УДК 681.3.07

О. А. ШЕВЧУК

АКТУАЛЬНІСТЬ АТАКИ НА ЗВ'ЯЗАНИХ КЛЮЧАХ ДЛЯ АПАРАТНИХ РЕАЛІЗАЦІЙ ЗАСОБІВ КЗІ

Вступ

З поширенням та впровадженням у щоденне життя засобів КЗІ зростає кількість виробників цих рішень, поширюється та поповнюється новими сторонами сітка дистрибуції. За таких умов кількість сторін, яким довіряє користувач засобів КЗІ, значно зростає. Постає питання – чи може користувач бути впевненним у тому, що на всіх етапах, від виробництва до впровадження рішень, не було допущено грубих помилок, або навмисно внесено лазівок? Загалом проблема довіри може бути частково вирішена незалежним аудитом та сертифікацією засобів, що зводить кількість зовнішніх сторін, яким треба довіряти, до однієї. Але чи можна бути впевненим, що продукти, які проходили сертифікацію, та продукти, що впроваджуються, – це ті самі продукти? У випадках з програмними засобами КЗІ можливо користуватись контрольною сумою, що наведено у матеріалах результатів сертифікації. Для апаратних та програмно-апаратних рішень неможливо обчислити геш значення. Неможливо дослідити, що чіпи та контроллери, що впроваджуються, – це саме ті контролери та чіпи, що було досліджено.

Така ситуація могла б не становити значної загрози, якщо б компроментація могла бути здійснена тільки за рахунок використання лазівок. Більш важлива можливість компроментації за результатами роботи засобів КЗІ, що здійснюються за рахунок невиконання важливих умов,на які розраховує алгоритм.

В статті на прикладі ЕЦП показана схема компроментації приватного ключа, що може бути використана у програмно-аппаратних засобах.

Модель загроз

Визначимо положення моделі загроз, у якій зловмисником виступає розробник.

- Розробник виконує програмно-аппаратну реалізацію схеми ЕЦП, передає засоби користувачеві.
- Розробник та користувач не можуть змінювати логіку роботи засобів після впровадження засобу.
- Розробник не має можливості взаємодіяти з засобом після впровадження.
- Вхідні дані та результат її обробки не є секретними, та можуть бути отримані зацікавленими сторонами.
- Вихідні дані можуть бути протестовані сторонніми засобами (наприклад – NIST STS).

Користувач не довіряє розробнику.

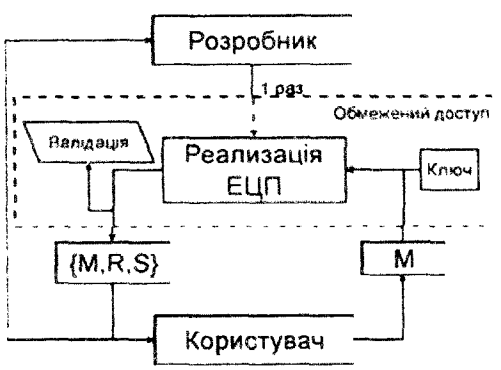

Рис. 1. Загальна модель загроз

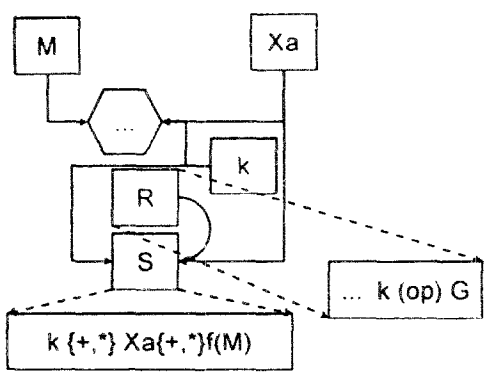

Рис. 2. Загальний вигляд

Задачі зловмисника та користувача такі:

• зловмисник – виробити засіб КЗІ таким чином, щоб сторонній засіб тестування не виявив наявності закладки. З наявних у публічному доступі пар даних та ЕЦП даних відновити приватний ключ користувача;

• користувач – виробити засіб та методику тестування вихідних значень засобу КЗІ таким чином, щоб мати можливість виявити наявність закладки або порушень у роботі пристрою, що можуть призвести до компроментації приватного ключа.

