Practical Cryptographic Systems

Symmetric Cryptography III

Instructor: Matthew Green

Housekeeping

- A1 still out
  - **Submit via Gradescope! Assignment will be set up by TA.**
  - ***Not Blackboard (I'm sorry)***
- Reading quiz/assignment coming tonight (it's short!)
  - Boneh/Shoup book readings
- Ditto for Projects list

News?

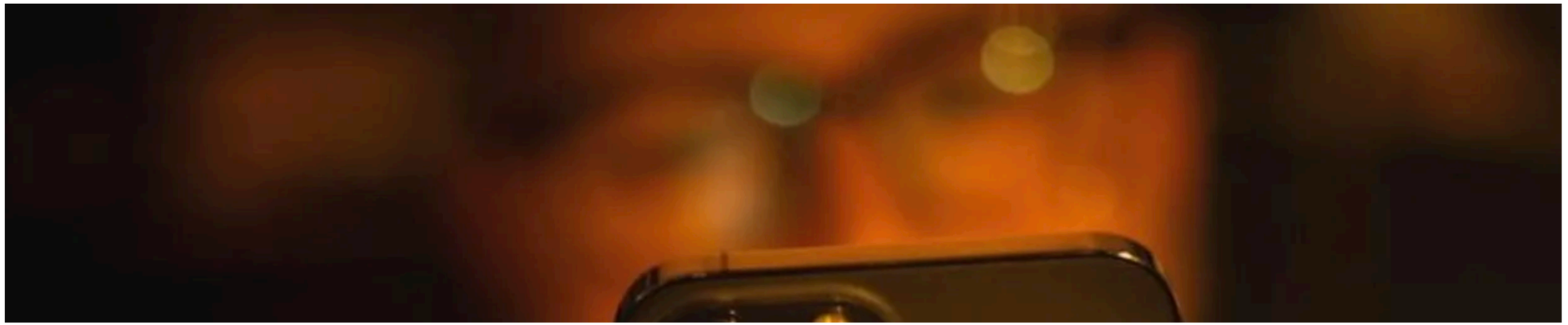
Nev

Apple says UK could 'secretly veto' global privacy tools

29th January 2024, 12:36 EST

Share

By Zoe Kleinman
Technology editor



Apple has attacked proposals for the UK government to pre-approve new security features introduced by tech firms.

Under the proposed amendments to existing laws, if the UK Home Office declined an update, it then could not be released in any other country, and the public would not be informed.

The government is seeking to update the Investigatory Powers Act (IPA) 2016.

The Home Office said it supported privacy-focused tech but added that it also had to keep the country safe.

A government spokesperson said: "We have always been clear that we support technological innovation and private and secure communications technologies, including end-to-end encryption, but this cannot come at a cost to public safety."

The proposed changes will be debated in the House of Lords tomorrow.

Apple says it is an "unprecedented overreach" by the UK government.

Review: block cipher

- Block cipher is a family of permutations
  - Indexed by a key (DES = 56 bit key, AES = 128/192/256)
  - Ideally: “Pseudo-random permutation (PRP)”

    (i.e., *computationally bounded* attacker who does not know the key can’t determine whether you’re using a random permutation, or a PRP)

Review: block cipher

Attack Game 4.1 (block cipher). For a given block cipher (E, D), defined over $(\mathcal{K}, \mathcal{X})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define:

Experiment b:

- The challenger selects $f \in \text{Perms}[\mathcal{X}]$ as follows:

  if $b = 0$: $k \xleftarrow{R} \mathcal{K}$, $f \leftarrow E(k, \cdot)$;
  if $b = 1$: $f \xleftarrow{R} \text{Perms}[\mathcal{X}]$.

