УДК 004:512.742.72

АНАЛІЗ ЗАГРОЗ ПРИ ПОВТОРНОМУ ВИКОРИСТАННІ НАЛАШТУВАННЯ У SNARK-ДОВЕДЕННІ ЗА ПРОТОКОЛОМ GRO-16

А. А. Бещук[1, a]

[1] *Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Фізико-технічний інститут*

Анотація

У даній роботі розглядається один із протоколів доведень без розголошення – GRO-16, та побудова атак на нього на основі повтороного використання множини початкових значень, на яких будується протокол доведення. Було показано, як зловмисник може використати такі відомості, щоб генерувати фальшиві доведення володіння одиницею криптовалюти.

Ключові слова: блокчейн, SNARK, білінійні відображення

Вступ

У багатьох протоколах доведень без розголошення існує множина початкових значень – *налаштування*([1]). Ця множина є вхідним значенням протоколу. Налаштування генерується спільними зусиллями значної частини учасників протоколу (можливо, усіма), щоб унеможливити неправомірні модифікації та значно зменшити імовірність змови [2]. Їх коректність відіграє значну роль у забезпеченні стійкості протоколів. GRO-16 не є винятком, і оскільки генерування налаштування може потребувати значних обчислювальних ресурсів, є доцільним розглядати питання використання одного й того ж налаштування. Проте з цим пов'язані деякі ризики та загрози, які будуть розглянуті у даній роботі.

У першій частині роботи розглядається означення та основні властивості білінійних відображень. Друга частина присвячена власне опису протоколу GRO-16 [3], а третя – загрозам повторного використання налаштування у доведеннях за протоколом GRO-16 володіння одиницею криптовалюти, яку надалі будемо називати *монетою*.

1. Білінійні відображення

Нехай $\mathbf{G_1}, \mathbf{G_2}, \mathbf{G_3}$ – циклічні мультиплікативні групи простого порядку p (не обов'язково однакові). Нехай G_i – генератор групи $\mathbf{G_i}$, $i = \overline{1,\ 3}$.

Означення. Білінійним відображенням називається таке відображення $e\colon \mathbf{G_1} \times \mathbf{G_2} \to \mathbf{G_3}$, що:

1) $\forall u \in \mathbf{G_1},\ v \in \mathbf{G_2},\ a, b \in \mathbb{Z}_p^* : e(u^a, v^b) = e(u, v)^{ab}$.
2) Існує алгоритм, що обчислює функцію e за поліноміальний час від довжини вхідних даних.
3) Відображення не повинно бути виродженим, тобто $e(G_1, G_2) \not\equiv 1_{\mathbf{G_3}}$, де $1_{\mathbf{G_3}}$ – нейтральний елемент групи $\mathbf{G_3}$.

[a] andrii.beshchuk@gmail.com

2. Протокол GRO-16

2.1. Постановка задачі

Надалі усі величини вважаються відомими всім учасникам протоколу, якщо окремо не сказано про їх секретність.

Нехай $\mathbb{F}_p$ – поле великої простої характеристики p. Значення p, елементи поля $u_{i,q}, v_{i,q}, w_{i,q} \in \mathbb{F}_p$, поліноми $u_i(x),\ v_i(x),\ w_i(x),\ t(x) \in \mathbb{F}_p[X], i = \overline{0, m}$ публікуються у нульовому блоці блокчейну та відомі всім учасникам. Учаснику необхідно довести без розголошення, що він знає такі значення $a_i, i = \overline{0, m}$, де $a_0 = 1$, та

$$\sum_{i=0}^{m} a_i u_i(x) \cdot \sum_{i=0}^{m} a_i v_i(x) \equiv \sum_{i=0}^{m} a_i w_i(x) \pmod{t(x)},$$

