#

# **РЕФЕРАТ З ТЕМИ:**

# **Шифрування з секретним ключем**

**1. Шифрування з секретним ключем**

Алгоритм RSA є алгоритмом асиметричної криптографії. Він був запропонований трьома дослідниками-математиками Рональдом Ривестом (R. Rivest) , Ади Шамиром (A. Shamir) і Леонардом Адльманом (L. Adleman) в 1977-78 роках.

Першим етапом будь-якого асиметричного алгоритму є створення пари ключів: відкритого і закритого та поширення відкритого ключа "по усьому світу". Для алгоритму RSA етап створення ключів складається з наступних операцій:

1. Вибираються два простих числа p та q
2. Обчислюється їх добуток n=(p\*q)
3. Вибирається довільне число e (e<n), таке, що НОД(e,(p-1)(q-1))=1, тобто e повинне бути взаємно простим із числом (p-1)(q-1).
4. Методом Євкліда в цілих числах знаходиться рішення рівняння e\*d+(p-1)(q-1)\*y=1. Тут невідомими є змінні d і y - метод Евкліда саме і знаходить безліч пар (d,y), кожна з яких є рішенням рівняння в цілих числах.
5. Два числа (e,n) - публікуються як відкритий ключ.
6. Число d зберігається в найсуворішому секреті - це і є закритий ключ, що дозволить читати всі послання, зашифровані за допомогою пари чисел (e,n).

Як же виробляється шифрування за допомогою цих чисел:

1. Відправник розбиває своє повідомлення на блоки, рівні k=[log2(n)] біт, де квадратні дужки позначають узяття цілої частини від дробового числа.
2. Подібний блок може бути інтерпретований як число з діапазону (0;2k-1). Для кожного такого числа (назвемо його mi) обчислюється ci=((mi)e)mod n. Блоки ci і є зашифроване повідомлення Їх можна спокійно передавати по відкритому каналу, оскільки операція зведення в ступінь по модулі простого числа, є необоротним математичним завданням. Зворотна їй операція – "логарифмування в кінцевому полі" є на кілька порядків більше складним завданням. Тобто навіть, якщо зловмисник знає числа e і n, те по ci прочитати вихідні повідомлення mi він не може ніяк, крім як повним перебором mi.

А от на прийомній стороні процес дешифрування все-таки можливий, і допоможе нам у цьому збережене в секреті число d. Досить давно була доведена теорема Ейлера, окремий випадок якої затверджує, що якщо число n представимо у вигляді двох простих чисел p і q, то для будь-якого x має місце рівність (x(p-1)(q-1))mod n = 1. Для дешифрування RSA-повідомлень скористаємося цією формулою. Зведемо обидві її частини в ступінь (-y): (x(-y)(p-1)(q-1))mod n = 1(-y) = 1. Тепер помножимо обидві її частини на x: (x(-y)(p-1)(q-1)+1)mod n = 1\*x = x.

А тепер згадаємо як ми створювали відкритий і закритий ключі. Ми підбирали за допомогою алгоритму Евкліда d таке, що e\*d+(p-1)(q-1)\*y=1, тобто e\*d=(-y)(p-1)(q-1)+1. А отже в останній рівності попереднього абзацу ми можемо замінити показник ступеня на число (e\*d). Одержуємо (xe\*d)mod n = x. Тобто для того щоб прочитати повідомлення ci=((mi)e)mod n досить звести його в ступінь d по модулі m: ((ci)d)mod n = ((mi)e\*d)mod n = mi.

Операції зведення в ступінь більших чисел досить трудомісткі для сучасних процесорів, навіть якщо вони виробляються по оптимізованим за часом алгоритмам. Тому звичайно весь текст повідомлення кодується звичайним блоковим шифром (набагато більше швидким), але з використанням ключа сеансу, а от сам ключ сеансу шифрується саме асиметричним алгоритмом за допомогою відкритого ключа одержувача і міститься в початку файлу.

# **2. Характеристики стандартних алгоритмів шифрування з секретним ключем**

**2.1 Симетричне шифрування**

Свою історію алгоритми симетричного шифрування ведуть зі стародавності: саме цим способом приховання інформації користувався римський імператор Гай Юлій Цезар в I столітті до н.е., а винайдений їм алгоритм відомий як "криптосистема Цезаря".

У наш час найбільш відомий алгоритм симетричного шифрування DES (Data Encryption Standard), розроблений в 1977 р. Донедавна він був "стандартом США", оскільки уряд цієї країни рекомендував застосовувати його для реалізації різних систем шифрування даних. Незважаючи на те, що DES планувалося використати не більше 10-15 років, його замінили тільки в 1997 р.

