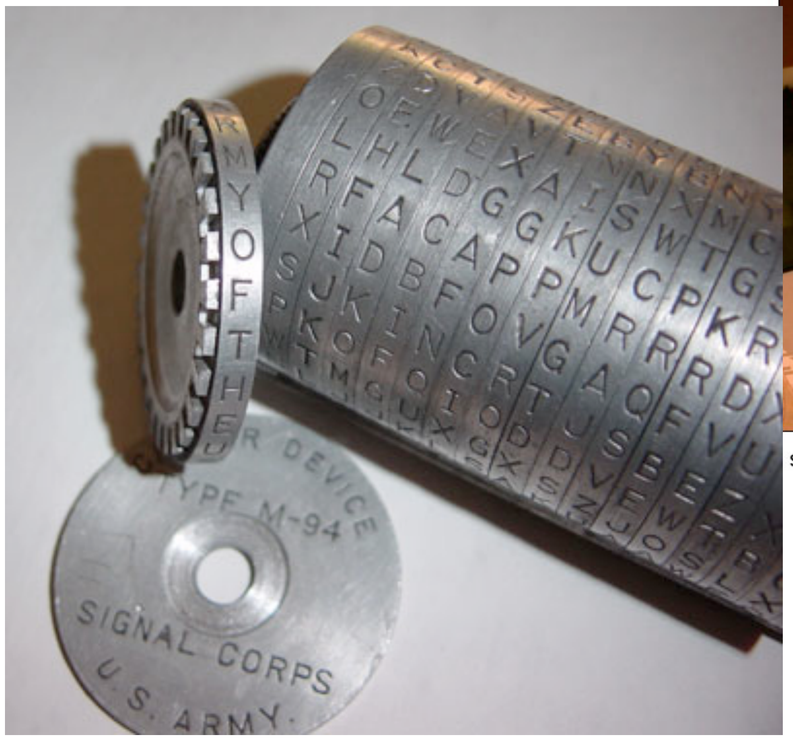
MACs

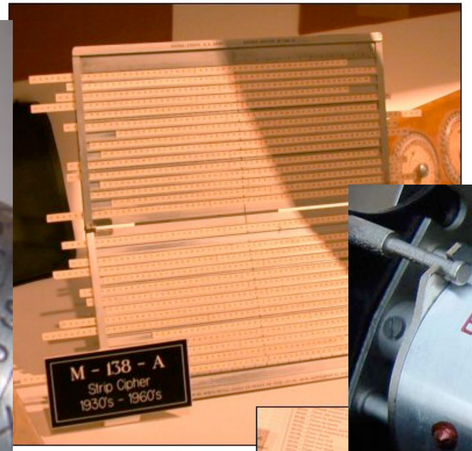
Authenticated Encryption



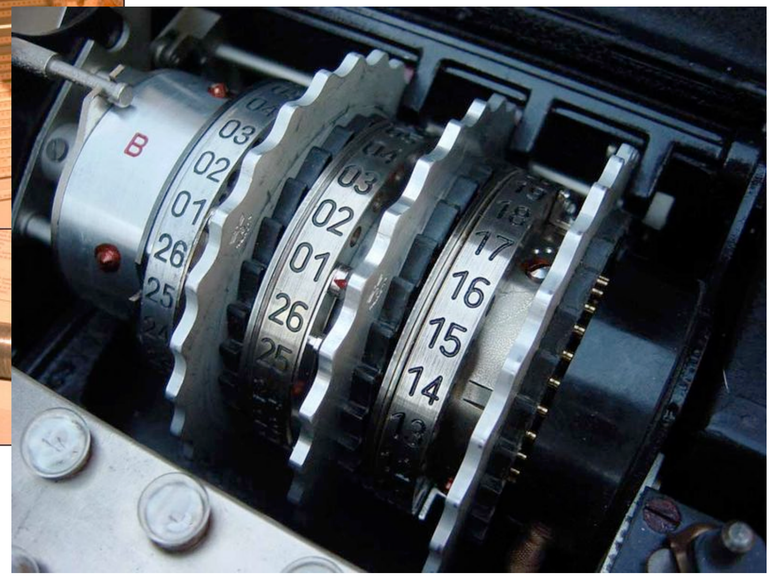
Convenience vs. Security

Mechanical Cryptography



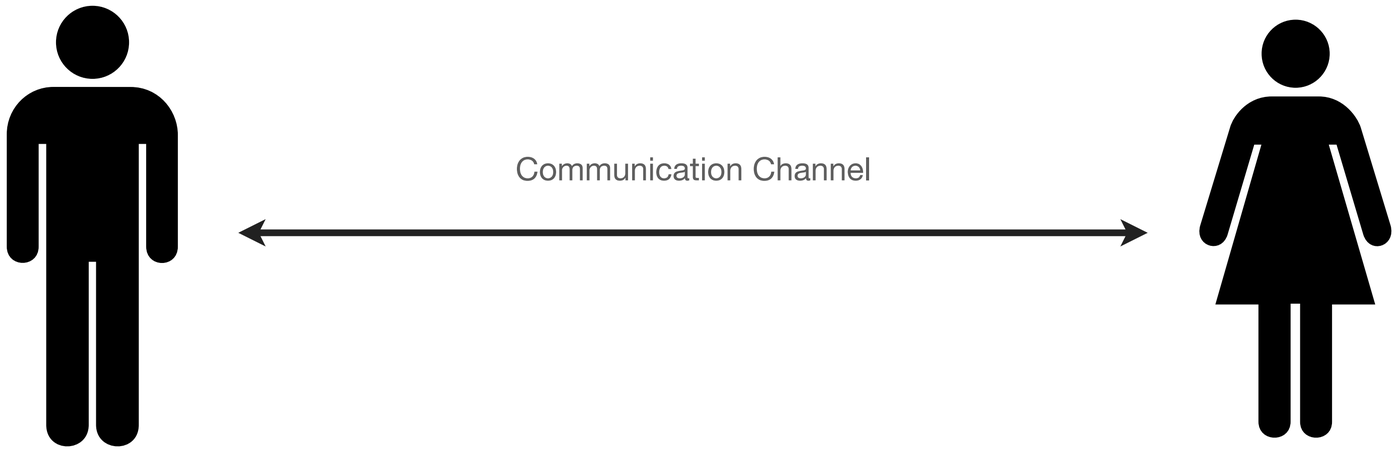


SIMPLE CIPHER DEVICES
(GVG / PD)

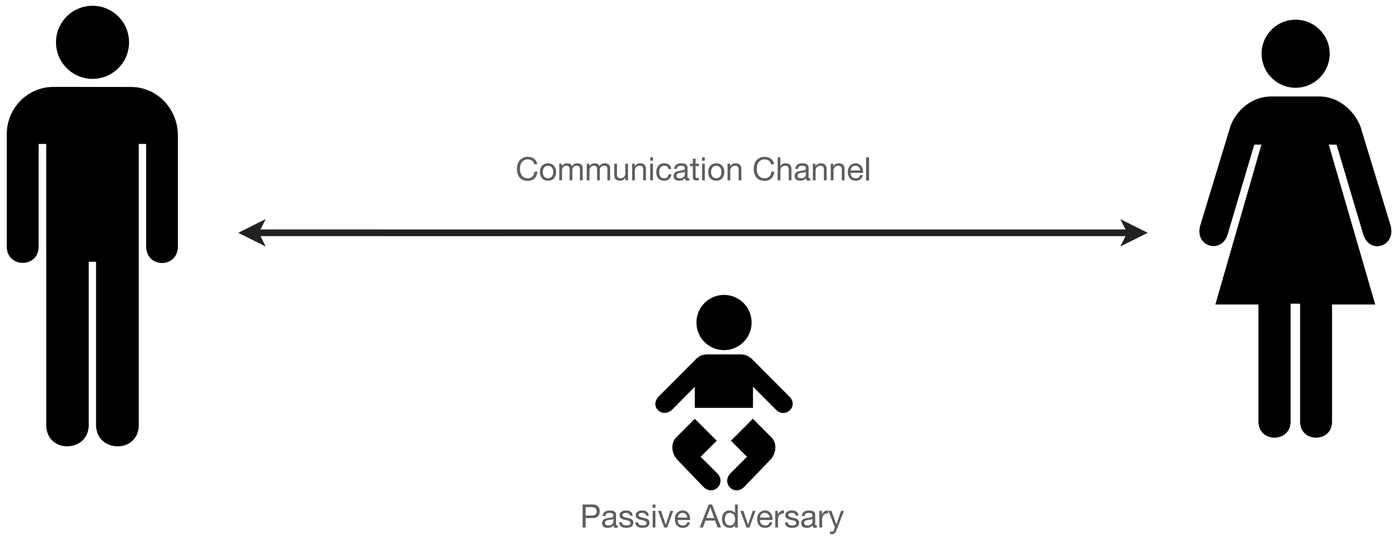


A quick note on ethics!

