«Системні технології» 2 (139) 2022 «System technologies»
DOI 10.34185/1562-9945-2-139-2022-01
УДК 004.056.53

Я.В. Іванчук, Ю.В. Горобець, К.О. Коваль

МЕТОД ПІДВИЩЕННЯ СТЕПЕНЮ ЗАХИСТУ ШИФРУВАННЯ ПОВІДОМЛЕНЬ НА ОСНОВІ АЛГОРИТМУ ЗА СТАЛОЮ СКЛАДОВОЮ ПО ЧАСУ

Аннотація. У статті розглянуто ураженість криптосистем типу DSA та ECDSA до атаки на основі аналізу змінної часу підписання повідомлення. Розроблена математична модель для перевірки даного типу ураженості, на основі решіткових атак. Показано, що при наявності достатньої кількості сигнатур з однаковим часом підписання можна ідентифікувати наявність загальних бітів ефемерних ключів, що дозволить відновити приватний ключ відправника. Визначено, що причиною ураженості є відсутність виконання операції обчислення оберненого модуля змінного по часу, що надає дані ефемерного ключа зловмиснику. Для вирішення поставленої задачі запропоновано розширений евклідовий алгоритм обчислення оберненого модуля за сталий час.
Ключові слова: тимчасова атака, решітка, зворотний модуль, постійний час, алгоритм, криптосистема.

Постановка проблеми. Криптосистема – це реалізація криптографічних методів та їх інфраструктури для надання послуг інформаційної безпеки [1]. Криптосистеми перетворюють вихідні дані в нечитабельну форму, використовуючи ключі шифрування і розшифрування, що у свою чергу дозволяє забезпечувати конфіденційність інформації. В основі будь-якої системи шифрування є наявність інформації обміну даними про відкритий текст тільки у відправника й отримувача.

У залежності від способу шифрування та дешифрування інформації, криптосистеми поділяються на два типи [1]:

1) Симетрична система шифрування базується на використанні однотипного ключа, інформація про використання якого узгоджується сторонами до початку сесії обміну даними. Прикладами даного типу систем шифрування є: DES, Triple-DES, BLOWFISH, IDEA, AES [2]. Основним недоліком даного типу системи шифрування є наявність складності керування ключами у великих мережах.

2) Асиметрична система шифрування (метод доступу на основі відкритого ключа) відкритого тексту використовує різні типи ключів, які пов'язані між со-

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

бою певною математичною залежністю. Даній підхід дозволяє отримати оригінал відкритого тексту шляхом використання обернених функцій математичних операцій. Основними представниками даного типу систем шифрування є DSA, DSS, RSA, ECDSA, El-Gamal [3]. Основними недоліками даного типу системи шифрування є наявність інформації про відкритий ключ відправника і отримувача при створенні зашифрованого повідомлення; рівень безпеки систем шифрування пропорційний рівню складності виконання операцій на основі різних математичних задач.

Для деяких систем шифрування, використовується додатковий ключ (ефермний ключ), який потрібен для додавання більшої складності в алгоритм шифрування, основною вимогою до нього, ефемерний ключ повинен використовуватися тільки один раз.

Для систем шифрування DSA, ElGamal, ECDSA – при створенні зашифрованих повідомлень додатково використовується ефемерний ключ, що у свою чергу призводить до атаки стороннього каналу мережі, аналізу часу створення сигнатури, скидання та перезавантаження даних ефемерного ключа [4], атак на генератори псевдорандомних чисел [5]. Усі ці типи криптоатаки призводять до перехоплення окремих бітів даних ефемерного ключа або отримання часу створення сигнатури, що надає третій стороні доступ до зашифрованих повідомлень, а це у свою чергу призводить до компроментації приватного ключа відправника.

Аналіз останніх досліджень і публікацій. У статтях [7, 8] показано криптоатаку на систему шифрування типу DSA, у якій за допомогою методів вирішення задач з теорії чисел, а саме проблему прихованих чисел, для розв’язання якої використовується метод коротких векторів або метод близьких векторів, можна знайти приватний ключ відправника, зібравши з n-кількості зашифрованих повідомлень числове значення менш або/і більш значущих бітів ефемерного ключа, використовуючи криптоатаки на кеш процессора, але дана криптоатака не є можливою, якщо числове значення цих бітів, є невідомим.

