КІБЕРНЕТИКА

УДК 519.7

А.В. АНІСІМОВ
Київський національний університет імені Тараса Шевченка, Київ, Україна,
e-mail: *a.v.anisimov@knu.ua.*

ЦИФРОВА АВТЕНТИФІКАЦІЯ «СВІЙ-ЧУЖИЙ»

Анотація. Запропоновано протокол багаторазової цифрової двораундової автентифікації типу «свій-чужий» для групи користувачів. В основу протоколу покладено таку конструкцію. В кожній сесії автентифікації члени групи підписом Вінтернітца підписують окремі w-блоки повідомлення, яке надає верифікатор. Він перевіряє валідність всього підпису. Поточні публічні ключі пересилаються верифікатору в попередній сесії. У такий спосіб публічні ключі утворюють структуру блокчейну. Безпека протоколу випливає з відомої безпеки підпису Вінтернітца і блокчейну публічних ключів. У криптографічній моделі випадкового оракула запропонований протокол також має властивість «доведення з нульовим розголошенням».

Ключові слова: автентифікація, коаліційна група, цифровий підпис, підпис Вінтернітца, блокчейн.

ВСТУП

Автентифікація — це процедурна ідентифікація об'єкта для перевірки прав доступу до визначеного ресурсу. За допомогою автентифікації об'єкт доводить свою автентичність. У процедурах автентифікації беруть участь дві сторони: сутність, що доводить свою ідентичність (прувер) і сутність, яка перевіряє це доведення і приймає рішення (верифікатор). Зазвичай у сучасних комп'ютерних системах доведення полягає в доведенні знання (можливості машинного обчислення) деяких ідентифікаційних параметрів без розкриття самих параметрів. При цьому вважають, що всі комунікаційні взаємодії відбуваються незахищеними каналами передавання даних. Через це безпековому аспекту автентифікації приділяють особливу увагу.

Віддалена автентифікація разом з процедурою встановлення ключів — це типова і добре досліджена задача криптографії з відкритими ключами. До найвідоміших відповідних протоколів можна віднести схему Нідхема–Шредера (типу Kerberos), яка потребує участі довіреного сервера [1]; у багатьох випадках, зокрема у мережі «Інтернет речей» (IoT), застосовують рішення, які ґрунтуються на процедурі встановлення початкового спільного секрету типу «рукостискання» Діффі–Хеллмана [2] або її модифікаціях. Безпековим недоліком базового протоколу Діффі–Хеллмана є вразливість до атаки «людина посередині» (man-in-the-middle attack). Як наслідок, з'явилися різні модифікації цього протоколу. Наприклад, порівняно з оригінальним протоколом Діффі–Хеллмана покращена схема створення спільного секрету — протокол Менезеса, Кью і Ванстоуна (MQV, HMQV) [3, 4], значно зменшує можливість несанкціонованого втручання в комунікаційні обміни. Але, як з'ясувалося, для забезпечення

гарантованого рівня безпеки і цей протокол потребує значних модифікацій [5–9]. Найбільш дослідженими є популярні трираундові Σ-протоколи для NP-відношень з математично доведеним нульовим розголошенням (zero-knowledge), зокрема сімейства протоколів автентифікації Фіата–Шаміра [10–12], Шнорра [13] та багато інших. Безпека цих протоколів гарантується складністю задач факторізації або дискретного логарифму для великих чисел.

Однак всі зазначені методи мають суттєві недоліки, що стосуються обчислювальних та безпекових аспектів — або велику кількість комунікацій, або відносно довгий час оброблення, а також вразливість до багатьох атак типу «людина посередині», включаючи атаку дублювання (replay attack). Крім того, як було доведено П. Шором, базові теоретико-числові проблеми, на складності розв'язання яких ґрунтується безпека алгоритмів, зокрема пошук множників заданого числа та дискретного логарифму в деяких числових групах, зводяться до задачі пошуку періоду степеневої функції. Цю задачу можна досить просто розв'язати в алгоритмічній моделі квантових обчислень [14]. Тому з появою промислових квантових комп'ютерів протоколи, які використовують припущення щодо неможливості обчислення базових теоретико-числових властивостей, вважатимуться вразливими. З іншого боку, в галузі криптографії з відкритими ключами добре розроблена технологія швидко обчислюваних односторонніх криптографічних геш-функцій, які за безпекових припущень не використовують теоретико-числові властивості. Завдяки цьому їх вважають безпечними навіть з появою квантових комп'ютерів.

Дві комунікації типу «запит-відповідь» — це традиційна особливість автентифікації типу «свій-чужий», що бере свій початок від визначення характеристик летючих об'єктів за допомогою спеціального оброблення радіолокаційних сигналів. Автор цієї роботи трактує поняття «свій-чужий» у ширшому розумінні, яке поширюється на коаліційні групи комп'ютерно керованих об'єктів. Зрозуміло, що в нинішню епоху розвитку комп'ютерних технологій використання традиційних незмінних таблиць типу «запит-відповідь» з константними або з тривалими у часі незмінними паролями не відповідає більшості безпекових вимог. Тому тут залишено форму традиційного рішення «свій-чужий» у двораундовій формі «запит-відповідь», але фактично за допомогою засобів криптографії з відкритими ключами закрито вміст кодових таблиць від читання сторонньою особою.