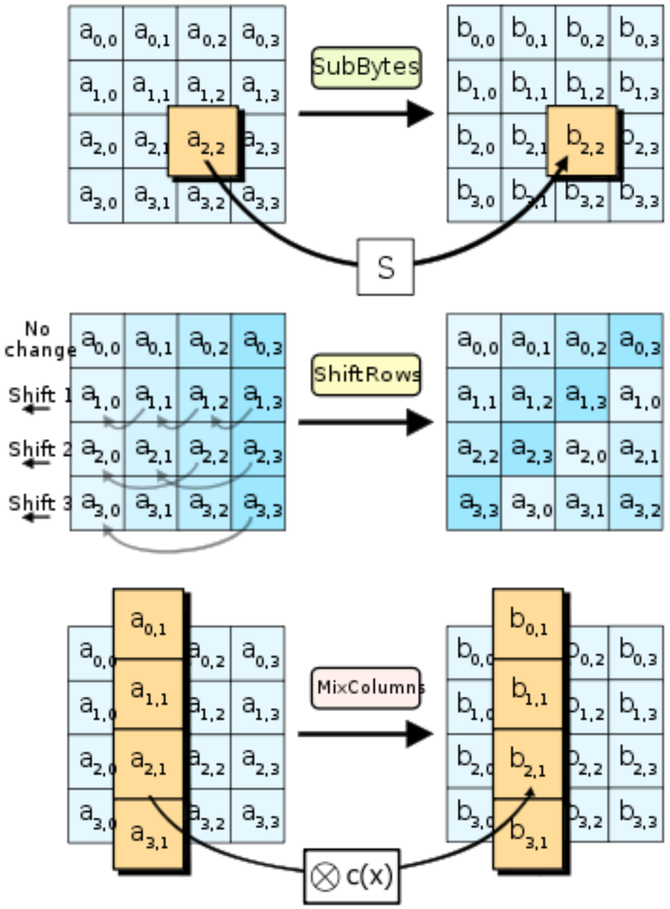
- The adversary submits a sequence of queries to the challenger.

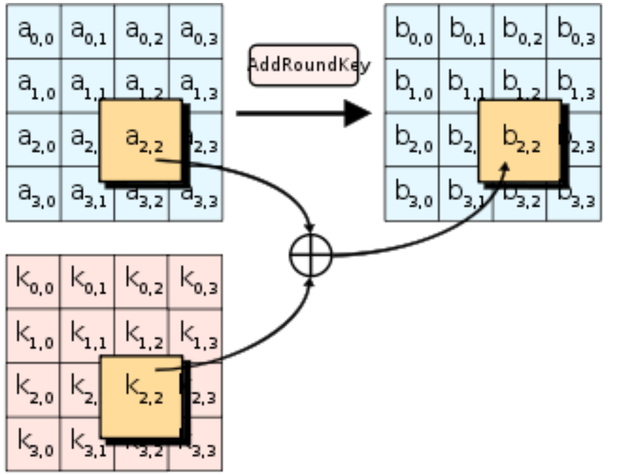
  For $i = 1, 2, \ldots$, the ith query is a data block $x_i \in \mathcal{X}$.

Review: AES

- NIST open competition (around 1998). Requirements:
  - Fast in software & hardware
  - Larger block size (128 bit)
  - Longer keys (128/192/256-bit)
  - Reviewed by academics & NSA
- 5 finalists:
  - MARS, RC6, Rijndael, Serpent, and Twofish

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





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 Initialization Vector)