або

$$\sum_{i=0}^{m} a_i u_i(x) \cdot \sum_{i=0}^{m} a_i v_i(x) = \sum_{i=0}^{m} a_i w_i(x) + h(x)t(x),$$

$$\text{де } x \in \{1, 2, ..., n\},\ t(x) = \prod_{q=1}^{n}(x - q),\ u_i(q) = u_{i,q},$$

$$v_i(q) = v_{i,q},\ w_i(q) = w_{i,q},$$

$$q = \overline{1,\ n}. \quad (1)$$

Тут і надалі кожна монета має власний унікальний набір значень $\{a_i\}_{i=0}^{m}$, який можна розбити на дві підмножини. Перша відома усім учасникам множина «тверджень» $\{a_i\}_{i=0}^{l}$, яка дозволяє однозначно ідентифікувати монету. Друга множина «свідків» $\{a_i\}_{i=l+1}^{m}$ відома тільки власнику, знаючи яку він може витратити цю монету.

2.2. Формування налаштування

Нехай $\mathbf{G_1}, \mathbf{G_2}, \mathbf{G_3}$ – циклічні групи великого простого порядку p; $e\colon \mathbf{G_1} \times \mathbf{G_2} \to \mathbf{G_3}$ – білінійне від-

ображення. $\mathbf{G_1}$ – підгрупа деякої еліптичної кривої $E(\mathbb{F}_r)$ над простим полем $\mathbb{F}_r$. $\mathbf{G_2}$ – підгрупа тієї ж еліптичної кривої $E(\mathbb{F}_{r^k})$ над деяким розширенням $\mathbb{F}_{r^k}$. $\mathbf{G_1} \cap \mathbf{G_2} = \{\mathbf{O}\}$, де $\mathbf{O}$ – нейтральний елемент $E(\mathbb{F}_{r^k})$. $\mathbf{G_3}$ – підгрупа $\mathbb{F}^*_{r^k}$. $G_1,\ G_2$ – генератори груп $\mathbf{G_1},\ \mathbf{G_2}$ відповідно.

Надалі «+» позначає операції у $\mathbf{G_1}$ та $\mathbf{G_2}$. Відповідно «*» позначає операцію в $\mathbf{G_3}$.

Задача генерації налаштування полягає в обчисленні елементів множини

$$\{G^{(1)}_\alpha, G^{(1)}_\beta, G^{(1)}_\delta, \{X^{(1)}_i = x^i G_1\}^{n-1}_{i=0}, \{G^{(1)}_{\alpha\beta\gamma,i}\}^{l}_{i=0}, \{G^{(1)}_{\alpha\beta\delta,i}\}^{m}_{i=l+1}, \{T^{(1)}_{\delta,i}\}^{n-2}_{i=0}\}, \quad (2)$$

та елементів множини

$$\{G^{(2)}_\beta, G^{(2)}_\gamma, G^{(2)}_\delta \{X^{(2)}_i = x^i G_2\}^{n-1}_{i=0}\}, \quad (3)$$

де

$$\begin{aligned} G^{(1)}_{\alpha\beta\gamma,i} &= \frac{\beta u_i(x) + \alpha v_i(x) + w_i(x)}{\gamma} G_1, \\ G^{(1)}_{\alpha\beta\delta,i} &= \frac{\beta u_i(x) + \alpha v_i(x) + w_i(x)}{\delta} G_1, \\ T^{(1)}_{\delta,i} &= \frac{x^i t(x)}{\delta} G_1. \end{aligned}$$

Для спрощення будемо позначати

$$G^{(j)}_\rho = \rho G_j,\ \rho \in \{\alpha, \beta, \delta, \gamma\},\ j = \overline{1,\ 2}.$$

Побудувати елементи множин (2) та (3) треба так, щоб виконувались такі вимоги:

- усі учасники знають всі значення елементів цих множин;
- кожний учасник бере участь в обчисленні елементів цих множин;
- якщо якийсь учасник діє не по протоколу, усі інші учасники це бачать;
- жоден учасник не знає про секретні значення $\alpha, \beta, \gamma, \delta, x^i,\ i = \overline{1, n-1}$.

Розглянемо роботу алгоритму генерування перших трьох елементів множин (2) та (3) на прикладі обчислення $G^{(1)}_\alpha = \alpha G_1$.