Автор праці [8] розглядає підхід для запобіганням криптоатакам по побічним каналам обміну даними у якому використовується технологія криптоопераций TPM [8]. Недоліком такого підходу є його не універсальність із-за апаратної реалізації, що знижує загальний рівень криптостійкості до зовнішніх несанкціонованих втручань. Тому, актуальним є підвищення рівня стійкості криптосистем до аналізу часу створення зашифрованого повідомлення, що дозво-

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

лить підвищити загальний рівень безпеки шифрування даних при обміну інформацією.

Мета дослідження. Підвищення степеню захисту шифрування даних в мережах обміну повідомлень на основі розробленого ефективного алгоритму створення сигнатури підпису.

Для досягнення поставленої мети необхідно вирішити наступні задачі: визначити тип враженості криптосистем на основі аналізу часу підписання повідомлення, а також розробити ефективний метод захисту криптосистем до атак за часом підписання повідомлення, що дозволить уникнути несанкціонованого доступу до зашифрованої інформації.

Викладення основного матеріалу дослідження. Для ідентифікації можливого потенційно небезпечного виду криптоатаки на приватний ключ, що базується алгоритмі шифрування DSA (ECDSA), необхідно проаналізувати математичний опис цього алгоритму.

Для використання DSA алгоритму, відправник обирає два простих числа *p* та *q*, та *g*, де *p* – просте число розміром від 512 до 1024 бітів, з кроком в 64 бітів; *q* – просте число, яке у відповідності до стандартів Національного інституту стандартів та технологій (NIST) [9] розміру не менше 160 бітів, і відповідає рівності $2^{N-1} < q < 2^N$, у якій $N \in \{160, 224, 256\}$, а також $q \mid p-1$; *g* – генератор порядку *q* підгрупи *G* кінцевого поля $\mathbb{F}_p^{\times}$ абелевої групи [10]. Далі, відправник вибирає $a \in \{1, ..., q-1\}$ та обчислює $A = g^a \bmod p$. Публічний ключ відправника це (p, q, g, A) та приватний ключ *a*. Далі, відправник обирає хеш-функцію $h : \{0,1\}^* \to \{0, ..., q-1\}$, яка відображає повідомлення будь-якої довжини у відповідності до $\{1, \ldots, q-1\}$.

Щоб підписати повідомлення *m*, відправник обирає випадкове число $k \in \{1, ..., q-1\}$, яке і є ефемерним ключем і обчислює залежності:

$$r = (g^k \bmod p) \bmod q; \quad s = k^{-1}(h(m) + ar) \bmod q. \qquad (1)$$

Сигнатура повідомлення *m*, це пара чисел (*r, s*). Сигнатура (1) дійсна тоді і тільки тоді, коли виконується рівність:

$$r = ((g^{s^{-1}h(m) \bmod q} A^{s^{-1}r \bmod q}) \bmod p) \bmod q. \qquad (2)$$

Для використання ECDSA алгоритму, відправник обирає еліптичну криву *E* на кінцевому полі $\mathbb{F}_p$, точка $P \in E(\mathbb{F}_p)$ (відповідно до DSA це генератор) з простим порядком *q* розміром як мінімум 160 бітів. Відповідно відповідності до FIPS 186-3 бінарна довжина простого числа *p* має бути в множині

$\{160, 224, 256, 512\}$. Крім того, для деякого випадкового вибраного $a \in \{1, ..., q-1\}$ обчислюється $Q = aP$. Публічний ключ відправника це (E, p, q, P, Q) і приватний ключ *a*. Також, відправник обирає хеш функцію $h: \{0,1\}^* \to \{0, ..., q-1\}$. Щоб підписати повідомлення *m*, відправник обирає випадкове число $k \in \{1, ..., q-1\}$, яке і є ефемерним ключем і обчислює $kP = (x, y)$ (де *x* та *y* розглядаються як цілі числа у діапазоні $\{0, ..., p-1\}$). Далі, обчислюється сигнатура повідомлення *m*, а саме пара чисел (*r*, *s*):

$$r = x \bmod q; \quad s = k^{-1}(h(m) + ar) \bmod q. \tag{3}$$

Для верифікації повідомлення, отримувач обчислює:

$$u_1 = s^{-1}h(m) \bmod q, \quad u_2 = s^{-1}r \bmod q, \quad u_1P + u_2Q = (x_0, y_0). \tag{4}$$

Сигнатура (3) валідна тоді і тільки тоді, коли $r = x_0 \bmod p$. Безпека цих криптосистем опирається на припущення, щоб єдиний спосіб підробити підпис – це або відновити приватний ключ *a*, або підробити ефемерний ключ *k*. Таким чином, параметри цих криптосистем вибрані таким чином, щоб обчислення дискретних логарифмів є обчислювально нездійсненим для груп розміром в 256 бітів.

