УДК 621. 391:519.2:519.7

АТАКА СПЕЦІАЛЬНОГО ВИДУ НА NTRU

М.Ф. БОНДАРЕНКО, Д.С. БАЛАГУРА, Д.В. ІВАНЕНКО

Наводиться порядок шифрування та розшифрування алгоритму NTRU. Розглядається атака спеціального виду за часом, демонструється можливість отримати знання про секретний ключ при відомій кількості звернень до геш-функції.

Ключові слова: алгоритм NTRU, атака спеціального виду, шифрування, розшифрування, шифр-текст, геш-функція, поліном.

ВСТУП

Алгоритм NTRUEncrypt(NTRU) був розроблений ще у середині 90-х математиками Jeffrey-Hoffstein, JillPipher и Joseph H. Silverman [1]. Він був заснований на решітчастій криптосистемі, яка у майбутньому могла би протидіяти атакам з використанням квантових комп'ютерів. Але у запропонованому на той час алгоритмі були суттєві недоліки: NTRU програвав у швидкодії та у продуктивності RSA [2] та криптосистемам на еліптичних кривих [2], тому одразу не отримав широкого розповсюдження. На сьогоднішній день основні недоліки, за твердженнями спеціалістів RSALabs та за результатами незалежних досліджень усунені. Фактом визнання якісних показників є прийняття NTRU технологічним стандартом для фінансових транзакцій та визнання Accredited Standards Committee X9 NTRU найшвидшим алгоритм асиметричного шифрування.

Зважаючи на швидке розповсюдження та визнання NTRU постає питання глибшого вивчення аспектів стійкості протоколів до будь-яких атак. Метою статті є аналіз можливості застосування відносно NTRU деякої атаки спеціального виду, а також обґрунтування пропозицій щодо попередження атаки спеціального виду за часом

1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА АЛГОРИТМУ NTRU

1.1. Зашифрування

У зашифруванні використовується дві геш-функції. Позначимо їх G та H, на практиці зазвичай використовується SHA-1 або SHA-256. Зашифрування здійснюється за таким алгоритмом:

Крок 1: Візьмемо відкритий текст M та переведемо його у β_N(бінарний поліном):

$$M \in \beta_N,$$

$$\beta_N := \{d_1 X^{N-1} + d_2 X^{N-2} + \ldots + d_{N-1} X^1 + d_N 1 \in R, d_i = \{0,1\}\}$$

$$\beta_N(d) := \{d_1 X^{N-1} + \ldots + d_N 1 \in R, d_i = \{0,1\} \text{ та } \sum d_i = d\} \quad (8)$$

Крок 2: за допомогою геш-функції G рандомізуємо текст M додаючи до нього бінарний поліном (d_r):

$$r = G(M) \in \beta_N(d_r). \quad (1)$$

Крок 3: на цьому етапі потрібно розрахувати замасковане повідомлення m':

$$m' = M \oplus H(r \cdot h \bmod q). \quad (2)$$

Крок 4: на останньому етапі отримуємо зашифрований текст e:

$$e = (r \cdot h + m') \bmod q. \quad (3)$$

1.2. Розшифрування

На етапі розшифрування відбувається не тільки розшифрування, але й перевірка справжності отриманої пари (m', e). Алгоритм розшифрування також використовує дві геш-функції G та H.

Крок 1: Алгоритм розшифрування відновлює отримане m':

$$m' = ((f \cdot e \bmod q) \bmod p) \cdot f_p^{-1} \bmod p. \quad (4)$$

Крок 2: На цьому етапі відновлюється текст M:

$$M = m' \oplus H(e - m' \bmod q). \quad (5)$$

Крок 3:Після визначення M та m', необхідно відтворити r:

$$r = G(M). \quad (6)$$

Крок 4: на останньому етапі перевіряємо чи (m', e) відповідає парі NTRU:

$$e = r \cdot h + m' \bmod q. \quad (7)$$

2. АТАКА ЗА ЧАСОМ НА NTRU

Нижче розглядається атака спеціального виду на особистий ключ алгоритму NTRU - атака за часом, яка базується на підрахуванні кількості звернень до геш-функції. Використовуючи слабкі сторони структури алгоритму, застосовуючи вже визначені спеціальні канали, які отримали назву атаки спеціального виду, можливо найти таємну інформацію.

