Symmetric Encryption: Modes of Operation, Semantic Security

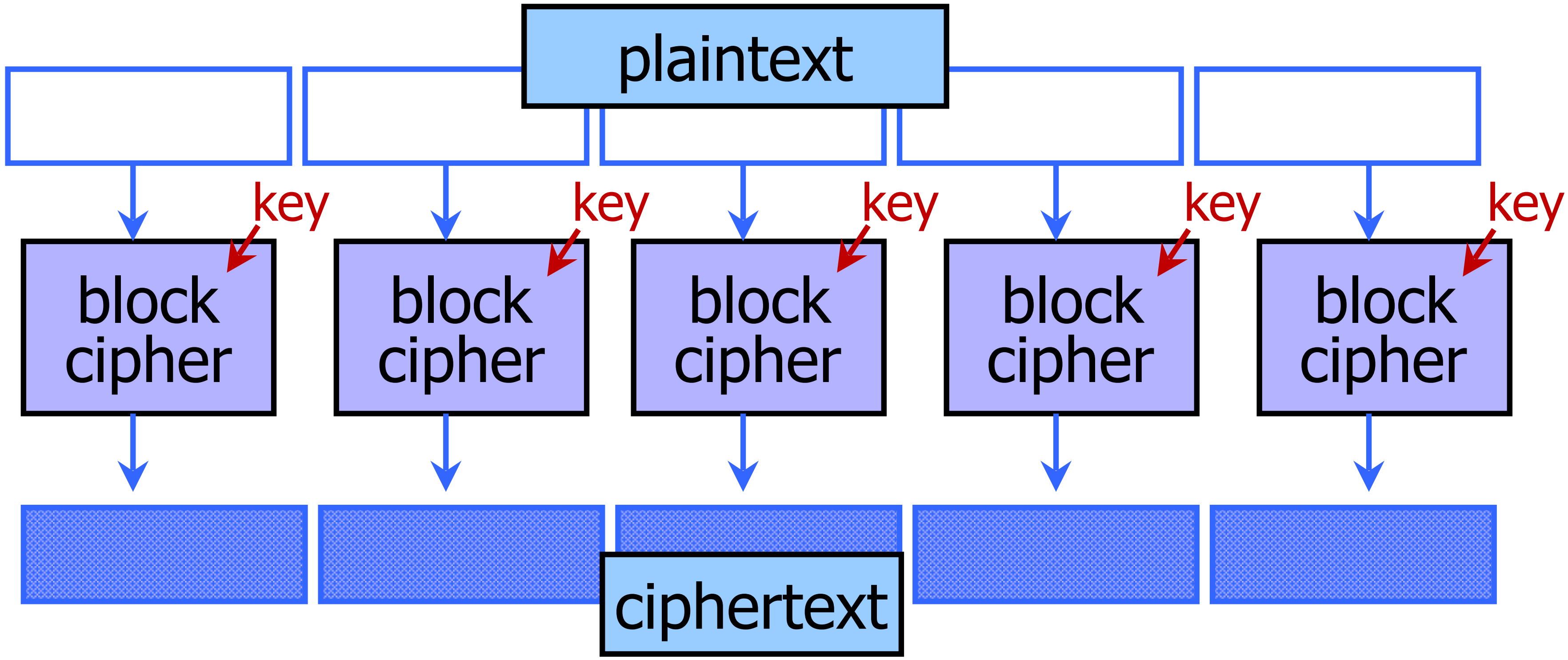
Yan Huang

Credits: Vitaly Shmatikov (Cornell Tech)

Encrypting a Large Message

- So, we've got a good block cipher, but our plaintext is larger than 128-bit block size
- Modes of Operation
  - Electronic Code Book (ECB) mode
    - Split plaintext into blocks, encrypt each one separately using the block cipher
  - Cipher Block Chaining (CBC) mode
    - Split plaintext into blocks, XOR each block with the result of encrypting previous blocks
  - Also various counter modes, feedback modes, etc.