Для здійснення потенційно-небезпечної криптоатаки зловмисником, що призведе до відкриття приватного ключа відправника, можна записати кількість часу, який необхідний для створення кожної сигнатури, що в залежності від значення ефемерного ключа *k*, є різним, тобто для ефемерного ключа *k*, розміром 254 бітів буде меншим чи при стороненні сигнатуром, з ключем в 256 бітів, при цьому буде відомо, що значення більш значущих бітів (MSB) буде рівнем нулю. Припустимо, що зловмиснику вдалось зібрати *n* кількість зашифрованих повідомлень m_i (*i*=1, …, *n*) з відповідними сигнатурами (r_i, s_i), такими що час створення сигнатури є дуже близькими один між одними, тоді можна припустити, що всі відповідні ефемерні ключі k_i поділяють загалом бітів між більш значущими бітами незалежно від поточної кількості повідомлень *i*. Таким чином, вони будуть відповідати наступній залежності для всіх *i*=1, …, *n*:

$$k_i = k + 2^t \tilde{k}_i + 2^{t'} k', \tag{5}$$

де $0 \le k < 2^t, 0 \le k' < 2^{N-t'}, \delta = \mathrm{N} - t' + t, 0 \le k_i < 2^{N-\delta}$, *k* та *k´* загальні для всіх k_i.

На рисунку 1 зображена схема розміщення бітів у ефемерному ключі:

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

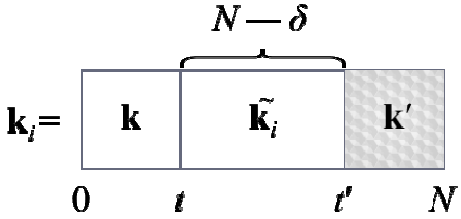

Рисунок 1 – Схема ефемерного ключа:
k – числове значення загальних більш значущих бітів (MSB),
k' – числове значення загальних менш значущих бітів (LSB), $\tilde{k}_i$ - це числове значення поточних бітів, t – кількість загальних більш значущих бітів,
t' – кількість загальних менш значущих бітів

Слід зауважити, що значення змінних $k_i, k, \tilde{k}_i, k'$ невідомі. У n-ої кількості рівнянь (4), що визначають підпис:

$$\begin{cases} m_1 + ar_1 - s_1 k_1 \equiv 0 \,(\mathrm{mod}\, q); \\ m_2 + ar_2 - s_2 k_2 \equiv 0 \,(\mathrm{mod}\, q); \\ \dots \\ m_n + ar_n - s_n k_n \equiv 0 \,(\mathrm{mod}\, q), \end{cases} \tag{6}$$

де m_i – це повідомлення, a – це приватний ключ відправника, r_i – числове значення першої частини сигнатури, s_i – числове значення другої частини сигнатури, k_i – ефемерний ключ, i-індекс повідомлення.

Якщо в системі рівнянь (6) замінити параметри m_i, r_i, s_i на значення (5) і виключити загальні змінні k та k', тоді отримаємо:

$$\begin{cases} (s_1^{-1} m_1 - s_2^{-1} m_2) + a(s_1^{-1} r_1 - s_2^{-1} r_2) - 2^t (\tilde{k}_1 - \tilde{k}_2) \equiv 0 (\mathrm{mod}\, q); \\ (s_1^{-1} m_1 - s_3^{-1} m_3) + a(s_1^{-1} r_1 - s_3^{-1} r_3) - 2^t (\tilde{k}_1 - \tilde{k}_3) \equiv 0 (\mathrm{mod}\, q); \\ \dots \\ (s_1^{-1} m_1 - s_n^{-1} m_n) + a(s_1^{-1} r_1 - s_n^{-1} r_n) - 2^t (\tilde{k}_1 - \tilde{k}_n) \equiv 0 (\mathrm{mod}\, q). \end{cases} \tag{7}$$