У цій роботі досліджено поняття цифрової геш-автентифікації типу «свій-чужий» для групи користувачів, яку названо коаліційною. Особливістю автентифікації «свій-чужий» є виконання двох швидких комунікаційних раундів верифікатора B з кожним членом групи A. Цього має бути достатньо для того, щоб верифікатор з великою ймовірністю розпізнав прувера як «свого», тобто такого, що належить коаліційній групі. При цьому можливий перехоплювач не отримує жодної інформації про рішення B і не може імітувати дії A. Своєю чергою, навіть верифікатор B не може обчислити ідентифікаційні параметри A і симетрично A не може обчислити секретні параметри верифікатора B. За умови «чесного» верифікатора така автентифікація належить класу взаємодій «доведення з нульовим розголошенням». Рішення «свій» означає, що A з імовірністю $1 - \varepsilon$ дійсно належить групі, де ε — нехтовно мале значення. Якщо веріфікатор B виробляє рішення «чужий», то це означає, що A або не знає потрібних параметрів належності групі, тобто він «чужий», або можливо він «свій», але інформація в його відповіді була спотворена перехоплювачем (хибно-негативне рішення).

Схеми цифрової ідентифікації тісно пов'язані зі схемами цифрових підписів. За допомогою перетворення Фіата–Шаміра будь-яку трираундову систему ідентифікації (Σ-протокол) у моделі випадкового оракула для геш-функцій (random oracle model) можна перетворити на цифровий підпис [15]. З іншого боку, в багатьох випадках двораундову ідентифікацію можна здійснювати у стандартний спосіб за допомогою цифрових підписів. Це роблять так. Верифікатор має публічний ключ прувера. Він надсилає випадкове повідомлення, прувер відповідає валідним цифровим підписом цього повідомлення. У такий спосіб прувер підтверджує наявність секретного ключа створення підпису. У разі групи користувачів цей спосіб має низку недоліків. Кожен користувач повинен мати змогу багатократно застосовувати свій власний цифровий підпис. Це створює значне навантаження на обчислювальні ресурси підписантів і верифікатора. Крім того цифровий підпис є публічним для перевірки, що не завжди є допустимим. Особливістю автентифікації «свій-чужий» є обмеження на верифікатора: тільки авторизований верифікатор може здійснювати групову автентифікацію.

У цій роботі запропоновано недетерміновану схему багаторазової групової двораундової геш-автентифікації типу «свій-чужий» з використанням модифікованої схеми одноразового підпису Вінтернітца (*W-OTS*) [16]. Кожен член групи створює унікальну частину підпису Вінтернітца. Веріфікатор перевіряє весь підпис.

КОАЛІЦІЙНА ГРУПА

Автентифікацію «свій-чужий» застосовують до членів коаліційної групи, які вважаються «своїми». Тут використано поняття коаліційної групи в такому розумінні.

Коаліційна група — це сукупність користувачів (процесів, пристроїв тощо), які об'єднані довірчими відносинами та взаємодіють між собою в комунікаційному кіберпросторі, якому загрожують атаки зловмисників.

Окремим випадком коаліційної групи є набір закодованих числових ознак одного об'єкта, який ідентифікують.

Коаліційна група передбачає наявність третьої сторони T, яка користується абсолютною довірою (Third Trusted Party, TTP) — контролера групи (це може бути штаб, банк, контролювальний сервер, особливий пристрій мережі IoT, тощо), з яким прувер A і верифікатор B у минулому мали захищену індивідуальну комунікацію. Контролер T генерує початкові таємні параметри (приватні ключі) окремо для кожного учасника схеми обмінів. Учасники A і B також можуть брати особисту участь у такій генерації. Програма T після взаємодій з A і B у разі потреби (яка визначається особливістю конкретних застосувань) «забуває» та знищує всі параметри своїх обмінів $(A \leftrightarrow T)$ і відповідно обмінів $(B \leftrightarrow T)$. Тому вважаємо, що витоків інформації від T не може бути. Надалі учасники A і B функціонують незалежно від T (швидше за все у вразливому сегменті кіберпростору) і спілкуються між собою тільки відкритими незахищеними каналами зв'язку. Наявність контролера T фактично визначає відносне поняття «свій».

Вимоги до протоколу швидкої автентифікації типу «свій-чужий». Сформулюємо такі досить сильні вимоги до автентифікації «свій-чужий».

1. Автентифікація виконується за два швидкі комунікаційні раунди. Інакше кажучи, сторона B (верифікатор) надсилає запит до A (прувера) і отримує швидку відповідь.

2. Сторона B нічого не знає і не може обчислити таємні ідентифікаційні параметри (секретні ключі) A і навпаки.