Алгоритм 1

1) Кожний учасник P_i вибирає випадкове секретне значення $\alpha_i \in \mathbb{F}_p$, обраховує $U_i = \alpha_i G_1$, $T_i = \alpha_i G_2$ та публікує U_i і T_i у блокчейн, $i = \overline{1, k}$, де k – кількість учасників.
2) Кожний учасник $P_j,\ j \neq i$, перевіряє рівність
$$e(U_i, G_2) = e(G_1, T_i).$$
3) Послідовно кожний учасник P_i, починаючи з $i \geq 2$, обраховує значення $U_{1,2,\dots,i} = \alpha_i U_{1,2,\dots,i-1}$ та публікує $U_{1,2,\dots,i}$ у блокчейн.
4) Паралельно, з кожною публікацією учасником P_i значення $U_{1,2,\dots,i}$, інші учасники перевіряють рівність:
$$e(U_{1,2,\dots,i}, G_2) = e(U_{1,2,\dots,i-1}, T_i).$$
5) Якщо всі учасники дотримувались протоколу, встановлюється $G^{(1)}_\alpha = U_{1,2,\dots,k}$.

Зауваження. Кожна генерація $x^i G_1$ відбувається як $x \cdot x^{i-1} G_1$.

Зауваження. Значення, наприклад, $u_i(x)G_1$ обчислюється таким чином. Нехай

$$u_i(x) = A^{(i)}_{n-1} x^{n-1} + A^{(i)}_{n-2} x^{n-2} + \dots + A^{(i)}_1 x + A^{(i)}_0.$$

Тоді

$$\sum_{j=0}^{m-1} A^{(i)}_j x^j G_1 = \sum_{j=0}^{m-1} A^{(i)}_j X^{(1)}_j = u_i(x) G_1.$$

Далі, щоб обчислити значення, наприклад, $\beta u_i(x) G_1$, також використовують Алгоритм 1, тільки замість T_i на кроці 4) використовують $u_i(x)G_1$. Учасник P_1 обчислює $\beta_1 u_i(x) G_1$, учасник P_2 обчислює $\beta_2 \beta_1 u_i(x) G_1$ і так далі, де $\beta_i,\ i = \overline{1, k}$, такі ж самі, що і при обчисленні $G^{(1)}_\beta$ і $G^{(2)}_\beta$.

Після того, як учасники сгенерували

$$U_i = (\beta u_i(x) + \alpha v_i(x) + w_i(x)) G_1,\ i = \overline{0, m},$$

їм потрібно помножити частину цих значень на γ^{-1} та на δ^{-1}, щоб отримати $\{G^{(1)}_{\alpha\beta\gamma,i}\}^{l}_{i=0}$ та $\{G^{(1)}_{\alpha\beta\delta,i}\}^{m}_{i=l+1}$ відповідно. Для цього вони використовують наступний алгоритм.

Алгоритм 2

1) k учасників протоколу використали Алгоритм 1 для побудови $G^{(2)}_\gamma = \gamma G_2$. Отже, кожному учаснику P_i вже відповідає значення $\gamma_i \in \mathbb{F}_p$ і множина $\{D_i,\ T_i\}$, де $D_i = \gamma_i G_1, T_i = \gamma_i G_2, i = \overline{1, k}$.
2) Учасник P_i обчислює значення $\gamma_i^{-1} \bmod p$, $i = \overline{1, k}$.
3) Учасник P_i обчислює значення $Z_i = \gamma_i^{-1} G_2$ і публікує Z_i у блокчейн, $i = \overline{1, k}$.
4) Учасник $P_j,\ i \neq j$, перевіряє рівність:
$$e(D_i, Z_i) = e(G_1, G_2),\ i = \overline{1, k},\ j = \overline{1, k}.$$
5) Послідовно кожний учасник P_i обчислює значення $J_{ij} = \gamma_i^{-1} J_{(i-1)j}$ і публікує $\{J_{ij}\}^{l}_{j=0}$, де $J_{0j} = U_j$.
6) Паралельно, з кожною публікацією учасником P_i значенньь $\{J_{ij}\}^{l}_{j=0}$, інші учасники перевіряють рівність:
$$e(J_{ij}, T_i) = e(J_{(i-1)j}, G_2),\ i = \overline{1, k},\ j = \overline{0, l}.$$
7) Якщо всі учасники дотримувались протоколу, встановлюється і публікується
$$\{G^{(1)}_{\alpha\beta\gamma,i} = J_{ki}\}^{l}_{i=0}.$$

Множина значень $\{G^{(1)}_{\alpha\beta\delta,i}\}^{m}_{i=l+1}$ обчислюється аналогічно. Також аналогічним чином будується $\{T^{(1)}_{\delta,i}\}^{n-2}_{i=0}$.