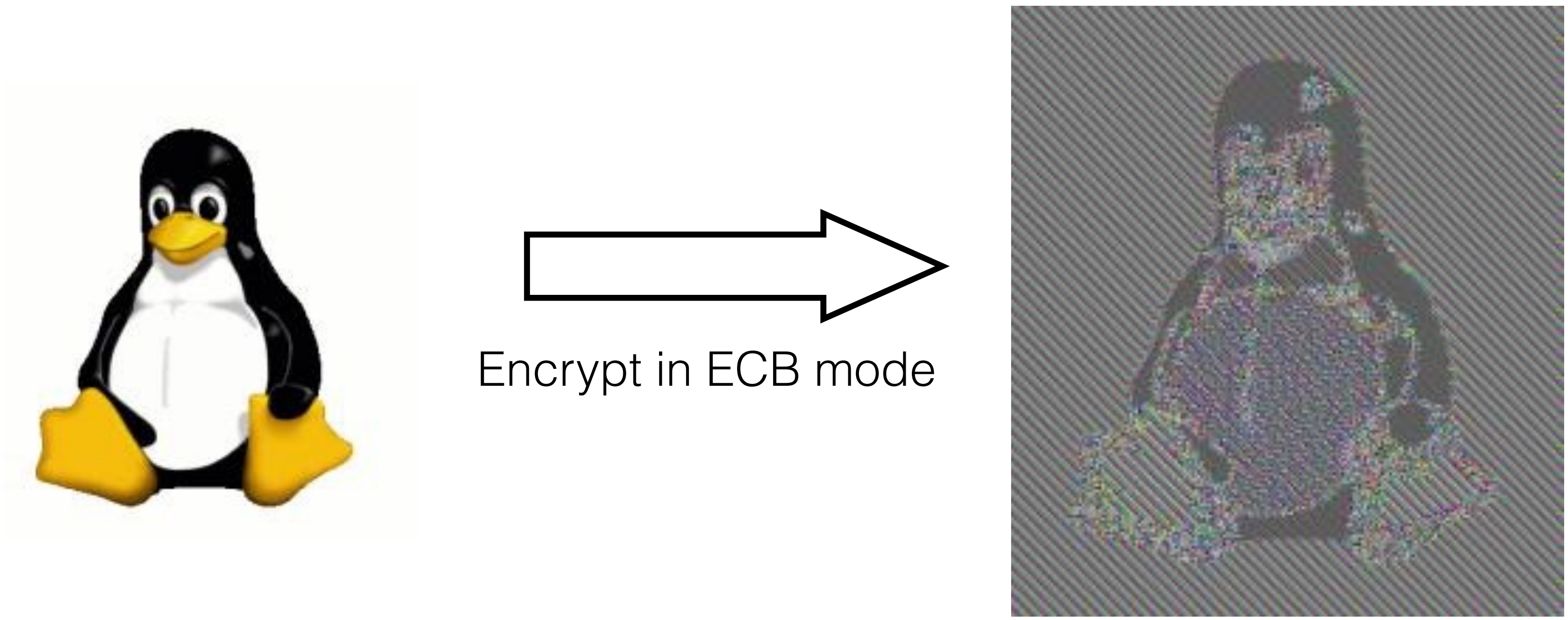
ECB Mode



- Identical blocks of plaintext produce identical blocks of ciphertext
- No integrity checks: can mix and match blocks

Information Leakage in ECB Mode

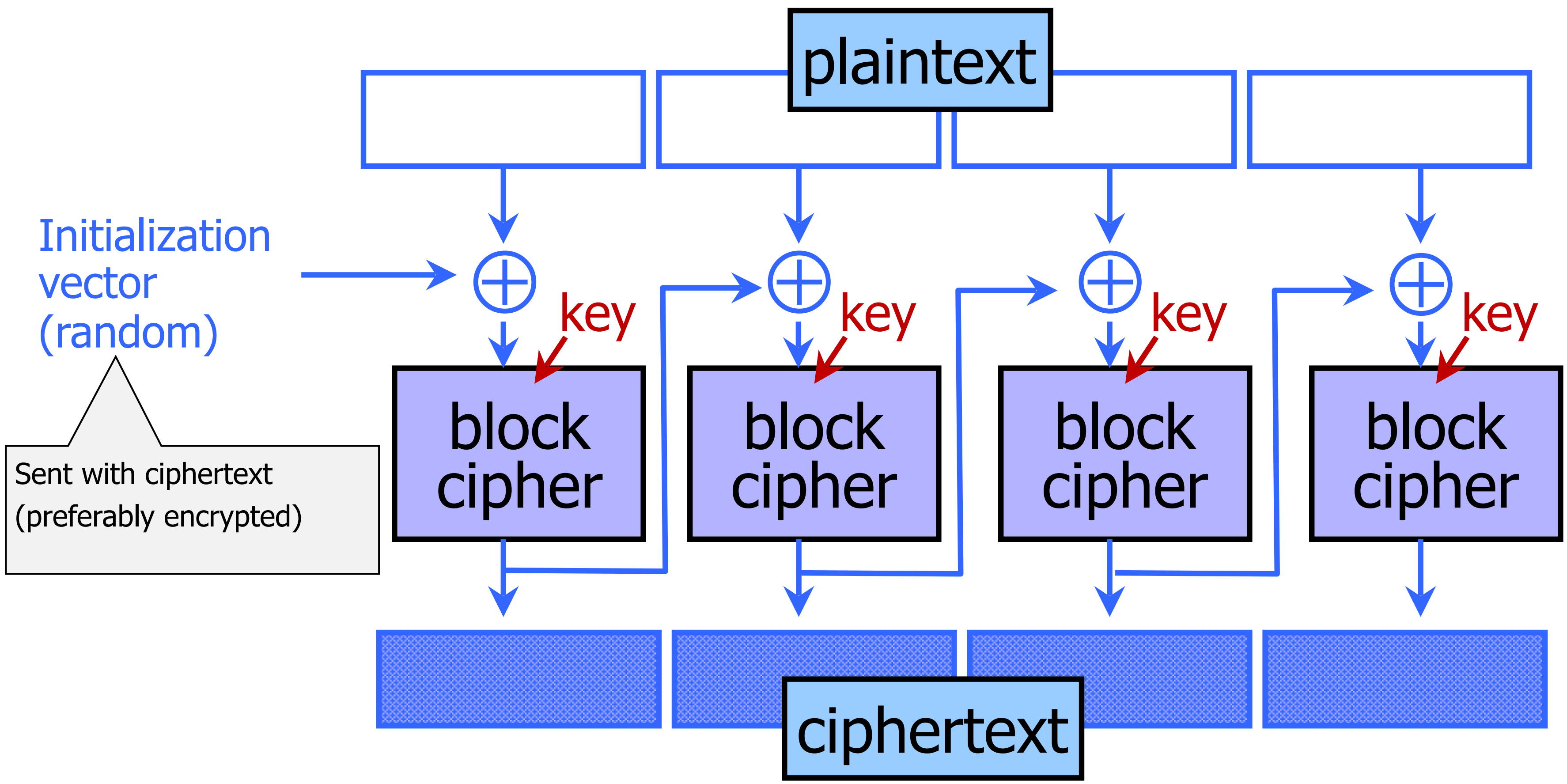
[Wikipedia]



