КОМП'ЮТЕРНИЙ ПРАКТИКУМ №3
З КУРСУ
МЕТОДИ КРИПТОАНАЛІЗУ 1

Криптоаналіз асиметричних криптосистем на прикладі атак на криптосистему RSA

1 Мета роботи

Ознайомлення з підходами побудови атак на асиметричні криптосистеми на прикладі атак на криптосистему RSA, а саме атаки на основі китайської теореми про лишки, що є успішною при використанні однакового малого значення відкритої експоненти для багатьох користувачів, та атаки «зустріч посередині», яка можлива у випадку, якщо шифротекст є невеликим числом, що є добутком двох чисел.

2 Необхідні теоретичні відомості

Часто криптосистеми, які вважаються надійними та активно використовуються на практиці, мають вразливості при деяких значеннях параметрів, тим самим роблячи можливими ситуації, коли криптосистема є незахищеною. Така проблема вирішується накладанням обмежень на вибір відкритих та/або особистих ключів, перевіркою необхідних умов на відкритий та шифротекст тощо.

Наприклад, при використанні криптосистеми RSA з відкритим ключем (n, e) та особистим ключем (d, p, q) перевіряються умови на значення p, q задля зменшення ймовірності успіху відомих алгоритмів факторизації для числа n.

Також розглядаються певні умови на значення d та e. Оскільки від розмірів значень e та d залежить час шифрування і розшифрування відповідно, то можна назвати ряд ситуацій, в яких бажано використовувати невеликі значення d та/або e. Наприклад, при використанні криптосистеми RSA для захисту електронних платежів зі застосуванням кредитних карток природною є вимога використання невеликих значень експоненти у власника картки й великого значення експоненти у центрального комп'ютера.

Однак вибір малих параметрів d, e є небезпечним з низки міркувань. Якщо малим є секретний параметр d, то можна використати метод перебору для дешифрування повідомлення. А якщо малим є параметр e, то досить велике число повідомлень, що задовольняють нерівності $M < \sqrt[e]{n}$ та зашифровані як $C = M^e \bmod n$, можна дешифрувати шляхом обчислення кореня степеня e в полі дійсних

чисел. Інша аналогічна ситуація може скластися, коли у декількох абонентів використовується однакове значення параметру e. В такому випадку стає можливою атака на основі китайської теореми про лишки.

2.1 Атака з малою експонентою на основі китайської теореми про лишки

Для підвищення швидкості роботи алгоритму шифрування криптосистеми RSA можуть використовувати як експоненту шифрування (параметр e) деяке невелике число (зокрема число з малою вагою Геммінга). Разом з тим, у кількох користувачів можуть виявитися однакові значення параметру e, що можна використати для реалізації атаки на основі китайської теореми про лишки. Розглянемо таку атаку далі.

Нехай користувач **B** хоче надіслати однакове повідомлення M кільком різним користувачам

$$\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k,\ k \in \mathbb{N},$$

причому відкритий ключ користувача $\mathbf{A}_i$, $i = \overline{1,k}$ – це пара чисел (n_i, e), де e – деяке мале число. Таким чином, користувачі $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k$ мають однакову експоненту, яка використовується для шифрування. Тоді **B** зашифровує повідомлення M, використовуючи відповідний відкритий ключ (n_i, e) для кожного з користувачів A_i, $i = \overline{1,k}$.

Нехай зловмисник **E** успішно підключається до каналу передачі даних і перехоплює шифротексти $C_1, C_2, \ldots, C_k$, де

$$C_i = M^e \bmod n_i,\ i = \overline{1,k}.$$

Будемо вважати, що усі значення n_i для $i = \overline{1,k}$ попарно взаємно прості. В іншому випадку можна обчислити найбільший спільний дільник і факторизувати, як мінімум, два модуля n_i, n_j, $i \neq j$, обчислюючи $\gcd(n_i, n_j)$. Також вважаємо, що $M < \min\{n_i, i = \overline{1,k}\}$. Тоді маємо:

$$\begin{cases} C_1 = M^e \bmod n_1; \\ C_2 = M^e \bmod n_2; \\ \quad \ldots \\ C_k = M^e \bmod n_k. \end{cases}$$

Якщо $k \geq e$, то зловмисник, перехопивши значення $C_1, \ldots, C_k$, може дешифрувати повідомлення M таким чином.

