Honey Encryption

Group 5 : 超級乳牛牧場

組員:
109550135:范恩宇
109550147:許竣凱
109550066:張瑋倫
110550135:黃孚翔

Outline

01 Motivation

02 Detail about Honey Encryption

03 Self-Implementation

04 Possible extensions

05 Conclusion

06 Reference

Motivation

- Good for defending brute-force attacks
  -> Often come into use

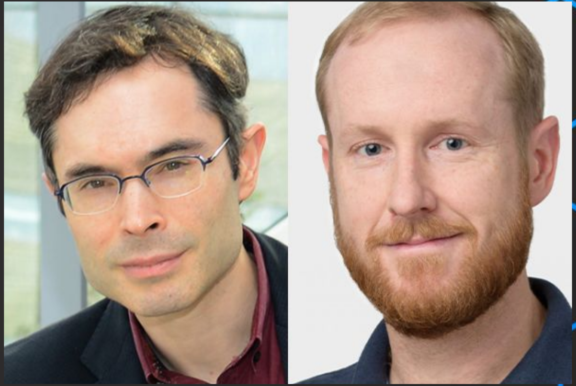
- Relatively new technique
  -> Much room for research and development

- Much related to data privacy & cyber security
  -> Quite important for this era

Detail about Honey Encryption

- Cryptographic technique that aims to address a specific security concern in encryption systems known as "ciphertext indistinguishability"

- First introduced in a 2014 research paper by Ari Juels & Thomas Ristenpart



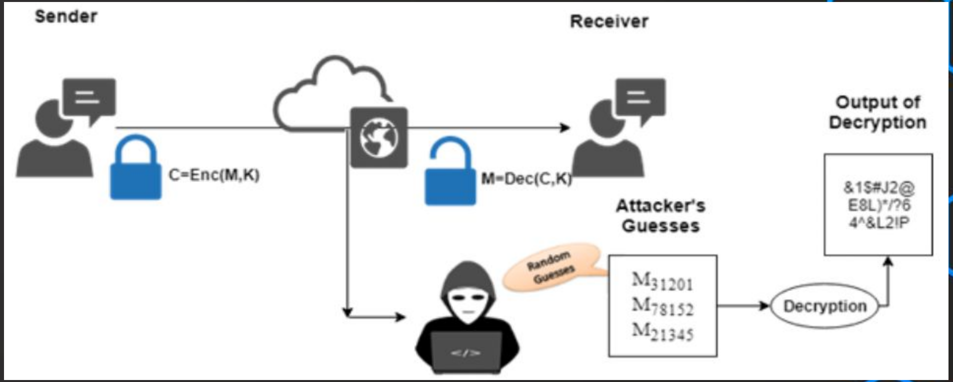
- Encryption technique that provides additional protection to ciphertext

Detail about Honey Encryption

- Two main steps :

1. Distribution Transforming Encoder (DTE)

   Map the plaintext required encrypted from the Message space (M) to Seed space (S) .



2. Password-Based Encryption (PBE)

   The seed is encrypted using this to obtain the ciphertext.

Detail about Honey Encryption

- Generates authentic-looking decoy values instead of producing errors or random output when an incorrect decryption key is used.

- Honey encryption protects against brute-force attacks by generating realistic but incorrect decoy values for incorrect decryption attempts.

- It provides additional security in breach scenarios by adding an extra layer of protection with plausible decoy values.

Detail about Honey Encryption

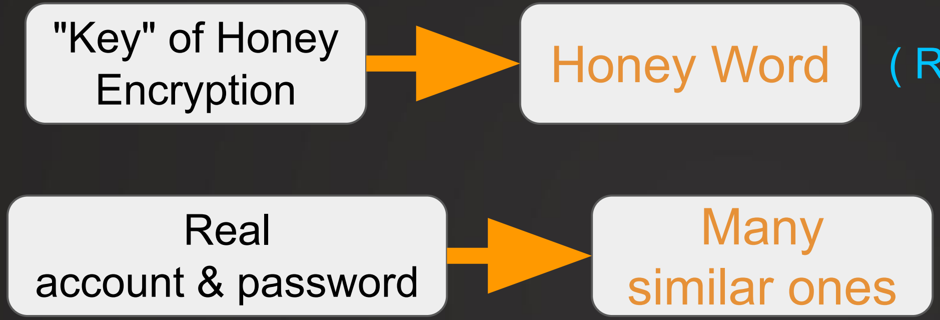
- It complements traditional encryption methods and can be integrated into existing encryption algorithms and protocols.
- The effectiveness of honey encryption depends on the careful design of decoy values and cryptographic algorithms used.

