УДК 621.391:519.2:519.7

А.П. БУБИР, І.Д. ГОРБЕНКО, д-р техн. наук

ОЦІНКА СТІЙКОСТІ НАПРАВЛЕНОГО ШИФРУ NTRU ДО АТАКИ З АДАПТИВНО ПІДІБРАНИМИ ШИФРОТЕКСТАМИ

Вступ

Використання криптосистем з відкритим ключем – основний спосіб захисту різноманітної інформації, що передається за допомогою комп'ютерних мереж. Найбільш широко використовуються такі криптосистеми, RSA яких заснована на складності факторизації великих чисел, схема Діфі – Гелмана, DSA, стійкість яких базується на складності вирішення задачі дискретного логарифмування в полі, сімейство алгоритмів у групі точок еліптичних кривих. Але всі вони мають певні недоліки, основні з яких – це або відносно низька швидкість (наприклад, алгоритми на базі ЕК), або низька стійкість (схема Діфі – Гелмана та інші алгоритми, засновані на проблемі пошуку дискретного логарифма в полі), або ці два недоліки одночасно (RSA). Вирішити проблему низької швидкості шифрування при збереженні високого рівня стійкості покликаний алгоритм NTRU, який було розроблено в середині 1990-х років та вперше представлено на конференції CRYPTO'96. NTRU було запатентовано компанією «NTRU Cryptosystems» (зараз – «Security Innovation») 24 липня 2000 р. В цьому алгоритмі всі операції виконуються у кільці зрізаних поліномів. Криптографічна стійкість алгоритму заснована на складності знаходження короткого вектора в заданій алгебраїчній решітці. В 2008 р. даний алгоритм був включений в стандарт [1], а його модифікована версія була взята за основу стандарту [2]. NTRU виконує операції шифрування, розшифрування, генерації ключів на порядки швидше [3] у порівнянні з іншими асиметричними криптосистемами, однак його криптографічна стійкість до різних класів атак ще недостатньо вивчена.

Опис атаки

В цій роботі була розглянута аналітична атака з адаптивно підібраними шифротекстами. Атака можлива за умови, що криптоаналітик отримає деяке число криптограм e, які являють собою одне й те ж повідомлення m, зашифроване на одному й тому ж відкритому ключі h, але із використанням різних сеансових ключів («засліплюючих поліномів») r_i [2]. Тоді криптоаналітик зможе відновити більшість або всі коефіцієнти полінома r_0, який відповідає криптограмі e_0, і потім обчислити значення зашифрованого повідомлення.

Нижче наведено узагальнений алгоритм атаки:

1) криптоаналітик перехоплює або будь-яким іншим чином отримує набір із C криптограм, які мають вищенаведену властивість;

2) на наступному кроці він обчислює різниці $r_i - r_0, 0 < i < C$ згідно з наступним співвідношенням:

$$(e_i - e_0) * h_q^{-1} = r_i - r_0 (\mathrm{mod} q), \quad (1)$$

де e_i – i-та криптограма.

Справедливість виразу (1) витікає з наступного. Якщо e_i визначимо як:

$$e_i = r_i * h + m (\mathrm{mod}\ q), \quad (2)$$

то

$$e_i - e_0 = (r_i * h + m) - (r_0 * h + m) (\mathrm{mod}\ q) = (r_i - r_0) * h (\mathrm{mod}\ q). \quad (3)$$

Помноживши обидві частини рівності (3) на інверсію відкритого ключа h_q^{-1}, отримуємо формулу (1);

3) аналізуючи коефіцієнти отриманих різниць, криптоаналітик проводить процедуру відновлення коефіцієнтів r_0. Якщо r_0 має число одиниць та мінус одиниць, яке дорівнює значенню d_r використаного набору параметрів, то він відновлений правильно, у противному випадку алгоритм сигналізує о про невдалу спробу розкриття й переривається;

