Звіт з виконання лабораторної роботи 3
Варіант 1С (Сліпий цифровий підпис)

ФІ-42мн Бондар Петро
ФІ-42мн Кістаєв Матвій

January 17, 2025

Огляд

Підгрупа: 1C.

Мета: Дослідити можливість реалізації схеми сліпого цифрового підпису.

Завдання (модифіковане):

1. Теоретично дослідити тему сліпих цифрових підписів.
2. Обрати схеми для аналізу/порівняння/реалізації.
3. Детальніше проаналізувати та теоретично порівняти обрані схеми сліпого цифрового підпису.
4. Реалізувати релевантні схеми на `Python`.

Сліпий цифровий підпис

Схема сліпого цифрового підпису надає можливість користувачу поставити свій цифровий підпис на отримане повідомлення, не маючи змоги отримати жодної інформації про це повідомлення.

В загальному випадку схема виглядає наступним чином:

1. $[S]$ Підписант S генерує криптосистему підпису $\Sigma = \langle Sign, Verify \rangle$ з публічним ключем pk.
2. $[U \leftrightarrow S]$ Якісь дії...
3. $[U \to S]$ Користувач U обирає повідомлення m для підпису та використовує до нього алгоритм «засліплення» $m' = Blind(m)$. Потім надсилає m' підписанту.
4. $[U \leftrightarrow S]$ Якісь дії...
5. $[S \to U]$ Підписант підписує засліплене повідомлення $\sigma' = Sign_{pk}(m')$ та надсилає σ' користувачу.
6. $[S]$ Користувач U «знімає» засліплення $Unblind(m', \sigma')$ та отримує валідний підпис (m, σ).
7. Підпис (m, σ) буде проходити перевірку: $Verify_{pk}(m, \sigma) = true$

Або у форматі протоколу:

Сліпий цифровий підпис

Підписант S		**Користувач** U
$\Sigma = \langle Sign, Verify \rangle$		m
	$\longleftarrow \cdots \longrightarrow$	
		$m' = Blind(m)$
	$m' \longleftarrow$	
$\sigma' = Sign(m')$		
	$\sigma' \longrightarrow$	
		$\sigma = Unblind(m', \sigma')$
		Підпис (m, σ)

Вимоги до схеми сліпого цифрового підпису

Основними вимогами стійкості для сліпого цифрового підпису є:

- **Неможливість підробки (Unforgeability):**

  За l сесій підпису зловмисник не може *ефективно* обчислити $l+1$ коректний підпис для $l+1$ обраних повідомлень, для будь-якого натурального $l = poly(n)$

- **Сліпота:**

  За підписом та повідомленням користувача (m, σ) підписант не здатний статистично ідентифікувати сесію, в якій цей підпис було утворено.

Першим підписом такого типу був сліпий підпис на основі `RSA`, запропонований Девідом Шаумом в 82 році разом із концепцією електронної готівки.

На практиці зараз переважно використовуються сліпі підписи на основі підпису Шнора (на групах точок еліптичних кривих). Схема сліпого підпису Шнора має наступний вигляд:

Сліпий підпис Шнора

..... **Публічні параметри:** $G = \langle\, g \,\rangle,\ \operatorname{ord} g = q$ – просте .....

Підписант S		**Користувач** U
$\Sigma = \langle Sign,\, Verify \rangle$		
$x \in \mathbb{Z}_q$ – секретний ключ		
$h = g^x \in G$ – публічний ключ		$m \in \mathcal{M}$ – повідомлення
$r \in_R \mathbb{Z}_q$	$\xrightarrow{\quad g^r \quad}$	
		$\alpha,\ \beta \in_R \mathbb{Z}_q$
	$\xleftarrow{\quad c' \quad}$	$c' = H(g^r g^\alpha h^\beta, m) + \beta$
$z' = r + c'x$	$\xrightarrow{\quad z' \quad}$	
		$\underline{\sigma = (c, z) = (c' - \beta,\ z' + \alpha)}$

$$Verify(m, \sigma):\ c \stackrel{?}{=} H(g^z h^{-c}, m)$$

Зауваження. В протоколі функція H – це «ідеальна» геш-функція вигляду:

$$H\ :\ G \times \mathcal{M} \to \mathbb{Z}_q$$

Схема сліпого підпису на основі підпису Шнора має багато переваг в порівнянні з `RSA`:

- Дозволяє використовувати групу точок еліптичної кривої – швидше та надійніше.
- Частина роботи підписанта є дуже ефективною з точки зору обчислень
- Має доведену стійкість (окрім ROS-атаки на паралельні сесії), не існує відомих практичних атак.

Схему сліпого підпису Шнора реалізовано в файлі `sign.py` із використанням генератора Маурера із попередньої лабораторної.