3. Система повинна бути стійкою до атак типу «людина посередині», а також до атак дублювання (replay attacks). Інакше кажучи, зловмисник, який контролює канал комунікацій від A до B, за умови, що він не володіє таємними ключами A, не може видати себе за A. Більш точно: ймовірність того, що зловмисник може видати себе за A без знання його таємного ключа, має бути нехтовно малою (negligibly small). Інакше кажучи, підміна A рівнозначна знанню зловмисником приватного ключа A. У протоколах з нульовим розголошенням ця вимога має назву «soundness».

З вимоги 3 відразу випливає вимога недетермінованості системи автентифікації «свій-чужий». Інакше кажучи, запити та відповіді не можуть повторюватись і мають бути непередбачуваними, бо інакше виникає можливість атаки дублюванням повідомлень (replay attack).

4. Прувер A може бути обчислювально «слабким» відносно верифікатора B. Особливо це стосується обмежень пам'яті для A. Наприклад, така ситуація типова для IoT, де сервер або призначений відносно потужний комп'ютер перевіряє багато спеціалізованих «слабких» пристроїв.

5. Якщо прувер належить коаліційній групі, то він не змінює процес обчислення своєї автентифікації. Інакше кажучи, він діє згідно з первинними інструкціями (програмою), які заклав контролер на етапі налаштування.

6. Тільки авторизований верифікатор B може автентифікувати A. Спосіб авторизації верифікатора B визначає контролер коаліції T. Інакше кажучи, він на етапі налаштування системи передає до B деякі дані, що залежать від таємних параметрів як A так і B, а також, можливо, самого T. Ці дані дозволяють тільки B однозначно автентифікувати A. Зауважимо, що цей пункт повинен корелювати з вимогою 2 про те, що B не може обчислити таємні параметри A.

7. Для взаємної автентифікації (B автентифікує A, при цьому A автентифікує B) вимоги є симетричними наведеним вище у разі заміни A на B та B на A.

ЦИФРОВИЙ ПІДПИС

Цифровий підпис $\Pi = (Gen, Sign, Vrfy)$ складається з двох імовірнісних поліноміальних за часом (PPT, Probabilistic Polynomial Time) алгоритмів: *Gen* — генерація ключів, *Sign* — створення підпису і детермінованого поліноміального за часом алгоритму *Vrfy*, який забезпечує перевірку відповідності підпису повідомленню.

1) *Gen.* **Вхід**: безпековий параметр 1^n.

Результат: пара (приватний (секретний) ключ для створення підпису sk, публічний ключ для перевірки відповідності підпису повідомленню pk), бітова довжина ключів не менша за n.

2) *Sign.* **Вхід:** sk, повідомлення m (число, рядок бітів).

Результат: підпис $\sigma = Sign_{sk}(m)$ — число, рядок бітів.

3) *Vrfy.* **Вхід:** pk, m σ.

Результат: $Vrfy_{pk}(m, \sigma) \in \{0, 1\}$.

Підпис вважають правильним (valid, legitimate), якщо $Vrfy_{pk}(m, \sigma) = 1$.

Для уніфікації довжин повідомлень, що підписуються, використовують підписи гешів від повідомлень, тобто підпис повідомлення m — це значення $\sigma = Sign_{sk}(H(m))$, де H — загальновідома фіксована криптографічна геш-функція.

Також вважаємо, що $Sign_{sk}(x, y) = Sign_{sk}H(x, y)$. Це не порушує безпекової властивості цифрових підписів. Доведення цього факту можна знайти в [15].

Безпеку підпису визначають через ігровий експеримент «ПІДРОБЛЕННЯ», який позначимо $Sigforge_{A^*,\Pi}(n)$ [15].

Експеримент $Sigforge_{A^*,\Pi}(n)$. Несанкціонований зловмисник A^* має публічний ключ перевірки підпису *pk*, за своїм бажанням створює повідомлення і може звертатися до оракула $Sign_{sk}(\cdot)$, який під час надання будь-якого повідомлення *m* із множини допустимих повідомлень видає правильні (валідні) підписи, що створюються згідно з протоколом П. Секретного ключа *sk* зловмисник не знає. Нехай Ω — множина повідомлень-звернень A^* до оракула. Зловмисник A^* виграє, тобто $Sigforge_{A^*,\Pi}(n) = 1$, тоді й тільки тоді, коли він знаходить таке повідомлення, що $Vrfy_{pk}(m, \sigma) = 1$ і при цьому $m \notin \Omega$.

Схему цифрового підпису П називають такою, що не піддається підробленню у разі атаки методом адаптивного підбору повідомлень (Existentially Unforgeable under an adaptive Chosen Message Attack, EU-CMA,) або просто безпечною, якщо для будь-якого ймовірнісного поліноміального алгоритму A^*, який виконує експеримент $Sigforge_{A^*,\Pi}(n)$, існує поліноміальна нехтовна функція (negligible function) *negl*(*n*) така, що імовірність вдалого експерименту для A^* буде меншою за *negl*(*n*), $\Pr[Sigforge_{A^*,\Pi}(n) = 1] \le negl(n)$.

