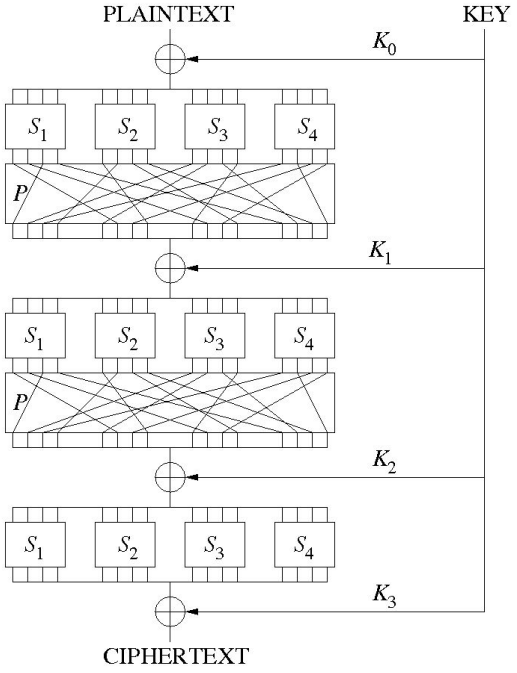
AES (Advanced Encryption Standard)

- Uses **substitution-permutation network:**
  1. Each plaintext block is divided into sub-blocks. Round key is derived from the general key, and combined with sub-blocks in each round to form input sub-blocks.
  2. Sub-blocks go through different substitution boxes (S-boxes) or permutation boxes (P-boxes) to produce the cipher text block. S/P boxes are applied in alternating rounds. S-boxes are like substitution cipher, whereas P-boxes distribute any bit to as many S-boxes input as possible.
  * Change in one input bit to S-boxes result in huge change of output bits
  * Satisfy confusion and diffusion principle
- Richard claims that it seems to not have been broken yet in feasible time
- *Note: might need to make this section a bit more comprehensive*



Buffer Overflow

- One CPU does one thing at a time
- Rapid *context switching* occurs between processes
  - Hardware, operating system *interruptions*
- *Stack* keeps track of all processes currently interrupted
  - Lets CPU know what processes to switch back to
- The latest process in the stack is at top
  - When it finishes it is thrown away and stack pointer moves down
- When a process is uninterrupted, its info must be restored
  - Info is stored in disk
- Current running program can also store data in stack
  - Hence, why not store data in stack rather than disk
- Writing to invalid memory/*beyond end of buffer* (eg. Overflowing an array) may cause memory of an interrupted process to be overwritten
  - Simplest example is when user input overflows an array but the program doesn't check
  - The mem overwritten may contain the data for the next instruction that process needs to carry out when uninterrupted
  - To perform attack, write to *return address* of next process to wake up and tell it go to the start of the array, where your malicious code is written

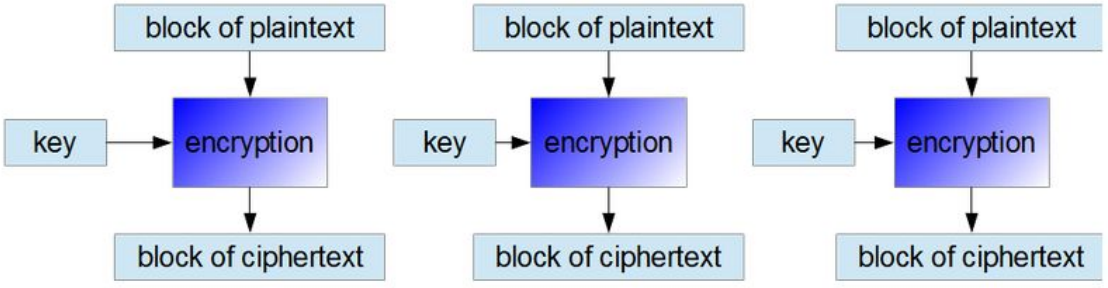
- If you don't know where in memory your array is located, you'll need to cause a memory leak to find out