2.1. Атака за часом на шифр-текст

При зашифруванні використовують сеансовий ключ r, який додавався до нашого повідомлення M. Для відтворення r потрібно підрахувати кількість звернень до геш-функції пари (m', e) відповідного повідомлення. Кількість цих звернень може бути різною для різних повідомлень. Тобто кількість звернень до геш-функції визначається для кожного повідомлення окремо. Зловмисник повинен виміряти кількість звернень до геш-функції підчас розшифрування шифр-тексту e. Існує таке число K, що є кількістю звернень до геш-функції, необхідних для визначення r, яке може дорівнюватися або k або k+1. На

наступному етапі потрібно визначити вихідне $r(m', e)$ для відповідної пари (m', e) кожного повідомлення з алгоритму розшифрування:

$$r(m',e) = G((m' + H(e - m' \bmod q)) \bmod 2. \quad (8)$$

Визначимо, що $\beta_N(m',e) \in \{0,1\}$. *Нуль* встановлюється у випадку коли потрібно більше ніж K геш значень при створенні $r(m', e)$, та *одиниця*, якщо потрібно K геш значень. За таких умов отримуємо $(X^i m', X^i e)$ при $i = 0,1...$. Зважаючи на вищезазначене, можна дати повне визначення функції атаки за часом. Це бінарний вектор, який має наступний вид:

$$T(m',e) = ((\beta(m',e), \beta(Xm', Xe), \beta(X^2 m', X^2 e),... \\ ..., \beta(X^{N-1} m', X^{N-1} e)). \quad (9)$$

Функція атаки за часом надасть нам кількість звернень до геш-функції потрібних для кожної пари (m', e). Вище було зазначено, що кількість звернень для різних повідомлень різна. Тому можна визначити , з якою імовірністю дві пари матимуть однакову функцію атаки за часом та відповідно і кількість звернень до геш-функцій. Припустимо, що P імовірність того, що для випадкової пари потрібно K звернень. Також припустимо, що $1-P$ імовірність іншої випадкової пари (m_2', e_2), якій потрібно як мінімум K звернень. Якщо P досить велика, то імовірність того, що дві пари матимуть ту ж функцію атаки за часом досить мала. Точніше імовірність може бути представлена наступною формулою:

$$\Pr ob(T(m_1', e_1') = T(m_2', e_2)) = (1 - 2P + 2p^2)^N . \quad (10)$$

Формула (10) може бути представлена у іншому вигляді. Представимо функцію атаки за часом у вигляді бінарного вектору довжиною N. Нехай P імовірність того що в i-й позиції бінарного вектору ми отримаємо 0. Ми отримаємо імовірність $1-P$ того що в i-й позиції буде 1. Тоді імовірність того, що в першій позиції двох випадкових функцій атак за часом буде таж сама:

$$\Pr ob(both = 0) + \Pr ob(both + 1) = p^2 + (1 - P)^2 . \quad (11)$$

Для того, щоб дві функції атаки за часом були ідентичні, вони повинні мати в усіх N позиціях деяке число (0 або 1). Таким чином, ми маємо:

$$\Pr ob(T(m_1', e_1') = T(m_2', e_2)) = (1 - 2P + 2p^2)^N . \quad (12)$$

Функція атаки за часом – це представлення ключа NTRU, за допомогою атаки за часом. Далі розглянемо цю атаку більш детально.