Adobe Passwords Stolen (2013)

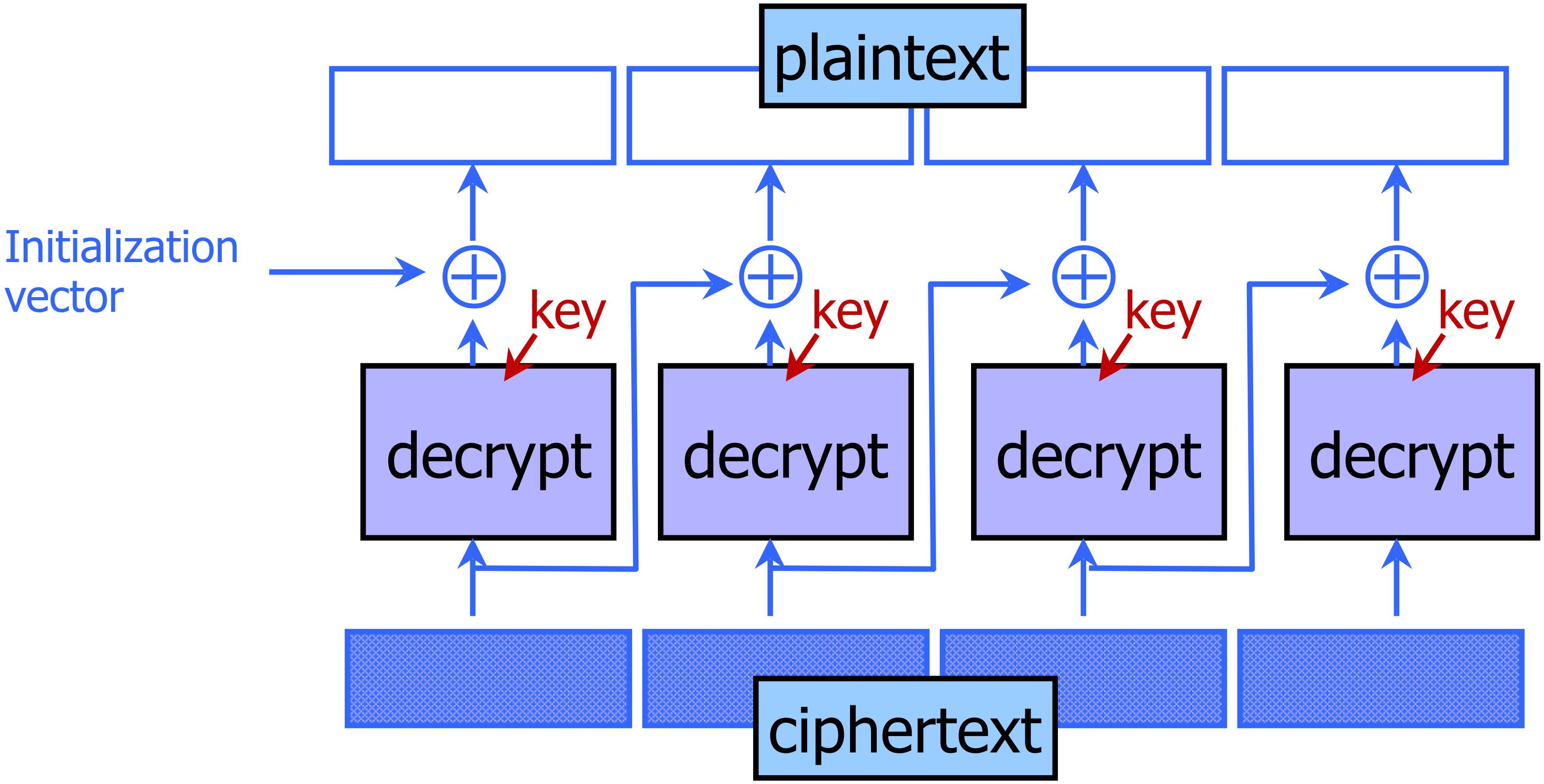
- 153 million account passwords
  - 56 million of them unique
- Encrypted using 3DES in ECB mode rather than hashed