З-поміж різноманітного сімейства цифрових підписів виділяють одноразові підписи (OneTimeSignatures, OTS). Така назва пов'язана з тим, що з форми підпису окремого повідомлення можна отримати інформацію, яка дає змогу побудувати підписи деяких інших повідомлень. Тому з міркувань безпеки схему підпису не можа використовувати двічі. У визначенні неможливості підроблення одноразового підпису в наведеному експерименті $Sigforge_{A^*,\Pi}(n)$ множина звернень Ω до оракула $Sign_{sk}(\cdot)$ обмежена разовим зверненням. Незважаючи на одноразовість, ці підписи виявилися дуже корисними у багатьох застосуваннях як криптографічні примітиви для побудови деяких криптографічних конструкцій. Зазвичай доведення неможливості підроблення схем одноразових підписів є простішим, ніж для багаторазових підписів.

Застосування геш-функцій для безпечних одноразових цифрових підписів та одноразової автентифікації вперше запропонував Л. Лемпорт у 1979 р. [17]. У схемі підпису Лемпорта підписують кожен біт повідомлення. Недоліки підпису Лемпорта — велика довжина підпису, а також велика кількість секретних та відповідних публічних ключів. Також незручною є властивість одноразовості підпису. Використовуючи дерево гешів Меркла, можна збільшити кількість повідомлень, що підписуються одноразовими схемами підпису типу підпису Лемпорта. Це буде одноразовий підпис Меркла (Merkle's signature) [18].

ОДНОРАЗОВИЙ ПІДПИС ВІНТЕРНІТЦА

Одноразовий підпис Вінтернітца (the Winternitz One Time Signature, *W – OTS*). Роберт Вінтернітц з математичного факультету Стенфордського університету через декілька місяців після появи роботи Лемпорта винайшов оригінальний варіант одноразового підпису, яким можна підписувати не окремі біти, а відразу блоки, що складаються з *w* бітів. Число *w* називають параметром Вінтернітца. Цей параметр дає змогу здійснити гнучку адаптацію протоколу підпису до обчислювальних та часових вимог. Якщо вибрати *w* великим, то

розмір підписів буде меншим, але при цьому збільшується час генерації ключів і перевірки відповідності підпису оригінальному повідомленню. Якщо $w=4$, $H=SHA-256$, то за умови виключення з повідомлення блоків, що складаються лише з одиниць, кількість ключів дорівнюватиме 64, максимальна кількість ітеративних гешувань у разі підписування блоків не перевищуватиме $2^4-1=15$. Для того, щоб під час кодування підпису гарантовано не виникали нульові ітерації гешування, можна збільшити на одиницю максимальну кількість ітерацій, що визначена числом w. Кількість ітеративних гешувань під час підписування блока не перевищуватиме 2^w. Тому максимальна кількість ітерацій для $w=4$, $H=$ SHA-256 становитиме $2^4=16$. Якщо $H=SHA-256$, $w=8$, то кількість ітеративних гешувань у разі підписування блоків не перевищує $2^8=256$.

Для пояснення підпису $W-OTS$ розглянемо найпоширеніший варіант $w=8$, $H=SHA-256$. Він забезпечує швидке байтове кодування підпису. Підписувач A вибирає 32 випадкових числа довжиною 256 бітів. Це приватний таємний ключ користувача A. Публічний ключ pk_A — це послідовність 32-х 256-бітових значень $pk_A=H^{256}(P_1), H^{256}(P_2),\ldots, H^{256}(P_{32})$. Нехай m — число у двійковому зображенні, $N=H(m)=N_1N_2\ldots N_{32}$ — розбиття двійкового запису 256-розрядного значення гешу N числа m на байти, $32\times 8=256$. Числове значення байта N_i позначаємо $\overline{N_i}$. $W-OTS$ підпис повідомлення m — це послідовність 32-х 256-розрядних чисел $Sign_{sk_A}(m)=$ $=H^{256-\overline{N_1}}(P_1), H^{256-\overline{N}_2}(P_2),\ \ldots\ H^{256-\overline{N}_{32}}(P_{32})$.

Перевірка відповідності підпису виконується за допомогою значення гешу $N=N_1N_2\ldots N_{32}$ і публічного ключа pk_A :

$$
\begin{aligned}
&pk_A=(H^{256}(P_1), H^{256}(P_2),\ldots, H^{256}(P_{32}))=\\
&=(H^{\overline{N_1}}(H^{256-\overline{N_1}}(P_1)))=H^{256}(P_1),\\
&H^{\overline{N_2}}(H^{256-\overline{N_2}}(P_2))=H^{256}(P_2),\\
&\qquad\ldots,\\
&H^{\overline{N_{32}}}(H^{256-\overline{N_{32}}}(P_{32}))=H^{256}(P_{32}).
\end{aligned}
$$

Підпис Вінтернітца одноразовий. Якщо $a>b$, то $H^{256-b}(\cdot)=H^{a-b}(H^{256-a}(\cdot))$. У цьому разі для будь-якого $b, b<a$, маючи a і $H^{256-a}(\cdot)$, можна обчислити $H^{256-b}(\cdot)$ і у такий спосіб створити підпис іншого повідомлення.