1. Обчислити значення $C = M^e \bmod (n_1 \cdot n_2 \cdot \ldots \cdot n_k)$, використовуючи китайську теорему про лишки;
2. Оскільки $M < \min\{n_i, i = \overline{1,k}\}$, то $M^e < n_1 \cdot n_2 \cdot \ldots \cdot n_k$. Тоді $C = M^e$.
3. Використовуючи алгоритми обчислення кореня m-го степеня для дійсних чисел (наприклад, метод дотичних), обчислити $M = \sqrt[e]{C}$.

Описана атака є найпростішим випадком атаки Хастада (англ. *Hastad's Broadcast Attack*). В загальному випадку атака успішна навіть якщо користувач **B** відправляє різні повідомлення виду $f_i(M)$ для кожного з користувачів $\mathbf{A}_i$, де f_i – деякі лінійні функції, $i = \overline{1,k}$.

2.2 Атака «зустріч посередині»

Нехай зловмисник **E** перехопив шифротекст C:

$$C = M^e \bmod n,$$

причому відомо, що $M < 2^l$, $l \ll \log_2 n$. З великою ймовірністю повідомлення M – це складене число, тобто його можна представити як добуток чисел $M_1 \cdot M_2$. Припустимо, що при цьому $M_1 \leq 2^{l/2}$ та $M_2 \leq 2^{l/2}$. Тоді маємо:

$$C = (M_1 \cdot M_2)^e \bmod n = M_1^e \cdot M_2^e \bmod n.$$

В цьому випадку зловмисник **E** зможе дешифрувати повідомлення M, виконуючи такі кроки.

1. Криптоаналітик **E** формує множину пар X:

$$X = \left\{(1,1), (2, 2^e \bmod n), (3, 3^e \bmod n), \ldots, \left(2^{l/2}, \left(2^{l/2}\right)^e \bmod n\right)\right\},$$

тобто кожна з пар множини має вигляд $(T, T^e \bmod n)$, $T = \overline{1, 2^{l/2}}$.

2. Послідовно обчислює значення

$$C_S = C \cdot S^{-e} \bmod n,\ S = \overline{1, 2^{l/2}},$$

причому $S^{-e} \bmod n$ можна не обчислювати повторно, а використовувати вже обраховані значення з множини X.

   (а) Для кожного значення C_S, одразу ж після його обчислення, **E** шукає в множині X таку пару, щоб $_S = (T^e \bmod n)$ для деякого значення $T = \overline{1, 2^{l/2}}$.

   (б) Якщо таке t не знайдено, повертаємось на крок 2 та обчислюємо наступне значення C_{S+1}. Якщо при цьому $S = 2^{l/2}$, то алгоритм дешифрування зупиняє роботу з відповіддю *«Відкритий текст не було визначено»*.

3. Для знайденого значення $T^e \bmod n$ виконується рівність:

$$T^e = C \cdot S^{-e} \bmod n.$$

Тоді маємо:

$$C = T^e \cdot S^e \bmod n;$$

$$C = (T \cdot S)^e \bmod n,$$

тобто шифротекст C було отримано внаслідок шифрування відкритого тексту $T \cdot S$.

3 Дані для аналізу

Вхідні дані для виконання комп'ютерного практикуму представлені у вигляді `.txt` файлів:

- з каталогу `SE_RSA_`*`size`*`_e`, де визначаються значення C_i, n_i при відкритій експоненті e для атаки на основі китайської теореми про лишки;
- з каталогу `MitM_RSA_`*`size`*`_l.txt`, де задані значення C та n для атаки «зустріч посередині» при параметрі l.

Файл з каталогу обирається відповідно до варіанту. Номер варіанту бригади залишається тим ж, як при виконанні комп'ютерних практикумів №1 та №2.