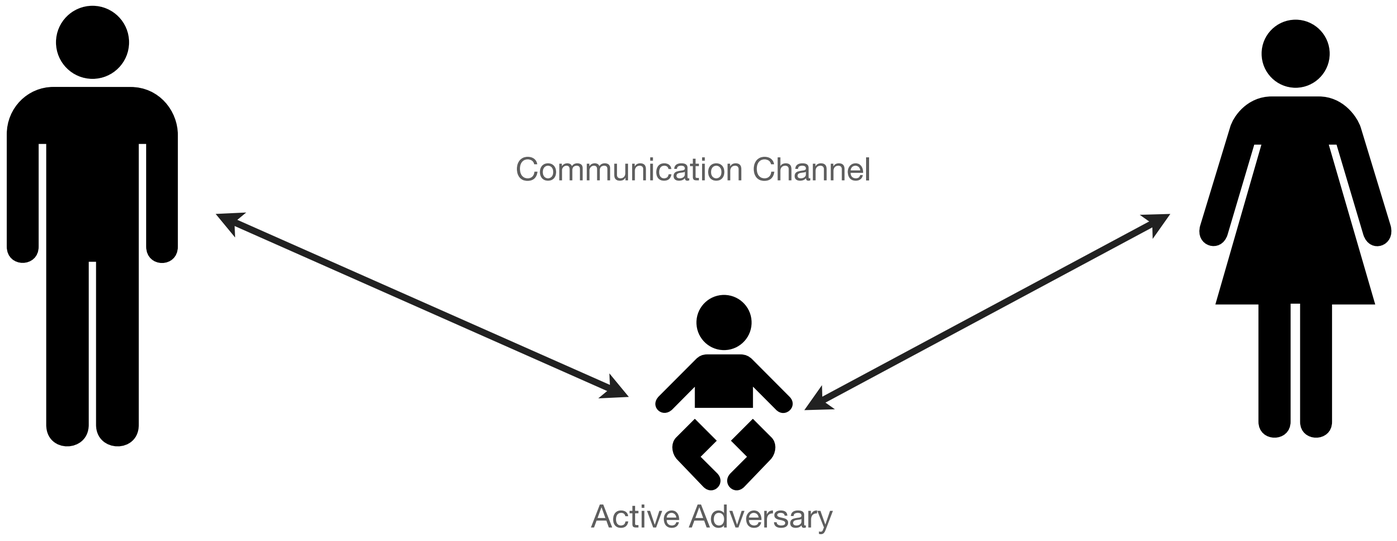
Communication Model



Communication Model



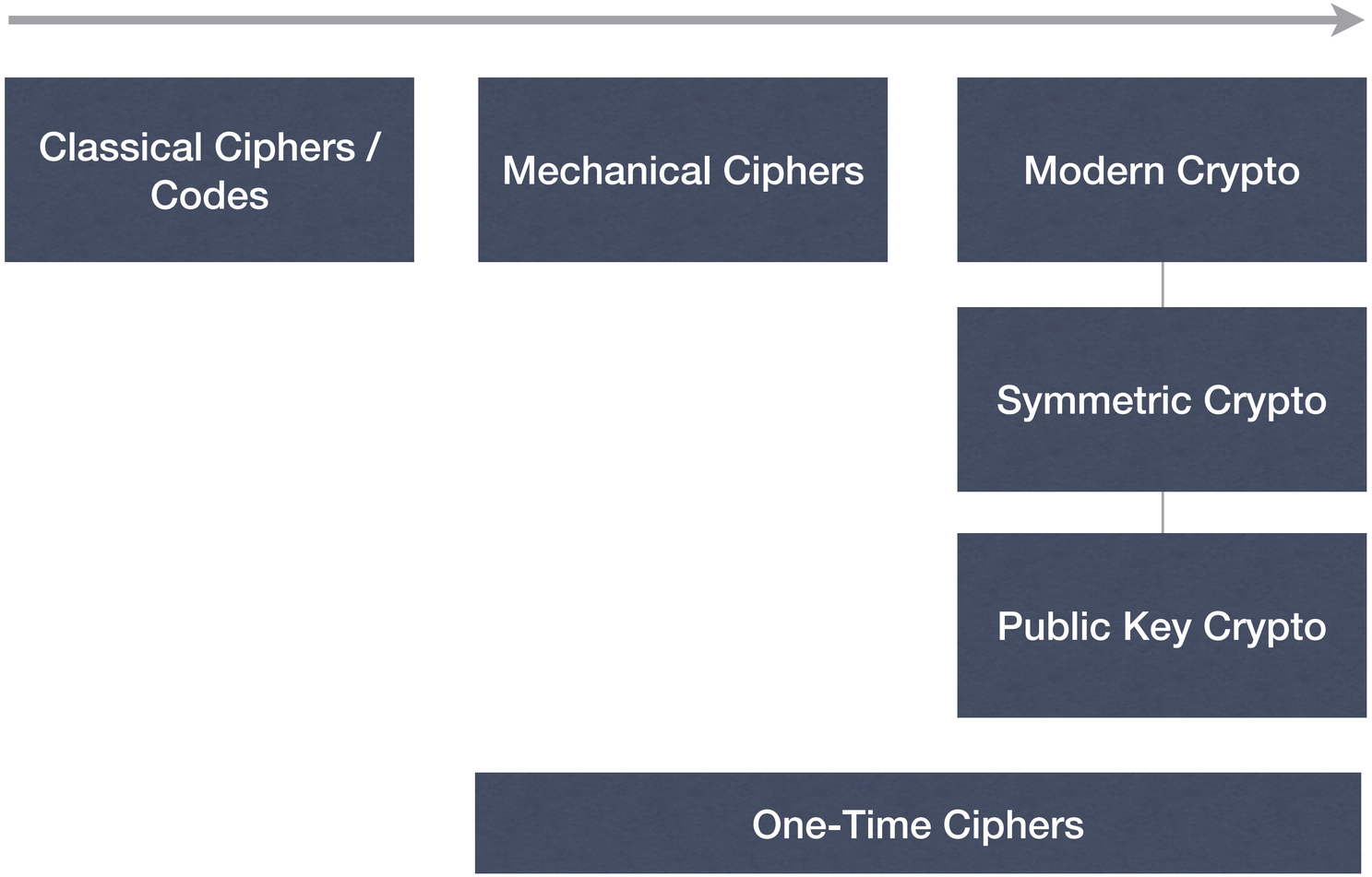
Communication Model



Secure Communication

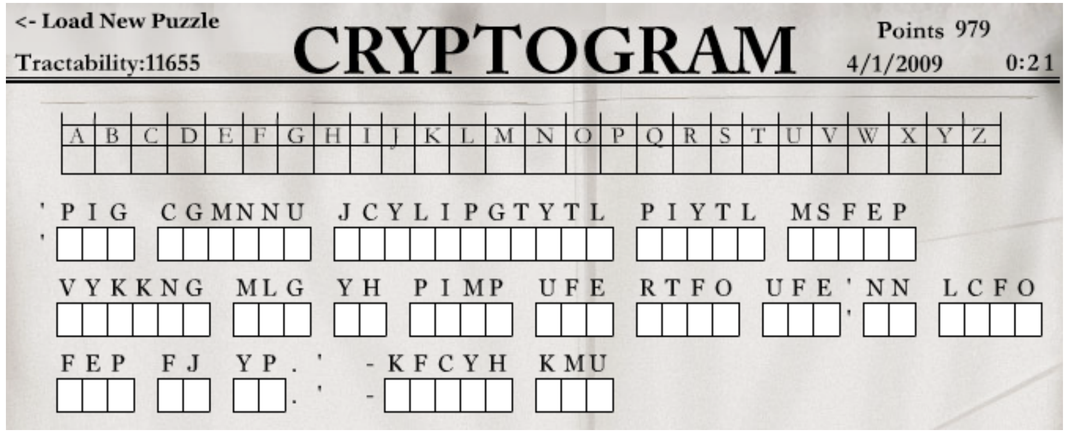
- Two basic properties we like to achieve:
  - Data <u>confidentiality</u>
  - Data <u>authenticity</u> (“integrity”)
- Tools:
  - Encryption
  - Message Authentication Codes
  - Digital Signatures