Зауважимо, що світ дізнався про підпис Вінтернітца у 1989 р. з доповіді Р. Меркла на одній із комп'ютерних конференцій [16]. Публікацій самого Вінтернітца на цю тему немає.

Є багато модифікацій підпису $W-OTS$, здебільшого пов'язаних із використанням спеціальних сімейств ітерувальних геш-функцій підпису блоків. Найбільш відомими є схеми $W-OTS^{\$}$ і $W-OTS^{+}$ [21, 22]. Ми вважаємо схему $W-OTS^{+}$ найбільш перспективною для автентифікації типу «свій-чужий». Але в цій роботі використано спрощений оригінальний варіант підпису Вінтернітца. Для підпису $W-OTS$ та його модифікацій доведено неможливість підроблення підпису у разі атаки методом адаптивного підбору повідомлень [19–22].

АВТЕНТИФІКАЦІЯ «СВІЙ-ЧУЖИЙ»

Основна ідея протоколу автентифікації «свій-чужий» за допомогою підпису Вінтернітца полягає в тому, що у процесі поточної автентифікації всі члени групи разом створюють один одноразовий підпис Вінтернітца випадкового повідомлення, який визначається верифікатором після отримання попередніх відповідей від усіх пруверів. Після цього всі члени групи змінюють секретні та відповідні публічні ключі підпису. Нові публічні ключі наступної сесії автентифікації передаються верифікатору. Тому підпис буде багаторазовим. У цьому полягає ідея ланцюгового підпису. Загальна методика перетворення одноразового підпису в багаторазовий ґрунтується на використанні цифрових підписів зі станами і детально описана в [15]. У конкретному випадку автентифікації типу «свій-чужий» за допомогою підпису Вінтернітца ситуація значно спрощена.

Учасник A_i з порядковим номером i підписує схемою $W-OTS$ тільки окремий i-й блок повідомлення, який містить w бітів. Правильність усього підпису визначається правильністю підпису кожного w-блока. Верифікатор знає своє повідомлення. Отримавши поточні публічні ключі всіх учасників групи, він має змогу перевірити правильність усього сесійного $W-OTS$ підпису.

Нехай маємо перенумеровану коаліційну групу з m учасників $G=\{A_1, A_2,\ldots, A_m\}$, а верифікатора позначимо B. Це може бути відокремлений від групи G об'єкт або член групи. Всі члени групи G доводять верифікатору B, що знають секретні індивідуальні ключі підпису, які одночасно є ідентифікаційними параметрами членів групи. Фіксуємо параметр Вінтернітца w. Він визначає максимальну довжину L повідомлень, що підписуються, $L=m\times w$. Відповідно цим визначено максимальну кількість учасників групи, яка не повинна перевищувати $\frac{L}{w}$. У разі вибору $m=\frac{L}{w}$ кожен учасник має принаймні один ключ. У спрощеному варіанті вважаємо, що w є дільником числа 256, тобто $L=256$. Наприклад, для w= 4 кількість учасників може досягати $\frac{256}{w}=64$. Узагальнення на випадок довільної кількості учасників групи є очевидним.

Нехай H — загальновідома геш-функція, наприклад H = SHA-256. Кожний член групи A_i має початковий секретний ключ $sk_i^0=P_i^0$ — випадкове число довжиною 256 бітів. Його початковий публічний ключ pk_i^0 — це пара значень (порядковий номер i, публічний ключ $H^{2^w}(P_i^0)$). Інакше кажучи, учасник A_i знає свій несекретний порядковий номер i, а також індивідуальне секретне ідентифікаційне значення P_i^0. Початковий безпечний розподіл секретних ключів P_i^0 здійснює ТТР T.

Розглянемо протокол автентифікації «свій-чужий».

Раунд 0. Початкове налаштування схеми. Контролер групи T здійснює початкове налаштування системи. Він генерує і розсилає у безпечний спосіб стартові ключі для створення $W-OTS$ підпису. Вважаємо, що кожний член групи отримує один секретний ідентифікаційний ключ. Узагальнення на випадок наявності багатьох ключів для одного учасника групи є очевидним. Учасник групи надсилає верифікатору початкові публічні ключі членів групи P_i^0 разом з їхніми порядковими номерами.

Сесія автентифікації проводиться у два раунди: запит/відповідь. Нехай t — номер поточної сесії автентифікації, $t=0,1,2,3,\ldots$.

Раунд 1. Запит. Верифікатор B починає чергову t–автентифікацію, $t = 0, 1, \ldots$.