Нехай $\alpha_i, \beta_i, \kappa_i \in Z$ є такими, що:

$$\begin{cases} \alpha_i := 2^{-t} (s_1^{-1} m_1 - s_i^{-1} m_i) \bmod q; \\ \beta_i := 2^{-t} (s_1^{-1} r_1 - s_i^{-1} r_i) \bmod q; \\ \kappa_i := \tilde{k}_1 - \tilde{k}_i. \end{cases} \tag{8}$$

Тоді, система рівнянь (7) приймає наступний вигляд:

$$\begin{cases} \alpha_2 + a\beta_2 - k_2 \equiv 0 \,(\mathrm{mod}\, q); \\ \alpha_3 + a\beta_3 - \kappa_3 \equiv 0 \,(\mathrm{mod}\, q); \\ \alpha_n + a\beta_n - \kappa_n \equiv 0 \,(\mathrm{mod}\, q), \end{cases} \tag{9}$$

де a, κ_i і α_i, β_i – значення відомих та невідомих аргументів відповідно.

Тоді систему розв'язків можна представити у наступному вигляді:

$$L = \{(x_0, x_1, ..., x_n) \in \mathbb{Z}^{n+1} \mid x_0\alpha_i + x_1\beta_i - x_i \equiv 0 \,(\mathrm{mod}\, q)\}\,, \qquad (10)$$

що утворює (*n*+1) мірну маска-матрицю, натягнуту на вектори-стрічки базисної матриці:

$$M = \begin{pmatrix} 1 & 0 & \alpha_2 & \dots & \alpha_n \\ 0 & 1 & \beta_2 & \dots & \beta_n \\ 0 & 0 & q & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & q \end{pmatrix}. \qquad (11)$$

Для знаходження короткого вектору у матриці (11), можуть бути застосовані LLL або BKZ алгоритми. Якщо система рівнянь (7) має розв'язок (визначення значення короткого вектора $\bar{\upsilon}_0$ з елементами $v_0 = (1, a, \kappa_2, \kappa_3, ..., \kappa_n)$, що і буде тотожним елементом системи розв'язків (10), тоді і буде отримано значення приватного ключа *a* відправника.

Розроблений підхід порядку отримання доступу до ключів повідомлень доводить, що алгоритм DSA (ECDSA) є вразливим до криптоатак за допомогою аналізу часу підписання повідомлення, у випадку коли ефемерні ключі кожного зашифрованого мають майже одинаковий час підписання повідомлення. На рисунку 2 показано залежність між кількістю загальних бітів та кількості необхідних повідомлень до відсотку успіху криптоатаки.

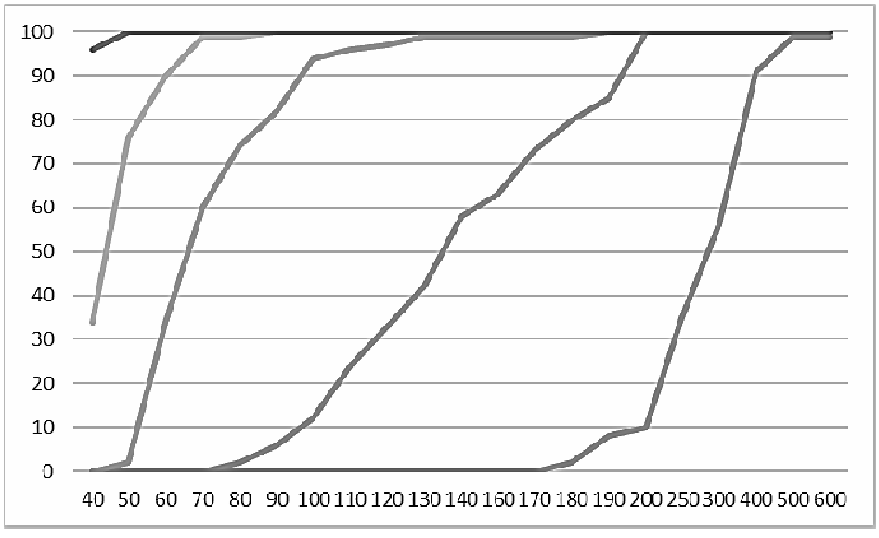

Рисунок 2 – Діаграма залежності між кількістю загальних бітів та кількості необхідних повідомлень до відсотку ймовірності успіху криптоатаки

Також, однією із вразливих точок при створення сигнатур (1) і (3) є розширений алгоритм Евкліда обчислення оберненого модуля у алгоритмі [11].