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

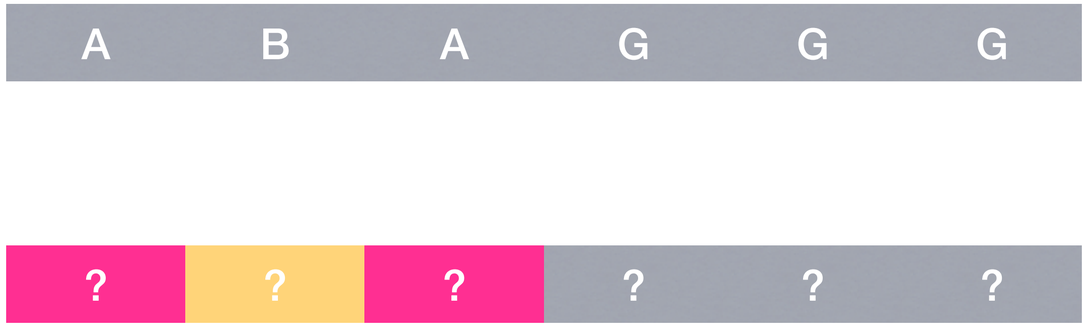
- Problem #1: ECB is deterministic

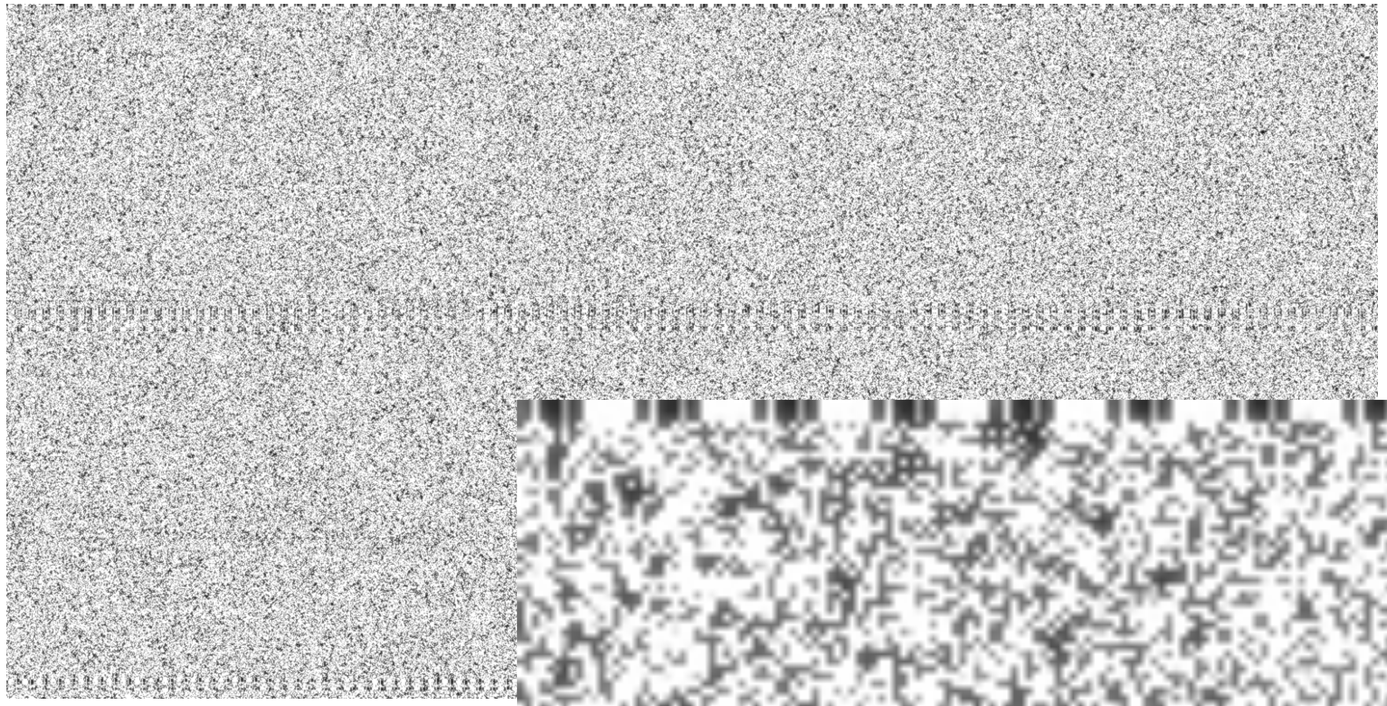
E(Attack Monster)