Далі він вибирає випадкове число x_t, обчислює

$$N_t = H(x_t, pk_1^t, pk_2^t, \ldots, pk_m^t), \; N_t = N_1^t \| N_2^t \| \ldots \| N_m^t,$$

де N_i^t — блок з w бітів, $i = 1, 2, \ldots, m$; H — загальновідома геш-функція, $H: \{0,1\}^* \to \{0,1\}^{m \times w}$. Методом широкомовлення (мультимовлення) верифікатор пересилає всім членам групи результівний геш N_t разом зі своїм ім'ям (координатами для комунікацій під час відповіді).

Раунд 2. Відповідь. Учасник групи A_i створює нові ключі (секретний та відповідний публічний) для наступної сесії автентифікації,

$$sk_i^{t+1} = H(N_t, sk_i^t); \; pk_{i,new}^{t+1} = H^{2^w}(sk_i^{t+1}).$$

Далі він знаходить у повідомленні N_t свій відповідний i-й w-блок N_i^t, за допомогою свого поточного секретного ключа sk_i^t підписує згідно з $W-OTS$ цей w-блок разом із публічним ключем наступної сесії,

$$y_i = Sign_{sk_i^t}(N_i^t, pk_{i,new}^{t+1}) = H^{2^w - \overline{B_i^t}}(sk_i^t),$$

де B_i^t — i-й блок гешу $H(N_i^t, pk_{i,new}^{t+1})$. Потім цей учасник пересилає верифікатору відповідь $(i, y_i, pk_{i,new}^{t+1})$.

Схема перевірки є такою. Верифікатор сортує отримані відповіді згідно з порядковими номерами членів групи, за згенерованим повідомленням x_t і сукупним набором поточних публічних ключів pk_i^t, $i = 1, \ldots, m$, отриманих на попередньому етапі, перевіряє валідність підпису Вінтернітца кожного блока N_t.

Наведемо формальний опис протоколу «свій-чужий». Уведемо такі позначення.

Запис $x = y$ означає присвоювання поточного значення y змінній x.

Запис $x \leftarrow random$ означає, що змінна x отримала випадкове значення з рівномірним розподілом із деякого визначеного інтервалу значень.

У разі передавання інформації запис $U \to V: x$ означає пересилання по відкритому каналу зв'язку об'єктом U повідомлення x до V. Елемент x може мати складну числову структуру (масив, множина чисел, множина елементів масиву, тощо). У розглядуваному випадку маємо справу тільки з натуральними числами (рядками в алфавіті {0, 1}). Часто отримувач V повинен знати ім'я відправника. У цьому разі до повідомлення x додають ім'я відправника. Тоді запис команди має вигляд $U \to V: x, U$. У разі такого пересилання адресат V отримує ім'я (координати) відправника.

Захищене передавання (по захищеному каналу або в інший безпечний спосіб) позначаємо двома стрілками: $U \rightrightarrows V: x$. Це означає, що значення x не доступно сторонній особі.

Широкомовлення (broadcasting, multicasting) позначимо $U \sim\!\!\to G: x$.

Це безадресне передавання x усім членам групи G. Члени групи, які отримали широкомовну інформацію x, не можуть знати, хто передав інформацію, якщо в повідомленні не вказано ім'я відправника.

Якщо $f(x)$ — функція, то запис $f^t(x) = f(f(\ldots(x)\ldots))$ означає ітеративне застосування функції f до аргументу x t разів.

Конкатенацію слів x та y позначаємо $x \| y$.

У цій роботі в описі алгоритму не використана конкретна мова програмування. Її застосовують на етапі конкретного впровадження. Скористаємося загальновідомими умовними та ітеративними конструкціями. В окрему змінну *результат* записуємо результат обчислень. Під час виконання будь-якого присвоювання змінній *результат* робота процедури зупиняється і видається значення цієї змінної.

Протокол групової автентифікації «свій-чужий». Дійові особи: коаліційна група $G = \{A_1, A_2, \ldots, A_m\}$, T — контролер групи (ТТР), B — верифікатор, sk_i^t — секретний ключ A_i, pk_i^t — відповідний публічний ключ сесії t, $pk_{i,new}^{t+1}$ — публічний ключ для наступної сесії, який виробляє особа A_i у сесії t.

Геш-функція H — це функція, яка задовольняє вимоги безпеки підпису W-OTS. Вважаємо, що H= SHA-256 або SHA-512, параметр Вінтернітца w — загальновідомі параметри, $t = 0, 1, \ldots$.

Протокол «свій-чужий» наведено нижче.

