Die Geheimſchriften

und

die Dechiffrir-Kunſt.

Mit beſonderer Berückſichtigung
der deutſchen und der franzöſiſchen Sprache

von

F. W. Kaſiski,
Major z. D.

Berlin, 1863.

Druck und Verlag von E. S. Mittler und Sohn.
(Zimmerſtraße 84. 85.)

CSL 505
CRYPTOGRAPHY

Lecture 3
Attack models

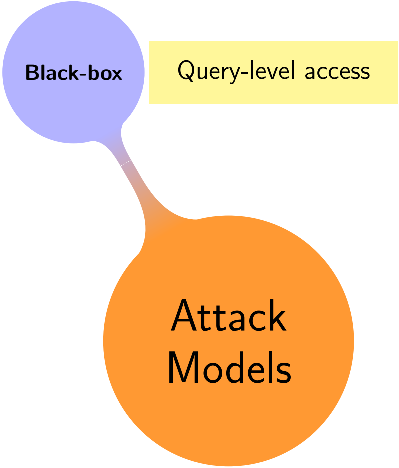
Instructor
Dr. Dhiman Saha

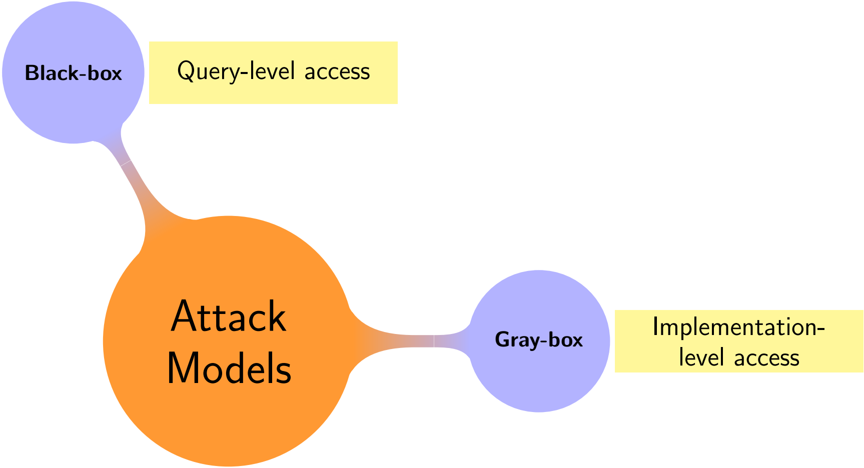
Image Source: Cryptography and the Art of Decryption by Friedrich W. Kasiski

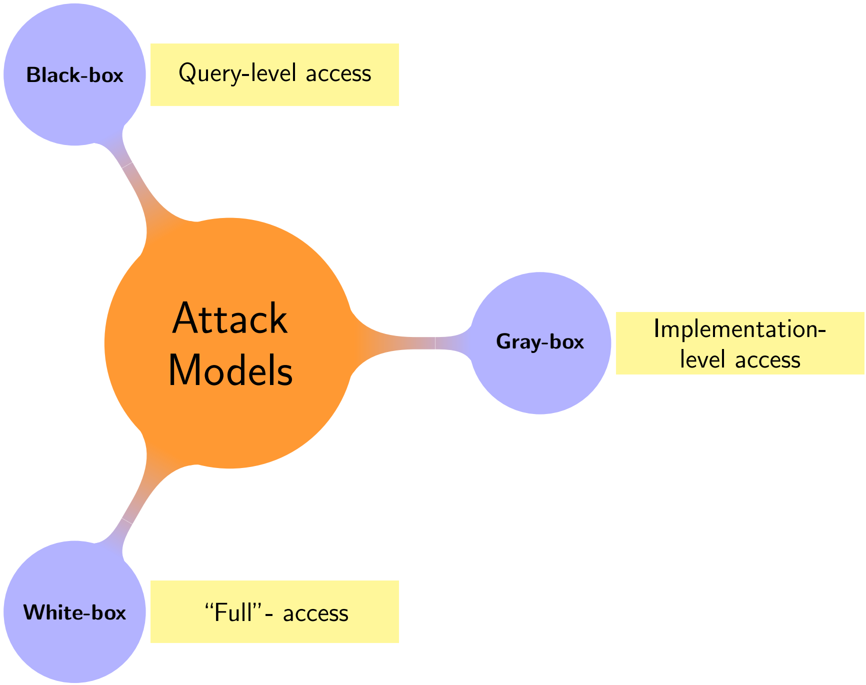
The Story So Far...