Основна причина заміни стандарту шифрування - його відносно слабка криптостойкість, причина якої в тім, що довжина ключа DES становить усього 56 значущий біт. Відомо, що будь-який криптостійкий алгоритм можна зламати, перебравши усі можливі варіанти ключів шифрування (так званий метод грубої сили - brute force attack). Легко підрахувати, що кластер з 1 млн. процесорів, кожний з яких обчислює 1 млн. ключів у секунду, перевірить 256 варіантів ключів DES майже за 20 ч. А оскільки по нинішніх мірках такі обчислювальні потужності цілком реальні, ясно, що 56-бітовий ключ занадто короткий і алгоритм DES необхідно замінити на більш ефективний.

Сьогодні усе ширше використовуються два сучасних крипостійких алгоритми шифрування: вітчизняний стандарт ГОСТ 28147-89 і новий криптостандарт США - AES (Advanced Encryption Standard).

#### 2.2 Стандарт ГОСТ 28147-89

Алгоритм, обумовлений ГОСТ 28147-89 (рис. 1), має довжину ключа шифрування 256 біт. Він шифрує інформацію блоками по 64 біт (такі алгоритми називаються блоковими), які потім розбиваються на два субблоки по 32 біт (N1 і N2). Субблок N1 обробляється певним чином, після чого його значення складається зі значенням субблока N2 (додавання виконується по модулю 2, тобто застосовується логічна операція XOR - "виключне або"), а потім субблоки міняються місцями. Дане перетворення виконується певне число разів ("раундів"): 16 або 32 залежно від режиму роботи алгоритму. У кожному раунді виконуються дві операції.

Рисунок 1 – Схема алгоритму ГОСТ 28147-89

Перша - накладення ключа. Вміст субблоку N1 складається по модулю 2 з 32-бітною частиною ключа Kx. Повний ключ шифрування представляється у вигляді конкатенації 32-біт підключів: K0, K1, K2, K3, K4, K5, K6, K7. У процесі шифрування використається один із цих підключів - залежно від номера раунду і режиму роботи алгоритму.

Друга операція - таблична заміна. Після накладення ключа субблок N1 розбивається на 8 частин по 4 біт, значення кожної з яких заміняється відповідно до таблиці заміни для даної частини субблока. Потім виконується побітовий циклічний зсув субблока вліво на 11 біт.

Табличні заміни часто використаються в сучасних алгоритмах шифрування, тому варто пояснити, як організується подібна операція. У таблицю записуються вихідні значення блоків. Блок даних певної розмірності (у нашому випадку - 4-біта) має своє числове подання, що визначає номер вихідного значення. Наприклад, якщо S-box має вигляд 4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1 і на вхід прийшов 4-біта блок "0100" (значення 4), то, відповідно до таблиці, вихідне значення буде дорівнює 15, тобто "1111" (0 а 4, 1 а 11, 2 а 2...).

Алгоритм, обумовлений ГОСТ 28147-89, передбачає чотири режими роботи: простої заміни, гамування, гамування зі зворотним зв'язком і генерації імітоприставок. У них використається те саме описане вище перетворення але, оскільки призначення режимів по-різному, здійснюється це перетворення в кожному з них по-різному.

У режимі простої заміни для шифрування кожного 64-бітног блоку інформації виконуються 32 описаних вище раунда. При цьому 32-бітні підключи використовуються в наступній послідовності:

K0, K1, K2, K3, K4, K5, K6, K7, K0, K1 і т.д. - у раундах з 1-го по 24-й;

K7, K6, K5, K4, K3, K2, K1, K0 - у раундах з 25-го по 32-й.

Розшифрування в даному режимі проводиться так само, але із трохи іншою послідовністю застосування підключей:

K0, K1, K2, K3, K4, K5, K6, K7 - у раундах з 1-го по 8-й;

K7, K6, K5, K4, K3, K2, K1, K0, K7, K6 і т.д. - у раундах з 9-го по 32-й.

Всі блоки шифруються незалежно друг від друга, тобто результат шифрування кожного блоку залежить тільки від його змісту (відповідного блоку вихідного тексту). При наявності декількох однакових блоків вихідного (відкритого) тексту відповідні їм блоки шифртексту теж будуть однакові, що дає додаткову корисну інформацію для крипто аналітика, що намагається розкрити шифр. Тому даний режим застосовується в основному для шифрування самих ключів шифрування (дуже часто реалізуються багатоключові схеми, у яких по ряду міркувань ключі шифруються друг на другу). Для шифрування інформації призначені два інших режими роботи - гамування та гамування зі зворотним зв'язком.