Т

ДЛЯ ВСІХ $A_i \in G$ РОБИТИ

$\{P_i^0 \leftarrow random;$

$sk_i^0 = P_i^0;\ pk_i^0 = H^{2^w}(P_i^0);\ T \rightrightarrows A_i: sk_i^0,\ pk_i^0;$

$T \rightrightarrows B: pk_i^0\}$

B СЕСІЯ t $(t = 0, 1, 2, \ldots)$

$\{x_t \leftarrow random;$

$N_t = H(x_t;\ pk_1^t, pk_2^t, \ldots, pk_m^t) = N_1^t \| N_2^t \| \ldots \| N_m^t;\ B \sim\!\!\rightarrow G: N_t, B\}$

A_i СЕСІЯ t $(t = 0, 1, 2, \ldots)$

ПРИЙНЯТИ $(N_t = N_1^t \| N_2^t \| \ldots \| N_m^t, B);$

$sk_i^{t+1} = H(N_t, sk_i^t);\ pk_{i,new}^{t+1} = H^{2^w}(sk_i^{t+1});$

$y_i = Sign_{sk_i^t}(N_i^t,\ pk_{i,new}^{t+1}) = H^{2^w - \overline{B_i^t}}(sk_i^t);$

$//\ H(N_i^t,\ pk_{i,new}^{t+1}) = B_1 \| B_2 \| \ldots \| B_m^n\ //$

$A \rightarrow B: (i,\ y_i,\ pk_{i,new}^{t+1})$

$B(Vrfy)$ СЕСІЯ t $(t = 0, 1, 2, \ldots)$

{ДЛЯ ВСІХ i РОБИТИ

$$\{\text{ПРИЙНЯТИ}(i,\ y_i,\ pk\ {}^{t+1}_{i,new});$$

$$//\ H(N_i^t,\ pk_{i,new}^{t+1}) = B_1\ \|B_2\ \|\ldots\|\ B_m\ //$$

$$\text{ЯКЩО}$$

$$(H^{\overline{B}_i^t}(y_i) \neq pk_i^t)$$

$$\text{ТО}\ \ результат\ =\ \text{«чужий»}\}$$

$$pk\ {}^{t+1}_{i} = pk\ {}^{t+1}_{i,new};\ результат\ =\ \text{«свій»}\}$$

Безпека протоколу. По суті наведений протокол автентифікації «свій-чужий» у кожній сесії t є розпаралелюванням одного модифікованого $W-OTS$ підпису повідомлення $N_t = H(x_t,\ pk_1^t,\ pk_2^t,\ldots,\ pk_m^t)$. Тому неможливість підроблення всього групового підпису випливає з неможливості підроблення одноразового підпису Вінтернітца. Це визначає безпеку протоколу автентифікації в кожній окремій сесії. Інакше кажучи, кожен член групи A_i за допомогою одноразового $W-OTS$ підпису доводить знання секретного сесійного ключа $sk_i^{\ t}$.

Публічні ключі поточної сесії t, $t \geq 1$, надходять від кожного A_i у попередній сесії $t-1$ і в сесії t зв'язуються геш-запитом $N_t = H(x_t,\ pk_1^t,\ pk_2^t,\ldots,\ pk_m^t)$, якщо A_i в сесії $t-1$ довів, що він «свій». Це конструкція блокчейну. Тому підроблення підпису в поточній сесії t «людиною посередині» можливе тільки за умови підміни всіх початкових генезисних ключів $sk_i^{\ 0}$, оскільки кожен сесійний секретний ключ $sk_i^{\ t}$ і відповідно публічний ключ pk_i^t однозначно визначаються історією $sk_i^0,\ x_0,\ldots,x_{t-1}$, а початкові ключі розповсюджував контролер групи безпечним шляхом.

Зауважимо, що x_t не передається. Тому за умови, що H — криптографічна геш-функція, навіть у разі перехоплення всіх публічних ключів по $N_y = H(x_t,\ pk_1^t,\ldots,\ pk_m^t)$ відновити $H(x_t,\ pk_1^t,\ pk_2^t,\ldots,\ pk_m^t)$ шляхом обчислень неможливо.

Крім того, якщо верифікатор вибирає запити випадково з рівномірним розподілом («чесний верифікатор»), то в моделі випадкового оракула геш-відповіді також вважаються випадковими з рівномірним розподілом. Тому існує незалежна від параметрів протоколу «свій-чужий» PPT-машина Тюрінга (алгоритм), яка моделює обміни між верифікатором і членами групи. Через це в моделі випадкового оракула цей протокол належить класу протоколів «доведення з нульовим розголошенням за умови чесного верифікатора» ("honest verifier zero-knowledge").

ВИСНОВКИ

Двораундова швидка цифрова автентифікація «свій-чужий» утворює важливий підклас процедур ідентифікації з доведенням і має широке коло застосувань від IoT та ройових розпізнавань «свій-чужий» до ранжування доступу до складних об'єктів критичної інфраструктури. Використання геш-підпису Вінтернітца дає змогу побудувати швидкий і доказово безпечний протокол зазначеної автентифікації, який належить класу криптографічних постквантових протоколів з нульовим розголошенням.