Detail about Honey Encryption

Some advantages of honey encryption :

- Plausible Deniability
- Active Defense
- Resists Statistical Attacks
- Usability and User Experience

Self - Implementation - Main Idea

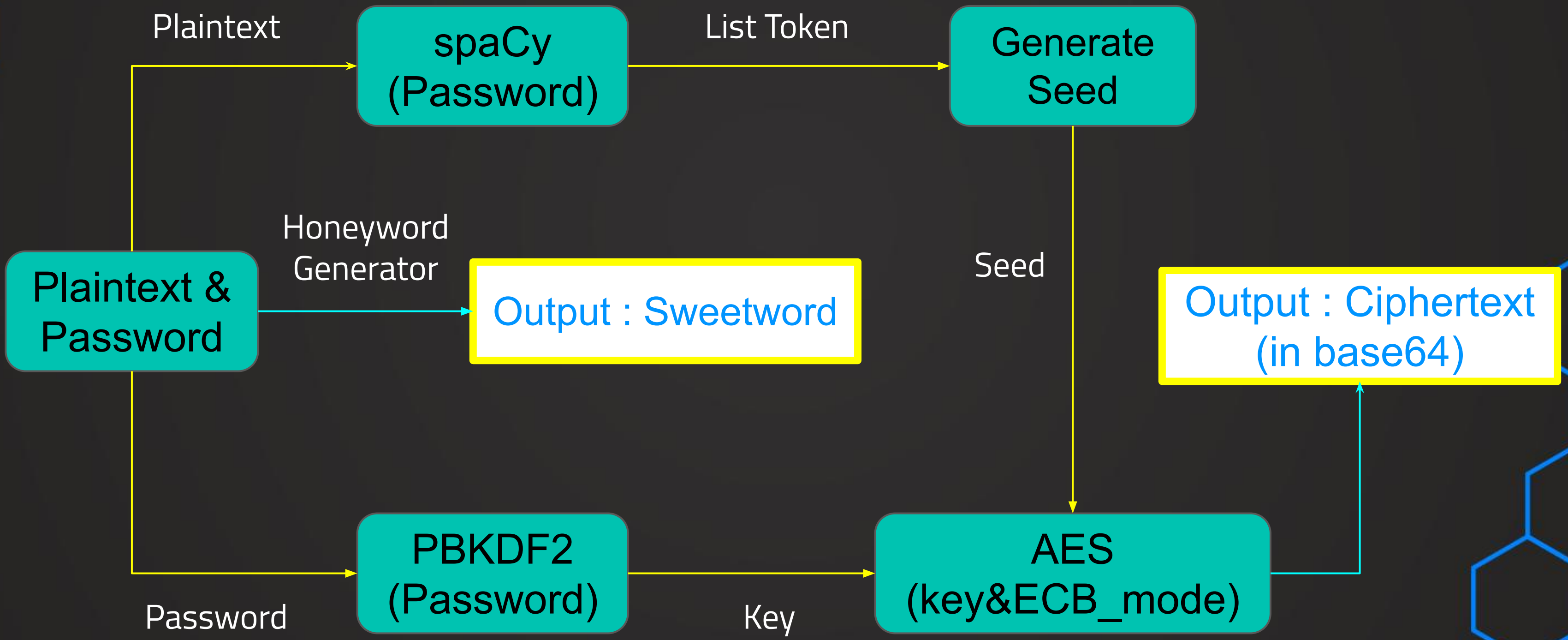


- "Key" of Honey Encryption → Honey Word (Replaced !)
- Real account & password → Many similar ones (Created by Honey Generator)

→ Attackers can’t find the real one easily

(since the resulting plaintexts all seem to be real)

Self - Implementation - Process



Self - Implementation - 1

- Generate the seed

1. Selected spaCy package from open-source NLP toolkit to use . After obtaining several attributes of our desired sentence , we then generate the seed.