4) на останньому кроці криптоаналітик обчислює значення вихідного повідомлення m:

$$m = e_0 - r_0 * h(\mathrm{mod}\ q). \qquad (4)$$

Опишемо більш детально ідею, яка лежить в основі процедури відновлення коефіцієнтів r_i. Якщо поліноми r_j є бінарними (коефіцієнти приймають значення {0, 1}), то кожний коефіцієнт різниці $r_i - r_0$ буде належати множині {-1, 0, 1}. Якщо він дорівнює 0, то даний факт не дасть додаткової інформації про значення відповідного коефіцієнта r_0. (1 – 1 = 0, 0 – 0 = 0). Але різниця, яка дорівнює 1, можлива лише тоді, коли коефіцієнт r_i дорівнює одиниці, а коефіцієнт r_0 дорівнює нулю (1 – 0 = 1). Аналогічно, різниця, яка дорівнює мінус одиниці, означає, що r_i має на даній позиції нуль, а r_0 – одиницю. Кількість одиниць та мінус одиниць у отриманих різницях визначає, чи буде атака успішною. Як тільки буде виявлено, що поліном, який відновлюється, містить d_r одиниць, то всі нерозкриті позиції заповнюються нулями, алгоритм зупиняється. На виході отримуємо сеансовий ключ r_0.

Якщо коефіцієнти r_j приймають значення з множини {-1, 0, 1}, то їх різниця може дорівнювати -2, -1, 0, 1, 2. Однозначно дозволяють визначити коефіцієнти сеансового ключа r_0 тільки значення -2 (-1 – 1) та 2 (1– (-1)). У першому випадку коефіцієнт r_0 дорівнює одиниці, у другому – мінус одиниці. Треба зауважити, що у випадку тернарних коефіцієнтів криптоаналітику необхідно використати більшу кількість криптограм (ймовірність того, що окремо взятий коефіцієнт різниці дозволить відновити коефіцієнт r_0, дорівнює 2/5, в бінарному випадку така ймовірність значно вища й складає 2/3).

Опис інструментального засобу

При програмному моделюванні було використано мову С й відкриту бібліотеку libntru, яка реалізує всі основні математичні операції в кільцях зрізаних поліномів, а також процедури генерації ключової пари, направленого шифруваннння за алгоритмом NTRU згідно із чорновою версією стандарту [1].

Вибір libntru у якості основної бібліотеки для реалізації атаки обумовлений тим, що на сьогоднішній день вона є єдиною бібліотекою з відкритим вихідним кодом, яка реалізує алгоритм НШ NTRU. Поточна версія бібліотеки працює тільки під UNIX-подібними операційними системами (GNU/Linux, MacOS X, FreeBSD, Android й т.п.). Це пояснюється використанням у якості джерела псевдовипадкових бітів спеціальних символьних псевдопристроїв /dev/random або /dev/urandom (на вибір користувача), які відсутні в сімействі ОС Windows. Компіляція даної бібліотеки під Windows потребує модифікації коду, який реалізує генерацію псевдовипадкових послідовностей. Для роботи деяких функцій libntru потрібна наявність у системі криптографічного пакету OpenSSL.

Практична робота складалася з двох етапів. Перший етап – реалізація алгоритму атаки з адаптивно підібраними шифротекстами на мові С. Ця програма моделює процес взаємодії між валідним користувачем та криптоаналітиком. Зі сторони користувача випадковим чином генерується повідомлення m, виконується його зашифрування по алгоритму NTRU із використанням декількох сеансових ключів r_j. Потім вважається, що набір таких криптограм потрапив у розпорядження криптоаналітика. Програма починає відновлення коефіцієнтів «засліплюючого» полінома r_0 та обчислює згідно із (3.4) повідомлення-кандидат m'. На останньому кроці проводиться порівняння вихідного повідомлення m та результату атаки m', виводиться повідомлення про їх спів падіння чи неспівпадіння.

Другий етап полягав у визначенні залежності між ймовірністю неправильного відновлення відкритого тексту $P_{пом}$ та кількістю криптограм спеціального виду C, які доступні зловмиснику. З цією метою на основі програми, яка реалізує атаку, було створено тест, який

для кожного значення C із заданого діапазону проводить серію спроб здійснення атаки й підраховує кількість невдалих результатів. Потім обчислюється ймовірність помилки, яка дорівнює відношенню кількості невдалих результатів до кількості усіх спроб проведення атаки для заданого C. Виходом програми тестування є набір пар $\{C, P_{пом}\}$.

Запуск програми тестування здійснюється із командного рядка. Перед запуском необхідно явно вказати місце розташування динамічної бібліотеки *libntru.so*, якщо вона не була раніше скопійована в один із стандартних каталогів, що зберігають бібліотеки (/lib, /usr/lib в ОС Linux). Дана процедура виконується шляхом встановлення змінної оточення LD_LIBRARY_PATH:

$ export LD_LIBRARY_PATH=”місце_розташування_бібліотеки”.