2.2. Атака за часом, яка базується на різній кількості звернень до геш-функції

Розглянемо ситуацію, коли є дві сторони *А* та *Б*, вони між собою обмінюються інформацією, припустимо, що *Б* – зловмисник, який намагається отримати секретний ключ іншої сторони. Щоб виконати своє завдання він може використовувати атаку спеціального виду за часом. Зловмисник вибирає множину ε (шифр-текст), яка є набором поліномів за *modq*. Далі *Б* повинен вибрати таке повідомлення M, щоб воно складалося з множини повідомлень створених A, та мало наступний вигляд:

$$\{((f \cdot e \bmod q) \bmod 2 \cdot (f^{-1} \bmod 2) : e \in \varepsilon\} .$$

Більш того, припустимо, що імовірність того, що наведене повідомлення створене стороною A буде знайдено у множині M дуже велика. Сторона *Б* проводить атаку та створює таблицю з функцією атаки для кожної пари $M \times \varepsilon$. Іншими словами він створює список-пошук бінарного вектору:

$$(T(m',e)) : m \in M,\ e \in \varepsilon .$$

Атака починається, коли *Б* посилає *А* деяке випадкове $e \in \varepsilon$. Коли шифр-текст буде відправлений, *Б* починає записувати як довго *А* буде розшифровувати цей шифр-текст. Якщо навіть відправлено підроблений шифр-текст, який у випадку успішної перевірки буде відхилено, це не вплине на головну задачу зловмисника, тому що зловмисник зацікавлений у часовій інформації. Тобто інформації, яка пов'язана з можливістю виміряти кількість звернень до геш-функції. Іншими словами він буде знати значення $m'(e)$, яке може бути представлене у вигляді:

$$((f \cdot e \bmod q) \bmod 2) \cdot (f^{-1} \bmod 2) . \quad (13)$$

Стороні *Б* не відомо значення $m'(e)$. Він знає тільки значення $\beta(m'(e), e)$, яке буде використовуватися після того, як буде порівняно зі значеннями отриманими від *А*, попередньо вирахуваними та записаними у таблицю, яка була створена *Б*. Для цього *Б* потрібно $N-1$ записів, які будуть зроблені функцією атаки. Ці значення він отримає після відправлення одного за іншим наступних поліномів:

$$Xe, X^2 e, X^3 e, ..., X^{N-1} e . \quad (14)$$

Після відправки шифр-тексту, зловмисник отримає значення $\beta(m'(X^i e), X^i e)$. Візьмемо випадкове $m'(X^i e)$. Нехай ми знаємо, що чому дорівнює $m'(e)$, тоді можна застосувати це також й до виразу $m'(X^i e)$:

$$m'(X^i e) = \left(\left(f \cdot X^i e \bmod q \right) \bmod 2 \right) \cdot \left(f^{-1} \bmod 2 \right),$$

оскільки усі X дорівнюють 0 чи 1, то ми можемо взяти таке X^i, що отримаємо

$$X^i \cdot ((f \cdot e \bmod q) \bmod 2) \cdot \left(f^{-1} \bmod 2 \right),$$

що насправді буде $X^i m'(e)$.

Сторона Б, знає функцію атаки $T(m'(e), e)$ від $(m'(e), e)$. На наступному етапі повинен відбутись пошук у сформованому списку, у якому з високою імовірністю знаходиться невелика кількість можливих пар із отриманих стороною *Б* пар. Насправді він має щось більше, наприклад *Б* має два полінома e, m' де *А* розшифровує e взявши другий, поліном m'. З цього зловмисник вираховує тільки $m' \cdot f \equiv (f \cdot e \bmod q) \bmod 2$, причому e та m' він знає.

В наступному підрозділі наводиться, як зловмисник може за допомогою наведеної атаки

отримати перевагу щодо визначення особистого(секретного) ключа *A*. Далі, використання секретного ключа *f* буде залежить від значення *e*. Наприклад, якщо елементи ε складаються із поліномів з ненульовими коефіцієнтами, то вираз:

$$m' \cdot f \equiv \left(f \cdot e \bmod q\right) \bmod 2$$

може дати інформацію відносно відстані між ненульовими коефіцієнтами *f*. В наступному підрозділі, описується специфічна множина ε у використанні атаки за часом на практиці, коли ключ *f* має форму $f = 1 + 2F$.