Для досягнення мети зловмисник має обрати такий елемент підпису, що використовується завжди, на який користувач не має ніякого впливу, та який отримано за допомогою функції отримання випадкового числа. Наприклад, доменні параметри ЕЦП дуже вразливі до невиконання вимог. Але, як правило, доменні параметри ЕЦП можуть бути визначені довіреною стороною.

Приватний ключ користувача – більш вдалий параметр для нав'язування. Його генерує пристрій, звичайно його неможливо отримати та дослідити. Але деякі пристрої мають можливість імпортувати приватний ключ до пристрою.

Найбільш придатним до модифікації є другий приватний ключ користувача. Подивимося, яку роль грає інший приватний ключ, так званий ключ сесії, на прикладі деяких ЕЦП у г.т. ЕК з відновленням та доповненням повідомлення.

Ключові данні у ЕЦП з доповненням та відновленням повідомлення

Для підписів ECDSA/ДСТУ/ГОСТ/ECNR/ECPV та інших [1 – 4] існує пара ключів, що формуються випадковим чином. Визначимо їх як $X_a = RNG()$ та $k = RNG()$. X_a формується один раз та називається секретним ключем, k формується для кожного підпису окремо та називається сесійним ключем. Але треба відзначити, що хоча k і називається *сесійним* ключем, він не має очікуваних властивостей сесійних ключів, подібних до тих, що мають схеми шифрування. Компроментація ключа k негайно призводить до компроментації X_a.

Подивимося докладніше. У всіх схемах ЕЦП k обчислюється і використовується двічі: як рандомізуючий компонент чи ключ KDF $k = RNG()$, $\Pi = \pi(kG)$; та як другий ключ у S компоненті [1, 2, 3, 4]

$\{R,S\}$:

- ECDSA: $\{r = \Pi; s = k^{-1}(f(M) + X_a r) : \mathrm{mod}\, N\}$
- ДСТУ: $\{r = f(M)\Pi; s = (k + X_a r) : \mathrm{mod}\, N\}$
- ECNR: $\{r = f(M) + \Pi; s = (k - X_a r) : \mathrm{mod}\, N\}$
- ECMR: $\{r = f(M) \oplus \Pi; s = (r * k - r - 1)/(X_a + 1) : \mathrm{mod}\, N\}$
- ...

Для підписів з доповненням $f(M) = Hash(M)$. Для підписів з відновленням $f(M)$ обчислюється відповідно до схеми.

Стандарти схем ЕЦП висувають k та $RNG()$ жорсткі вимоги щодо випадковості значення. Стандарт DSA, наприклад, визначає k як непередбачуваний секретний параметр, що ніколи не повторюється. На практиці ця вимога іноді не виконується, а у деяких джерелах вимоги скорочуються до k як «випадкового»' числа.

Проблема є актуальною. За останні декілька років трапилось два досить відомих випадки, коли невиконання вимог щодо випадковості k призвело до створення массових загроз безпеки та зашкодило репутації корпорації.

У 2008 р. група, що виконує обов'язки з аудиту програмного забезпечення дистрибутиву Debian (Debian Security Team), виявила значні недоліки у пакеті з реалізацією ЕЦП – openssl. Проблема полягала у тому, що генератор псевдовипадкових послідовностей ініціалізувався ідентифікатором потока користувача, що міститься у множині значень $s \in (0;32768)$. Помилка мала місце у функціонуючих системах два роки. В результаті було масово відкликано не

тільки ті сертифікати, які було сформовані за час функціонування версій ПЗ, що мали недолікі, але й ті, що *використовувалися* у той проміжок часу[5].

У 2010 р. на 27-й конференції Chaos Communication Congress було зроблено доповідь про значний успіх у роботі над завантаженням до Sony PlayStation 3 довільного ПЗ. Корпорація Sony реалізує у продаж мільйони пристроїв PlayStation 3 нижче собівартості пристрою, розраховуючи на прибутки від продажу програмного забезпечення. Ігровий засіб не виконує не підписане корпорацією ПЗ. Дослідники змогли відновити закритий ключ корпорації завдяки невиконанню вимог до випадкової компоненти k у реалізації ЕЦП ECDSA. Відновлений ключ відкриває шлях до завантаження та виконання довільного ПЗ на пристрої. Можна констатувати, що корпорація понесе значні збитки [6].

