Татьянчиков А.О.

Студент 5 курсу 1 групи фізико-математичного факультету СДПУ

Дослідження проблеми стійкості криптографічних систем

Побудовано обчислювально захищену криптосистему, яку протестовано на повідомлені довжиною в $1028 \cdot k$ ($k = 1, 5, 10, 20, 100$) символів. Показано залежність часу роботи алгоритмів шифрування (T_C), дешифрування ($T_D C$) та алгоритму ламання коду (T_D) від довжини повідомлення (N).

Ключові слова: *криптосистема, теоретико-інформаційна стійкість, обчислювальна захищеність, поліноміальні алгоритми, експоненційні алгоритми*

Стрімкий розвиток обчислювальної техніки та розвиток телекомунікаційних мереж веде до необхідності створення систем безпечного зберігання та передавання конфіденційної інформації. Ці завдання вирішуються за допомогою криптографічних алгоритмів захисту інформації.

Сімейство відображень шифру в сукупності з протоколами, що використовуються, утворюють криптосистему. Здатність криптосистеми протистояти атакам одержала назву криптографічної стійкості криптосистеми. Стійка криптосистема забезпечує захист інформації протягом тривалого часу, незважаючи на зусилля супротивника, що володіє значними матеріальними, інтелектуальними й обчислювальними ресурсами. Існує різниця між теоретико-інформаційною стійкістю (або теоретичною стійкістю) і обчислювальною захищеністю (або практичною стійкістю) криптосистеми. Криптографічна система називається обчислювально захищеною (або обчислювально стійкою), якщо найкращий з можливих алгоритмів, що зламують її, вимагає невиправдано високих витрат обчислювальних ресурсів. Беручи до уваги потужність сучасних комп'ютерів, можна вважати, що 2^{80} операцій, необхідних для зламу шифру, це та межа, виходячи за яку алгоритми зламу стають занадто дорогими. Таким чином, якщо мінімальне число N операцій, необхідних алгоритму, що атакує дану криптосистему, більше за 2^{80}, то говорять, що вона обчислювально захищена. Зауважимо, що ніяку реальну систему не можна обґрунтовано вважати обчислювально захищеною, оскільки ми не зможемо довести оптимальність знайденого методу зламу. Тому на практиці ми визначаємо її захищеність в обчислювальному відношенні в тому випадку, якщо кращий з відомих алгоритмів для її зламу вимагає неприпустимо великої кількості обчислень.

Таким чином, обчислювально, або доказово, стійка криптосистема є стійкою стосовно супротивника, чиї обчислювальні ресурси обмежені. Навіть у

тому випадку, коли супротивник володіє більшими, але обмеженими ресурсами, він усе ще не зможе зламати систему.

При вивченні обчислювально захищених схем необхідно:

- подбати про довжину ключа, бо якщо розмір ключа малий, то в супротивника цілком може вистачити ресурсів для зламу криптосистеми;
- стежити за останніми алгоритмічними досягненнями й розвитком комп'ютерної техніки.

Більшість криптосистем, що активно експлуатуються в наш час, обчислювально захищені.

З іншого боку, система називається абсолютно стійкою або досконалою, якщо ми не обмежуємо обчислювальної потужності супротивника. Інакше кажучи, криптосистема досконала, якщо її не можна зламати навіть за допомогою нескінченного числа операцій. Отже, незалежно від алгоритмічних досягнень і досконалості обчислювальної техніки, абсолютно стійку схему зламати неможливо.

До обчислювально стійких систем можна віднести криптосистеми:

- DES;
- RSA;
- ЭльГамаля.

Однак ці системи не є абсолютно стійкими.

Щоб оцінити обчислювальну складність задачі (швидкість росту об'єму потрібних для розв'язання цієї задачі обчислень через зростання її розмірності) потрібно оцінити ефективність алгоритму, за допомогою якого можна знайти розв'язок задачі. Алгоритм є ефективним, якщо він розв'язує задачу на прийнятних для користувача умовах, і неефективним – в противному разі.

Поняття “ефективності” пов'язане з усіма обчислювальними ресурсами, потрібними для роботи алгоритму. При використанні комп'ютерів це може бути кількість спожитої електроенергії, об'єм використаної пам'яті (зокрема, оперативної), час роботи алгоритму і т.ін. Але найчастіше домінуючим фактором є обмеження на час, тому “найефективніший” як правило означає “найшвидший”.

Часова складність даного алгоритму A – функція $f_A(n)$, яка визначається як найменший час, достатній для розв'язання будь-якої задачі із входом довжини n. Реалізація алгоритму на реальному комп'ютері дуже погано піддається опису і аналізу загальноматематичними засобами. Зокрема, вона значною мірою залежить від конструктивних особливостей даного комп'ютера. Тому при теоретичному аналізі ефективності алгоритмів зазвичай обмежуються порівняно простими обчислювальними моделями з обмеженим набором елементарних операцій, керуючись головним чином зручністю їх використання. Найпопулярнішою з них є машина Тьюрінга. Кількість кроків роботи машини Тьюринга до результативної зупинки називається *часом роботи* машини Тьюрінга на вході x.

