Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа 4

«Атака на алгоритм шифрования RSA методом повторного шифрования»

Вариант № 8

Группа: P34102

Выполнил: Лапин А.А.

Проверил:
Рыбаков С.Д.

Санкт-Петербург
2024г.

Оглавление

Введение

Цель работы: изучить атаку на алгоритм шифрования RSA посредством повторного шифрования.

Текст задания

Вариант	Модуль, N	Экспонента, e	Блок зашифрованного текста, C
8	290716329017	497729	1135414239 169213008965 175441050863 109545918774 123669279758 149542889269 43068653151 32806195453 285151390718 137668394392 140567677417 176736386447 218957656245

Ход работы

Будем строить последовательность: $c_1 = c,\ c_i = c_{i-1}^e \mod N, i > 1$.

$$
\begin{array}{l|l}
N = 290716329017 & c_1 = c^e \mod N = 1135414239^{497729} \mod 290716329017 = 50864408514 \\
e = 497729 & c_2 = c_1^e \mod N = 50864408514^{497729} \mod 290716329017 = 219212785551 \\
c = 1135414239... & \Rightarrow \\
\vdots & \vdots \\
 & c_i = c_{i-1}^e \mod N
\end{array}
$$

Программная реализация

Listing 1: main.py

```python
import math
from omegaconf import DictConfig
import hydra
from tqdm import tqdm
import logging
logger = logging.getLogger(__name__)

# RSA cryptanalysis using repeated encryption
def int_to_bytes(m):
    """
    Convert an integer to bytes.
    """
    hex_str = hex(m)[2:]
    if len(hex_str) % 2:
        hex_str = '0' + hex_str
    return bytes.fromhex(hex_str)

def repeated_encryption_attack(y, e, N):
    """
    Perform RSA cryptanalysis using repeated encryption attack.
    Constructs the sequence:
        y1 = y
        yi = y_{i-1}^{e} mod N for i > 1
    Continues until y_i = y, then returns y_{i-1} as the plaintext x.
    """
    y_current = y
    i = 1
    y_sequence = [y_current]

    while True:
        y_next = pow(y_current, e, N)
        if y_next == y:
            logger.debug(f"Cycle detected at iteration {i+1}.")
            break
        y_sequence.append(y_next)
        y_current = y_next
```

```
        i += 1

    if i == 0:
        logger.debug("No repetition detected.")
        return None

    plaintext = y_sequence[-1]
    logger.debug(f"Plaintext x found: {plaintext}")
    return plaintext


@hydra.main(version_base=None, config_path=".", config_name="config")
def main(cfg: DictConfig):
    N = cfg.N
    e = cfg.e
    ciphertexts = cfg.c

    print(f"N = {N}")
    print(f"e = {e}")
    print(f"Ciphertexts = {ciphertexts}")

    # Decrypt each ciphertext block using repeated encryption attack
    print("Performing repeated encryption attack on ciphertext blocks...")
    decrypted_bytes = b''
    for idx, c in tqdm(enumerate(ciphertexts, start=1), total=len(ciphertexts), desc
        ="Decrypting Ciphertext Blocks"):
        logger.debug(f"Decrypting ciphertext block {idx}: {c}")
        plaintext_int = repeated_encryption_attack(c, e, N)
        if plaintext_int is None:
            print(f"Failed to decrypt ciphertext block {idx}.")
            continue
        decrypted_bytes += int_to_bytes(plaintext_int)

    try:
        plaintext = decrypted_bytes.decode('cp1251')
        print(f"Plaintext: {plaintext}")
    except UnicodeDecodeError:
        print("Decrypted bytes could not be decoded to cp1251. Raw bytes:")
        print(decrypted_bytes)

if __name__ == "__main__":
    main()
```

Listing 2: config.yaml

```
# RSA Configuration Parameters

# The modulus N, which is the product of two primes p and q.
N: 290716329017

# The public exponent e.
e: 497729

# The list of ciphertext blocks to be decrypted.
c:
  - 1135414239
  - 169213008965
  - 175441050863
```

```
  - 109545918774
  - 123669279758
  - 149542889269
  - 43068653151
  - 32806195453
  - 285151390718
  - 137668394392
  - 140567677417
  - 176736386447
  - 218957656245
```

Результаты работы программы

Listing 3: Вывод в консоль

```
> python main.py
N = 290716329017
e = 497729
Ciphertexts = [1135414239, 169213008965, 175441050863, 109545918774, 123669279758,
    149542889269, 43068653151, 32806195453, 285151390718, 137668394392,
    140567677417, 176736386447, 218957656245]
Performing repeated encryption attack on ciphertext blocks...
Decrypting Ciphertext Blocks: ████████████████████████100%|| 13/13 [00:01<00:00,
    9.29it/s]
Plaintext: тестер. Он позволяет измерить уровень шумов. В про__
```

Заключение

В ходе выполнения лабораторной работы была реализована атака на алгоритм шифрования RSA методом повторного шифрования.