- Cryptographic goals: Privacy, Integrity, Authenticity
- Historical ciphers
- Idea of Cryptanalysis

Attack Models


Black-box
Query-level access
Attack Models


Black-box
Query-level access
Attack Models
Gray-box
Implementation-level access

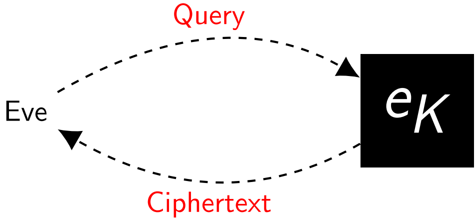

Black-box
Query-level access
Attack Models
Gray-box
Implementation-level access
White-box
"Full"- access

Black-box Models

Notion of query — In current context

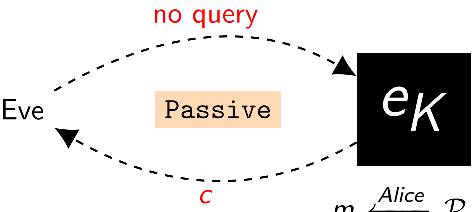
Operation that sends an input value to some function and gets the output in return, without exposing the details of that function.

- Assume cipher behaving as a black-box
- Eve only sees what goes in and out of the cipher
- Different models based on what Eve can query for



Ciphertext Only Attack (COA)

- Eve has access to the ciphertext c only
- But **does not** know the associated plaintexts
- Or how they were selected
- The default scenario
- Weakest adversarial model

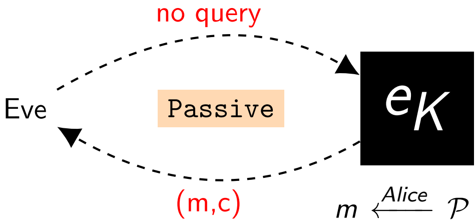


$m \xleftarrow{Alice} \mathcal{P}$

$c = e_K(m)$

Known Plaintext Attack (KPA)

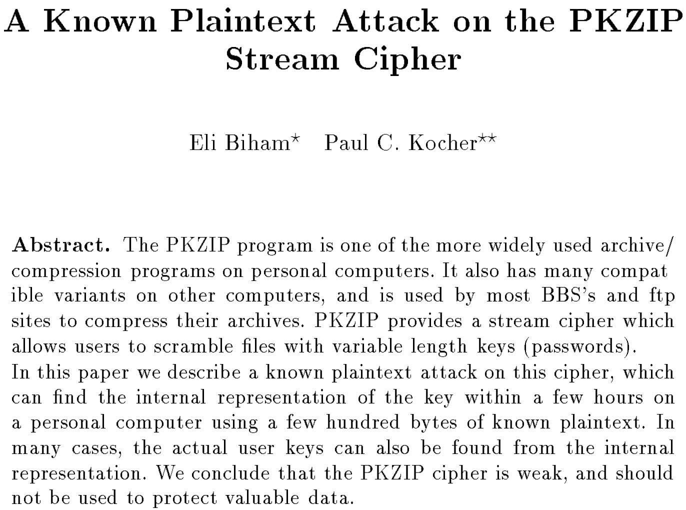
- Eve can get her hands on both a ciphertext c and its corresponding plaintext m,
- Where plaintexts are **assumed to be randomly selected.**
- Stronger than COA
- Is Hill Cipher KPA-vulnerable?



