НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Звіт з виконання комп'ютерного практикуму №1

ДИФЕРЕНЦІАЛЬНИЙ КРИПТОАНАЛІЗ БЛОКОВИХ ШИФРІВ

Виконав студент
групи ФІ-32мн
Мельник Ілля

Перевірив:
Столович М.В.

Київ — 2024

1 КОМП'ЮТЕРНИЙ ПРАКТИКУМ №1

Мета роботи: дослідити методи побудови та реалізації атак на блокові шифри з використанням диференціального криптоаналізу. Дослідження стійкості шифрів до атак такого виду.

Постановка задачі: для заданого умовою блокового шифру Хейса та заданою перестановкою S, обчислити високоімовірні диференціали 5-го раунду з допомогою методу "Границь та гілок". Провести атаку на сьомий раундовий ключ використовуючи статистичний матеріал, згенерований допоміжною програмою.

Варіант: 3.

1.1 Хід Роботи

1) Реалізувати 6-ти раундовий шифр Хейса для довжини 16 з блоками S заданими варіантом. Ключем шифрування вважати довільний 112-бітний бітовий рядок, який розбивається на сім незалежних 16-бітних раундових підключа.

2) Реалізувати пошук високоімовірних п'ятираундових диференціалів шифру Хейса методом «гілок та границь».

3) Реалізувати атаку на сьомий раундовий ключ шифру Хейса. Для побудови атаки використати знайдені на попередньому кроці диференціали із високою імовірністю. Необхідний статистичний матеріал (шифровані тексти) одержується із тестової програми Heys.exe, що додається.

1.2 Опис труднощів

Основною проблемою постала реалізація правильного обрахунку високоімовірних диференціалів. Навіть з теоритичної точки зору, ця задача

не є тривіальною для звичного обрахунку в силу можливого різкого збільшення числа диференціалів з кожним кроком. При цьому необхідно зберігати диференціали попереднього раунду, що використовуються для обрахунку нових. Було обрано Hashmap, як тип даних, для зберігання цих структур, що з однієї сторони дає змогу зручно витягати імовірності диференціалів, але при цьому цей тип даних має мінус – ми не можемо витягнути значення ключів (значення диференціалу), що містяться.

Окрім цього виникли великі проблеми із зчитуванням та записуванням чисел у байтовий вид, оскільки Java сприймає байти як знакові числа. В якості вирішення цього, були використані хитрощі переводу до беззнакового виду.

1.3 Пошук високоімовірних диференціалів

Для реалізації цієї задачі необхідні наступні кроки:

Спочатку виберемо альфу (для обрання кращої пари, варто перебирати усі можливі альфи) та зафіксуємо її, розбивши на блоки. Далі, використовуючи таблицю диференціальних імовірностей для S-блоку, ми можемо обрати значення блоків (умовні бета, в які даний блок може перейти з деякою ненульовою імовірністю) та сформувати усі можливі варіанти міксування їх. Причому, ймовірність такого міксованого вектора буде як добуток ймовірностей переходу кожного початкового блоку альфи у дані бета. Оскільки на першому раунді початкова альфа обрана єдиним варіантом, тому її ймовірність одиниця. А от з кожним наступним кроком ймовірність вектору у який перейде наша гама (назвемо так умовну точку на і-тому раунді) буде відповідно ймовірність переходу з і-1 раунду на і раунд та домножити на ймовірність переходу початкової альфи у нашу гаму. Таким чином, ми будемо формувати список на кожному раунді в що може перейти альфа з деякою ненульовою ймовірністю. І звичайно суть метода полягає у тому, що після кожного раунду ми можемо зменшувати цей список, відкидаючи бета, що будуть малойвірними.

Для нашого варіанта було обрано поріг 0.0008. Цей поріг виявився оптимальним, оскільки він урізає інші можливі варіанти різниць, залишаючи лише одну високоімовірну бету.

Варто додати, що окрім цього відсіюються пари, в яких бета має нульовий блок. Оскільки для нас це критично (не зможемо відновити однозначно ключ через недостатність інформації), тому наша отримана різниця є високоімовірною при ненульових блоках бета.