2. The approach involved :
   Noting index of the chosen word in the dictionary collected from an open-source dataset , along with the sentence pattern.

3. These numbers were then concatenated, and XORed with a randomNumber to generate the seed.

Self - Implementation - 2

- Generate the key

  To prevent dictionary attacks , we also salted the password to generate the key .

Self - Implementation - 2

- Salt

A specific string inserted at a random position in the content (ex : password) before hashing it . For ensuring the resulting hashed value be different from the original unsalted one

→ Enhancing security in various applications.

However, if hashed result needs to be verified in future (ex : when authenticating a user's password input) , the salt used during hashing must be recorded.

Self - Implementation - 3

- Encryption

Using AES encryption algorithm → Easy to use and widely adopted. Then , encode encrypted ciphertext using base64 before output.

- Decryption

If key is correct/incorrect , get the true/fake plaintext. The fake one looks similar to the true one.

Self - Implementation - Problem

- Semantic Problem :

➡ Plaintext length too long / having contextual relevance & using only Honey Encryption :

➡ Generated plaintext less fluent , likely to be detected as a decoy."

- Typo-safety Problem

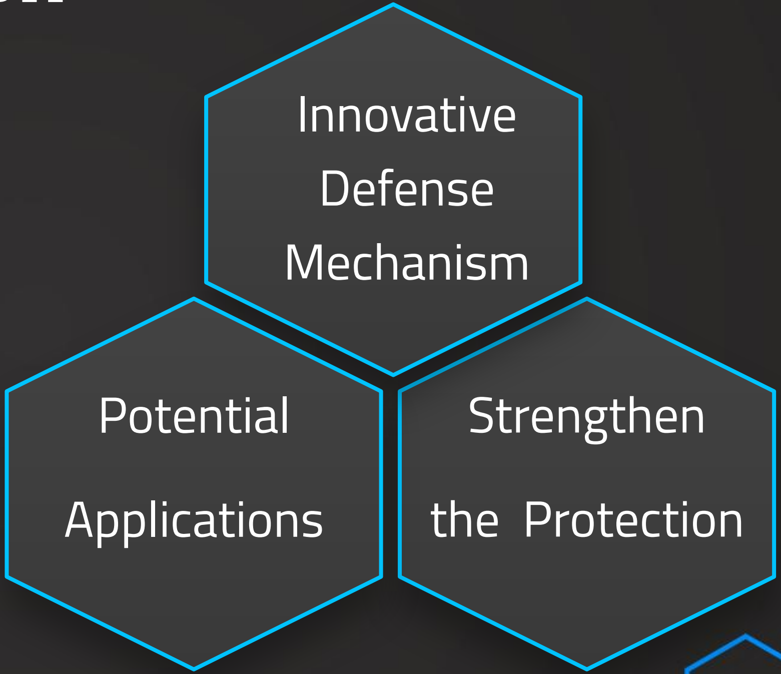
➡ Small error/typo in input can result in quite different decryption

➡ Poses challenge in ensuring accuracy and reliability of the decrypted information when using HE.

Possible extensions

- Adaptive honey encryption
- Application to different types of data

Conclusion



Innovative Defense Mechanism

Potential Applications

Strengthen the Protection

Reference

[1] Burgess, J. (2017). Honey Encryption Review.

[2] Honey Encryption Security Techniques: A Review Paper . Ammar Abdul Majed Gharbi ,Ahmed Sami Nori

[3] https://medium.com/smucs/honey-encryption-e56737af081c

[4] HONEY ENCRYPTION ALGORITHM TO SAFEGUARD AGAINST BRUTE FORCE ATTACKS . Prithvik H C, Naman Jain, Rakesh K R

[5] Natural Language Processing by Enhanced Honey Encryption Technique , Mrinal Paliwal International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-12S, October 2019

[6] Differential Privacy and Natural Language Processing to Generate Contextually Similar Decoy Messages in Honey Encryption Scheme , Kunjal Panchal ,University of Massachusetts, Amherst ,kpanchal@umass.edu ,November 2, 2020

THE END