Модель атаки на реалізацію з використанням Г(П)СЧ

Визначимо суть та мету атаки: для функції RNG необхідно виробити таку функцію відображення $a()$ ($a(): (k = RNG()) \to k'$), вектор результатів виконання $[k_0, k_1, \ldots k_n]$ якої буде задовольняти наступним умовам:

- $\exists\{k_i, k_j\} \in [k_0, k_1, \ldots k_n] : k_i - k_j = \varepsilon = const, n = const$
- $\forall\{k_i, k_j\} \in [k_0, k_1, \ldots k_n] : k_i - k_j \neq 0, n = const$
- $K = a(RNG(N)); \forall K : STS([k_0, k_1, \ldots k_n], nN) = 0, n = const$, де $STS(K, N)$ – функція статистичного тестування бітової послідовності довжиною N біт, що вертає 0, якщо послідовність відповідає критеріям випадковості, та 1 – якщо не відповідає; $RNG(N)$ – вертає випадковій бітовий рядок довжиною N біт.

Наведемо приклад алгоритму функції $a()$, що задовольняє двом умовам з трьох:

$$\forall\{k_0, k_n\} \in a() : k_0 - k_n = 0 \qquad (1)$$

Дійсно, якщо $n = const$, тоді $i - j = 0 - n = const$. Тестування подібної послідовності з використанням комплекту NIST STS 2.1 для функції $STS()$ показує, що для $L_n(k_i) = 160$ та $n = 1000$ та вказаних до NIST STS рекомендаціях послідовність, що отримана за рахунок відображення функцією $a()$ вважається випадковою.

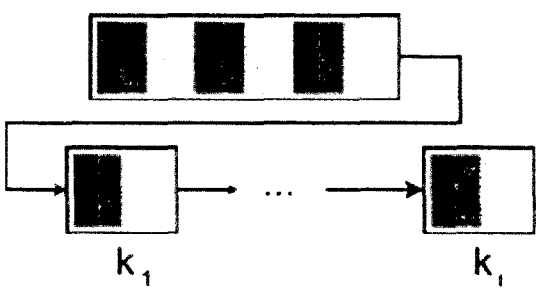

Рис. 3. Повторювання послідовності

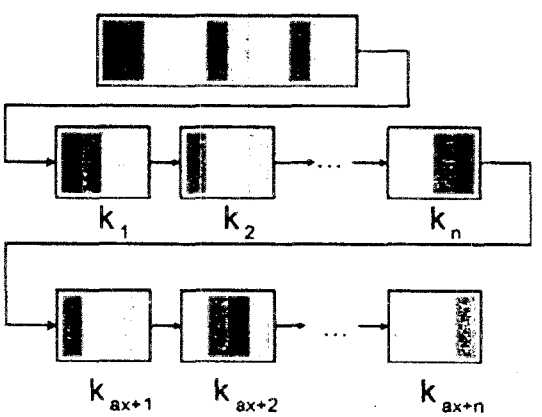

Рис. 4. Невипадкова послідовність

Для функції (1) відновлення ключа x буде мати такий вигляд:

- ECDSA:

$$k = \frac{d_2 - d_1}{s_1 - s_2}; x = \frac{sk - d}{r}; \bmod N.$$

- ДСТУ:

$$x = \frac{(s_1 - s_2)}{(r_1 - r_2)}; \bmod N.$$

- ECNR, ECPV, ECAO, ECKNR:

$$x = \frac{s_1 - s_2}{r_2 - r_1}; \bmod N$$

- ECMR:

$$x = \frac{r_2 - r_1}{r_1 s_2 - r_2 s_1}; \bmod N$$

Модифікуємо (1) таким чином, щоб нове рівняння задовольняло усім умовам:

$$\forall \{k_a, k_n\} \in a() : k_a - k_n = \varepsilon = const. \quad (2)$$

Для послідовностей, де $\varepsilon = 2^{\lfloor \log_2 n \rfloor - 1} - 1$ та $n > 900$, функція відображення $a()$ значно не змінює статистичну картину, що отримано за допомогою NIST STS 2.1, та зовсім не змінює кількість тестів на випадковість що не пройдено.

Для функції (2) відновлення ключа x буде мати такий вигляд.