Означення 1. *Алгоритм, який на нескінченній послідовності входів робить таку кількість кроків, залежну від n, яка може бути обмежена за допомогою поліному, де n довжина входу, називається поліноміальним. Про такий алгоритм кажуть, що він вимагає поліноміального часу.*

Поліноміальні алгоритми відповідають загальним уявленням про швидкі або ефективні алгоритми. Задачу, яку можна розв`язати за допомогою поліноміального алгоритму слід вважати легкою. В гіршому випадку такий алгоритм можна вважати швидким лише асимптотично, а на відносно невеликих входах алгоритм може працювати довго.

Означення 2. *Алгоритм, який на нескінченній послідовності входів робить більше як 2^{n^c} кроків, де n довжина входу, а $c > 0$ – деяка константа, називається експоненційним. Про такий алгоритм кажуть, що він вимагає експоненційного часу.*

Експоненційні алгоритми відповідають загальним уявленням про повільні, неефективні на праці алгоритми. Задачу, яку можна розв`язати лише за допомогою експоненційного алгоритму слід вважати важкою.

Якщо натуральне число n подається в системі числення за основою k, то його запис має довжину $[\log_k n]+1$, а алгоритм є поліноміальним тоді і тільки тоді, коли час його роботи на вході n (довжини $[\log_k n]+1$) обмежений функцією $c(\log_k n)^d$ для деяких констант $c>0$ і $d>1$. Алгоритм є експоненційним, якщо на вході n (точніше, на нескінченній послідовності таких входів) час його роботи перевищує cn^d для деяких констант $c,d>0$.

Означення 3. *Алгоритм називається швидким у середньому, якщо для переважної більшості входів довжини n він є швидким.*

Швидкий у середньому алгоритм є хоч і експоненційним, але може бути цілком придатним для практичних потреб.

Теорема *Для зламання $\Gamma(*,\circ)$ шифру потрібно здійснити не менше ніж 2^n кроків.*

Побудовано обчислювально захищену криптосистему, яку протестовано на повідомлені довжиною в $1028 \cdot k$ (k = *1, 5, 10, 20, 100*) символів. Досліджено можливості так званої пасивної атаки, тобто атаки на шифр, при якій атакуючий може дослідити тільки шифротекст і за ним повинен поновити таємний ключ.

Таблиця 1 ілюструє зміну часу роботи алгоритмів шифрування (T_C), дешифрування (T_DC) та алгоритму ламання коду (T_D) в залежності від довжини повідомлення (N):

Таблиця 1

k	**N**	$\mathbf{T_C}$	$\mathbf{T_DC}$	$\mathbf{T_D}$
1	1028	0.7	0.7	>0.9
5	5140	1.8	1.8	>3.7
10	10280	4.5	4.5	>8.2
20	20560	10.0	10.0	>21.1
100	102800	137.1	137.1	>293.4

В останньому стовпчику стоїть знак > тому, що алгоритм тестувався у випадку найсприятливішому для суперника, тобто припустили, що довжину ключа вгадали одразу. Тестування проводилося на ЕОМ Celeron, 1100 Hz, 256 ОЗУ.

Література

1. *Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В.* Основы криптографии. М.: Гелиос АРВ, 2001. – 479 с.
2. Введение в криптографию (под общей ред. Ященко В.В.). М.: МЦНМО - ЧеРо, 1999. – 271 с.
3. *Вербіцький О.В.* Вступ до криптології, Львів: Науково-технічна література, 1998. – 247 с.
4. *Виноградов И.М.* Основы теории чисел. М.: Наука, 1972. – 167 с.
5. *Гашков С.Б., Чубариков В.Н.* Арифметика. Алгоритмы. Сложность вичислений. – М.: Высшая школа, 2000. – 320 с.
6. *Кнут Д.* Искусство программирования для ЭВМ. Т.2. Получисленные алгоритмы. – М.: Мир, 1977.
7. *Смарт Н.* Криптографія. – М.: Техносфера, 2005. – 528 с.
8. *Koblitz Neal* Algebraic aspects of cryptography. – Berlin: Springer, 1998. – 200 p.
9. *Koblitz N* A Course in Number Theory and Cryptography. Springer-Verlag, New York, Inc., 1994 (є польські переклади: Wyk\lad z teorii liczb i kryptografii. Warszawa, Wydawnictwa Naukowo-Techniczne, 1995, 2000 і рос. переклад)
10. *Salomaa Arto*. Public-Key Cryptographi. – Berlin: Springer-Verlag, 1996. – 271 p.