Block Modes http://www.crypto-it.net/eng/theory/modes-of-block-ciphers.html

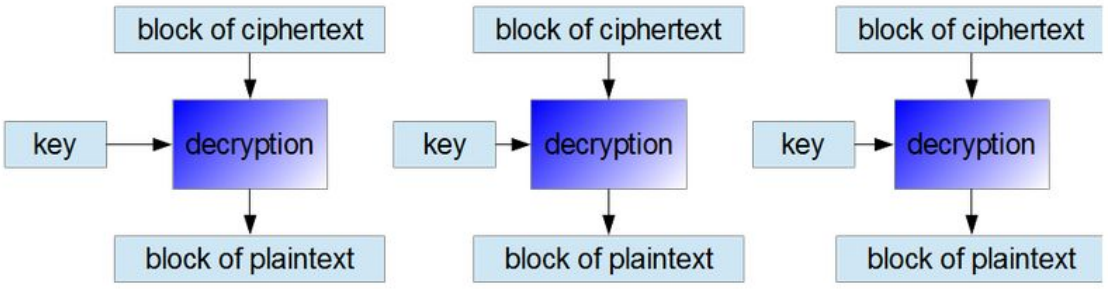
- Allow block ciphers to work with large data streams without the risk of compromising the provided security
- **initialization vector** is randomly generated used to ensure distinct ciphertexts are produced even when the same plaintext is encrypted multiple times independently with the same key

ECB (Electronic Codebook Mode)

- Each plaintext block is encrypted separately
- Each ciphertext block is decrypted separately

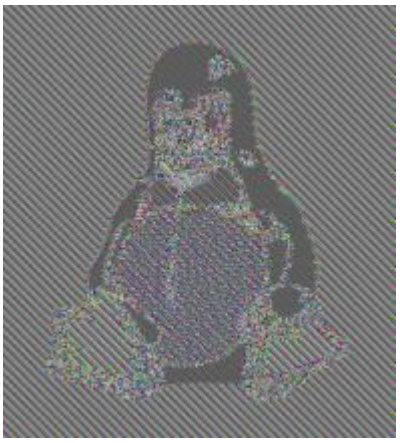


Encryption in the ECB mode



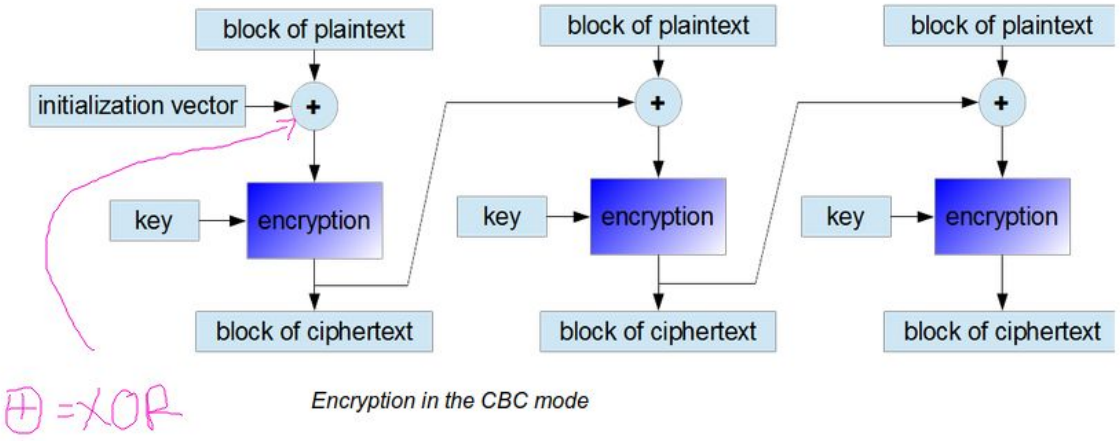
Decryption in the ECB mode

- Never use ECB cuz of the penguin image (on wikipedia) - like excrypts to like. Linux penguin logo encrypted with ECB clearly shows outline of original image

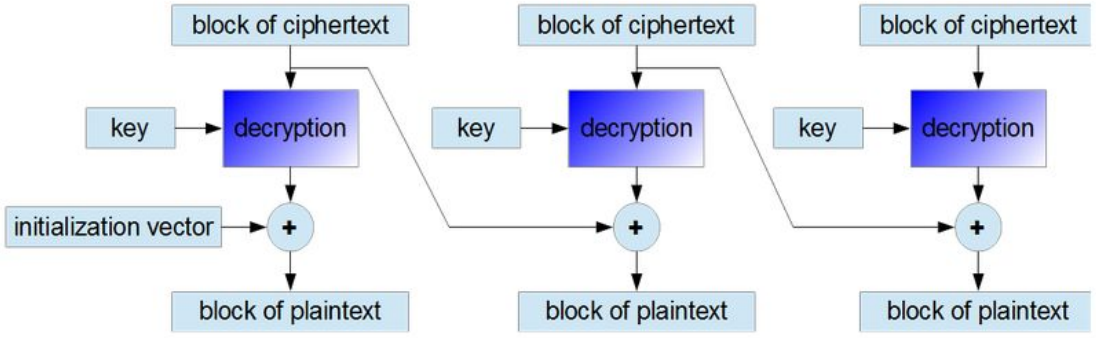


CBC (Cipher-Block Chaining)