CBC Mode: Encryption



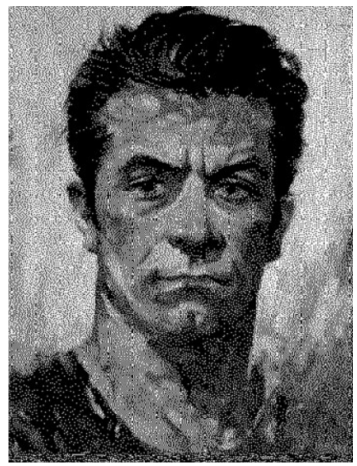
- Identical blocks of plaintext encrypted differently
- Last cipherblock depends on entire plaintext
  - Still does not guarantee integrity

CBC Mode: Decryption



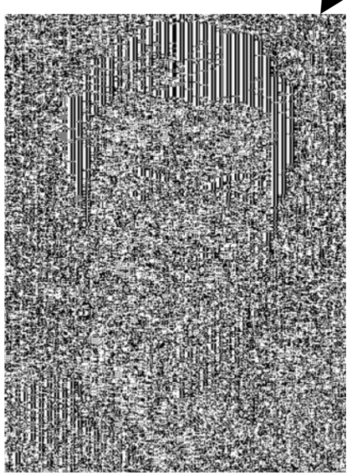
ECB vs. CBC

[Picture due to Bart Preneel]

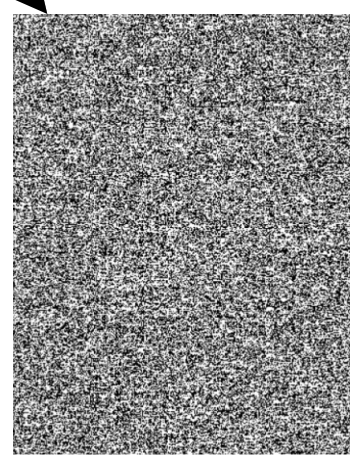


AES in ECB mode

AES in CBC mode



Similar plaintext blocks produce similar ciphertext blocks (not good!)