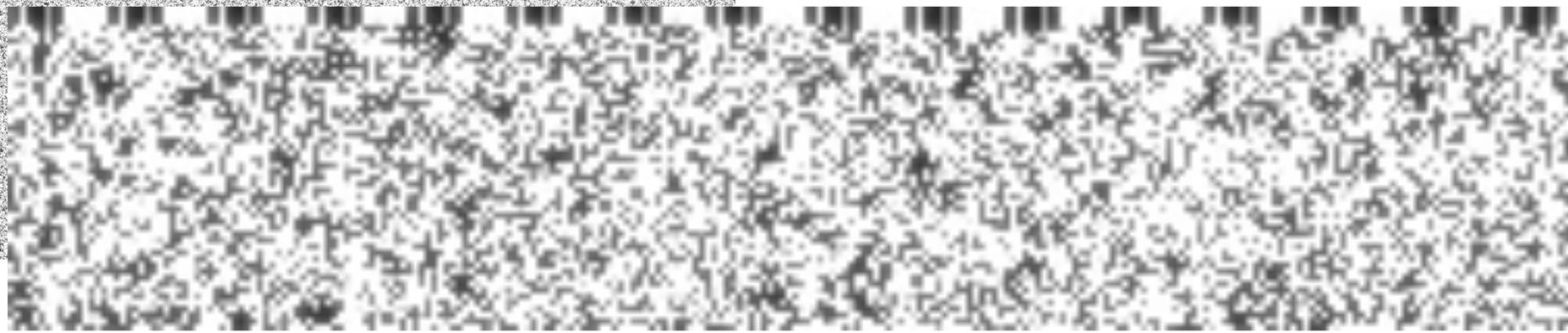
E(Monster Attacks)

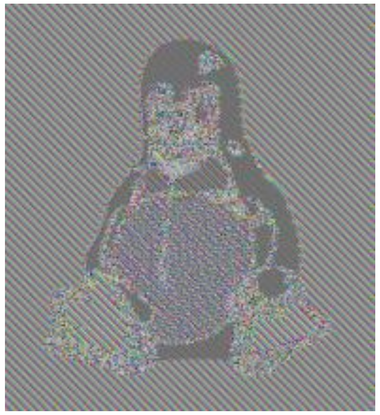
ECB Mode

- Problem #2: Leakage of plaintext patterns









Security of Encryption

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

•Even if adversary can request chosen plaintexts

  - How do we state this formally?

Semantic security

- Semantic Security (IND-CPA)

$M_1, ..., M_n$

$E(k, M_1), ..., E(k, M_n)$

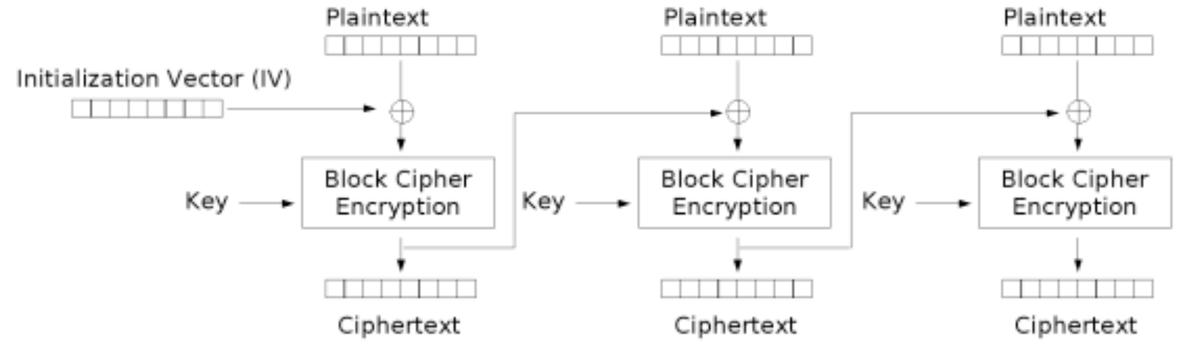
M_0, M_1

$E(M_b)$

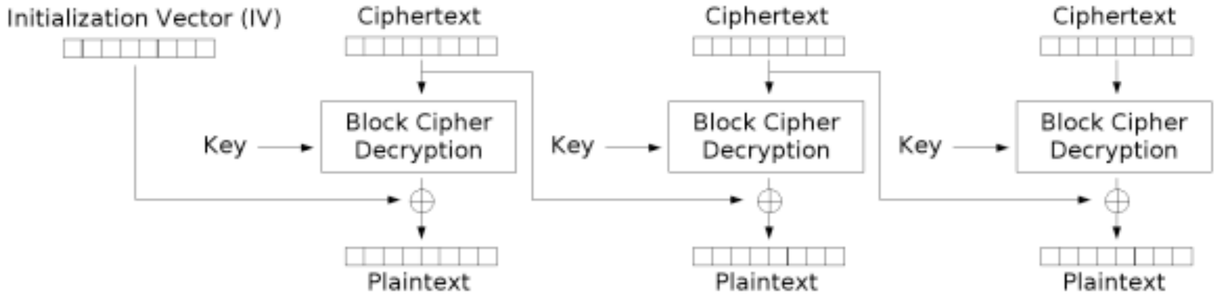
Using Block Ciphers

- 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 Initialization Vector)

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 passive adversary)
  - Correct (random) IV generation
  - Can prove 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!

