

# Коротка криптографія

Тарас Фітьо

1 серпня 2004 — 2 вересня 2004 р.

Конфігурація тестового комп'ютера: AMD 2200+, 512MB.

## Зміст

1	Дефініції	1
2	Ознаки надійного шифра	2
3	Типи атак	2
4	Системи з відритим ключем	3
5	Проблеми генерування випадкових чисел	3
6	Історичні шифри та їх зломи	4
7	Як уникнути неякісного криптографічного програмного забезпечення	6
8	Блокове шифрування	7
9	Швидкості кодування для систем з симетричним ключем	7

## 1 Дефініції

Криптосистема або шифр (cipher) — метод так перетворити повідомлення, щоб тільки адресат зміг його прочитати.

Криптографія — наука створення і використання шифрів.

Криптоаналіз — наука зламування: вміння прочитати закодоване повідомлення, навіть якщо воно не призначалося вам.

Оригінальне повідомлення називається — простим текстом або повідомленням.

Закодоване — криптотекстом.

## 2 Ознаки надійного шифра

1. Надійність системи більше ґрунтується на секретності ключа ніж на уявлюваній секретності алгоритму<sup>1</sup>.
2. Надійний шифр має великий простір ключів.
3. Надійний шифр продукує криптотекст, який звичайно виглядає випадковим для всіх стандартних тестів<sup>2</sup>.
4. Надійний шифр є стійким до всіх відомих атак. Шифр, який не піддавався перевірці, — підозрілий.
5. Навіть шифр, що задовольняє всім вище переліченим пунктам, не обов'язково надійний.

Слід зауважити — тільки один шифр є незламний (метод одноразового блокноту), щодо решти сильних шифрів існують припущення.

## 3 Типи атак

Метою атаки не є знайти ключ, а прочитати простий текст (попередній/даний/наступний) або згенерувати криптотекст для заданого повідомлення. Вважається, що алгоритм кодування відомий. Деколи хакеру потрібно “просто” змінити/замінити повідомлення.

1. Хакеру відомий лише криптотекст.

---

<sup>1</sup>ЦРУ не повідомляє МОССАД, який алгоритм вона використовує для внутрішніх комунікацій, але остання при потребі взнає це

<sup>2</sup>Зокрема, це означає неможливість стиснути результат кодування

2. Хакер знає криптотекст так і відповідний простий текст. Хакер не може вибрати його.
3. Хакер для довільного простого тексту може знайти криптотекст.
4. Для довільного криптотексту може знайти відповідний простий текст (Має сенс для систем з відкритим ключем, фальшива аутентифікація).
5. Хакер може знайти криптотекст для вибраного простого тексту в інтерактивному чи в ітеративному процесі базованому на попередніх результатах. Це диференціальний криптоаналіз. *незрозуміло!*

## 4 Системи з відкритим ключем

В системах з відкритим ключем кожна особа має два ключа: один публічний, відомий всім, а другий таємний відомий тільки цій особі. Якщо  $A$  хоче надіслати повідомлення  $B$ , то вона, використовуючи відомий ключ  $B$ , створює повідомлення, котре не затрачаючи значних зусиль може прочитати тільки  $B$  використовуючи свій таємний ключ. Для інших прочитати повідомлення  $B$  — надзвичайно складна обчислювальна задача.

Перевага: ключ знає тільки  $B$ , не потрібно робити ніяких припущень щодо секретності передачі ключа, що має місце для стандартних криптосистем (де ключ ділять двоє, або більше осіб).

Недоліки: низька швидкодія у порівнянні з системами зі спільним ключем. Використовується такий вихід. Через систему з відкритим ключем співрозмовники обмінюються закодованим повідомленням, яке вони надалі використовують як спільний ключ.

Найвідоміший — RSA. Базується на модульній арифметиці для великих чисел, що і зумовлює його повільність.

## 5 Проблеми генерування випадкових чисел

Сильний шифр повинен мати великий простір ключів. Інакше його можна зламати звичайним перебором ключів (brute-force attack). При за-

даному розмірі найнадійнішим ключем служать випадкові дані<sup>3</sup>. Тому постає проблема їх генерування.