У режимі гамування кожен блок відкритого тексту побітно складається по модулю 2 із блоком гами шифру розміром 64 біт. Гама шифр - це спеціальна послідовність, що виходить у результаті певних операцій з регістрами N1 й N2 (див. рис. 1).

1. У регістри N1 і N2 записується їх початкове заповнення - 64-бітна величина, називана синхропосилання.

2. Виконується шифрування змісту регістрів N1 й N2 (у цьому випадку - синхропосилання) у режимі простої заміни.

3. зміст регістра N1 складається по модулю (232 - 1) з константою C1 = 224 + 216 + 28 + 24, а результат додавання записується у регістр N1.

4. зміст регістра N2 складається по модулю 232 з константою C2 = 224 + 216 + 28 + 1, а результат додавання записується в регістр N2.

5. зміст регістрів N1 і N2 подається на вихід як 64-бітовий блок гами шифру (у цьому випадку N1 і N2 утворять перший блок гами).

Якщо необхідно наступний блок гами (тобто необхідно продовжити шифрування або розшифрування), виконується повернення до операції 2.

Для розшифрування гама виробляється аналогічним способом, а потім до біт зашифрованого тексту і гами знову застосовується операція XOR. Оскільки ця операція оборотна, у випадку правильно виробленої гами виходить вихідний текст (таблиця 1).

#### Таблиця 1 - Зашифрування й розшифрування в режимі гамування

|  |  |  |
| --- | --- | --- |
|   | Операція | Результат |
| Вихідний текст |   | 100100 |
| Гама | XOR | 111000 |
| Шифртекст | = | 011100 |
| Гама | XOR | 111000 |
| Вихідний текст | = | 100100 |

Для отримання потрібної для розшифровки гами шифру у користувача, що розшифровує криптограму, повинен бути той же ключ і те ж значення синхропосилання, які застосовувалися при шифруванні інформації. У іншому випадку одержати вихідний текст із зашифрованого не вдасться.

У більшості реалізацій алгоритму ГОСТ 28147-89 синхропосилання не секретне, однак є системи, де синхропосилання - такий же секретний елемент, як і ключ шифрування. Для таких систем ефективна довжина ключа алгоритму (256 біт) збільшується ще на 64 біт секретного синхропосилання, що також можна розглядати як ключовий елемент.

У режимі гаміровання зі зворотним зв'язком для заповнення регістрів N1 і N2, починаючи з 2-го блоку, використається не попередній блок гами, а результат шифрування попереднього блоку відкритого тексту (рис. 2). Перший блок у даному режимі генерується повністю аналогічно попередньому.

Рисунок 2 – Формування гами шифру в режимі гамування із зворотним зв'язком

Розглядаючи режим генерації імітоприставок, варто визначити поняття предмета генерації. Імітоприставка - це криптографічна контрольна сума, що обчислюється з використанням ключа шифрування і призначена для перевірки цілісності повідомлень. При генерації імітоприставки виконуються наступні операції: перші 64-біта блок масиву інформації, для якого обчислюється імітоприставка, записується в регістри N1 й N2 і шифрується в скороченому режимі простої заміни (виконуються перші 16 раундів з 32). Отриманий результат додається по модулю 2 до наступного блока інформації зі збереженням результату в N1 і N2.

Цикл повторюється до останнього блоку інформації. 64-біт, що вийшло в результаті цих перетворень, у регістрів N1 й N2 називається імітоприставкою. Розмір імітоприставки вибирається, виходячи з необхідної вірогідності повідомлень: при довжині імітоприставки r біт імовірність, що зміна повідомлення залишиться непоміченою, дорівнює 2-r.Найчастіше використається 32-бітна імітоприставка, тобто половина змісту регістрів. Цього досить, оскільки, як будь-яка контрольна сума, імітоприставка призначена насамперед для захисту від випадкових спотворень інформації. Для захисту від навмисної модифікації даних застосовуються інші криптографічні методи - у першу чергу електронний цифровий підпис.

Алгоритм ГОСТ 28147-89 вважається дуже сильним алгоритмом - у цей час для його розкриття не запропоновано більш ефективних методів, чим згаданий вище метод "грубої сили". Його висока стійкість досягається в першу чергу за рахунок великої довжини ключа - 256 біт. При використанні секретного синхропосилання ефективна довжина ключа збільшується до 320 біт, а засекречування таблиці замін додає додаткові біти. Крім того, кріптостійкість залежить від кількості раундів перетворень, яких за ГОСТ 28147-89 повинне бути 32 (повний ефект розсіювання вхідних даних досягається вже після 8 раундів).