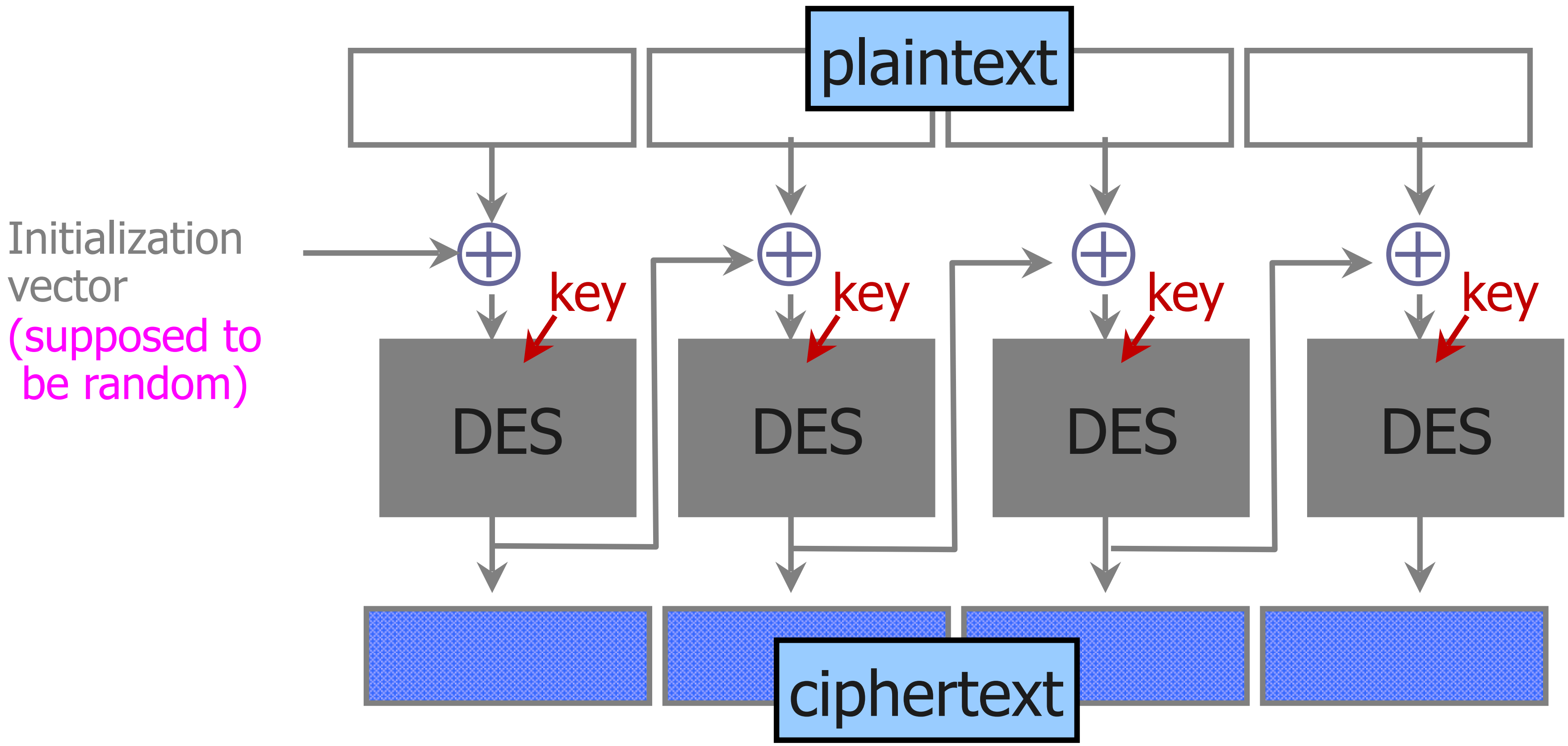


Choosing the Initialization Vector

- Key used only once
  - No IV needed (can use IV=0)
- Key used multiple times
  - Best: fresh, random IV for every message

CBC and Electronic Voting

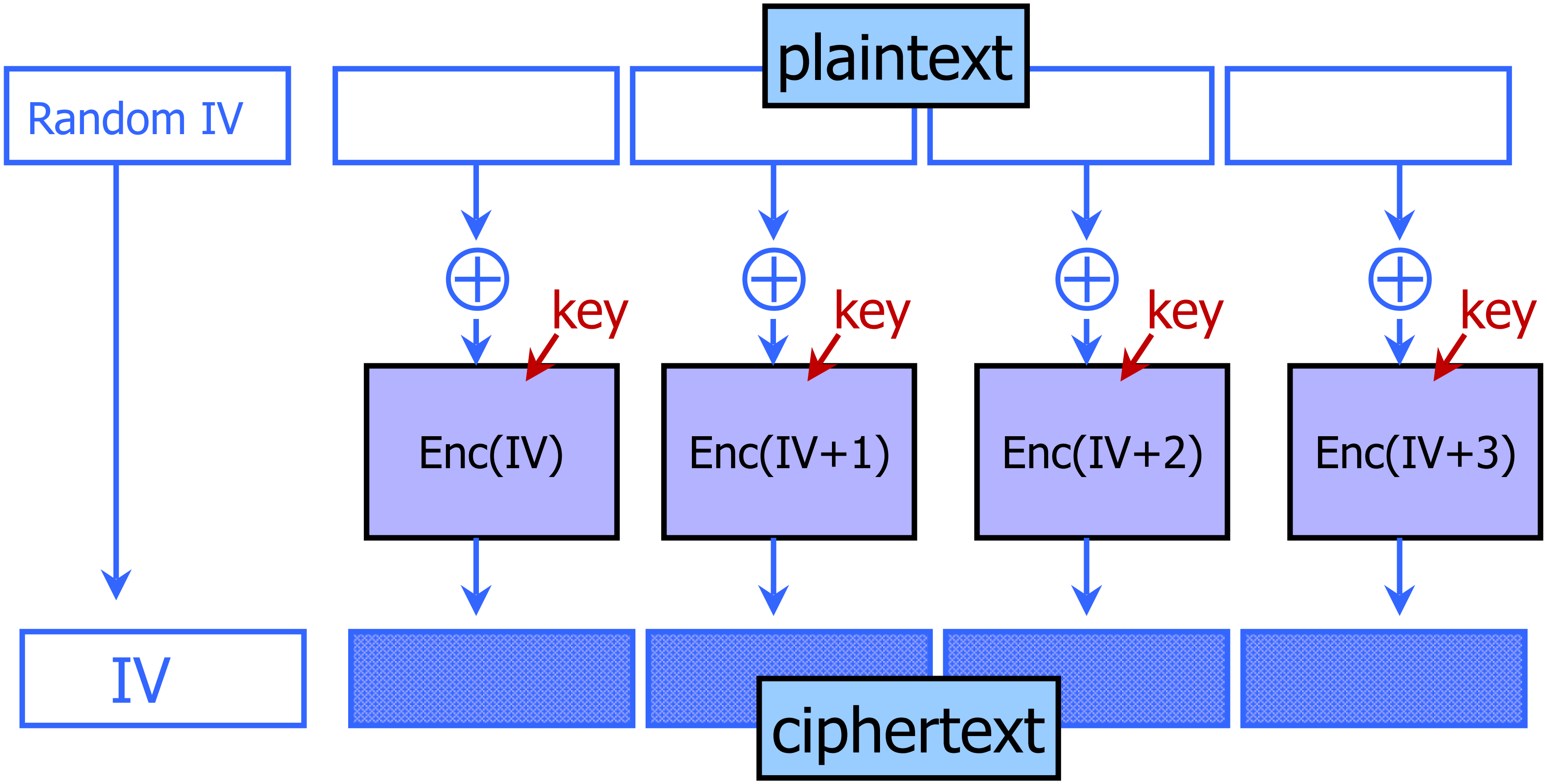
[Kohno, Stubblefield, Rubin, Wallach, IEEE S&P'04]



Found in the source code for Diebold voting machines:

```
DesCBCEncrypt((des_c_block*)tmp, (des_c_block*)record.m_Data,
              totalSize, DESKEY, NULL, DES_ENCRYPT)
```

CTR (Counter Mode)



- Still does not guarantee integrity
- Fragile if counter repeats

When Is a Cipher "Secure"?