Отже, маємо:

$$\sigma_1 G_1 = \{G^{(1)}_\alpha, G^{(1)}_\beta, G^{(1)}_\delta, \{X^{(1)}_i\}^{n-1}_{i=0}, \{G^{(1)}_{\alpha\beta\gamma,i}\}^{l}_{i=0}, \{G^{(1)}_{\alpha\beta\delta,i}\}^{m}_{i=l+1}, \{T^{(1)}_{\delta,i}\}^{n-2}_{i=0}\},$$
$$\sigma_2 G_2 = \{G^{(2)}_\beta, G^{(2)}_\gamma, G^{(2)}_\delta \{X^{(2)}_i\}^{n-1}_{i=0}\}.$$

Таким чином учасники генерують множину налаштування

$$Setup = \{\sigma_1 G_1, \sigma_2 G_2\}. \quad (4)$$

2.3. Формування доведення

Нехай учасник P хоче довести володіння своєю монетою зі значеннями $\{a_i\}_{i=0}^m$. *Доведенням* будемо називати трійку $\langle A^{(1)}, B^{(2)}, C^{(1)}\rangle$. Алгоритм формування доведення полягає у тому, щоб спочатку випадково обрати значення $r, s \in \mathbb{F}_p^*$, та потім обчислити і опублікувати значення:

$$A^{(1)} = AG_1, B^{(2)} = BG_2, C^{(1)} = CG_1, \quad (5)$$

$$\text{де } A = \alpha + \sum_{i=0}^{m} a_i u_i(x) + r\delta,$$

$$B = \beta + \sum_{i=0}^{m} a_i v_i(x) + s\delta,$$

$$C = \frac{\sum_{i=l+1}^{m} a_i(\beta u_i(x) + \alpha v_i(x) + w_i(x))}{\delta} + \\ + \frac{h(x)t(x)}{\delta} + sA + rB - rs\delta,$$

тоді як значення A, B та C невідомі нікому, в тому числі й учаснику, який генерує доведення.

Нижче наведено алгоритм для формування доведення.

Алгоритм 3

1) P випадково генерує два значення $r, s \in \mathbb{F}_p^*$.
2) P обчислює
$$AG_1 = \alpha G_1 + \sum_{i=0}^{m} a_i u_i(x) G_1 + r\delta G_1 = A^{(1)}$$
із використанням дійсного на момент налаштування (4).
3) P обчислює
$$BG_1 = \beta G_1 + \sum_{i=0}^{m} a_i v_i(x) G_1 + s\delta G_1 = B^{(1)}.$$
4) P обчислює
$$BG_2 = \beta G_2 + \sum_{i=0}^{m} a_i v_i(x) G_2 + s\delta G_2 = B^{(2)}.$$
5) P обчислює
$$CG_1 = \frac{\sum_{i=l+1}^{m} a_i(\beta u_i(x) + \alpha v_i(x) + w_i(x))}{\delta} G_1 + \\ + \frac{h(x)t(x)}{\delta} G_1 + sAG_1 + rBG_1 - rs\delta G_1 = \\ = \sum_{i=l+1}^{m} a_i G^{(1)}_{\alpha\beta\delta,i} + \sum_{i=0}^{n-2} h_i T^{(1)}_{\delta,i} + sA^{(1)} + rB^{(1)} - \\ - rsG^{(1)}_{\delta} = C^{(1)},$$
де $\sum_{i=0}^{m} a_i u_i(x) \cdot \sum_{i=0}^{m} a_i v_i(x) = \sum_{i=0}^{m} a_i w_i(x) + h(x)t(x)$,
$\{h_i\}_{i=0}^{n-2}$ – секретні коефіцієнти $h(x)$ із (1).
6) P публікує для перевірки $\langle A^{(1)}, B^{(2)}, C^{(1)}\rangle$.