#### 2.3 Стандарт AES

На відміну від алгоритму ГОСТ 28147-89, що довгий час залишався секретним, американський стандарт шифрування AES, покликаний замінити DES, вибирався на відкритому конкурсі, де всі зацікавлені організації і приватні особи могли вивчати і коментувати алгоритми-претенденти.

Конкурс на заміну DES був оголошений в 1997 р. Національним інститутом стандартів і технологій США. На конкурс було представлено 15 алгоритмів-претендентів, розроблених як відомими в області криптографії організаціями (RSA Security, Counterpane і т.д.), так і приватними особами. Підсумки конкурсу були підведені в жовтні 2009 р.: переможцем був оголошений алгоритм Rijndael, розроблений двома кріптографами з Бельгії, Винсентом Риджменом (Vincent Rijmen) і Джоан Даймен (Joan Daemen).

Алгоритм Rijndael не схожий на більшість відомих алгоритмів симетричного шифрування, структура яких зветься "мережа Фейстеля" і аналогічна російському ГОСТ 28147-89. Особливість мережі Фейстеля полягає в тому, що вхідне значення розбивається на два і більше субблоків, частина з яких у кожному раунді обробляється за певним законом, після чого накладається на необроблювані субблоки (див. рис. 1).

На відміну від вітчизняного стандарту шифрування, алгоритм Rijndael представляє блок даних у вигляді двомірного байтового масиву розміром 4X4, 4X6 або 4X8 (допускається використання декількох фіксованих розмірів шифруєемого блоку інформації). Всі операції виконуються з окремими байтами масиву, а також з незалежними стовпцями і рядками. Алгоритм Rijndael виконує чотири перетворення: BS (ByteSub) - таблична заміна кожного байта масиву (рис. 3); SR (ShiftRow) - зсув рядків масиву (рис. 4). При цій операції перший рядок залишається без змін, а інші циклічно побайтно зсуються вліво на фіксоване число байт, що залежить від розміру масиву. Наприклад, для масиву розміром 4X4 рядки 2, 3 і 4 зсуються відповідно на 1, 2 і 3 байти. Далі йде MC (MixColumn) - операція над незалежними стовпцями масиву (рис. 5), коли кожен стовпець за певним правилом множиться на фіксовану матрицю c(x). І, нарешті, AK (AddRoundKey) - додавання ключа. Кожен біт масиву складається по модулю 2 з відповідним ключем раунду, що, у свою чергу, певним чином обчислюється із ключа шифрування (рис. 6).

Рисунок 3 – Операція BS

Рисунок 4 – Операція SR

Рисунок 5 – Операція MC

Рисунок 6 – Операція AK

У кожному раунді (з деякими виключеннями) над даними, що шифруються, по черзі виконуються перераховані перетворення (рис. 7). Виключення стосуються першого і останнього раундів: перед першим раундом додатково виконується операція AK, а в останньому раунді відсутній MC. У результаті послідовність операцій при шифруванні виглядає так:

AK, {BS, SR, MC, AK} (повторюється R-1 раз), BS, SR, AK.

Рисунок 7 – Раунд алгоритму Rijndael

Кількість раундів шифрування (R) в алгоритмі Rijndael змінне (10, 12 або 14 раундів) і залежить від розмірів блоку і ключа шифрування (для ключа також передбачено кілька фіксованих розмірів).

Розшифрування виконується за допомогою наступних зворотних операцій. Виконується обіг таблиці і таблична заміна на інверсній таблиці (щодо застосовуваної при шифруванні). Зворотна операція до SR - це циклічний зсув рядків вправо, а не вліво. Зворотна операція для MC - множення за тим же правилом на іншу матрицю d(x), що задовольняє умові: c(x) \* d(x) = 1. Додавання ключа AK є зворотним самому собі, оскільки в ньому використається тільки операція XOR. Ці зворотні операції застосовуються при розшифруванні в послідовності, зворотної тієї, що використалася при шифруванні.

Rijndael став новим стандартом шифрування даних завдяки цілому ряду переваг перед іншими алгоритмами. Насамперед він забезпечує високу швидкість шифрування на всіх платформах: як при програмній, так і при апаратній реалізації. Його відрізняють незрівнянно кращі можливості розпаралелювання обчислень у порівнянні з іншими алгоритмами, представленими на конкурс. Крім того, вимоги до ресурсів для його роботи мінімальні, що важливо при його використанні в пристроях, що володіють обмеженими обчислювальними можливостями.

Недоліком же алгоритму можна вважати лише властиву йому нетрадиційну схему. Справа в тому, що властивості алгоритмів, заснованих на мережі Фейстеля, добре досліджені, а Rijndael, на відміну від них, може містити сховані уразливості, які можуть виявитися тільки через деякий час з моменту початку його широкого поширення.