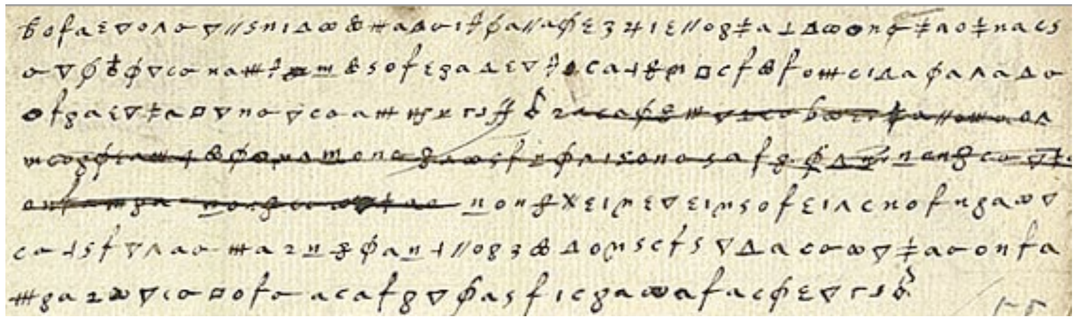
History of Encryption

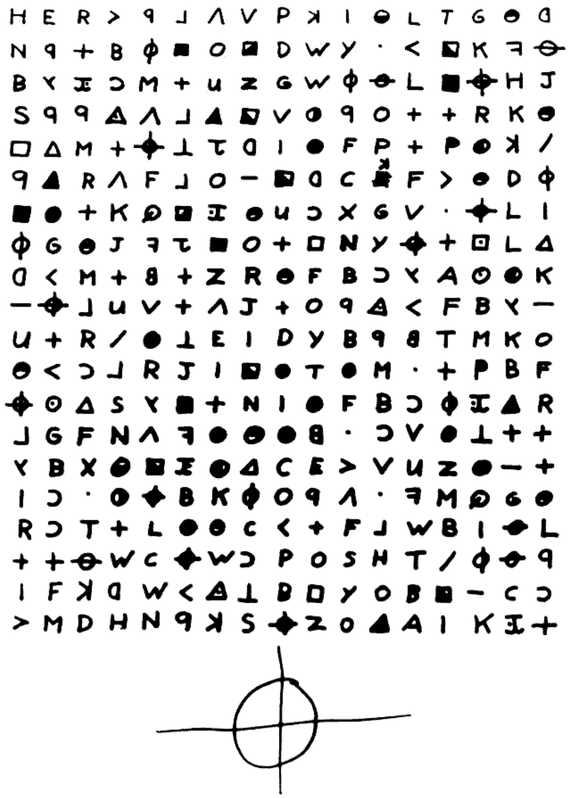


Classical Cryptography

- Beginning of time to 1900s or so
  - Shift (Caesar) cipher
  - Substitution ciphers
  - Polyalphabetic ciphers (Vigenère)
  - Digraph ciphers (Playfair)
  - A multitude of others...







A	G	R	P	T
B	I	K	C	Q
S	L	D	M	E
N	Y	W	F	X
G	J	H	O	Z

S	E	N	D	R	E	I	N	F	O	R	C	E	M	E	N	T	S
V	I	G	E	N	E	R	E	V	I	G	E	N	E	R	E	V	I
N	M	T	H	E	I	Z	R	A	W	X	G	R	Q	V	R	O	A

One-Time Ciphers

- 1900s
  - Vernam & Mauborgne's "Unbreakable" cipher

•Based on Baudot code for Teletypes

•Added (XORed) a random Key (sequence of bits) to a binary message

  - Perfectly secure, provided:

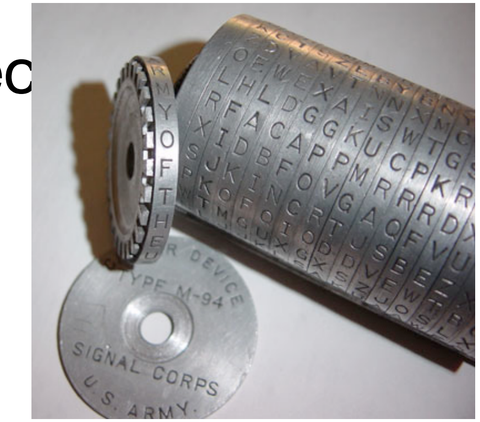
•key is perfectly random

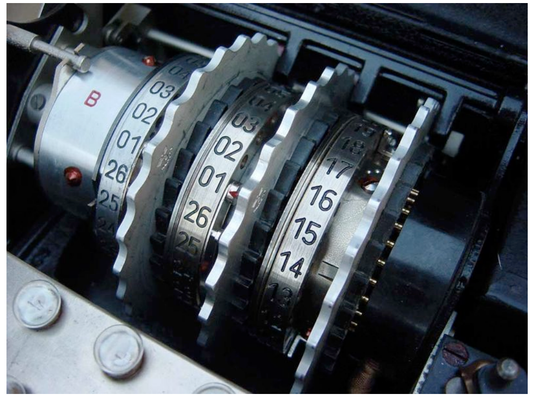
•key is <u>at least as long</u> as the message

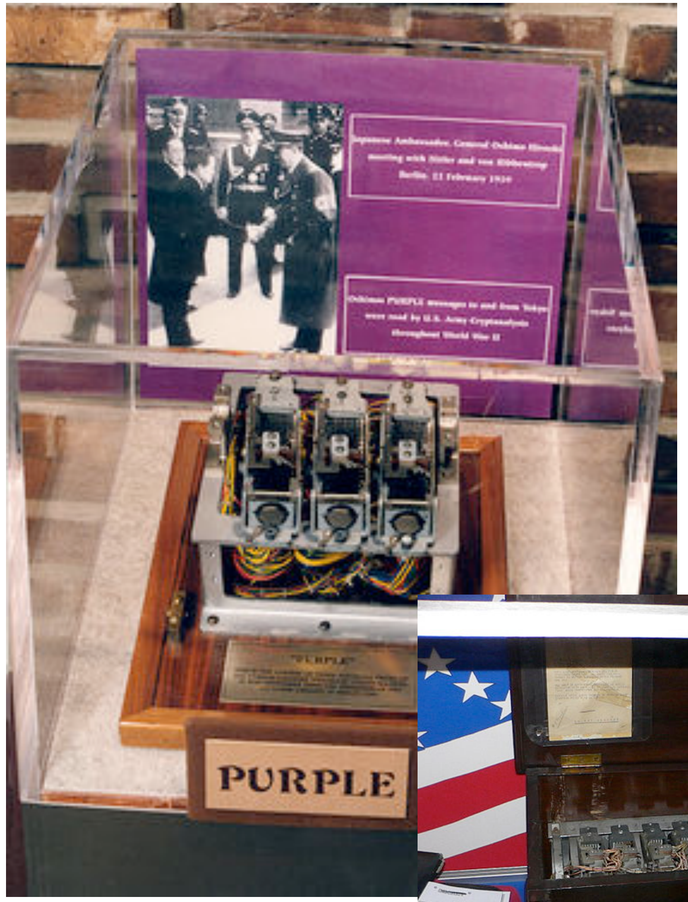
•key is never re-used

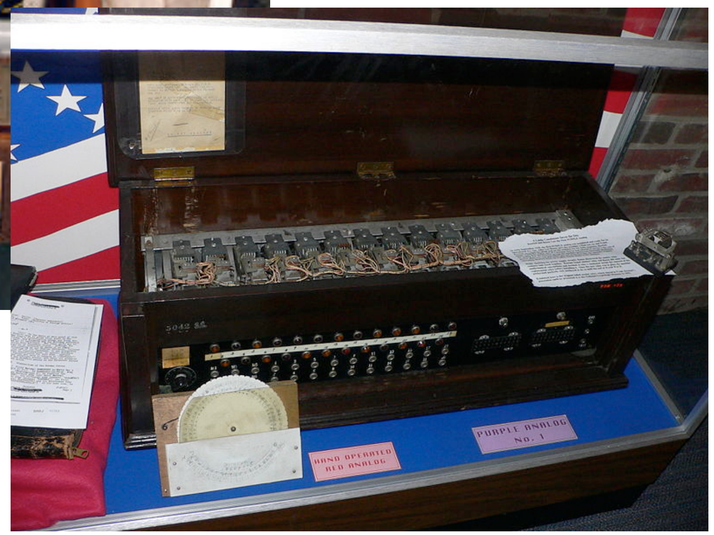
Mechanical Cryptography

- 1900s
  - Mass production and usage of cipher devices
  - Rotor ciphers
  - Elec



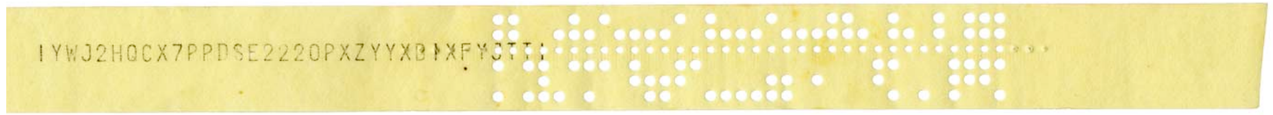






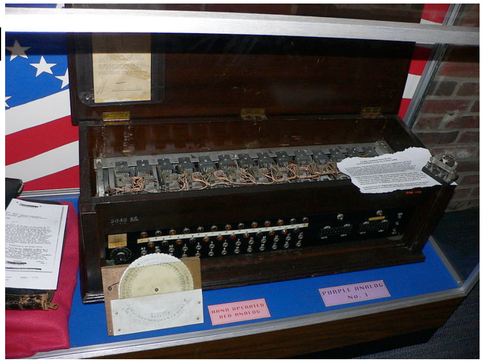
HAGELIN M-209 CIPHER MACHINE (GVG / PD)

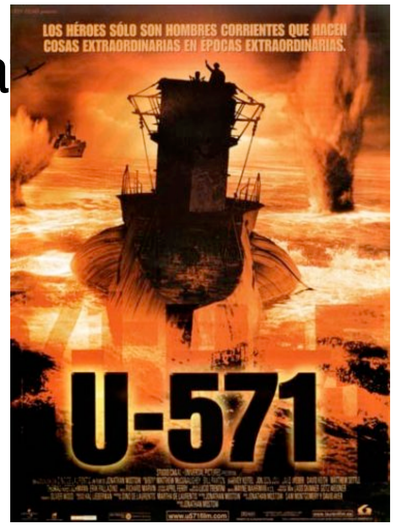
0015
AB CD EF GH IJ KL MN OP QR ST UV WX YZ
AB CD EF GH IJ KL MN OP QR ST UV WX YZ
0 AB
1 CD
2 EF
3 GH
4 IJ
5 KL
6 MN
7 OP
8 ST
9 UV


IYWJ2HQCX7PPDSE2220PXZYYXB1XFY3T

Summary

- Most cryptosystems ultimately broken
  - Sophistication of the attackers outpaces that of the cryptosystem
  - Security relies on <u>secrecy of design</u>
  - Not evaluated for chosen plaintext, known plaintext a
  - Key generation/distribution procedu
  - It’s an arms race...