2.4. Перевірка доведення

Коли якийсь учасник P протоколу опублікував доведення (5) та стверджує, що він володіє монетою з ідентифікатором $\{a_i\}_{i=0}^l$, кожний інший учасник V має змогу перевірити його коректність, використовуючи тільки надані значення та налаштування (4). Нижче приведено алгоритм перевірки.

Алгоритм 4

1) V обчислює
$$V_{AB} = e(A^{(1)}, B^{(2)}).$$
2) V обчислює
$$V_{\alpha\beta} = e(G^{(1)}_{\alpha}, G^{(2)}_{\beta}),$$
$$V_{\alpha\beta\gamma,i} = e(G^{(1)}_{\alpha\beta\gamma,i}, G^{(2)}_{\gamma}), i = \overline{0,l},$$
$$V_{C\delta} = e(C^{(1)}, G^{(2)}_{\delta}).$$
3) V Перевіряє рівність
$$V_{AB} = V_{\alpha\beta}\left(\sum_{i=0}^{l} a_i V_{\alpha\beta\gamma,i}\right) V_{C\delta}. \quad (6)$$
Якщо рівність виконується, то доведення коректне та учасник P дійсно володіє монетою з ідентифікатором $\{a_i\}_{i=0}^l$.

2.5. Коректність доведення

Розглянемо ліву частину рівності (6):

$$V_{AB} =$$
$$= e(G_1, G_2)^{(\alpha+\sum_{i=0}^{m} a_i u_i(x)+r\delta)(\beta+\sum_{i=0}^{m} a_i v_i(x)+s\delta)} =$$
$$= e(G_1, G_2)^{P_{left}},$$
$$\text{де } P_{left} = \alpha\beta + \sum_{i=0}^{m} a_i \alpha v_i(x) + \alpha s\delta +$$
$$+ \sum_{i=0}^{m} a_i \beta u_i(x) + \sum_{i=0}^{m} a_i u_i(x) \cdot \sum_{i=0}^{m} a_i v_i(x) +$$
$$+ s\delta \sum_{i=0}^{m} a_i u_i(x) + r\delta\beta + r\delta \sum_{i=0}^{m} a_i v_i(x) + rs\delta^2 =$$
$$= \alpha\beta + \sum_{i=0}^{m} a_i(\beta u_i(x) + \alpha v_u(x)) +$$
$$+ s\delta\left(\alpha + \sum_{i=0}^{m} a_i u_i(x)\right) + r\delta\left(\beta + \sum_{i=0}^{m} a_i v_i(x)\right) +$$
$$+ rs\delta^2 + \sum_{i=0}^{m} a_i u_i(x) \cdot \sum_{i=0}^{m} a_i v_i(x).$$

Розглянемо тепер праву частину рівності (6):

$$V_{\alpha\beta}\left(\sum_{i=0}^{l} a_i \cdot V_{\alpha\beta\gamma,i}\right) V_{C\delta} = e(G_1, G_2)^{\alpha\beta} \times$$
$$\times e(G_1, G_2)^{\sum_{i=0}^{m} a_i(\beta u_i(x)+\alpha v_i(x)+w_i(x))} \times$$
$$\times e(G_1, G_2)^{h(x)t(x)+s\delta\left(\alpha+\sum_{i=0}^{m} a_i u_i(x)+r\delta\right)} \times$$
$$\times e(G_1, G_2)^{r\delta\left(\beta+\sum_{i=0}^{m} a_i v_i(x)+s\delta\right)-rs\delta^2} =$$
$$= e(G_1, G_2)^{P_{right}},$$
$$\text{де } P_{right} = \alpha\beta + \sum_{i=0}^{m} a_i(\beta u_i(x) + \alpha v_i(x)) +$$
$$+ rs\delta^2 + s\delta\left(\alpha + \sum_{i=0}^{m} a_i u_i(x)\right) + rs\delta^2 +$$
$$+ r\delta\left(\beta + \sum_{i=0}^{m} a_i v_i(x)\right) - rs\delta^2 +$$
$$+ \sum_{i=0}^{m} a_i w_i(x) + h(x)t(x).$$

Нарешті, легко бачити, що, скорочуючи однакові доданки та у силу (1), $P_{left} = P_{right}$, при коректних $\{a_i\}_{i=0}^{m}$, тобто рівність (6) виконується.