Програма тестування має повноцінний інтерфейс командного рядка, який дозволяє задати всі основні параметри атаки. Формат запуску програми:

$ blindings_test [-N max_power] [-f num_ones] [-m mess_num_ones] [-b] [-C ct_num] [-e num_of_test_iters] [-h].

Більш детальну інформацію щодо призначення опцій програми можна отримати, задавши при її виклику ключ –h.

```
ander@ander-laptop:~/libntru$ export LD_LIBRARY_PATH="."
ander@ander-laptop:~/libntru$ ./blindings_test -N 1087 -f 120 -m 120 -C 7 -e 100
 -b
----- Starting test -----
Number of experiments: 100
Number of fails: 3
Failure probability: 0.030000
----- End of testing -----
ander@ander-laptop:~/libntru$
```

Рис. 1. Приклад запуску тесту (ОС Linux)

Результати роботи програми:

– Number of experiments – кількість проведених експериментів (визначається параметром -e);

– Number of fails – кількість невдалих результатів експериментів;

– Failure probability – ймовірність помилки відновлення вихідного повідомлення.

Результати досліджень

Дослідження проводились з метою визначення мінімальної кількості криптограм, при наявності якої ймовірність неправильного відновлення повідомлення прямує до нуля. Тестування проводилось для наборів параметрів ees401ep1, ees449ep1, ees677ep1, ees1087ep1 із стандарту [2], які є представниками всіх можливих рівнів стійкості (112, 128, 192 та 256 біт відповідно). Ймовірність обчислювалась як для бінарного, так і для тернарного представлень сеансових ключів. Такий вибір методики тестування обумовлений необхідністю визначення загальносистемних параметрів, від яких може залежати ймовірність невдачі, а також тим фактом, що в теорії для відновлення коефіцієнтів, які мають значення {0, 1} за їх різницями знадобиться менша кількість вибраних криптограм (див. п. 3.1).

За отриманими результатами було побудовано графіки залежності ймовірності помилкового відновлення повідомлення $P_{пом}$ від кількості шифротекстів C для бінарного й тернарного випадків (рис. 2, 3).

Експериментальні дані підтвердили теоретичне припущення про те, що для успішного відновлення бінарних коефіцієнтів сеансового ключа криптоаналітику необхідно одержати меншу кількість криптограм спеціального вигляду. Наприклад, для набору параметрів ees1087ep1 (рівень стійкості 256 біт) ймовірність помилки відновлення бінарних коефіцієнтів наближається до нуля вже при кількості криптограм, рівній 10, тоді як для тернарних поліномів цей показник наближається до 90.