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Час виконання алгоритму [11, 12], залежить від обох значень a і b, а саме кількість ітерацій циклу while, а також степені двійки, які потрібно видалити вкінці алгоритму, значно відрізняються для різних вхідних даних. Тому для перетворення алгоритму 2 в алгоритм, що виконується за постійний час для заданого w_n-бітового модуля m, необхідно запезбечити такі вимоги:

1. Одна ітерація завжди обчислюється таку ж кількість часу, це означає обчислити всіх чотирьох гілкок із алгоритму 2 і вибір правильних значень за постійний час. Це гарантує, що час виконання однієї ітерації не залежить від взятої гілки, але означає, що час обчислення збільшується до часу обчислення всіх гілок;

2. Алгоритм повинен виконувати однакове число ітерацій(за постійний час), це означає, що необхідно обчислення найгіршого числа $2w_n$ ітерацій. Це можна реалізувати, визначивши коли алгоритм 2 завершується (коли v=1). В залежності від цієї умови, створюємо бітову маску і обираємо вхідне значення для цієї ітерації (коли v=1), або значення, обчисленні за допомогою ітерації з постійним часом ($v \neq 1$).

Після внесених модифікацій в алгоритм [11], отримаємо новий Алгоритм 1, який буде виконувати обчислення оберненого модуля за сталий час:

Алгоритм 1. Постійно часовий алгоритм оберненого ділення по модулю

Input: $a, b \in \mathbb{Z}_{>0}$ з $\gcd(a,b) = 1, 0 \le b < a$,

Output: $\begin{cases} (b^{-1} \cdot 2^k \bmod a, k) \\ [\log_2(a)] \le k \le 2[\log_2(a)]. \end{cases}$

$u \leftarrow a,\ v \leftarrow b, r \leftarrow 0, s \leftarrow 1, k \leftarrow 0$

for $\mathbf{i} = 1$ *to* $2[\log_2(a)]$ **do**

$\mathbf{uv}_< \leftarrow sub(u', u, v)$

$\mathbf{uv}_= \leftarrow equal(u', 0)$

$\mathbf{d} \leftarrow 0 - \mathbf{uv}_=$ $\quad \#\mathbf{d} = \begin{cases} 0 \ \ if\ u \neq v \\ 2^{\omega} - 1 if\ u = v \end{cases}$

$$\left.\begin{array}{l} lshift_1(\tilde{s}, s) \\ add(rs, r, s) \\ rshift_1(\tilde{u}, u) \\ \mathbf{m_1} \leftarrow \mathbf{d} \vee (0 - (\mathbf{u_0} \wedge 1)) \\ \mathbf{m_2} \leftarrow bitflip(\mathbf{m_1}) \\ select(u, \tilde{u}, \mathbf{m_2}, u, \mathbf{m_1}) \\ select(s, \tilde{s}, \mathbf{m_2}, s, \mathbf{m_1}) \end{array}\right\} \#\mathbf{if}\ u \equiv 0$$

$$
\left.\begin{array}{l}
lshift_1(\tilde{r}, r) \\
\mathbf{S} \leftarrow (\mathbf{d} \vee bitflip(\mathbf{m_1})) \\
\mathbf{m_3} \leftarrow S \vee 0 - (\mathbf{v_0} \wedge 1) \\
\mathbf{m_4} \leftarrow bitflip(\mathbf{m_3}) \\
rshift_1(\tilde{v}, v) \\
select(v, \tilde{u}, \mathbf{m_4}, v, \mathbf{m_3}) \\
select(r, \tilde{r}, \mathbf{m_4}, r, \mathbf{m_3})
\end{array}\right\} \# \textbf{ else if } v \equiv 0
$$