Kerckhoffs' Principle(s)

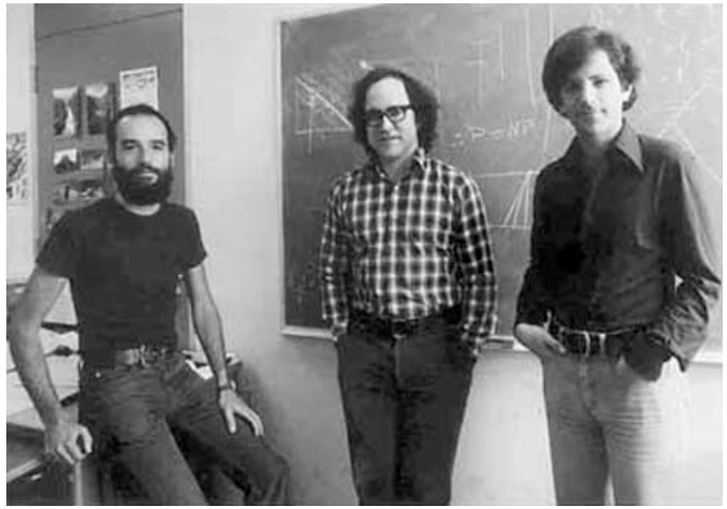
- Auguste Kerckhoffs (1835–1903)

Kerckhoffs' Principle(s)

2. It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience;

"The enemy knows the System"
-- Claude Shannon's Maxim

The 1970s



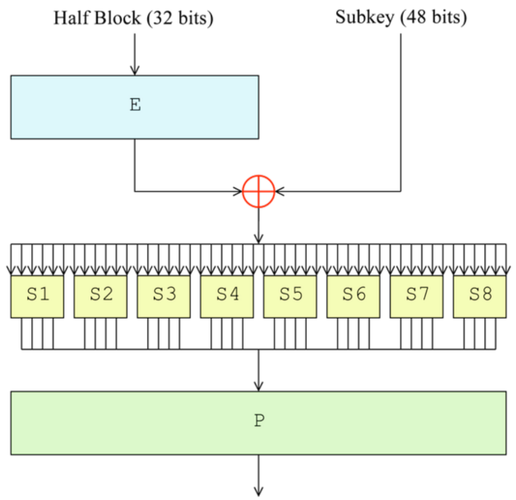
U.K. GCHQ

The Implications

- Exponential increase in study & usage of cryptography in industry, academia
- Wide-scale deployment of cryptographic systems
- <u>Provable Security</u>
  - Cryptographic Systems can be <u>reduced</u> to some hard mathematical problem

Data Encryption Standard

- Commercial-grade Block Cipher
  - 64-bit block size
  - 56 bit key (+ 8 bits parity)
  - “Feistel Network” Construction



Permutation

Permutation

Permutation Families

- Can’t have just <u>one</u> permutation
  - Alice & Bob know the permutation
    Adversary doesn’t
  - Permutation is “random” (ish)
  - But there are possible permutations
  - DES has a 56 bit key...

Block Cipher

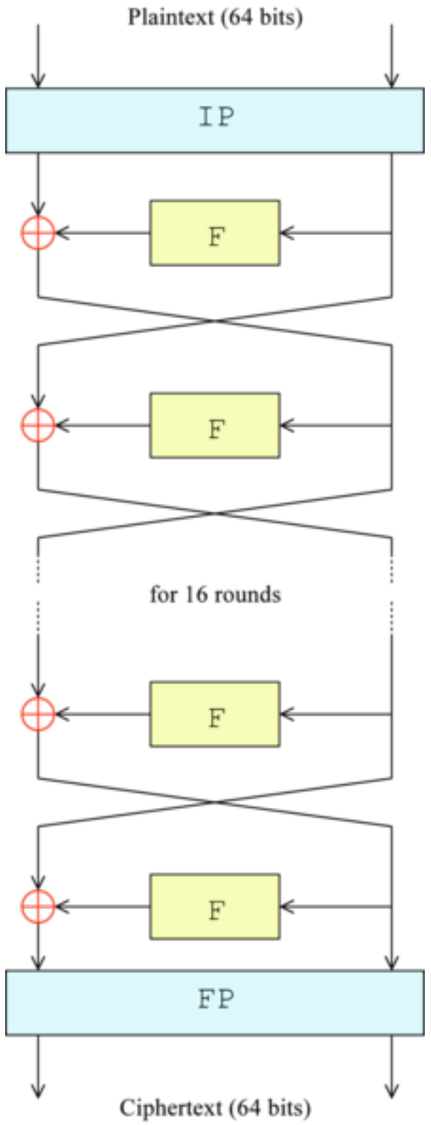
- Block cipher is a family of permutations
  - Indexed by a key (DES = 56 bit key)
  - “Pseudo-random”

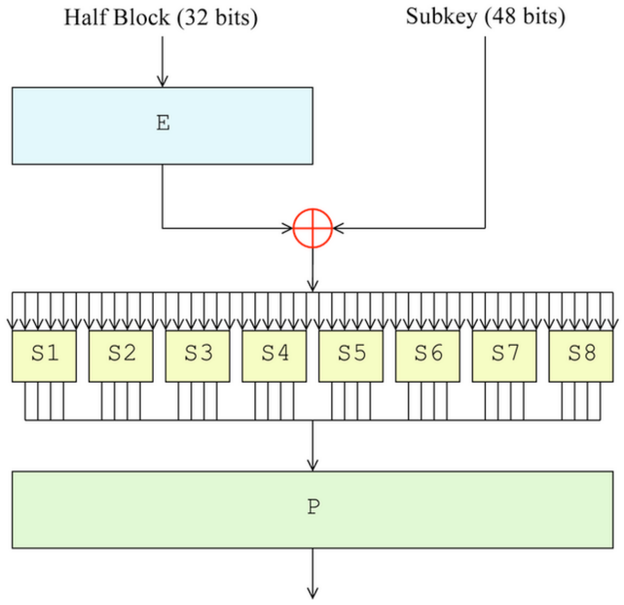
Block Cipher

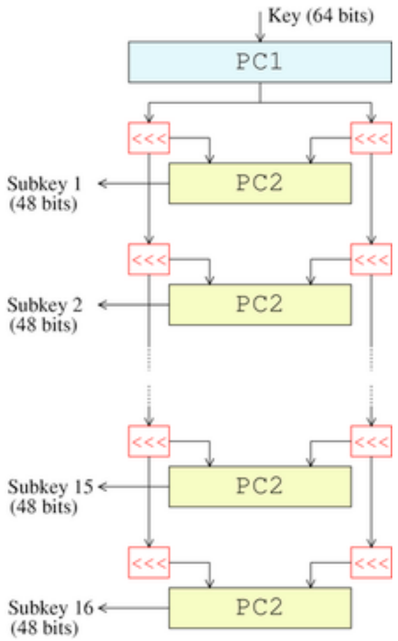
- Block cipher is a family of permutations
  - Indexed by a key (DES = 56 bit key)
  - Ideally: “Pseudo-random permutation (PRP)”

    (i.e., attacker who does not know the key can’t determine whether you’re using a <u>random</u> permutation, or a PRP)

DES: 64-bit Block, 56-bit Key

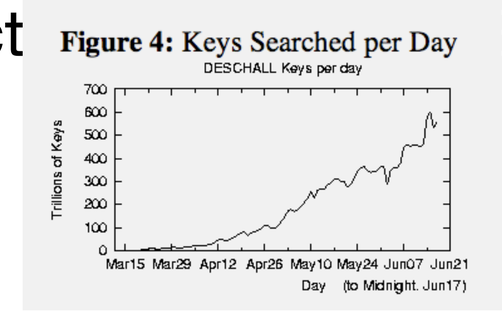


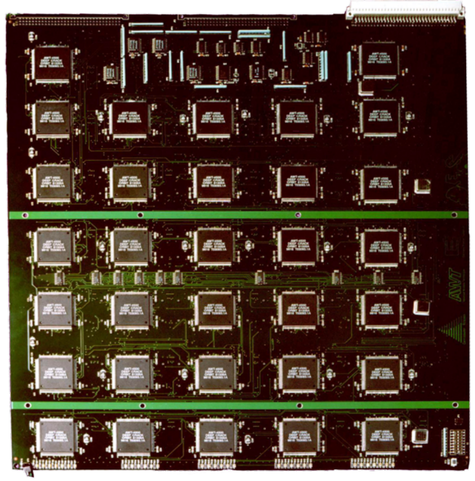




DES

- Some “clever” attacks on DES
  - However: practical weakness = 56 bit key size
  - Pract ES (now being deprecated)