$m \xleftarrow{Alice} \mathcal{P}$

$c = e_K(m)$

KPA — Practical Example



A Known Plaintext Attack on the PKZIP Stream Cipher

Eli Biham[*] Paul C. Kocher[**]

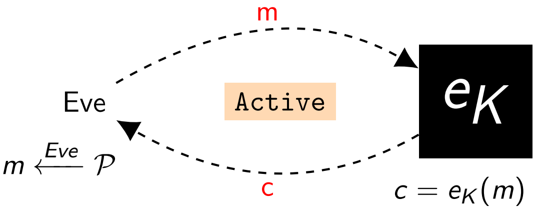
Abstract. The PKZIP program is one of the more widely used archive/compression programs on personal computers. It also has many compatible variants on other computers, and is used by most BBS's and ftp sites to compress their archives. PKZIP provides a stream cipher which allows users to scramble files with variable length keys (passwords).

In this paper we describe a known plaintext attack on this cipher, which can find the internal representation of the key within a few hours on a personal computer using a few hundred bytes of known plaintext. In many cases, the actual user keys can also be found from the internal representation. We conclude that the PKZIP cipher is weak, and should not be used to protect valuable data.

`https://www.unix-ag.uni-kl.de/~conrad/krypto/pkcrack.html`

Chosen Plaintext Attack (CPA)

- Eve can perform encryption queries for plaintexts of her choice
- And observe the resulting ciphertexts.
- Notion of **active** adversary
- Stronger than KPA
- Affine Cipher is CPA-vulnerable. Why?



m

Eve

`Active`

e_K

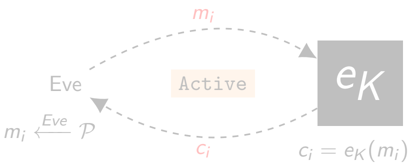
$m \xleftarrow{Eve} \mathcal{P}$

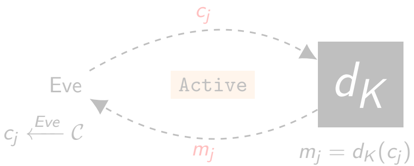
c

$c = e_K(m)$

Chosen Ciphertext Attack (CCA)

- Eve can perform encryption as well as decryption queries.
- Does this make sense?
- Decrypting something is not always enough to break a system.
- Stronger than CPA

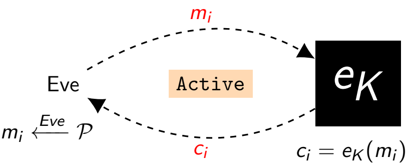


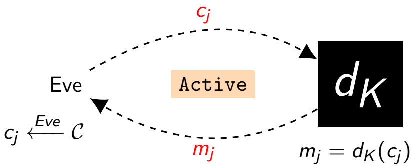


- More advanced: Adaptive CCA

Chosen Ciphertext Attack (CCA)

- Eve can perform encryption as well as decryption queries.
- Does this make sense?
- Decrypting something is not always enough to break a system.
- Stronger than CPA





- More advanced: Adaptive CCA

Formally

KPA, CPA, CCA

KPA-Security

Given one or more pairs of plaintexts and their corresponding ciphertexts, $(m_1, c_1), (m_2, c_2), \cdots, (m_n, c_n)$ it must be very difficult to decrypt any ciphertext c that is **not in the given list** without knowing k.

CPA-Security

For any list of plaintexts $m_1, m_2, \cdots, m_n \in \mathcal{P}$ chosen by the adversary, even with knowledge of the corresponding ciphertexts $e_K(m_1), e_K(m_2), \cdots, e_K(m_n)$ it is very difficult to decrypt any ciphertext c that is **not in the given list** without knowing k.