$$
\left.\begin{array}{l}
\mathbf{S} \leftarrow \mathbf{S} \vee bitflip(m_3) \\
\mathbf{m_5} \leftarrow \mathbf{S} \vee (0 - \mathbf{uv}_<) \\
\mathbf{m_6} \leftarrow bitflip(\mathbf{m_5}) \\
rshift_1(u', u') \\
select(u, u', \mathbf{m_6}, u, \mathbf{m_5}) \\
select(r, rs, \mathbf{m_6}, r, \mathbf{m_5}) \\
select(r, \tilde{s}, \mathbf{m_6}, s, \mathbf{m_5})
\end{array}\right\} \# \textbf{ else if } u > v
$$

$$
\left.\begin{array}{l}
\mathbf{S} \leftarrow \mathbf{S} \vee bitflip(\mathbf{m_5}) \\
\mathbf{m_7} \leftarrow bitflip(\mathbf{S}) \\
sub(\tilde{v}, v, u) \\
rshift_1(\tilde{v}, \tilde{v}) \\
select(v, \tilde{v}, \mathbf{m_7}, v, \mathbf{S}) \\
select(s, rs, \mathbf{m_7}, s, \mathbf{S}) \\
select(r, \tilde{r}, \mathbf{m_7}, r, \mathbf{S})
\end{array}\right\} \# \textbf{else}
$$

$$\mathbf{k} \leftarrow ((\mathbf{k} \wedge \mathbf{d}) \vee ((\mathbf{k} + \mathbf{1}) \wedge bitflip(\mathbf{d})))$$

end for

return $(s, \mathbf{k})$

Висновки.

1. Визначено тип враженості криптосистем до атак по побічним каналам на основі часового аналізу створення сигнатур, який заключається у зборі підписаних повідомлень та часу їх підписання. Використовуючи розроблену математичну модель, приватний ключ відправника може бути визначений при наявності достатньої кількості повідомлень менш ніж за хвилину обходячи вирішення задачі дискретного логарифмування, а це у свою чергу повністю компроментує стійкість криптосистеми.

2. Розроблено ефективний метод захисту криптосистем до атак на основі часового аналізу створення зашифрованих повідомлень для запобігання несанкціонованому доступу до зашифрованої інформації, на основі удосконаленого

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

розширеного алгоритму Евкліда обчислення оберненого модуля ефемерного ключа, що виконується за сталий час, а саме, забезпечує постійне число тактів процесора, при виконанні алгоритму для різного розміру (у бітах) чисел.

3. При використанні не стало часового розширеного алгоритму Евкліда, при обчисленні оберненого модуля числа розміру бітів 256, число тактів буде нерівномірним 50-60. При використанні удосконаленого алгоритму число тактів завжди дорівнює 486, що і забезпечує стійкіть до атаки у цілому.

ЛІТЕРАТУРА

1. Menezes A. J., Van Oorschot P.C., Vanstone S.A. Handbook of applied cryptography // Handbook of Applied Cryptography. 1996.

2. Ubaidullah M., Makki Q. A Review on Symmetric Key Encryption Techniques in Cryptography // Int. J. Comput. Appl. 2016. Vol. 147, № 10.

3. Elgamal T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms // IEEE Trans. Inf. Theory. 1985. Vol. 31, № 4.

4. Liu M., Chen J., Li H. Partially known nonces and fault injection attacks on SM2 signature algorithm // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2014. Vol. 8567.

5. Fouque P.A., Tibouchi M., Zapalowicz J.C. Recovering private keys generated with weak prngs // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2013. Vol. 8308 LNCS.

6. Boneh D., Venkatesan R. Hardness of computing the most significant bits of secret keys in diffie-hellman and related schemes // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 1996. Vol. 1109.

7. Howgrave-Graham N.A., Smart N.P. Lattice Attacks on Digital Signature Schemes // Des. Codes, Cryptogr. 2001. Vol. 23, № 3.

8. Moghimi D. et al. TPM-FAIL: TPM meets timing and lattice attacks // Proceedings of the 29th USENIX Security Symposium. 2020.

9. Gallagher P.D., Romine C. FIPS PUB 186-4 Digital Signature Standard (DSS) // Encycl. Cryptogr. Secur. 2013. № July.

10. Rostislav D. Iskovych-Lototsky, Yaroslav V. Ivanchuk, Natalia R. Veselovska, Wojciech Surtel, Samat Sundetov. "Automatic system for modeling vibro-impact unloading bulk cargo on vehicles", Proc. SPIE 10808, Photonics Applications in As-

tronomy, Communications, Industry, and High-Energy Physics Experiments 2018, 1080860 (1 October 2018). doi: 10.1117/12.2501526.
11. Kaliski B.S. The Montgomery Inverse and Its Applications // IEEE Trans. Comput. 1995. Vol. 44, № 8.
12. Wójcik, W., & Pawłowska, M. (Eds.). (2020). Biomass as Raw Material for the Production of Biofuels and Chemicals (1st ed.). Routledge. https://doi.org/10.1201/9781003177593.