U.S. Data-Scrambling Code Cracked With Homemade Equipment

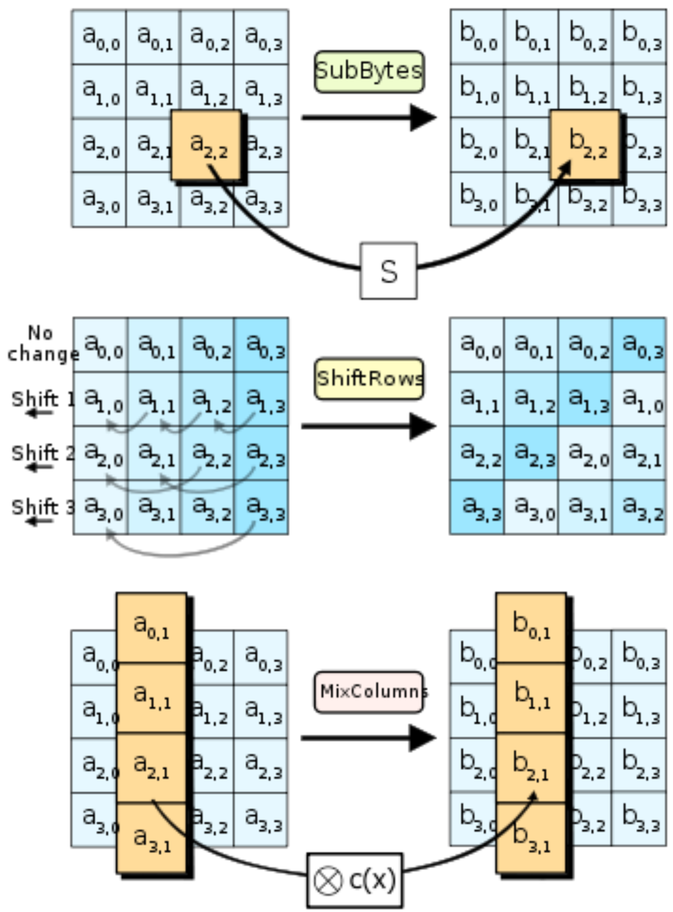
By JOHN MARKOFF

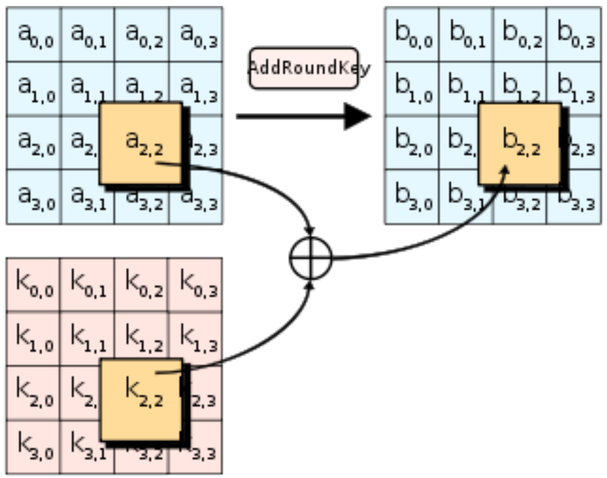
SAN FRANCISCO -- In a 1990s variant of a John Henry-style competition between man and machine, researchers using a homemade supercomputer have cracked the government's standard data-scrambling code in record time -- and have done it by out-calculating a team that had harnessed thousands of computers, including some of the world's most powerful.

AES

- NIST open competition:
  - Fast in software & hardware
  - Larger block size (128 bit)
  - Longer keys (128/192/256-bit)
- 5 finalists:
  - MARS, RC6, Rijndael, Serpent, and Twofish

AES: 128-bit Block, 128/192/256-bit Key



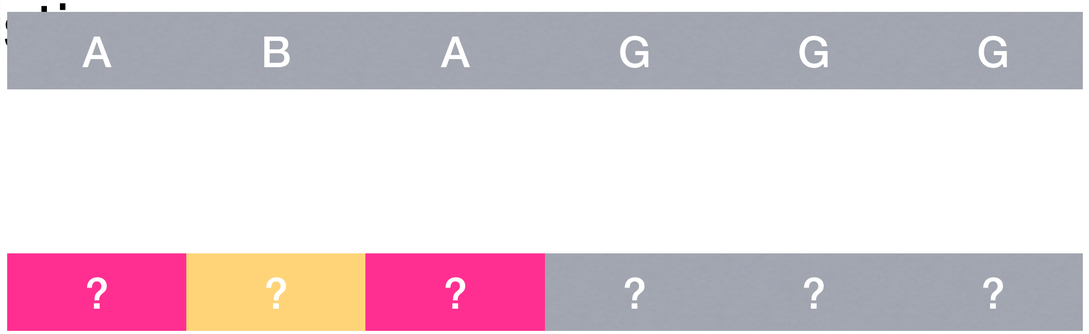


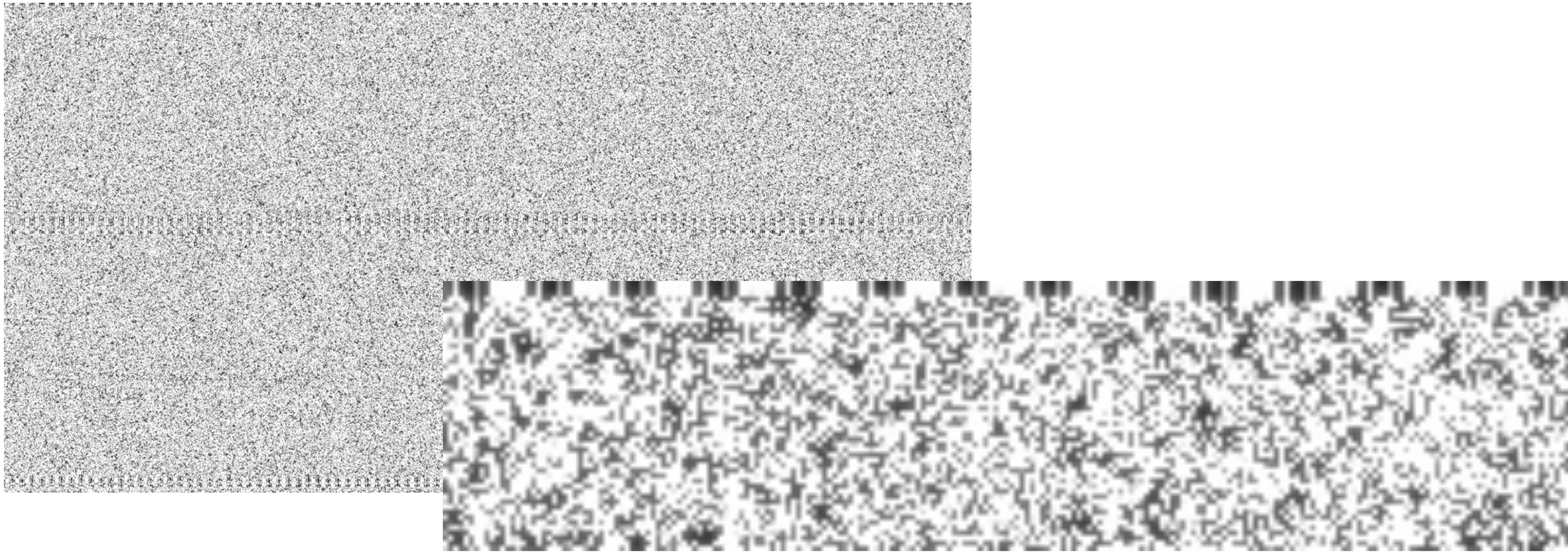
Attacks on AES

- Attacks on the design:
  - All theoretical
  - Work on reduced-round versions
  - Some debate about whether estimates are correct
- Side channel attacks:
  - Timing, cache hits & timing

Using Block Ciphers

- ECB Mode: Encrypt each block separately
  - Bad idea!
  - May reveal distribution of the plaintext
  - Totally determini





Defining Security

- What we'd like from our encryption scheme?
  - Ciphertext doesn't leak *any* info about plaintext
  - Even if Adversary knows a lot about the plaintext distribution
  - Even if Adversary can choose the plaintext distribution

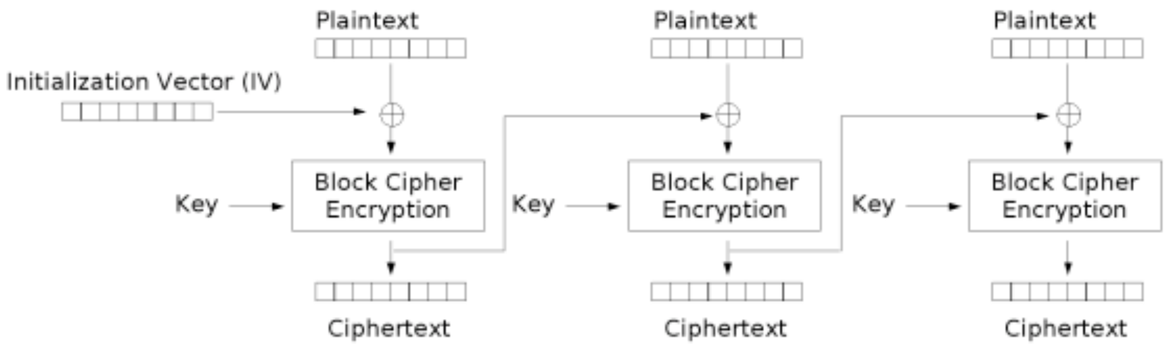
Defining Security

- Semantic Security
  - “Indist. under chosen plaintext attack” (IND-CPA)
  - Adversary w/ the ciphertext learns as much as Adversary w/o the ciphertext