3. Атаки на GRO-16

Атаки на протокол будуються у вигляді тверджень. У цій роботі вони приведені без доведень.

Отже, нехай учасники сгенерували дійсне налаштування (4). Після аналізу протоколу, мають місце наступні твердження.

Твердження. Якщо учасник M знає значення α, β, δ із налаштування (4), то він може генерувати фальшиві доведення, які пройдуть перевірку.

Потім було виявлено значне послабшення цього твердження.

Твердження. Якщо учасник M знає хоча б одне значення із множини $\{\alpha, \beta, \delta, x\}$ з налаштування (4), то він може генерувати фальшиві доведення, які пройдуть перевірку.

Наприклад, нехай учасник M знає α і хоче довести володіння монетою із ідентифікатором $\{a_i\}_{i=0}^{l}$. Він виконує наступні кроки для генерації фальшивого доведення.

1) Випадково обирає значення $A,\ C \in \mathbb{F}_p$.
2) Обчислює $(\alpha^{-1} \bmod p)$ та $(A^{-1} \bmod p)$.
3) Обчислює $A^{(1)} = A\alpha G_1$.
4) Обчислює $C^{(1)} = CG_1$.
5) Обчислює $(\alpha^{-1}A^{-1}\beta u_i(x))G_2$.
6) Обчислює $(A^{-1}v_i(x))G_2$.
7) Обчислює $(\alpha^{-1}A^{-1}w_i(x))G_2$.
8) Обчислює $B^{(2)} = BG_2$, де
$$B = A^{-1}\beta + \sum_{i=0}^{l} a_i(\alpha^{-1}A^{-1}\beta u_i(x) + A^{-1}v_i(x) +$$
$$+\alpha^{-1}A^{-1}w_i(x)) + \alpha^{-1}A^{-1}C\delta.$$
9) Публікує значення $\langle A^{(1)}, B^{(2)}, C^{(1)}\rangle$.

Не важко перевірити, що будь-який учасник V, який буде перевіряти доведення $\langle A^{(1)},\ B^{(2)},\ C^{(1)}\rangle$, прийме його за дійсне.

Висновки

У даній роботі була показана доволі суттєва загроза валідності протоколу при повторному використанні налаштування (4). Звісно, навряд чи зловмисник має достатньо обчислювальних ресурсів, щоб швидко знайти невідомі значення з множини $\{\alpha, \beta, \delta, x\}$ при відомому налаштуванні, доки протокол взагалі не перестане бути дійснім. Але можуть існувати інші методи, які дозволяють знайти це значення, не вирішуючи задачу дискретного логарифмування. Зокрема, підслуховувати кожного учасника при генерації. Отже, необхідно робити регулярну генерацію нового налаштування (4), щоб значно зменшити ризик та кількість вкрадених монет у випадку успішної атаки.

Перелік використаних джерел

1. Buterin Vitalik. Zk-SNARKs: Under the Hood. — 2017. — P. 5 c. — Access mode: `https://medium.com/@VitalikButerin/zk-snarks-under-the-hood-b33151a013f6`.
2. Menezes Alfred J., van Oorschot Paul C., Vanstone Scott A. Handbook of Applied Cryptography. — 5 edition. — 1996. — P. 816 c. — Access mode: `http://cacr.uwaterloo.ca/hac/`.
3. Groth Jens. On the Size of Pairing-based Non-interactive Arguments. — 2016. — P. 25 c. — Access mode: `https://eprint.iacr.org/2016/260.pdf`.