- Hard to recover plaintext from ciphertext?
  - What if attacker learns only some bits of the plaintext? Some function of the bits? Some partial information about the plaintext?
- Fixed mapping from plaintexts to ciphertexts?
  - What if attacker sees two identical ciphertexts and infers that the corresponding plaintexts are identical?
  - What if attacker guesses the plaintext – can he verify his guess?
  - Implication: encryption must be randomized or stateful

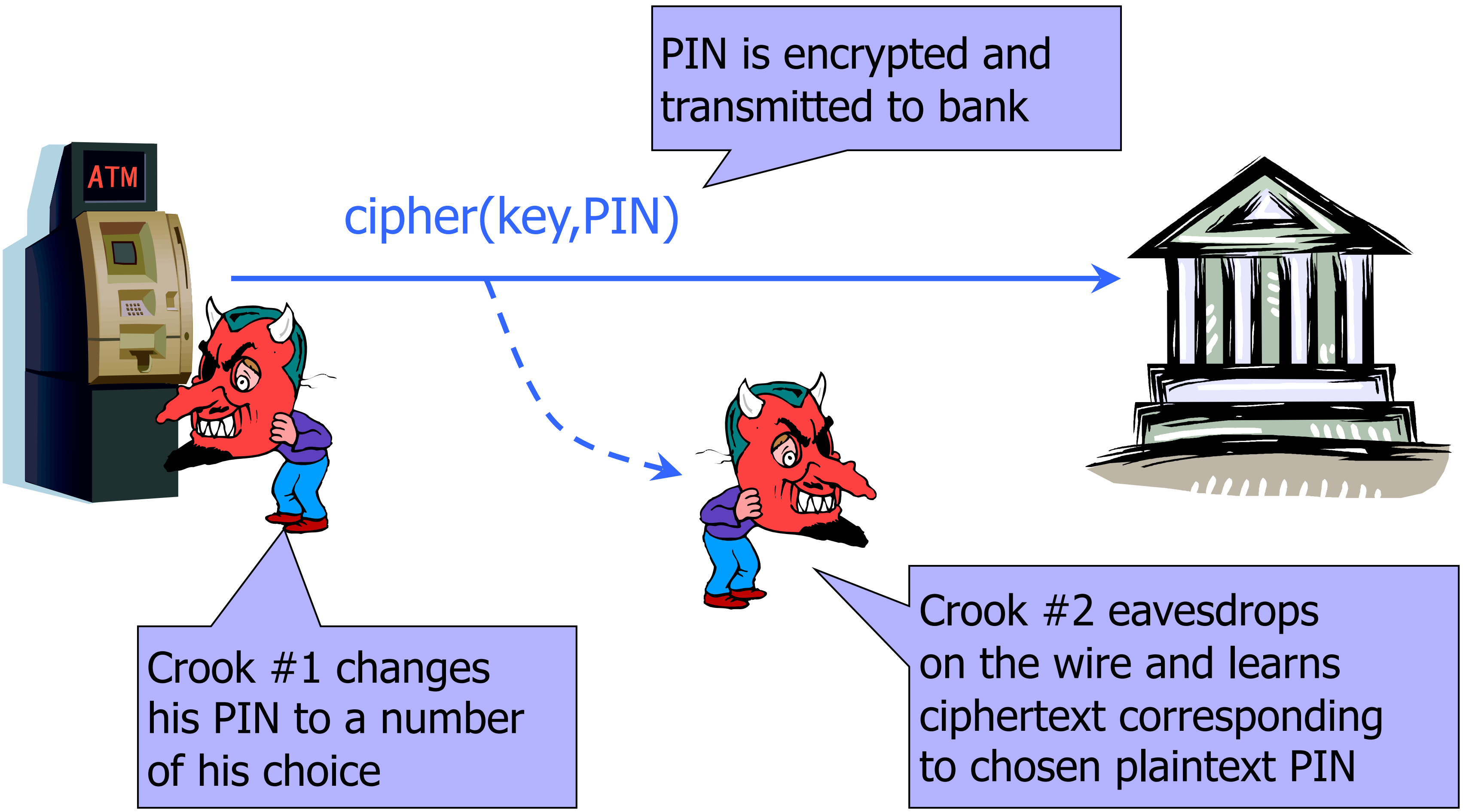
How Can a Cipher Be Attacked?

- Attackers knows ciphertext and encryption algorithm
  - What else does the attacker know? Depends on the application in which the cipher is used!