- ECNR, ECPV, ECAO, ECKNR:

$$x = \frac{s_1 - s_2 + \varepsilon}{r_2 - r_1}; \bmod N.$$

- ECMR:

$$x = \frac{r_2 - r_1 + r_1 r_2 \varepsilon}{r_1 s_2 - r_2 s_1}; \bmod N.$$

Можна зробити висновок, що k становить для зловмисника оптимальний вектор атаки: користувач не впливає на процес формування цього параметру, він випадковий, його неможливо відновити та складно проаналізувати.

Виявлення небезпечних ключових послідовностей

Визначимо небезпечну ключову послідовність як бітовий рядок B, який відповідає умові

$$\exists \{i, j\} : f(k_i, \varepsilon) = k_j \quad (3)$$

де $k_n \| k_1 \| \ldots \| k_0 \| pad = B$, $f(x, y) = x'$ – деяка відома лінійна функція, що може бути виражена груповими операціями від змінної та константи, та є дистрибутивною операцію щодо групової операції в групі точок еліптичної кривої; ε – деяке відоме значення.

Для $\{k_n, k_1, \ldots k_0\}$ згідно до (2) відновлення секретного параметру можливе.

Визначимо можливу методику використання послідовності B. По-перше, сформулюємо у загальному вигляді процес використання пов'язаних значень. Зловмисник для відновлення секретного параметру x використовує відомі параметри двох підписів $\{r_1, s_1, d_1\}, \{r_2, s_2, d_2\}$ та відомий параметр ε, та вирішує рівняння

$$S(r_1, d_1, k_1, x) = S(r_2, d_2, f(k_1, \varepsilon), x). \quad (4)$$

відновлюючи секретний параметр x, де $S(...)$ – функція формування s компоненти підпису. Якщо вимоги до функції $f()$ виконані, тоді (4) має тривіальне вирішення для існючих схем ЕЦП. Випадок нелінійності функції $f()$ не було досліджено.

Розглянемо можливі способи виявлення таких послідовностей. k-компонента підпису використовується в обох частинах ЕЦП: $\{r, s\}$. В загальному вигляді r може бути представлене як $r = OS2IP(EC2OSP(kG))$, де група функцій $OS2IP(EC2OSP())$ відіграє роль функції відображення точки еліптичної кривої до скінченого поля, яка в більшості випадків не є ізоморфною. Таким чином, у загальному вигляді r компонента підпису не має джерел для аналізу, за випадком деяких тривіальних відображень (ДСТУ 4145, ECDSA), де r представлена як $r = x_1(x, y) = kG$. Завжди доступна можливість дослідження kG у час

перевірки ЕЦП. В усіх схемах ЕЦП з k компонентою у групі точок еліптичної кривої на останніх етапах проводиться відновлення передпідпису.

Визначимо методику дослідження передпідпису. Для дослідження точки kG як елемента групи точок еліптичної кривої ми можемо використовувати операції $\{=,+\}$. З (4) зрозуміло, що об'єктом дослідження є можливість визначення для деяких пар $\{k_i, j_i\}$ такого ε, що $k_i = f(k_i, \varepsilon)$. Єдиний спосіб аналізувати зв'язок k компонент за допомогою відновленого передпідпису – перевіряти

$$k_i G = f(k_i G, \varepsilon' G) \tag{5}$$

де ε' – обране для перевірки випадкове (чи визначене) число. Якщо (5) дійсне, тоді для k_i та k_j існує зв'язок, що визначено функцією $f()$ з параметром ε'. Треба зауважити, що складність пошуку ε' за допомогою оракула дорівнює складності пошуку k. Модель пошуку за (5) з використанням оракулу накладає додаткові обмеження на функцію $f()$. Легко побачити, що функція $f()$ має бути виражена тільки за допомогою операції '+'.