2.3. Атака за часом у випадку $f=1+2f$

З наведеного вище відомо, що у більшості випадків параметр *p* дорівнює 2. Це значить, що ***q*** непарне, і далі секретний ключ $f = 1 + 2F$ для деякого бінарного поліному $F \in \beta_N(d_F)$. Після підрахування, отримуємо, що $F = \sum F_i X^j$, де $F_j \in \{0,1\}$. Визначимо новий параметр λ = [q/4]. В нашому випадку відправною точкою буде, коли Б відправить А шифр-текст *e,* але у такому випадку ми будемо мати

$$\left\{\lambda + \lambda X^i : 1 \le i < N\right\}.$$

Це значить, що випадкове значення *e* теж поліном, але в цьому випадку з двома коефіцієнтами рівними λ та всі іншими коефіцієнтами –0.

Стисло розглянемо, як *Б* проводить атаку на секретний ключ А:

1. Вибираємо змінну δ
2. Нехай

$$\varepsilon = \left\{e_i = \lambda + \lambda X^i : 0 \le i \le (n-1/2)\right\} \text{та}$$

$$M = \beta_N (0 < d \le \delta).$$

Розрахуємо та збережемо усі значення функції атаки $T(m', e)$.

3. Від А передано $X^j e_i$, де j = 0, 1, ·, N – 1. Зробимо деяке число розшифрування, заміряючи це функцією $T(m'(e_i), e_i)$.

4. Далі зробимо пошук можливих кандидатів.

5. Використовуючи отриманні значення $m'(e_i)$, відновлюємо *F*.

В цей момент необхідно знайти можливе значення $m'(e_i)$, що отримане на кроці 2, тобто (1). З цією метою здійснимо розшифрування, як це робить *A*. З початку *A* вираховує:

$$a = f \cdot e \bmod q \equiv (1 + 2F) \cdot \left(\lambda + \lambda X^i\right) (\bmod q) \equiv$$

$$\equiv \lambda + \lambda X^i + \sum_{j=0}^{N-1} 2\lambda \left(F_j + F_{j-i}\right) X^i.$$

Нехай для *j*-го коефіцієнту *a*, ми маємо наступні функції:

$$a_j = \begin{cases} \lambda(1 + 2F_o + F_{-i} \bmod q) \ \textit{if } j = 0, \\ \lambda(1 + 2F_i + 2F_o) \bmod q \ \textit{if } j = i, \\ \lambda(2F_i + 2F_{j-i}) \bmod q \ \textit{if } j \ne 0, i. \end{cases}$$

Розглядати функцію вище ми повинні чітко, так як λ =2[*q*/8], це значення λ не набагато більше ніж *q*/4, и тому права сторона цього виразу приймає значення менше, ніж *q*–1. Тому не потрібно зменшувати $a \,(mod\, q)$, за винятком випадку $F_i = F_{j-i} = 1$. Застосуємо зменшення після якого отримуємо нові значення для a_j:

$$a_j = \begin{cases} \lambda, 2 \textit{ or } 3\lambda & \textit{if } F_j = 0 \textit{ or } F_{j-i} = 0, \\ 4\lambda - q \textit{ or } 5\lambda - q & \textit{if } F_j = 0 \textit{ or } F_{j-i} = 0. \end{cases}$$

Після зниження $a \bmod 2$, отримуємо значення 0 та 1, і більш того λ – парне, а *q* – непарне, після шага зниження отримаємо:

$$a_j \bmod 2 = \begin{cases} 0 & \textit{if } F_j = 0 \textit{ or } F_{j-i} = 0, \\ 1 & \textit{if } F_j = F_{j-i} = 1. \end{cases}$$

Тепер можемо визначити чітко $m'(e_i)$:

$$m'(e) = \sum_{j=0}^{N-1} \begin{pmatrix} 1 & F_i = F_{j-i} = 1 \\ 0 & otherwise \end{pmatrix} X^j. \qquad (15)$$

у полі якого маємо часткову інформацією про *F*:

$$F(e_i) = \sum_{j=0}^{N-1} \begin{pmatrix} 1 & \textit{if } m'(e_i)_j = 1 \\ & \textit{or } m'(e_i)_{i+j} = 1 \\ 0 & \textit{if } m'(e_i)_{j-i} = 1 \textit{ and } m'(e_i)_j \ne 1 \\ & \textit{or } m'(e_i)_{j+2i} = 1 \textit{ and } m'(e_i)_{j+i} \ne 1 \\ ? & otherwise \end{pmatrix} \qquad (16)$$