1.4 Таблиця диференціальних імовірностей для S-блоку

$\beta\|\alpha$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.125	0.125	0.0	0.0	0.0	0.125	0.0	0.125	0.25	0.125	0.125	0.0
2	0.0	0.0	0.0	0.125	0.25	0.0	0.0	0.125	0.0	0.0	0.125	0.0	0.0	0.0	0.125	0.25
3	0.0	0.125	0.0	0.25	0.0	0.0	0.0	0.125	0.125	0.0	0.125	0.125	0.0	0.0	0.0	0.125
4	0.0	0.125	0.125	0.0	0.0	0.0	0.0	0.0	0.0	0.125	0.125	0.0	0.125	0.125	0.125	0.125
5	0.0	0.125	0.25	0.0	0.125	0.125	0.125	0.0	0.125	0.0	0.0	0.0	0.0	0.0	0.125	0.0
6	0.0	0.125	0.125	0.125	0.0	0.125	0.0	0.0	0.0	0.0	0.0	0.125	0.0	0.25	0.125	0.0
7	0.0	0.25	0.0	0.0	0.0	0.125	0.125	0.0	0.0	0.0	0.125	0.125	0.125	0.0	0.125	0.0
8	0.0	0.0	0.125	0.125	0.0	0.0	0.25	0.0	0.0	0.125	0.0	0.125	0.125	0.0	0.0	0.125
9	0.0	0.0	0.125	0.0	0.0	0.125	0.0	0.25	0.0	0.125	0.125	0.125	0.0	0.0	0.0	0.125
10	0.0	0.125	0.0	0.0	0.0	0.0	0.125	0.0	0.25	0.125	0.0	0.0	0.125	0.125	0.0	0.125
11	0.0	0.0	0.0	0.25	0.125	0.25	0.125	0.0	0.125	0.0	0.125	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.125	0.125	0.125	0.125	0.0	0.0	0.0	0.125	0.0	0.125	0.0	0.125	0.0	0.125
13	0.0	0.0	0.0	0.0	0.125	0.0	0.125	0.25	0.125	0.125	0.0	0.0	0.0	0.125	0.125	0.0
14	0.0	0.125	0.0	0.0	0.125	0.0	0.125	0.125	0.0	0.0	0.0	0.125	0.125	0.125	0.125	0.0
15	0.0	0.0	0.125	0.0	0.0	0.0	0.0	0.125	0.25	0.125	0.25	0.0	0.125	0.0	0.0	0.0

Таблиця 1.1

1.5 Результати пошуку високоімовірних диференціалів

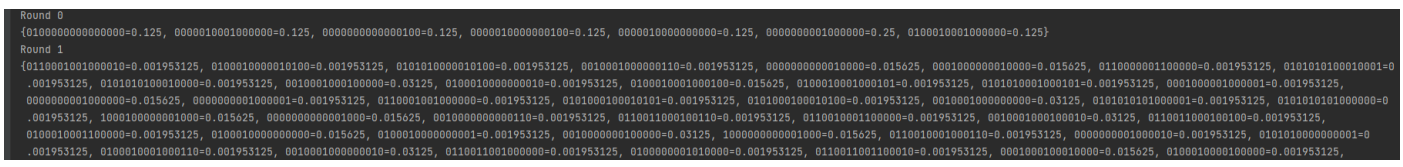

Рисунок 1.1 – Результати відпрацювання програми

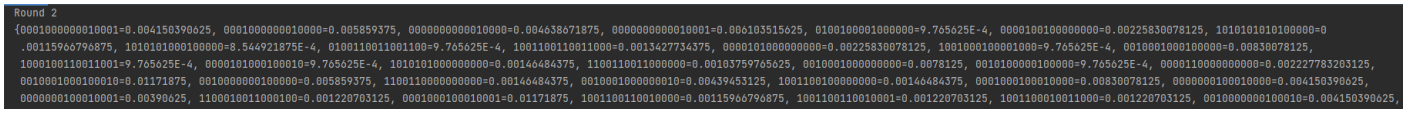

Рисунок 1.2 – Результати відпрацювання програми

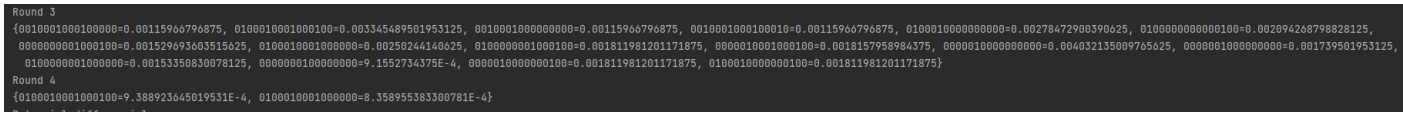

Рисунок 1.3 – Результати відпрацювання програми

1.6 Результати атаки

Для реалізації атаки було згенеровано 10000 текстів звичайних та 10000 текстів заксорених зі високоімовірною альфою, в сумі 20000. В ході атаки було знайдено ключ к7= 1011001100001101. Кількість співпадінь, що $Y' = Y \oplus \beta$ дорівнює 11, що є найкращим результатом і ймовірність правильності ключа точно значно перевищує імовірність 2^{-16}.

1.7 Код програми

Наведений у файлі "$Heys.java$"

ВИСНОВКИ

У даній роботі ми програмно реалізували атаку на останній ключ з допомогою диференціального криптоаналізу, використовуючи високоімовірні диференціали, знайдені методом гілок та границь.

Підсумовуючи усе, можна сказати, що диференціальний криптоаналіз дійсно може бути дієвим для деяких видів шифрів, але при цьому він доволі затратний по обчислювальним ресурсам.