REFERENCES

1. Menezes A.J., Van Oorschot P.C., Vanstone S.A. Handbook of applied cryptography // Handbook of Applied Cryptography. 1996.
2. Ubaidullah M., Makki Q. A Review on Symmetric Key Encryption Techniques in Cryptography // Int. J. Comput. Appl. 2016. Vol. 147, № 10.
3. Elgamal T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms // IEEE Trans. Inf. Theory. 1985. Vol. 31, № 4.
4. Liu M., Chen J., Li H. Partially known nonces and fault injection attacks on SM2 signature algorithm // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2014. Vol. 8567.
5. Fouque P. A., Tibouchi M., Zapalowicz J.C. Recovering private keys generated with weak prngs // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2013. Vol. 8308 LNCS.
6. Boneh D., Venkatesan R. Hardness of computing the most significant bits of secret keys in diffie-hellman and related schemes // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 1996. Vol. 1109.
7. Howgrave-Graham N.A., Smart N.P. Lattice Attacks on Digital Signature Schemes // Des. Codes, Cryptogr. 2001. Vol. 23, № 3.
8. Moghimi D. et al. TPM-FAIL: TPM meets timing and lattice attacks // Proceedings of the 29th USENIX Security Symposium. 2020.
9. Gallagher P.D., Romine C. FIPS PUB 186-4 Digital Signature Standard (DSS) // Encycl. Cryptogr. Secur. 2013. № July.
10. Nguyen P.Q., Nguyen P.Q. Public-key Cryptanalysis. 2008.
11. Kaliski B.S. The Montgomery Inverse and Its Applications // IEEE Trans. Comput. 1995. Vol. 44, № 8.

Received 03.02.2022.
Accepted 07.02.2022.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Method for increasing the degree of protection of message encryption based on an algorithm for a constant component in time

Currently, asymmetric cryptosystems are used everywhere, in document management for cryptocurrencies, providing a high level of protection to end users, relying on the mathematical complexity of calculating a discrete algorithm. But, it is possible to make a cryptocurrency attack on the so-called ephemeral key, which is an auxiliary key when creating a signature. Recent works have shown examples of cryptocurrencies on the random number generator, processor cache, timing attacks. However, these attacks do not work when the numerical value of the bits is unknown. Also, recent work shows the main vulnerability in the case signature, namely the inverse module calculation algorithm that is vulnerable to timing attacks. The article considers the damage of cryptosystems such as DSA and ECDSA before the attack based on the analysis of the variable time of signing the message. A mathematical model has been developed to test this type of lesion, based on lattice attacks. It is shown that if there are enough signatures with the same signing time, it is possible to identify the presence of common bits of ephemeral keys, which will restore the sender's private key. It is proved that the cause of the lesion is the lack of execution of the operation of calculating the inverse module of the time variable, which provides ephemeral key data to the attacker. To solve this problem, an extended Euclidean algorithm for calculating the inverse module for a fixed time is proposed. In this paper, the advanced Euclidean algorithm for calculating the inverse module is improved, namely, its constant time execution is achieved, which prevents timed attacks.

Іванчук Ярослав Володимирович – д.т.н., професор кафедри комп'ютерних наук Вінницького національного технічного університету.

Коваль Костянтин Олегович – к.т.н., завідувач кафедри інтеграції навчання з виробництвом Вінницького національного технічного університету.

Горобець Юрій Володимирович – аспірант кафедри комп'ютерних наук Вінницького національно технічного університету.

Ivanchuk Yaroslav Volodymyrovych – Doctor of Technical Sciences, Professor Department of Computer Science Vinnytsia National Technical University of Ukraine.

Koval Kostyantyn Olegovych – Ph. D., Head of Department of integration of study with production, Vinnytsia National Technical University of Ukraine.

Horobets Yurii Volodymyrovych – graduate student of the Department of Computer Science of The Vinnytsia National Technical University of Ukraine.