Апаратна генерація: найнадійніша. Починаючи від робота, котрий підкидує монету і визначає як та впала і закінчуючи квантовими ефектами проходження одиночного фотона крізь подільник. Недолік: низька швидкодія.

Програмна реалізація (генератори псевдо-випадкових чисел): задається початкове число і на його основі за певним алгоритмом генерується їх послідовність, яка і буде ключем. Велика швидкодія. Недолік: при відомому алгоритмі простір ключів не більший за простір початкового числа. Для більшості алгоритмів генерування існують ефективні методи злому (це якщо алгоритм невідомий).

В літературі рекомендується, по-можливості, використовувати чисто апаратні джерела, які гарантують випадковість. Інакше використовувати декілька нескорельованих комп'ютерних джерел (типу фону колонок, час доступу вінчестера, інтервали між натисками клавіатури і т.д.), якщо є перекося між частотами бітів — усунути їх<sup>4</sup>, перемішати ці дані<sup>5</sup>, а пізніше додатково стиснути їх<sup>6</sup>.

## 6 Історичні шифри та їх зломи

**Шифр частоколу** Ключ ціле число. При ключі 3 простий текст “sturto-graphy” перетворюється так  $c^r y p^{t^o} g^{r^a} p^{h^y}$ . Спочатку записуються найнижчі літери, пізніше середні, далі найвищі. Отримуємо “сргprtrhуоay”.

Потужність простору ключів менша ніж довжина повідомлення. Отже, криптоаналіз елементарний — простий перебір ключів.

**Шифр Цезаря** — зсув кожної літери на три позиції (“A” → “D”, “B” → “E”, ... “Z” → “C”). Можна узагальнити: при кодуванні кожна буква замінюється якоюсь іншою, декодування — зворотня заміна (квадрат 5 на 5). Для 26 літер алфавіту простір ключів доволі потужний (містить 26! ключів, це більше  $10^{26}$ ).

<sup>3</sup>Для осмисленого англійського тексту завдовжки 20 символів на кожних 8 біт, що припадають на символ, містить близько 2 біт інформації. Для українського ще менше.

<sup>4</sup>Наприклад з допомогою алгоритму Шенона: якщо два біти співпадають то вони не враховуються інакше враховується тільки перший

<sup>5</sup>Наприклад з допомогою операції XOR

<sup>6</sup>Наприклад використовуючи hash функцію MD5

Криптоаналіз доволі простий. Кожна мова характеризується статистичним розподілом літер. Для великого криптотексту можна відразу визначити найбільш частотні букви. Метод можна посилити замінюючи не кожен окрему букву, а блок — блоком, це розвиває статистику (частоти великих блоків або нуль, або мале число). Приклад: шифр чотирьох квадратів.

**Шифр Віженера** Винайдений в 16 столітті і протягом 300 років вважався незмінним. Кожна буква алфавіту нумерується починаючи від нуля. Ключ - певна послідовність літер. Кодується так: ключ періодично повторюється, щоб повністю покрити повідомлення; рахується сума номера кожної літери повідомлення з номером відповідної літери ключа; від цієї суми береться остача від ділення на кількість букв в алфавіті; остача — номер літери в криптотексті.

Приклад<sup>7</sup>:

Повідомлення	БОРОНІТЬКОРОЛІВНУВІДВОРОГІВ
Повторений ключ	КЛЮЧКЛЮЧКЛЮЧКЛЮЧКЛЮЧКЛЮЧКЛЮ
Криптотекст	ЛАОЇЮЦРФШАОЇЩЦАІГНЗЯМАОЇНЦА

Криптоаналіз: в криптотексті фіксуються комбінації літер, котрі повторюються. Обчислюється період повторів, який є кратним довжині ключа. Тоді для кожної підпослідовності літер криптотексту проводиться частотний аналіз.

Шифр Віженера можна підсилити: ключ використовується один раз, а далі в ролі ключа виступає простий текст. Якщо ключ є довшим за повідомлення і є цілком випадковим, то цей шифр є незламним<sup>8</sup> і називається **шифром одноразового блокноту** (one-time pad). Слід зауважити, що ключ можна використовувати тільки один раз (або користуватися, ще невживаними ділянками ключа).

**Шифр, який ми використовуємо** На основі ключа будується псевдовипадкова послідовність, яка додається до повідомлення-малюнка за модулем 2 (кожен біт виступає як окрема літера: 0 або 1).