- Each plaintext block is XOR'd with the previous ciphertext block that was produced.
- The first plaintext block is XOR'd with a random initialization vector which is the same size as the plaintext block

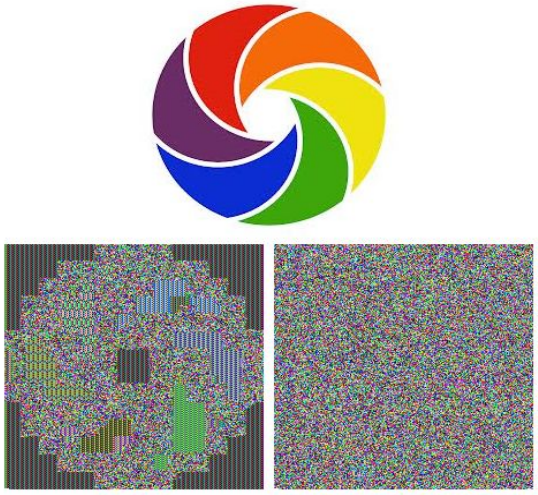


Encryption in the CBC mode



Decryption in the CBC mode

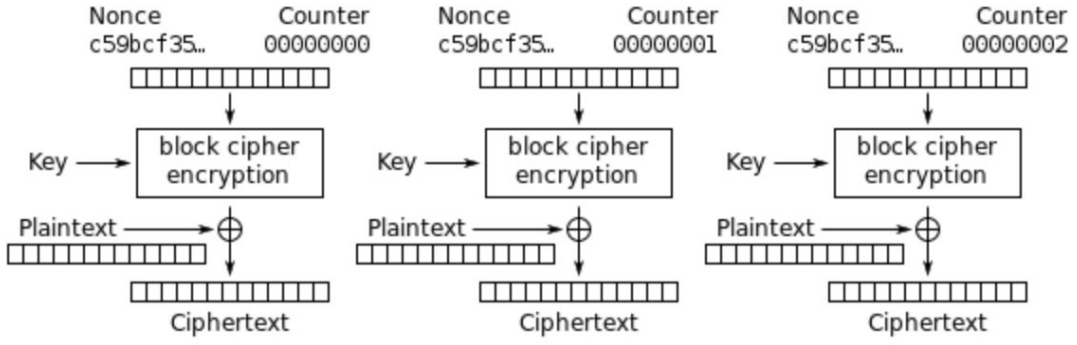
- ECB vs CBC



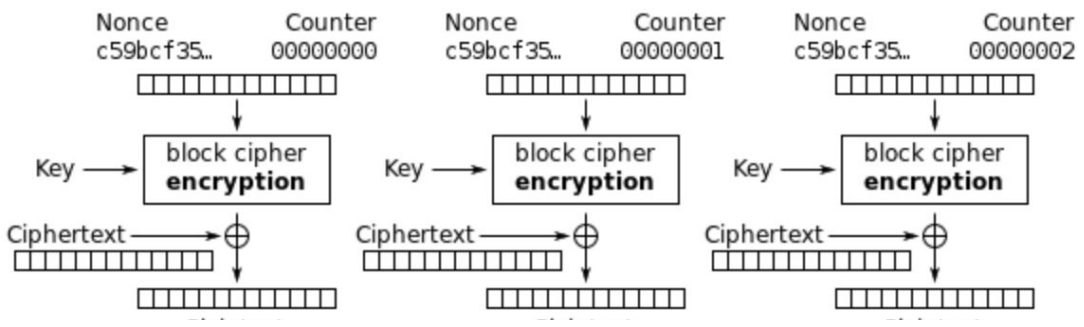
The bitmap image encrypted using DES and the same secret key. The ECB mode was used for the left image and the more complicated CBC mode was used for the right image.