- Known-plaintext attack (stronger)
  - Knows some plaintext-ciphertext pairs
- Chosen-plaintext attack (even stronger)
  - Can obtain ciphertext for any plaintext of his choice
- Chosen-ciphertext attack (very strong)
  - Can decrypt any ciphertext <u>except</u> the target
  - Sometimes very realistic

OpenSSL
Cryptography and SSL/TLS Toolkit

Chosen-Plaintext Attack



… repeat for any PIN value

Very Informal Intuition

Minimum security requirement for a modern encryption scheme

- Security against Chosen-Plaintext attack
  - Ciphertext leaks no information about the plaintext
  - Even if the attacker correctly guesses the plaintext, he cannot verify his guess
  - Every ciphertext is unique, encrypting the same message twice produces completely different ciphertexts
- Security against chosen-ciphertext attack
  - Integrity protection – it is not possible to change the plaintext by modifying the ciphertext

How to formalize CPA-Security?

Can you recall Shanon's definition of perfect security?

Shannon's perfect secrecy

Let (E, D) be a cipher over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$

(E,D) has perfect secrecy if $\forall\ m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

$$\{ E(k,m_0) \} = \{ E(k,m_1) \} \quad \text{where} \quad k \leftarrow \mathcal{K}.$$

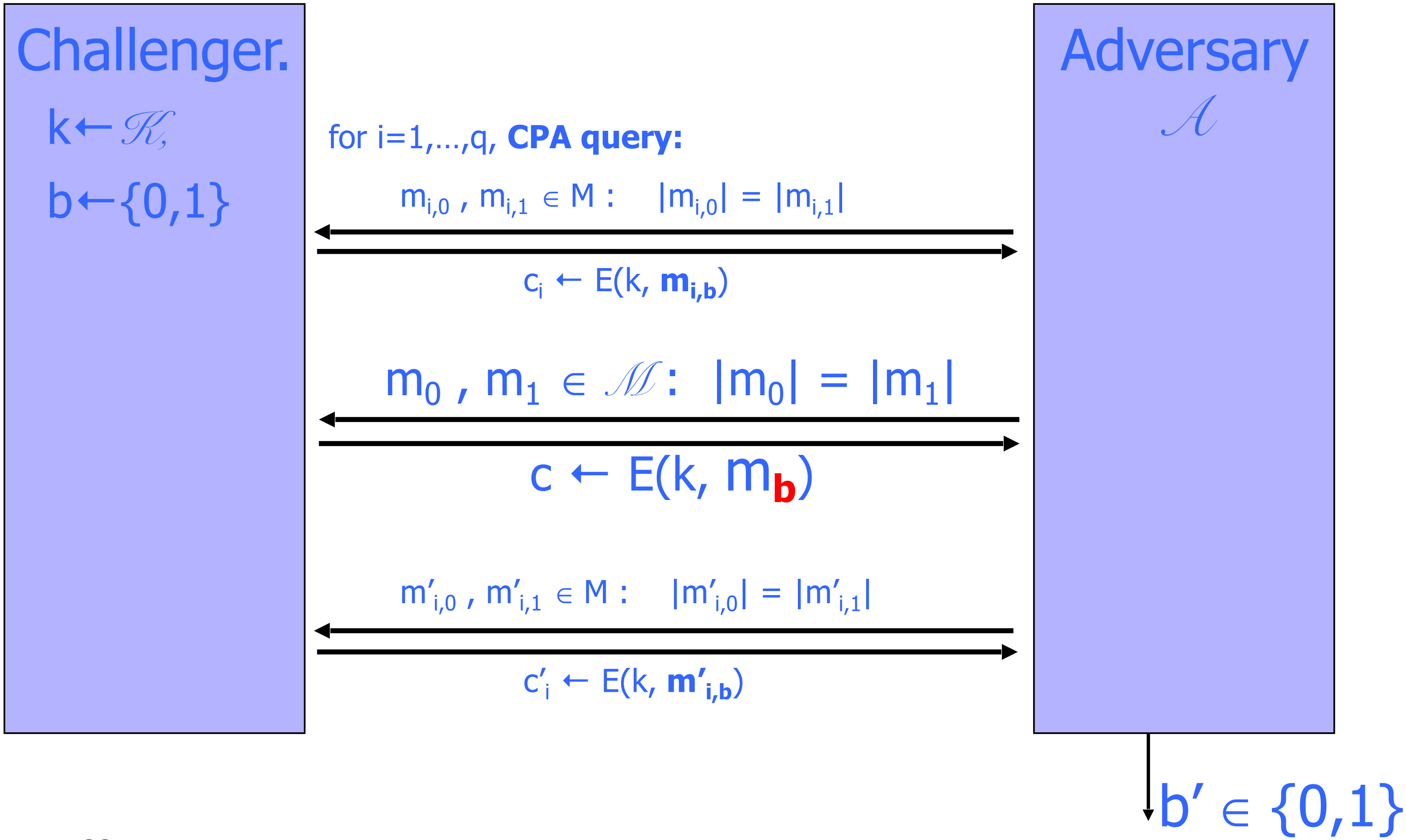
Does this help to define CPA-Security?

The Chosen-Plaintext Game

1. k←KeyGen(1^n). b←{0,1}. Give Enc(k, ·) to $\mathscr{A}$.
2. $\mathscr{A}$ chooses as many plaintexts as he wants, and receives the corresponding ciphertexts via Enc(k, ·).
3. $\mathscr{A}$ picks two plaintexts M_0 and M_1. (Picking plaintexts for which A previously learned ciphertexts is allowed!)
4. $\mathscr{A}$ receives the ciphertext of M_b, and continues to have accesses to Enc(k, ·).
5. $\mathscr{A}$ outputs b'.