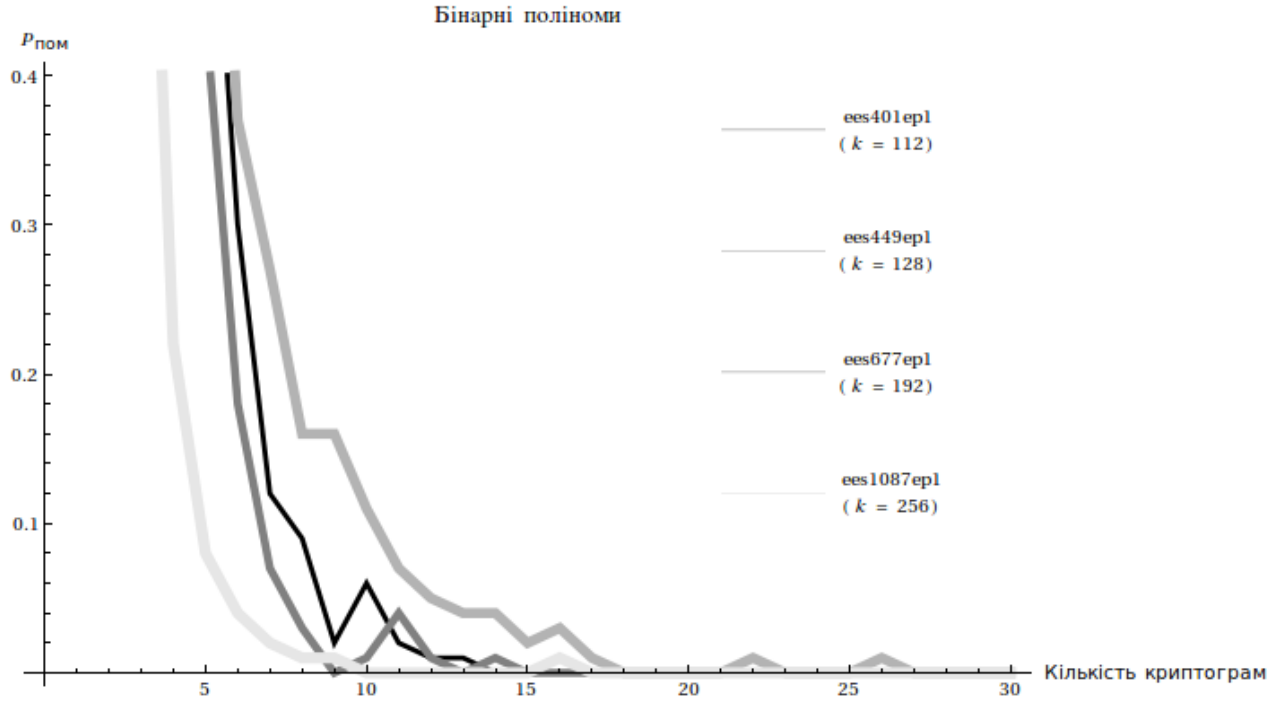

Рис. 2. Залежність ймовірності невдалого результату атаки від кількості шифротекстів (бінарні поліноми)

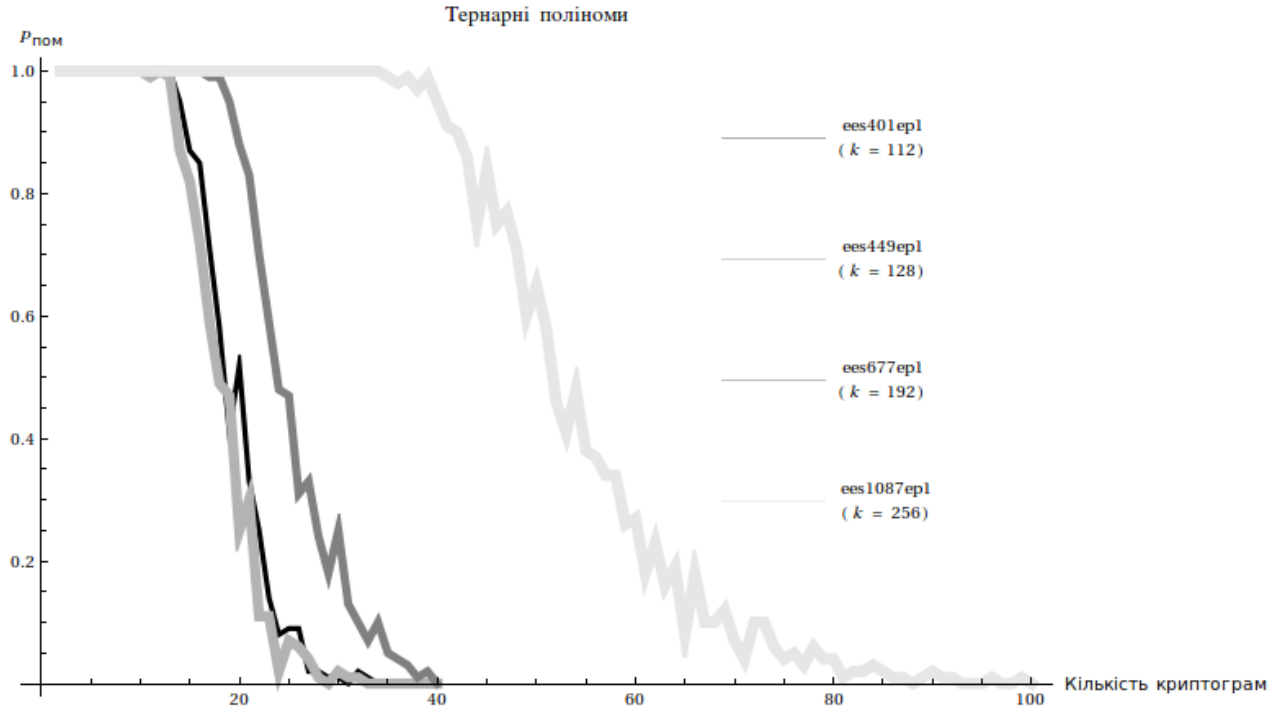

Рис. 3. Залежність ймовірності невдалого результату атаки від кількості шифротекстів (тернарні поліноми)

Було також виявлено, що у випадку бінарних коефіцієнтів мінімальна кількість шифротекстів, при якій $P_{пом} \approx 0$, залежить тільки від кількості одиниць d_f в сеансовому ключі r. Максимального значення вона досягає для набору параметрів ees667ep1, де $d_f = 157$, мінімального – для наборів ees401ep1 и ees1087ep1 (113 и 120 одиниць відповідно). Параметр N,

який визначає рівень стійкості, для бінарних коефіцієнтів практично не впливає на ймовірність успішної реалізації атаки, а обумовлює лише незначне збільшення часу її виконання із зростанням N (через обчислення різниць векторів, які мають більшу кількість коефіцієнтів).