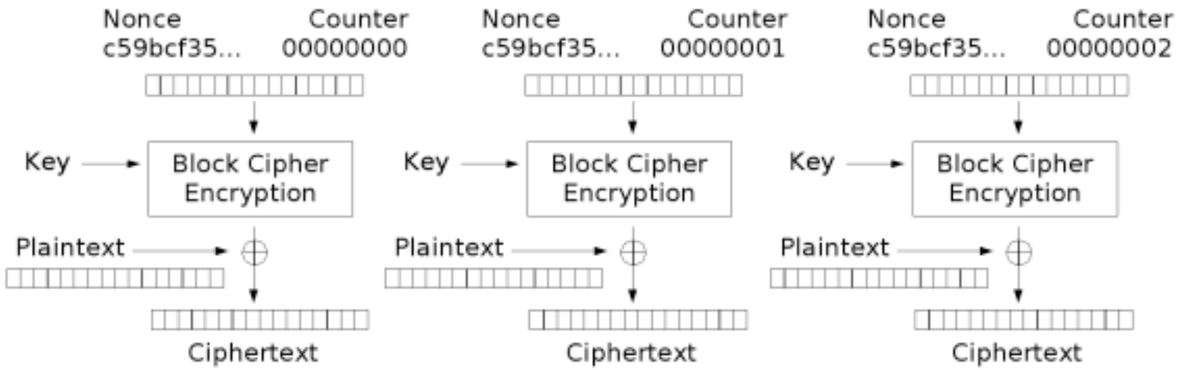
The size of the frame of data to be encrypted or decrypted (i.e. how often a new CBC chain is started) depends on the particular application, and is defined for each in the corresponding format specific books of this specification. Unless otherwise specified, the Initialization Vector used at the beginning of a CBC encryption or decryption chain is a constant, $iv_{0,}$ which is:

$0BA0F8DDFEA61FB3D8DF9F566A050F78_{16}$

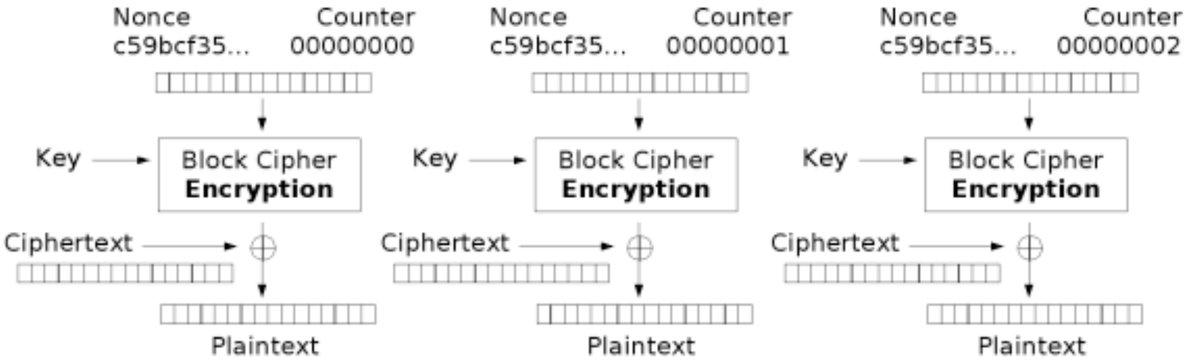
Advanced Access Content System (AACS)