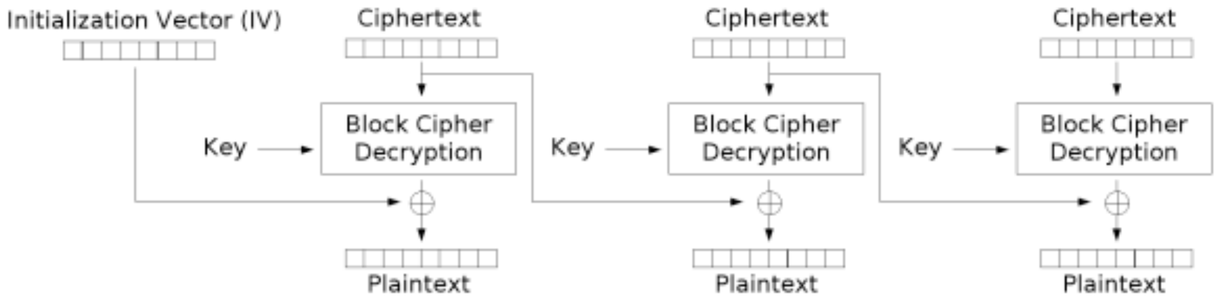
Using Block Ciphers

- ECB is not semantically secure, hence we use a “mode of operation”
  - e.g., CBC, CTR, CFB, OFB (and others)
- These provide:
  - Security for multi-block messages
  - Randomization (through an <u>Initialization Vector</u>)

CBC Mode



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

Security of CBC

- Is CBC secure?
  - Yes, assuming a secure block cipher
  - IND-CPA
- Easy to use wrong...
  - Most imporant: use a unique & random IV!

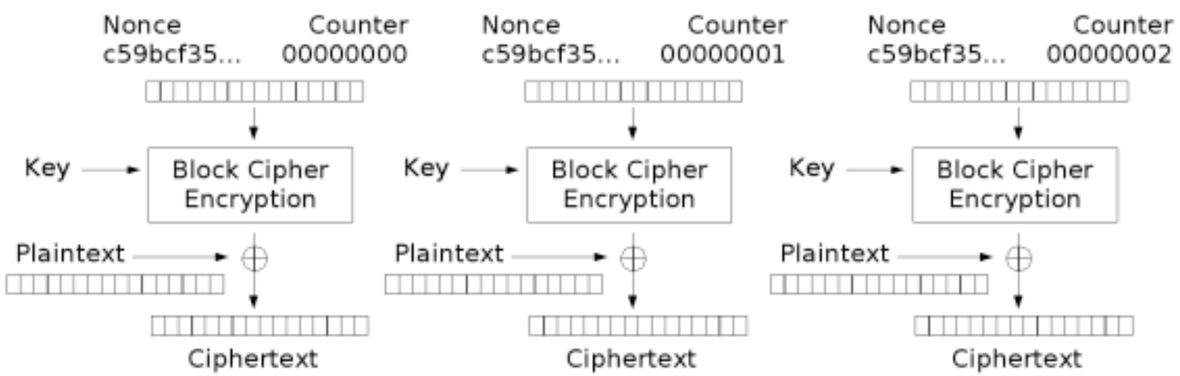
The size of the frame of data to be encrypted or decrypted (i.e. how often a new CBC chain is started) depends on the particular application, and is defined for each in the corresponding format specific books of this specification. Unless otherwise specified, the Initialization Vector used at the beginning of a CBC encryption or decryption chain is a constant, iv_0, which is:

$$0BA0F8DDFEA61FB3D8DF9F566A050F78_{16}$$

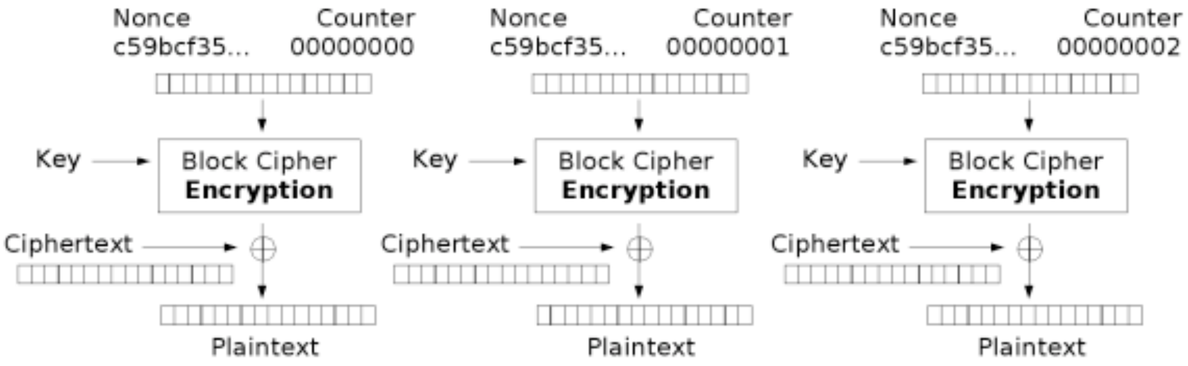
Advanced Access Content System (AACS)

Introduction and
Common Cryptographic Elements

CTR Mode



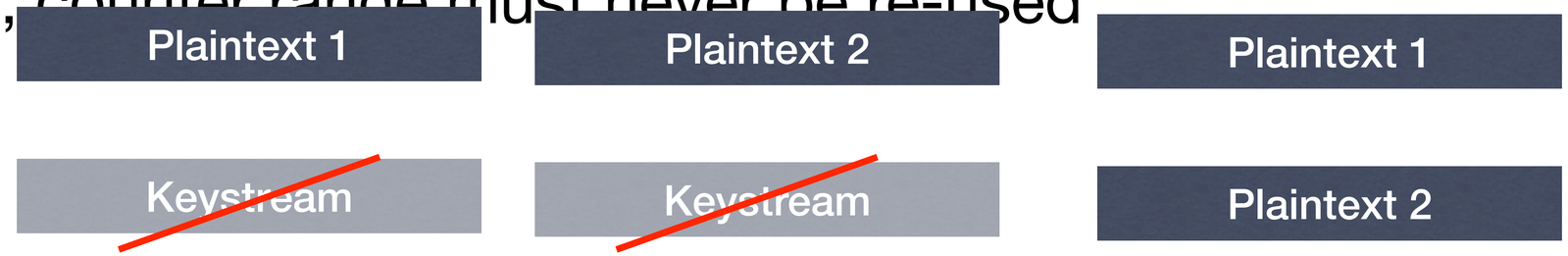
Counter (CTR) mode encryption



Counter (CTR) mode decryption

Security of CTR

- Yes, assuming secure block cipher
- However, counter range must never be re-used



- Similar example: MS Word 2003
  - (they used RC4, but same problem)

Point of order

- Proofs of security:
  - We <u>don't</u> know how to prove that DES or AES are secure block ciphers
  - But if we assume that the block ciphers <u>are</u> secure:

•We can prove that CBC & CTR are secure encryption modes.

Malleability

- The ability to modify a ciphertext
  - Such that the plaintext is meaningfully altered
  - CTR Mode (bad)
  - CBC Mode (pretty bad)
- The solution:
  - Authenticated Encryption

MACs

- Symmetric-key primitive
  - Given a key and a message, compute a “tag”
  - Tag can be verified using the same key
  - Any changes to the message detectable
- To prevent malleability:
  - Encrypt <u>then</u> MAC
  - Under separate keys

MACs

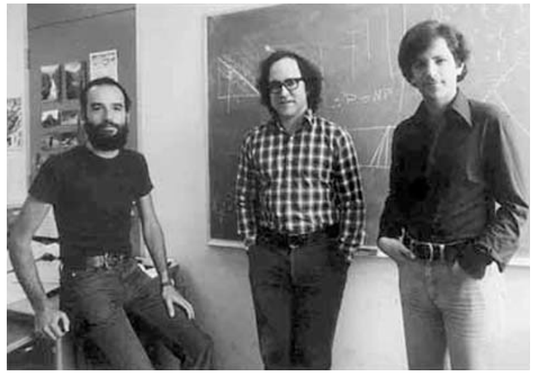
- Definitions of Security
  - Existential Unforgeability under CMA
- Examples:
  - HMAC (based on hash functions)
  - CMAC (block ciphers)

Presentations

- Lest I forget... (2/19 or 2/26)
- My suggestions:
  - Random Number Generation
  - Attacks on WinZip
  - Cracking Passwords (Rainbow Tables)
  - But... other ideas welcome!