2.4. Обґрунтування вибору

Початкова множина ε відносно несильно знижена попередніми розрахунками. Функція атаки покликана повідомити зловмиснику, що він ідентифікує $m'(e_i)$, яке відповідає e_i в його базі. Проте, якщо нетривіально можливо те, що функція атаки – унікальна, то Б зможе вимагатиме перевірку, що він коректно визначив e_i. Відмітимо, якщо $e_i = \lambda + \lambda X^i$ і розшифруємо m', то це привело до альтернативної форми:

$$e_i^* = (\lambda + 2) + \lambda X^i, or\, \lambda + \lambda X^i \, or \ldots., \qquad (17)$$

або множина з $\lambda_1 + \lambda_2 X^i$ з цілими λ_1 та λ_2 у діапазоні $q/4$ і задовольняє $\lambda_1 + \lambda_2 > q/2$. Тому Б вибирає один з можливих e_i^* підраховує $T\left(m', e_i^*\right)$ для повідомлень m' так, щоб він вважав, що проводить розшифрування дійсного e_i^*, та що для представленого e_i^* знаходить кількість звернень за допомогою функції атаки. Якщо функція атаки вимірить та вирахує «подібну пару», то зловмисник сформулює припущення про m'. Проте дійсна пара функції атаки це простий збіг.

2.5. Практичні аспекти атаки за часом для $f=1+2F$

Далі на практиці буде показано як виконується аналогічна атака для секретного ключа виду $f=1+2F$, для цього буде використовуватися певний набір параметрів. Почнемо з опису використання геш-фукнції при вирахуванні *r*, та потім вирахуємо імовірність того, скільки раз цей процес використовує геш. Значення *r* (бінарного поліному з простих d_r) обчислюється з геш-функцій, через повторне використання деяких версій SHA. В результаті:

1. Маємо значення C, що задовольняє умові $2^c > N$. Також нехай $b = [c/8]$ та $n = [2^c/N]$, причому b це найменше ціле число таке, що b біт міститься принаймні у c біт. Таким чином n найменше множина N, що менше ніж $2c$.

2. Будемо використовувати SHA та поділимо вихід на найменші проміжки довжини b. В межах кожного такого проміжку, зберігається нижній регістр c біт та відкидається верхній регістр $8b–c$ біт. Конвертуємо нижній регістр c біт у прості числа $i_1, i_2, \cdot\cdot, i_t$.. На виході SHA складається з tb біт, де t просте число.

3. Створемо список $j_1, j_2, \cdot\ldots$ шляхом перебору i-х значень на кроці2. Якщо $i<n$ та $i \bmod N$ не знаходиться у списку, тто необхідно $i \bmod N$ додати до списку, в іншому випадку відкинути i. Продовжуємо список поки він містить j значення множини d_r. Якщо будь-яка точка не має i значення, то викликається SHA та створюється додаткове значення i як на кроці 2.

У порядку вирахування r, у алгоритмі описано необхідність створення списку $d_{r,}$, яке б задовольняло $0 \le i < N$. Кожного разу алгоритм використовує геш, він отримує t число, яке задовольняє $0 \le i < 2^c$. Звідси, імовірність того, що достатньо визвати SHA s раз, дорівнює імовірності, що випадкових st раз у межах $[0, 2^c)$ міститься принаймні значення d_r у діапазоні $[0, n)$, значення якої визначено модулем N. Звідси

$$\Pr ob\left(\textit{потрібних звернень } SHA\right) =$$

$$= \Pr ob\left(\begin{array}{c}\textit{випадкові } st \textit{ вибрані у диапазоні } [0,2^c) \\ \textit{включаючи не менше } d_r \textit{ у межах } [0,n) \\ \textit{узяті за модулем } N\end{array}\right).$$

Перша імовірність ближче до 0,5 це значить, що існує велика імовірність того що функція атаки унікальна.