CCA-Security

HW-Problem

Formally

KPA, CPA, CCA

KPA-Security

Given one or more pairs of plaintexts and their corresponding ciphertexts, $(m_1, c_1), (m_2, c_2), \cdots, (m_n, c_n)$ it must be very difficult to decrypt any ciphertext c that is **not in the given list** without knowing k.

CPA-Security

For any list of plaintexts $m_1, m_2, \cdots, m_n \in \mathcal{P}$ chosen by the adversary, even with knowledge of the corresponding ciphertexts $e_K(m_1), e_K(m_2), \cdots, e_K(m_n)$ it is very difficult to decrypt any ciphertext c that is **not in the given list** without knowing k.

CCA-Security

HW-Problem

Formally

KPA-Security

Given one or more pairs of plaintexts and their corresponding ciphertexts, $(m_1, c_1), (m_2, c_2), \cdots, (m_n, c_n)$ it must be very difficult to decrypt any ciphertext c that is **not in the given list** without knowing k.

CPA-Security

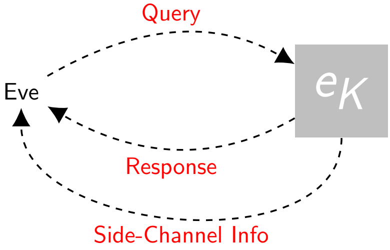
For any list of plaintexts $m_1, m_2, \cdots, m_n \in \mathcal{P}$ chosen by the adversary, even with knowledge of the corresponding ciphertexts $e_K(m_1), e_K(m_2), \cdots, e_K(m_n)$ it is very difficult to decrypt any ciphertext c that is **not in the given list** without knowing k.

CCA-Security

HW-Problem

Gray-box Models

- Eve has access to a cipher's **implementation**.
- Captures practical scenarios
- Models physical access for applications such as smart cards, embedded systems
- **Notion of Side-channel attacks**[1]
- Software - timing, errors, return values
- Hardware - power, EM, faults
- Non-invasive/Invasive



[1] Will be explored later in the course

White-Box Models

- Worst-case attack model
- Eve has **full control** over the cryptographic program and its **execution environment**
- Consider ability to decrypt ciphertexts under a certain key **without sharing the key itself** - Applications like DRM
- How is that even possible?

How about embedding the key in the code itself?

Are you serious!!!

- What if this is reverse engineered?

White-Box Models

- Worst-case attack model
- Eve has **full control** over the cryptographic program and its **execution environment**
- Consider ability to decrypt ciphertexts under a certain key **without sharing the key itself** - Applications like DRM
- How is that even possible?

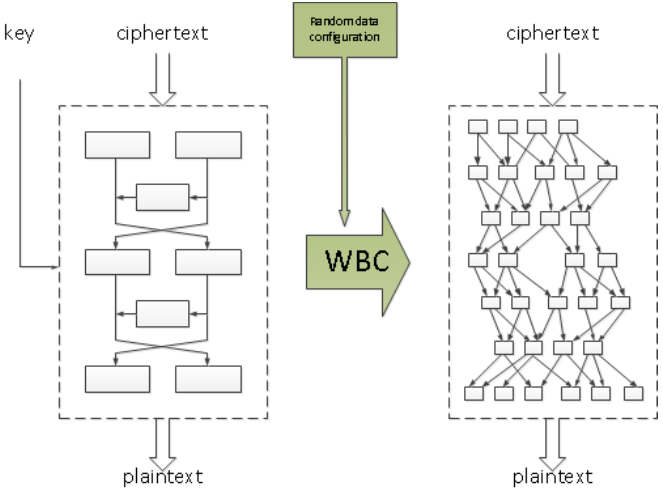
How about embedding the key in the code itself?

Are you serious!!!

- What if this is reverse engineered?

Idea

White-Box Crypto



- Idea: obfuscate the code to prohibit reverse engineering

Image Source: http://www.whiteboxcrypto.com/