Introduction and Common Cryptographic Elements

CTR Mode



Counter (CTR) mode encryption



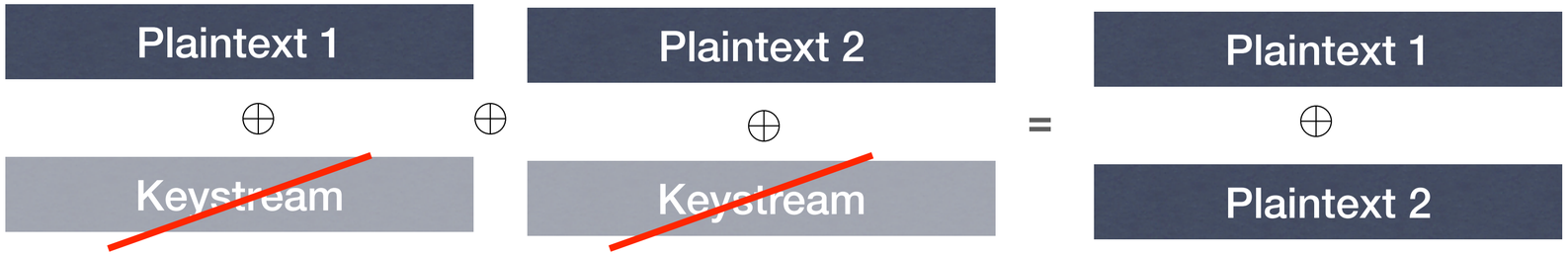
Counter (CTR) mode decryption

CTR (intuition)

- CTR uses the cipher to expand a short cipher key (K) into a long string of pseudorandom output bits
  - 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 never be re-used



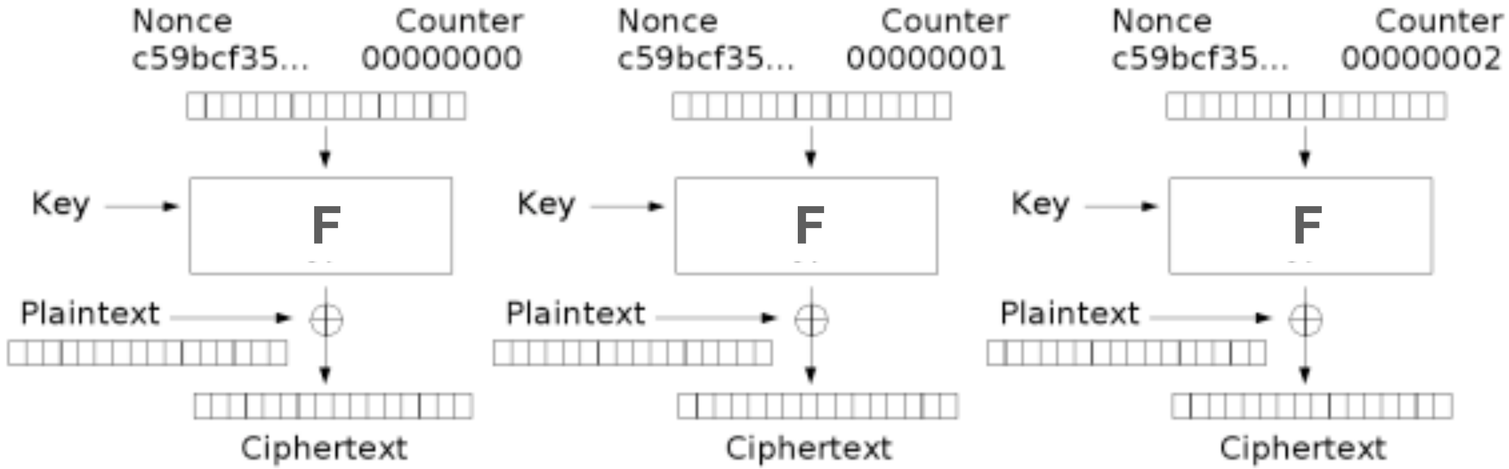
- Similar example: MS Word 2003
  - (they used RC4, but same problem)

CTR with other functions

- We've been assuming a cipher that is invertible (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 invertible (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?