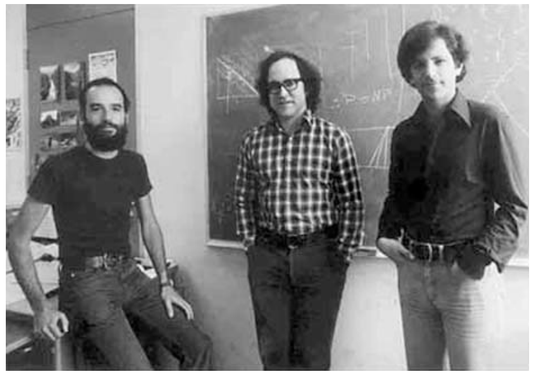
Next Time

- On through the 70s!
  - Key agreement
  - Public Key Encryption



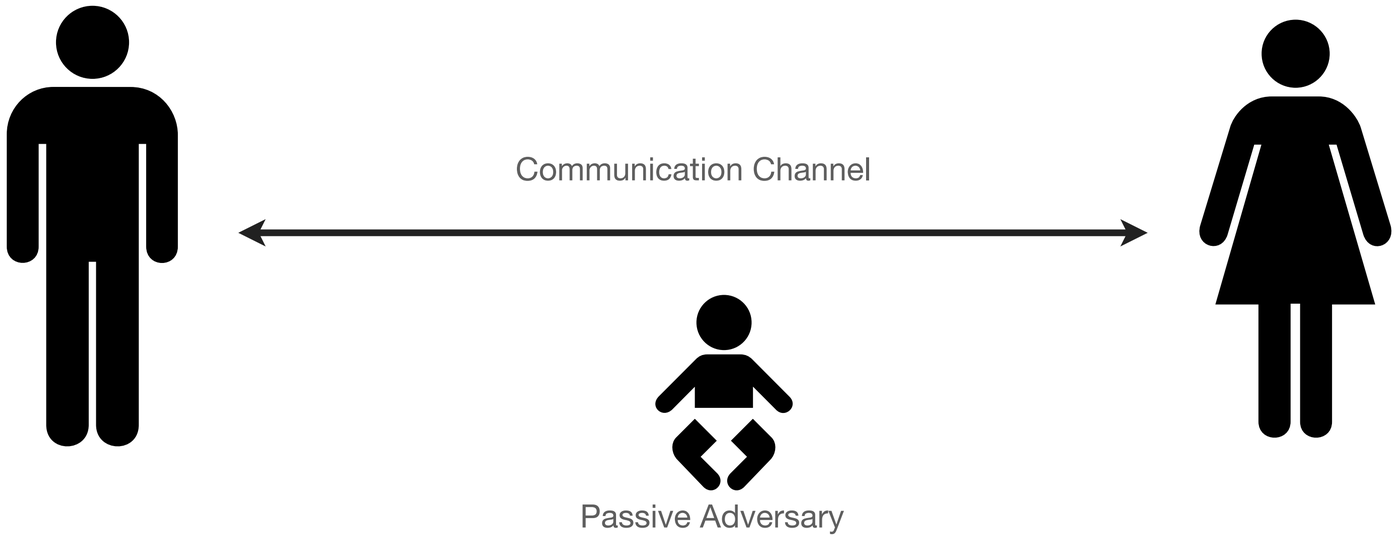
Asymmetric Crypto

- So far we've discussed <u>symmetric</u> crypto
  - Requires both parties to share a key
  - Key distribution is a hard problem!
- But there is anothe

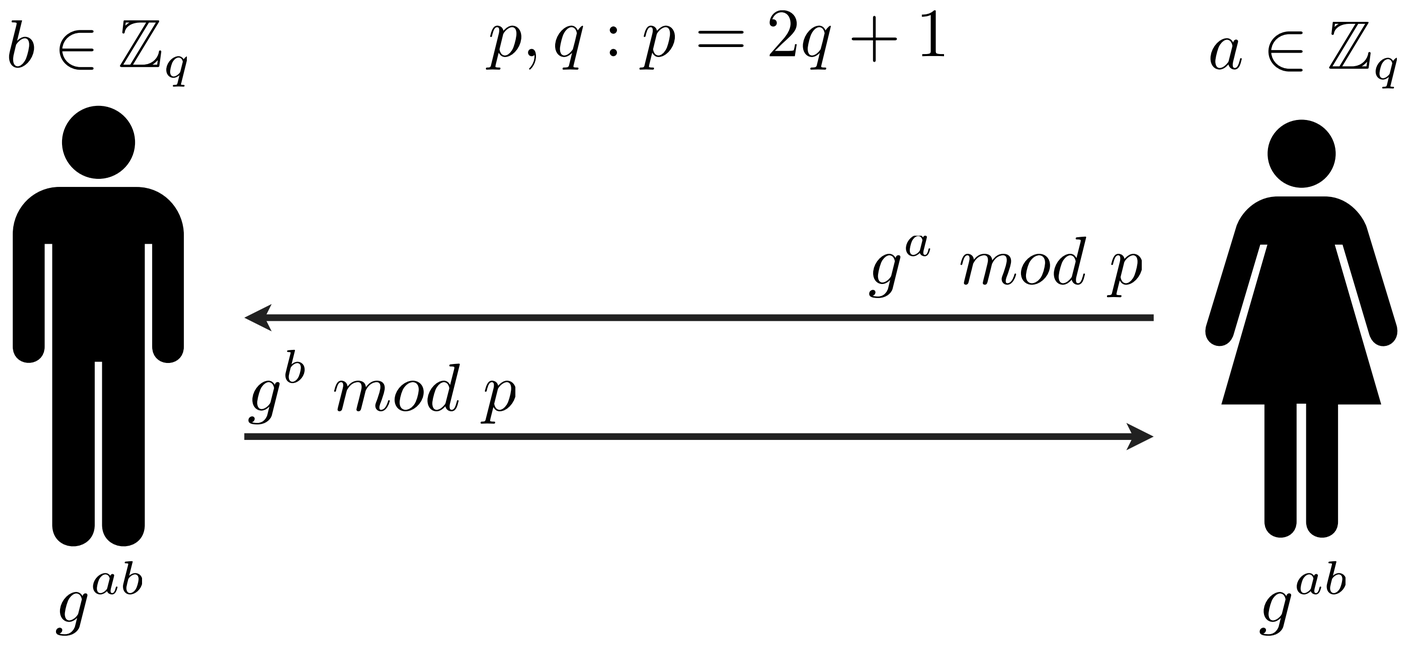


Key Agreement

- Establish a shared key in the presence of a passive adversary

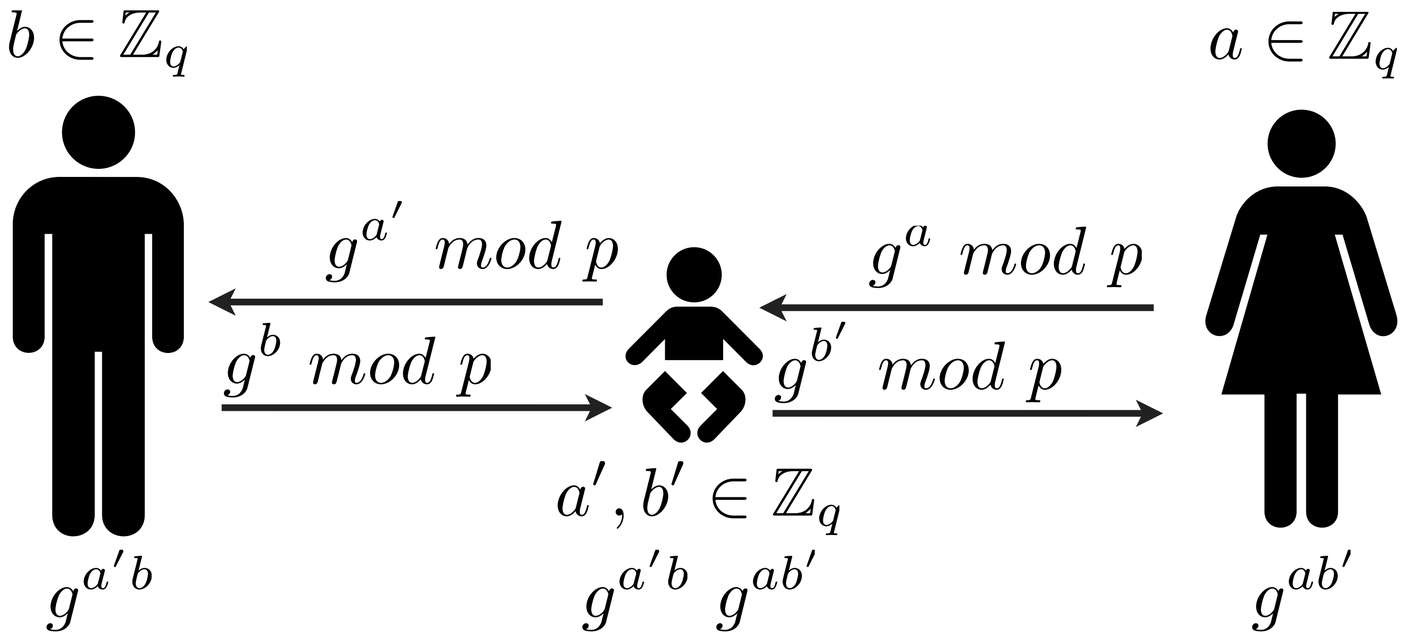


D-H Protocol



Man in the Middle

- Assume an active adversary:



Man in the Middle

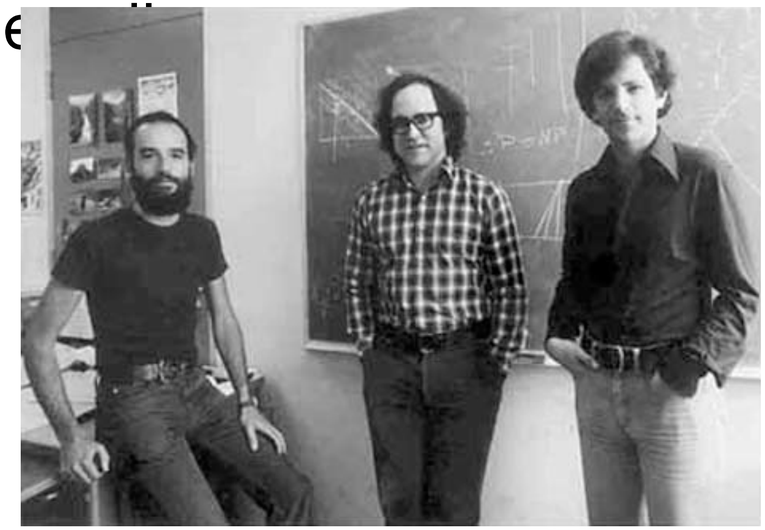
- Caused by lack of <u>authentication</u>
  - D-H lets us establish a shared key with anyone... but that’s the problem...
- Solution: Authenticate the remote party

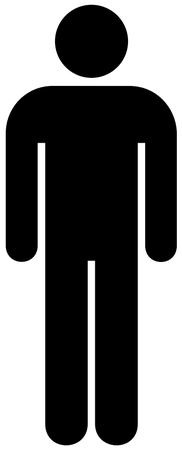
Preventing MITM

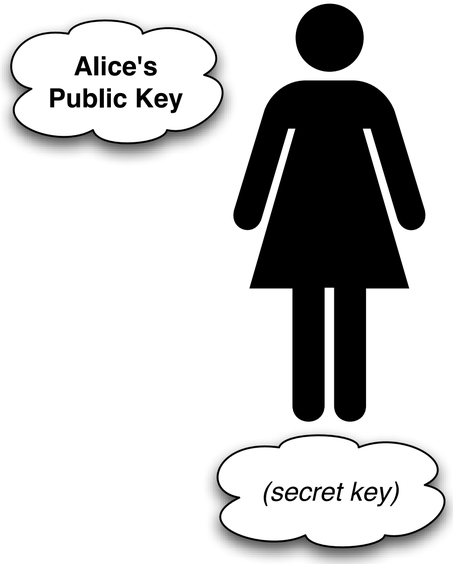
- Verify key via separate channel
- Password-based authentication
- Authentication via PKI

Public Key Encryption

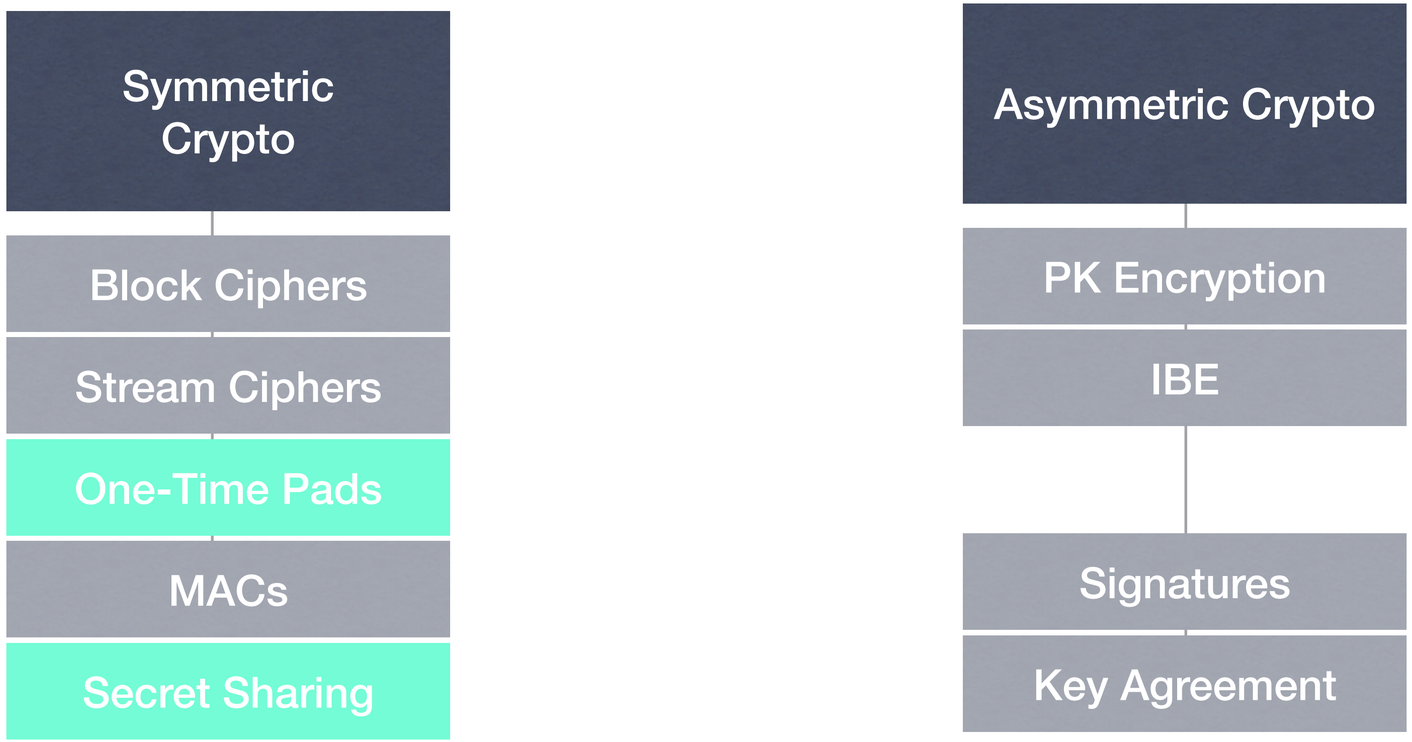
- What if our recipient is <u>offline?</u>
  - Key agreement protocols are interactive
  - e.g., want to send an e






Alice's
Public Key
(secret key)

Basic Primitives

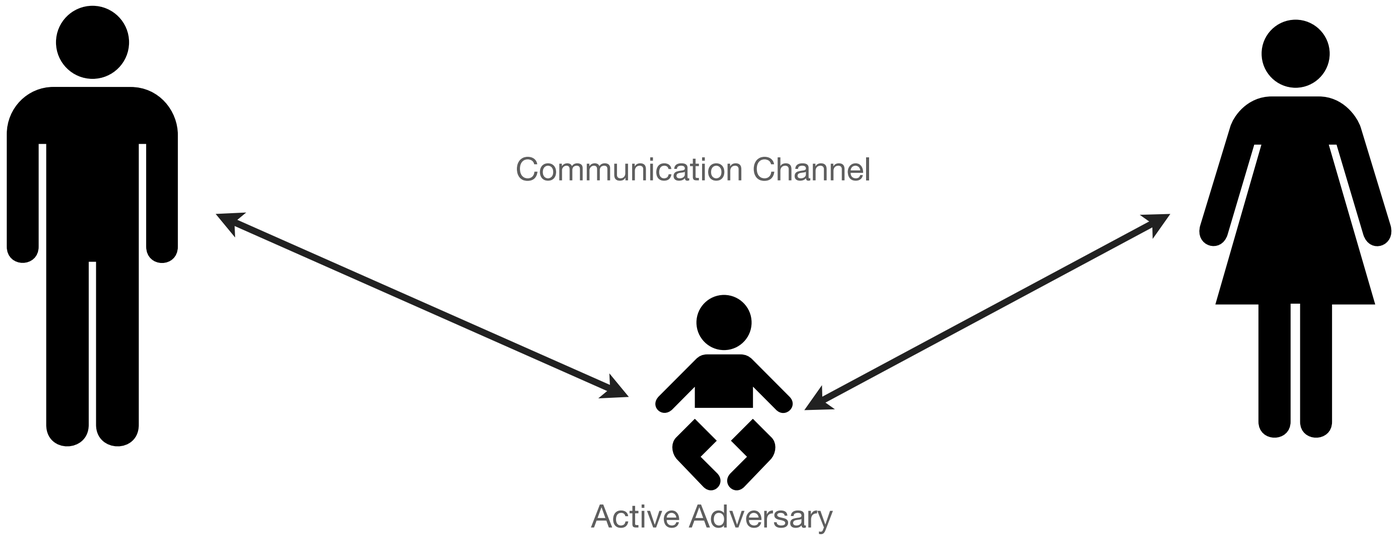


Information Theoretic Complexity The

Next Time

MACs

Authenticated Encryption



Convenience vs. Security