При реалізації криптосистеми RSA для випадку малих експонент (для подальшої побудови атаки з використанням китайської теореми про лишки) використовувався паддинг для цифрового підпису RSA, що описаний у специфікації RFC 8017. При реалізації криптосистеми RSA для подальшої побудови атаки «зустріч посередині» використовувався параметр $e = 65537$.

4 Порядок виконання роботи і методичні вказівки

-1. Ознайомитись з порядком виконання комп'ютерного практикуму та відповідними вимогами до виконання роботи.

0. Уважно прочитати необхідні теоретичні відомості до комп'ютерного практикуму.

1. Створити новий репозиторій в системі контролю версій `Git` (бажано використовувати вебсервіс `GitHub` *). Важливо:
   (а) репозиторій створюється перед початком роботи над програмним кодом (якщо репозиторій приватний, то перед початком роботи має бути надано доступ викладачу до даного репозиторію);
   (б) весь процес створення програмного коду має бути відображений у відповідних комітах проекту (для кожної атомарної зміни коду має бути власний коміт);
   (в) програмна реалізація не допускається до захисту при недотриманні вищевизначених вимог.
2. Реалізувати атаку з малою експонентою на основі китайської теореми про лишки.
3. Реалізувати атаку «зустріч посередині» та порівняти її швидкодію з повним перебором можливих відкритих текстів.
4. Оформити звіт до комп'ютерного практикуму.

Додаткове завдання # 1: Самостійно реалізувати алгоритм обчислення кореня m-го степеня для дійсних (цілих) чисел, що використовується в атаці з використанням китайської теореми про лишки.

Додаткове завдання # 2: Реалізувати атаку типу «зустріч посередині» для значення $l = 56$ (потребує значної оптимізації та, можливо, застосування додаткових обчислювальних ресурсів).

Комп'ютерний практикум виконується у такому ж складі бригади, як виконувались комп'ютерні практикуми №1 та №2. Зміна складу бригади та способу виконання роботи протягом семестру можлива лише при узгодженні цього з викладачем комп'ютерних практикумів.

5 Оформлення звіту

Звіт про виконання комп'ютерного практикуму оформлюється згідно зі стандартними правилами оформлення наукових робіт за допомогою системи набору і верстки LaTeX, причому дозволяється використовувати розмір шрифту 12pt та одинарний міжрядковий інтервал. Звіт обов'язково має містити:

- мету комп'ютерного практикуму;
- постановку задачі та варіант завдання;
- хід роботи;
- результати проведених атак, включно з часом їх виконання;
- опис труднощів, що виникали при виконанні комп'ютерного практикуму, та шляхи їх розв'язання;
- висновки.

Лістинги програми дозволяється не включати у звіт.

6 Порядок захисту комп'ютерного практикуму

Для зарахування комп'ютерного практикуму студенту необхідно виконати захист теоретичної та практичної частин роботи (за умови своєчасного надання доступу викладачеві до `Git`-репозиторію, що містить код програми). Студент має можливість здавати теоретичну та практичну частини комп'ютерного практикуму в різні дні в довільному порядку.

*Використання інших сервісів необхідно попередньо узгодити з викладачем

7 Контрольні питання

1. Опис роботи криптосистеми RSA.
2. Атака на криптосистему RSA при використанні малої експоненти *одним* користувачем.
3. Атака на криптосистему RSA при використанні малої експоненти *багатьма* користувачами.
4. Чи може паддинг захищати від атаки на криптосистему RSA на основі китайської теореми про лишки?
5. Атака «зустріч посередині» на криптосистему RSA.

Оцінювання комп'ютерного практикуму

Можлива кількість рейтингових балів 12

- Програмна реалізація 6
- Теоретичний захист роботи 6
- Виконання атаки «зустріч посередині» для $l = 56$ (оптимізація) ..... +4 бали
- Власна реалізація алгоритму обчислення кореня m-го степеня ..... +0,5 бала

Несвоєчасне виконання роботи ..... -1 бал за кожен тиждень пропуску

Академічний плагіат .....-10 балів до рейтингу з вимогою виконати комп'ютерний практикум повторно та без можливості складання іспиту на основній сесії