CTR (Counter Mode)

- Makes a block cipher work in a similar way to a stream cipher
- A pseudo-stream cipher
- Difference between a stream cipher and CTR: Instead of XOR with a bit of pseudorandom key stream, each bit of plaintext is combined with a bit of random nonce and counter value encrypted together.
- Since counter is increased for each subsequent block, same ce encryption of each block doesn’bit of input results in different bit of output. Sint require previous output, hence is parallelizable.



Counter (CTR) mode encryption



Moore's Law 1965

- He noticed the number of transistors in a chip doubled every x (rewatch lec to find out what Richard said) years
    - Later also computing power; roughly doubles every 18 months
    - Hence you lose one bit of security every 18 months

Disk Encryption

- How Richard says encryption keys are stored:
    - Generate a random key to encrypt the disk
    - Encrypt the key (that can be decrypted with a password) and store it somewhere NOT ON THE DISK
- What Windows OS use to do:
    - Ask the disk if it can encrypt itself
    - Let the disk encrypt itself
- Forensics
    - RAM can still store data even after shutting off power (for a short period of time).
    - If RAM is frozen, data leaks even slower, and thus can be retrieved.
    - Police are then able to analyse it for say, the encryption key so that they can get to the hard disk.
    - Some links *(please don't delete me)*:
        - https://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1162&context=adf
        - https://en.wikipedia.org/wiki/Cold_boot_attack

Authentication

- What is Authentication? Suppose that we are in an isolated environment, where our only job was to provide "authentication" given data from the outside world, and nothing else. How do we authenticate?
- Factors of Authentication (Multifactor authentication)
    - Defense in depth
    - Something you know
        - Secrets (e.g. password)
        - Flaw: easy to prove that you know a secret, hard to prove that no one else knows the secret
    - Something you have
        - Straight-forward: do you have something that only the correct person should have
        - Commonly implemented as 2-factor authentication (i.e. with mobile phones)

        - Flaw: we know how to intercept messages now (more of an implementation flaw)
    - Something you are
        - Something that defines you
        - Mostly implemented as biometrics
        - Flaw: easy/viable to fake e.g. biometrics (again, more of an implementation flaw)
        - Flaw: Once its leaked/stolen, it cannot be changed
    - According to Richard: These are all just different forms of “Something you know”
- Hardware bypass of authentication (again, implementation flaw). Because authentication is done with computers/electronics; theoretically, you just have to send the correct signals along the correct wire(s) and thus bypass authentication.
- These are all challenges related to authenticating identity. If we’re just authenticating information, the problem becomes much simpler: we simply just combine information with encryption.

Some stuff mentioned (in passing)

- Feistel cipher
- Weaknesses arise from the implementation of the algorithm more than from the algorithm itself
- The express envelopes story
  Using the post office to do what you want, while it is just carrying out its normal function
- Look up how current hardware chips are designed, then look at Turning (? Turing? Didn’t catch the exact word *it’s Turing*) design which has separate stacks for data and control
- ‘Smashing the Stack for Fun and Profit’
- Kahoot questions this week:
    - Bits of security
    - Distinguishing collision attack vs preimage attack vs 2nd preimage attack
    - which nsw law requires gov depts provide docs for public requests?
        - Gipa
- NIST story about how DES came to be

Key concepts from Cryptocurrency Presentation

- Facilitates direct transactions between individual

- Removes the need for third party intermediaries

What is a blockchain?

- A chain of blocks
- Each block contains
    - Data
        - Contains hundreds of transactions
        - In bitcoin, you're looking at about 2000 transactions
    - Hash of the block
    - Hash of the **previous** block
- Hashes make the block **tamper resistant**

How blocks are formed

1. Transactions are grouped into a **transaction pool**
2. Miners gather transaction to form a **candidate block**
3. Candidate block is given metadata known as a **block header**
4. We hash the block with a nonce - a randomly generated number
5. If the hash is lower than a certain target value the block is added to the chain
6. Miners race to find nonces

Historical flaws

- 51% Attack
- Attack on exchanges
    - Find weakness in exchanges to gain coins
    - MtGox was subject to an SQL injection
    - User databases were attacked
- Bugs in smart contracts
    - Allow developers to program their own smart contracts
    - A smart contract is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts allow the performance of credible transactions without third parties. (Blockgeeks)