ВИСНОВОК

В цій статті було показано атаку спеціального виду за часом, яка може розкрити секретний ключ NTRU. Ця атака можлива завдяки тому, що у розшифруванні різних шифр-текстів використовується різна кількість звернення до геш-функції. Хоча стаття було присвячена атаці на секретний ключ вид у $f = 1 + 2F$, це справедливо запропонувати і для ключів загального виду. Цей метод оснований на розшифруванні кількості звернень до геш-функції для кожного шифр-тексту. Тобто підраховується кількість звернень до геш-функції для кожного можливого шифр-тексту. Візьмемо максимальне значення та назвемо його $K_{max.}$ У випадку випадкового шифр-тексту, коли число звернень буде менше ніж K_{max}, то знадобитися додаткове використання геш-функції. Якщо шифр-тексту потрібно K – звернень до геш-функції, то $K_{max}–K$ буде потрібно додаткових звернень. Тому кількість звернень повинна бути однаковою для усіх шифр-текстів для попередження цієї атаки. Цей метод отримання кількості звернень на одне повідомлення, також по іншому можна назвати «доповненням». Для того, щоб попередити атаку спеціального виду потрібно збільшити стійкість самого алгоритму від атаки за часом, щоб кожен біт додавання впливав на кожен біт кількості звернень.

Література

[1] *J. Hoffstein, J. Pipher, J.H. Silverman* / NTRU: A new high speed public key cryptosystem, Algorithmic Number Theory (ANTS III)//, Portland, OR, June 1998, Lecture Notes in Computer Science 1423, J.P. Buhler (ed.), Springer-Verlag, Berlin, 1998, 267–288.

[2] *Jens Hermans, Frederik Vercauteren, Bart Preneel* / Speed Records for NTRU. // Topics in Cryptology - CT-RSA 2010, The Cryptographers Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010. Proceedings. Volume 5985 of Lecture Notes in Computer Science, pages 73-88, Springer, 2010.

[3] *Joseph H. Silverman, William Whyte* / NTRU Cryptosystems Technical Report Report 021, Version 1Timing Attacks on NTRUEncryption via Variation in the Number of Hash Calls // http://grouper.ieee.org/groups/1363/lattPK/ submissions /021NTRUTechReport-sha-timing.pdf

Надійшла до редколегії 20.03.2012

Бондаренко Михайло Федорович, член-кореспондент НАН України, лауреат державної премії України, доктор технічних наук, професор, ректор Харківського національного університету радіоелектроніки.

Балагура Дмитро Сергійович, фото та відомості про автора див. на с. 199.

Іваненко Дмитро Вікторович, аспірант кафедри БІТ ХНУРЕ. Область наукових інтересів: інформаційні технології, захист інформації, методи та засоби автентифікації.

УДК 621. 391:519.2:519.7

Атака специального вида на NTRU/ М.Ф. Бондаренко, Д.С. Балагура, Д.В. Иваненко // Прикладная радиоэлектроника: науч.-техн. журнал. – 2012. – Том 11. № 2. – С. 216–219.

Приводиться порядок шифрования и расшифрования алгоритма NTRU. Рассматривается атака специального вида по времени, демонстрируется возможность получить знание секретного ключа NTRU при известном количестве обращений до хеш-функции.

Ключевые слова: алгоритм NTRU, атака специального вида, шифрование, расшифрование, шифр-текст, хеш-функция, полином.

Библиогр.: 3 назв.

UDC 621. 391:519.2:519.7

Side channel attack on NTRU / M.F. Bondarenko, D.S. Balagura, D.V. Ivanenko // Applied Radio Electronics: Sci. Journ. – 2012. Vol. 11. № 2. – P. 216–219.

The paper considers the order of encryption and decryption of the NTRU algorithm as well as it reviews the side channel attack and shows the possibility of obtaining knowledge about the secret key with a certain amount of hash-function calls.

Keywords: NTRU algorithm, side channel attack, encryption, decryption, cipher-text, hash function, polynomial.

Ref.: 3 items.