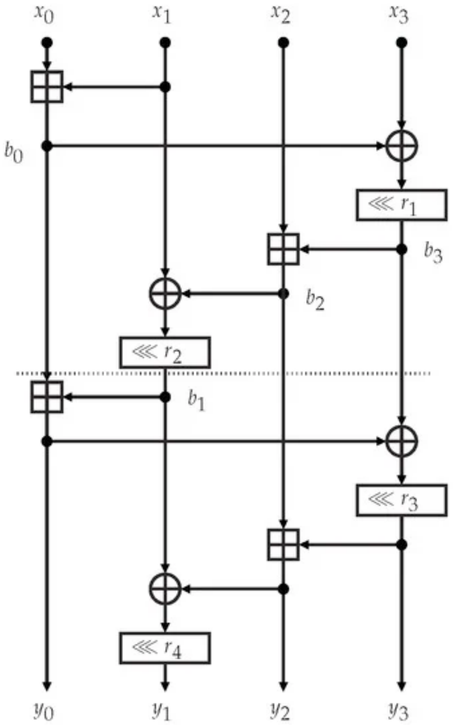
Counter (CTR) mode encryption

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

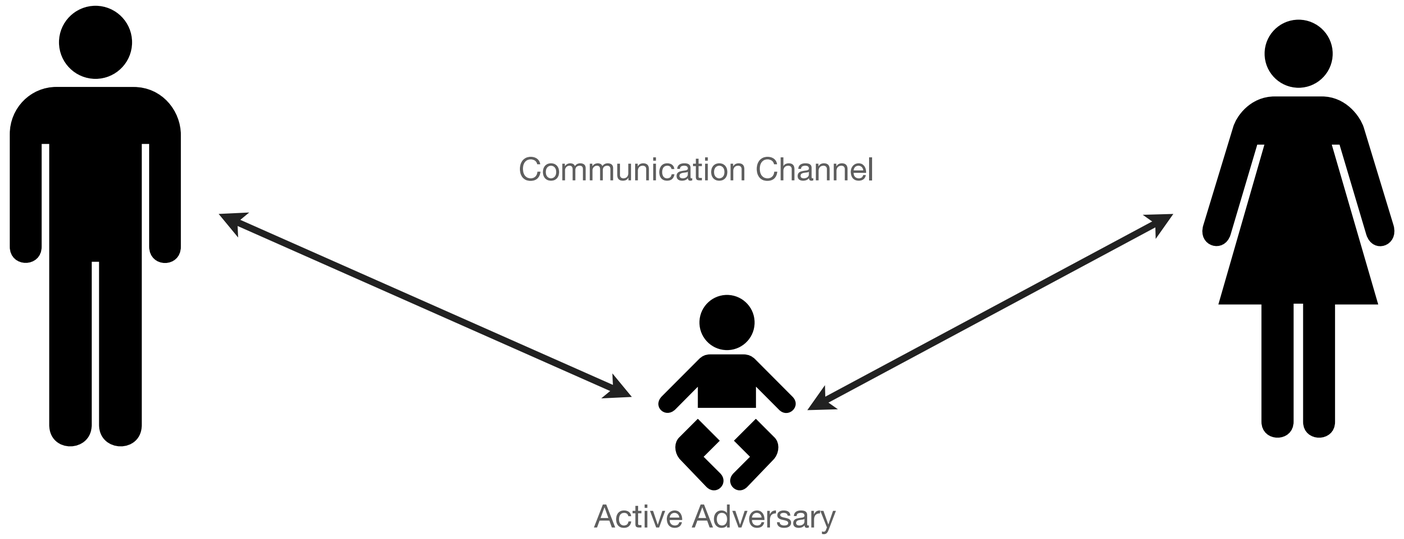
Malleability

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

Malleability

- The ability to modify a ciphertext
  - Such that the plaintext is meaningfully altered
  - CTR Mode, and stream ciphers (very bad!)
  - CBC Mode (somewhat bad)

Authenticated Encryption



Message Authentication Codes (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 then 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:
  - Generic composition
    Encrypt (e.g., CBC mode) then MAC

•two different keys, multiple primitives

- Authenticated mode of operation

•Integrates both encryption & authentication

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

•Ex: CCM, OCB, GCM modes