Для тернарних коефіцієнтів, на відміну від двійкових, чітко відстежується зв'язок між рівнем стійкості параметрів та величиною N (див. рис. 3). Збільшення N поряд із незначним зростанням часу атаки призводить до пропорційного збільшення кількості криптограм, необхідних для їх успішної реалізації.

Рекомендації із захисту від атаки з адаптивно підібраними шифротекстами

Єдиний спосіб убезпечити поточну версію алгоритму НШ NTRU від такої атаки, який гарантовано не вплине на стійкість до інших видів атак, – не пересилати те ж саме повідомлення, зашифроване на одному й тому ж відкритому ключі, більш ніж один раз. Для стандарту ANSI X9.98 ця вимога повинна виконуватись «апріорі», тому що вважається, що алгоритм NTRU буде використовуватись виключно для безпечного обміну ключами симетричного шифрування, які є випадковими. В цьому випадку ймовірність повтору повідомлення m повністю визначається властивостями джерела ключових даних для симетричних криптоалгоритмів.

Ще один метод захисту від можливості проведення саме цієї атаки – використання одного й того ж «засліплюючого» полінома r для пари {повідомлення, відкритий ключ}. Згідно із стандартом [2] r є результуючим значенням геш-функції

$$r(x) = H(m(x) \,||\, b), \tag{5}$$

де b – сіль – бітова послідовність, яка повинна формуватися випадковим чином.

Для забезпечення константного значення r необхідно, щоб сіль b стала не випадковою, а детермінованою. В такому випадку будь-яке повідомлення m буде при повторному шифруванні відображатися в ту ж саму криптограму e. Ця умова несумісна із семантичною стійкістю та стійкістю від атаки із вибраними шифротекстами (зловмисник не повинен мати можливість взнати будь-яку інформацію про відкритий текст, якщо він має в розпорядженні лише криптограму). Ми прийшли до протиріччя: захист від атаки з підібраними шифротекстами таким методом зробить шифр вразливим до цілого ряду інших атак.

Висновки

Стійкість до атаки з адаптивно підібраними шифротекстами можлива тільки тоді, коли для кожного повідомлення m генерується унікальне, єдине значення полінома r. У зв'язку з тим, що r є результатом використання геш-функції до конкатенації повідомлень m із сіллю b, ймовірність того, що для двох різних пар (m, b) буде згенеровано одне й те ж значення r, повністю визначається ймовірністю колізії значень геш-функції. Повторне шифрування повідомлення на одному й тому ж відкритому ключі неприпустиме. В іншому випадку шифр є абсолютно нестійким до атаки із адаптивно підібраними шифротекстами. Тестування програмної моделі атаки показало, що в середньому достатньо перехопити 30-40 таких криптограм, щоб, не знаючи особистий ключ, відновити початкове повідомлення.

Таким чином, направлений шифр NTRU, як і всі інші класи асиметричних алгоритмів, підпорядковується фундаментальному правилу побудови криптосистем, яке забороняє використання НШ для безпосереднього шифрування повідомлень.

Список літератури: 1. *IEEE* 1363.1 IEEE Standard Specification for Public-Key Cryptographic Techniques Based on Hard Problems over Lattices [Електронний ресурс]. – 2008. 2. *American* National Standard for Financial Services ANSI X9.98-2010 Lattice Based Polynomial Public Key Establishment Algorithm for the Financial Services Industry [Електронний ресурс]. – 2010. 3. *Jens Hermans, Frederik Vercauteren, Bart Preneel.* Speed records for NTRU [Электронный ресурс]. – Режим доступа: www / URL: https://www.securityinnovation.com/uploads/ntru_speed_benchmark_research.pdf. – 2010.

Харківський національний
університет радіоелектроніки

Надійшла до редколегії 15.09.2012