$\mathscr{A}$ wins if b'=b.

CPA Secure (one-time key)

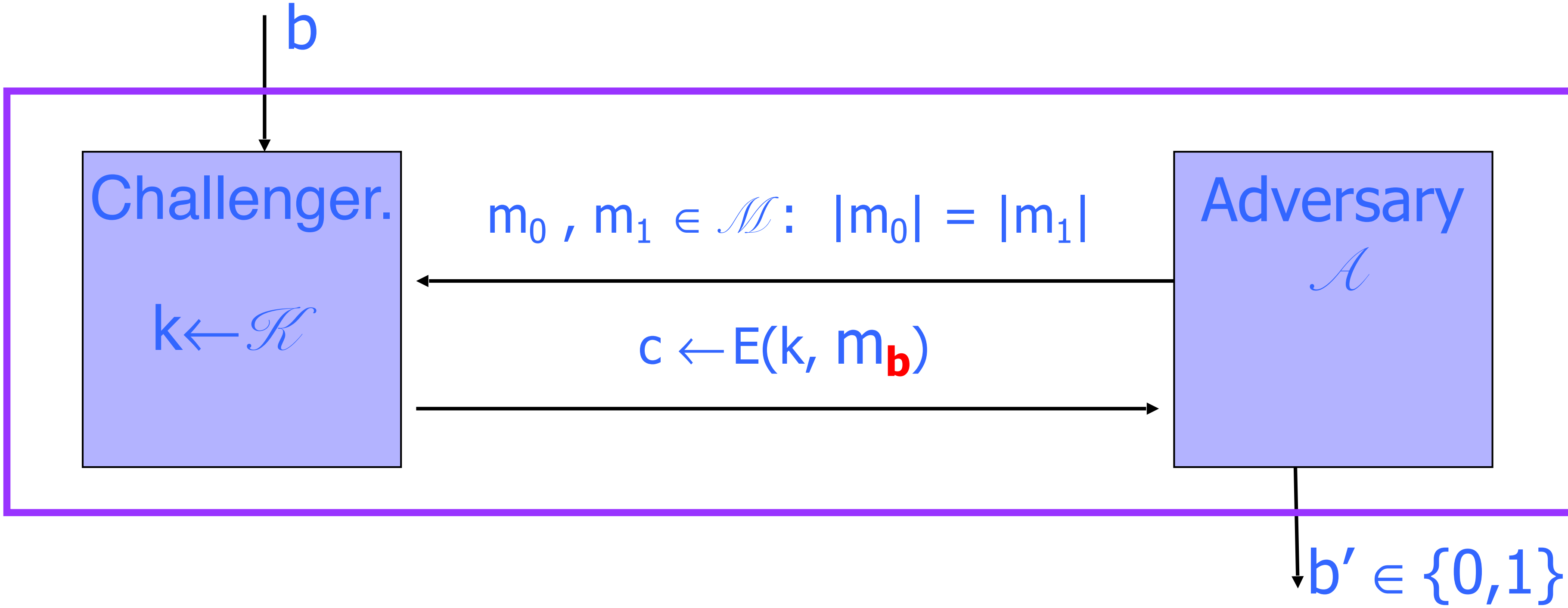


For all efficient adversary $\mathcal{A}$,

$$|\Pr[\, b=b' \,] - 1/2 \,| \text{ is “negligible”.}$$

Alternative Definition of CPA-Security (one-time key)

For $b \leftarrow \{0,1\}$, define experiment EXP(b) as:



Define W_b := [event that EXP(b)=1].

$$\mathrm{Adv}(\mathcal{A}, \mathbf{E}) := \left| \Pr[W_0] - \Pr[W_1] \right| \in [0,1]$$

Alternative Definition of CPA-Security (one-time key)

E is **computational secure** if for all efficient adversary $\mathscr{A}$

Adv ($\mathscr{A}$, E) is "negligible".

Negligible

- Concrete sense:
  e.g., $< 2^{-40}$
- Asymptotic sense:
  negl(n) < any inverse polynomial of n, as long as n is sufficiently large.

Defining Perfect Security (one-time key)

E is **perfectly secure** if for all adversary $\mathscr{A}$

$$\text{Adv}(\mathscr{A}, \text{E}) \text{ is } 0.$$

$\Leftrightarrow$ For all explicit $m_0, m_1 \in M$:

$$\{ E(k, m_0) \} = \{ E(k, m_1) \}, \text{ where } k \leftarrow \mathscr{K}.$$

A Simple Example

- Any deterministic, stateless symmetric encryption scheme is insecure
  - Attacker can easily distinguish encryptions of different plaintexts from encryptions of identical plaintexts
  - This includes ECB mode of common block ciphers!

Attacker A interacts with Enc(-)

query Enc(0)

Let x=0, y=1 be any two different plaintexts

Send x, y to the challenger

If C_1=Enc(0) then b=0 else b=1

- The advantage of this attacker A is 1