Використання зв'язку для рівномірної вибірки даних

Визначимо ознаку рівномірності. Нехай $y = RNG_D(x)$ – функція, що вертає деяке $y \in [0, x-1]$ з джерела D. Тоді рівномірною вибіркою даних назвемо такий кортеж $\{(x_0, y_0), (x_1 y_1), \ldots (x_n y_n)\}$, для яких

$$\forall x \in \{x_0, x_1, \ldots x_n\} : log_2 x = C = const \tag{6}$$

Нехай джерело D постачає кожні 163 біти, з проміжком 163000 бітів, як $y_i = 2^{163} - 1 - y_{(i+163000,163)}$. У разі, коли $f(\xi) = \xi + \varepsilon$, та $y = RNG_D(x) = d \bmod x$, $d \in D$, $\lfloor log_2 d \rfloor = \lfloor log_2 x \rfloor$, $D = \{d_0, d_1, \ldots d_n\}$ можливі наступні варіанти. Якщо $d < x$, тоді $k = y$, та $k' = f(k) = k + \varepsilon = y + \varepsilon$. Коли $d > x$, тоді $k = x - d$, та $k' = f(k) = k + \varepsilon - x = y - x + \varepsilon$.

Таким чином, в загальному випадку для перевірки гіпотези щодо значення ε' необхідно перевірити рівняння (2) з параметрами $\varepsilon = \varepsilon'$ та $\varepsilon = \varepsilon' - x$.

У разі, коли джерело D є постачальником скомпроментованих значень k, постає питання – що буде, коли в (6) $x \neq const$?

По-перше, виникає проблема визначення того факту, що використане значення k містить частину невипадкового d. Встановити це можливо, якщо результати використання усіх k наявні.

По-друге, необхідно переформатувати d таким чином, щоб рівняння (6) було дійсним. Нехай $D = \{\ldots, d_{i+1}, d_i, d_{i-1}, \ldots\}$, де $\delta = (d_{i+1} \| d_i \| d_{i-1})$, для якого є відоме значення ε, $y = d_i$, та $f(), \varepsilon'$ пов'язують між собою d. Така нерівномірність з обох боків призводить до ускладнення загальної схеми.

Висновки

Визначена проблема можливості створення функції відображення є загальною, та не має відношення тільки до конкретних схем ЕЦП. Суть проблеми становить композиція характеристик схеми, що поєднана з фізичною реалізацю. Визначимо важливі тези.

- Наявність компонент у криптографічних схемах, які приймають значення з деякою вірогідністю, унеможливлюють створення пакету тестування, що буде функціонувати за методикою чорної скриньки.
- Пов'язаність безпеки приватних даних, а не безпеки однієї транзакції. У разі компроментації сесійного ключа компроментується секретний компонент схеми.
- Можливість визначення приватної компоненти за можливості визначення такої функції $f(k_1) = k_2$, що робить можливим обчислення одного сесійного ключа з іншого, та яка визначенна для усіх можливих k.

• Діючі вимоги до апаратних засобів та діючі вимоги до сесійної компоненти k (випадковий ключ сесії k має бути знищений одразу після виконання криптографічної операції) унеможливлюють ані отримання, ані аналіз.

• Принципова неможливість валідації існуючого засобу КЗІ, що функціонує як чорна скринька. Можна казати, що існує так звана «проблема чорного лебідя». Повний аналіз пристрою виведе його з робочого стану та не буде гарантувати того, що інші пристрої є безпечними. Необхідно внести вимоги щодо можливості фізичної декомпозиції засобу КЗІ.

Проблемні питання з цього напряму включають наступні пункти:

- вирішення рівняння пов'язаних ключів для нерівномірних вибірок данних;
- отримання дифференційной моделі джерела випадкових послідовностей;
- внесення змін вимог до реалізації апаратних засобів КЗІ;
- пошук оптимального співвідношення $\{\varepsilon, i-j\}$ для рівняння (3).

Список літератури: 1. *ДСТУ 4145-2002* Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевірка". – К. : Держстандарт України, 2003. 2. *ГОСТ Р 34.10-2001.* Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. Госстандарт России, 2001. 3. *Iso/iec 9796-3:* Discrete logarithm based mechanisms / ISO/IEC. URL:http://www.iso.org/. 2006, вересень. 4. *American* National Standards Institute, 1120 Connecticut Ave., N.W.Washington, DC 20036. Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 1998, вересень. 5. US-CERT/NIST. Cve-2008-0166. 6. *fail0verlow.* Console hacking 2010. URL: http://events.ccc.de/congress/2010/Fahrplan-/attachments/-1780_27c3_-console_-hacking_2010.pdf.

Харківський національний університет радіоелектроніки

Надійшла до редколегії 12.08.2011