Криптоаналіз: якщо використовувати той самий ключ до двох різних малюнків то при попіксельному накладанні криптомалюнків проступа-

<sup>7</sup>За Вербіцьким: “Вступ до криптології”, Львів 1998

<sup>8</sup>Шифр є незламним в теоретичному сенсі, на практиці виникають проблеми згенерувати великий масив випадкових даних. Шифри типу RSA чи DES є незламними в обчислювальному сенсі — на сучасних комп'ютерах вони не ламаються за тривалий проміжок часу.

ють риси і першого, і другого малюнків, а якщо хоча б один з них відомий то автоматично розкодується й інший. Також враховуючи те що простір ключів доволі малий ( $2^{32} \approx 4 \cdot 10^9$  і те, що існують ефективні оцінки чи декодування правильне (псевдовипадкова послідовність часто змінює колір) атака перебором ключів є доволі ефективною<sup>9</sup>.

Підсилення: послідовно кодувати з кількома різними ключами, два послідовно застосовані ключа вже не піддаються перебору. Весь час змінювати ключ.

## 7 Як уникнути неякісного криптографічного програмного забезпечення

Нижче наведено список заяв розробника, які вказують на можливі проблеми.

1. *Технолетет*: Опис системи є незрозумілим, можливо навіть для експертів. Характерне вживання торгових марок та нової термінології. Якщо проспекти є заплутані то чому цього очікувати від документації.
2. *Секретні алгоритми*: Якщо розробник хоче зберегти реалізацію алгоритмів в секреті це свідчить про можливі діри. Користувач не може перевірити надійності реалізації, а хакер все-одно зможе дизасемблювати програму.
3. *Революційний прорив*: Кожен новий шифр повинен випробовуватися криптоаналітиками. Надійність шифру випробовується часом. Уникайте програмного забезпечення з новими парадигмами обчислень (нейронні мережі, і т.д.). Новий тип обчислень дає такі ж результати, що й старий.
4. *Сертифікати, відгуки*: Не покладайтеся на відгуки в газетах, неспеціалізованих журналах. Те, що алгоритм патентований не робить його надійним.

---

<sup>9</sup>На тестовому комп'ютері повний перебір для зображення шириною 327 пікселів займає 57 хвилин

5. *Незламність*: про неї можна говорити тільки при виконанні певних умов (якщо комп'ютер не вкрадений, і т.д.).
6. *Одноразовий блокнот*: метод дійсно безпечний, але будь-яка модифікація може значно послабити його. Ключ має бути дійсно випадковим.
7. *Воєнне використання*: Це твердження здебільшого не вдається перевірити.

## 8 Блокове шифрування

В класичних шифрах (які розглядалися в 6 розділі) кожний символ кодувався окремо, незалежно від літер сусідів. Блокові шифри закодовують відразу блок з кількох символів (здебільшого блок завдовжки 8 байт). Так як блокові шифри використовують однакові алгоритми кодування для всіх блоків то якщо в простому тексті будуть наявні однакові блоки то однакові блоки з'являться і в криптотексті. Щоб уникнути цього кожен блок перед шифруванням певним чином змінюється в залежності від попереднього.

Є наступні режими зміни:

1. *Electronic Codebook (ECB)*: кожен блок шифрується незалежно (зі всіма вище наведеними недоліками).
2. *Cipher Block Chaining (CBC)*: перед кодуванням кожен блок простого тексту комбінується операцією XOR з попереднім блоком криптотексту.
3. *Cipher Feedback (CFB)*:

## 9 Швидкості кодування для систем з симетричним ключем

Виміри проводилися кодуючи та розкодовуючи файли завдовжки 314KB на програмному забезпеченні написаному Hagen'ом Reddmann'ом. З фрази "We are using a very, very long password." генерувався ключ відповідної довжини. Використовувався режим кодування Output Feedback.

Алгоритм	Довжина ключа, біт	час кодування, мс
Blowfish	448	69
DES	64	125
TripleDES	64	306
TripleDES	192	850
IDEA	128	231
RC5	2048	65
RC6	2048	168
Skipjack	80	233
Square	128	100
Twofish	256	183