СПИСОК ЛІТЕРАТУРИ

1. Needham R., Schroeder M. Using encryption for authentication in large networks of computers. *Commun. ACM.* 1978. Vol. 21, Iss. 12. P. 993–999. https://doi.org/10.1145/359657.359659.
2. Diffie W., Hellman M. New directions in cryptography. *IEEE Trans. Inf. Theory.* 1976. Vol. 22, Iss. 6. P. 644–654. https://doi.org/10.1109/TIT.1976.1055638.
3. Menezes A., Qu M., Vanstone S. Key agreement and the need for authentication. Presentation at PKS ‘95. 1995. Toronto, Canada.
4. Law L., Menezes A., Qu M., Solinas J., Vanstone S. An efficient protocol for authenticated key agreement. *Designs, Codes and Cryptography*. 2003. Vol. 28, N 2. P. 119–134. https://doi.org/10.1023/A:1022595222606.
5. Kaliski B. An unknown key-share attack on the MQV key agreement protocol. *ACM Transactions on Information and System Security*. 2001. Vol. 4, N 3. P. 275–288. https://doi.org/10.1145/501978.501981.
6. Menezes A., Ustaoglu B. On the importance of public-key validation in the MQV and HMQV key agreement protocols. *Proc. 7th International Conference on Cryptology “Progress In Cryptology — INDOCRYPT 2006”* (11-13 December 2006, Kolkata, India). Kolkata, 2006. LNCS. 2006. Vol. 4329. P. 133–147. https://doi.org/10.1007/11941378_11.
7. Krawczyk H. HMQV: a high-performance secure Diffie–Hellman protocol. *Proc. 25th Annual International Cryptology Conference “CRYPTO 2005”* (14–18 August 2005, Santa Barbara, California, USA). Santa Barbara, 2005. LNCS. 2005. Vol. 3621. P. 546–566. https://doi.org/10.1007/11535218_33.
8. Menezes A. Another look at HMQV. *Journal of Mathematical Cryptology*. 2007. Vol. 1, №. 1. P. 47–64. https://doi.org/10.1515/JMC.2007.004.
9. Hao F. On robust key agreement based on public key authentication. *Proc. 14th International Conference on Financial Cryptography and Data Security* (25-28 January 2010, Tenerife, Spain). Tenerife, 2010. LNCS. 2010. Vol. 6052. P. 383–390. https://doi.org/10.1007/978-3-642-14577-3_33.
10. Fiat A., Shamir A. How to prove yourself: practical solutions to identification and signature problems. *Proc. Conference in the Theory and Application of Cryptographic Techniques ”Advances in Cryptology — CRYPTO 1986”* (11–15 August 1986, Santa Barbara, USA). Santa Barbara, 1986. LNCS. 1987. Vol. 263. P.186–194. https://doi.org/10.1007/3-540-47721-7_12.
11. Fiege U., Fiat A., Shamir A. Zero knowledge proofs of identity. *Proc. 19th Annual ACM Symposium on Theory of Computing (STOC ‘87)* (25–27 May 1987, New York, USA). New York, 1987. P. 210–217. https://doi.org/10.1145/28395.28419.
12. Guillou L., Quisquater P. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. *Proc. Workshop on the Theory and Application of of Cryptographic Techniques “Advances in Cryptology (EUROCRYPT ‘88)”* (25–27 May 1988, Davos, Switzerland). Davos, 1988. LNCS. 1988. Vol. 330. P. 123–128. https://doi.org/10.1007/3-540-45961-8_11.
13. Schnorr C.P. Efficient signature generation by smart cards. *Journal of Cryptology*. 1991. Vol. 4, № 3. P. 161–174. https://doi.org/10.1007/BF00196725.
14. Shor P.W. Algorithms for quantum computation: Discrete logarithms and factoring. *Proc. 35th Annual Symposium on Foundations of Computer Science* (20-22 November 1994, Santa Fe, NM, USA). Santa Fe, 1994. P. 124–134. https://doi.org/10.1109/sfcs.1994.365700.

15. Katz J., Lindell Y. Introduction to Modern Cryptography. Second Edition. New York: Chapman and Hall/CRC, 2015. 603 p. https://doi.org/10.1201/b17668.

16. Merkle R.C. A certified digital signature. *Proc. Conference on the Theory and Application of Cryptology "Advances in Cryptology — CRYPTO'89"* (20-24 August 1989, Santa Barbara, CA, USA). Santa Barbara, 1989. LNCS. 1990. Vol. 435. P. 218–238. https://doi.org/10.1007/0-387-34805-0_21.

17. Lamport L. Constructing Digital Signatures from a One Way Function. Technical Report. Computer Science Laboratory, SRI International, 1979. SRI-CSL-98. 8 p. URL: https://lamport.azurewebsites.net/pubs/dig-sig.pdf.

18. Merkle R.C. Secrecy, Authentication and Public Key Systems: Ph.D. Thesis. Stanford: Stanford University, 1979. 187 p. URL: https://www.ralphmerkle.com/papers/Thesis1979.pdf.

19. Hevia A., Micciancio D. The provable security of graph-based one-time signatures and extensions to algebraic signatures chemes. *Proc. 8th International Conference on the Theory and Application of Cryptology and Information Security "Advances in Cryptology — ASIACRYPT 2002"* (1–5 December 2002, Queenstown, New Zealand). Queenstown, 2002. LNCS. 2002. Vol. 2501. P. 379–396. https://doi.org/10.1007/3-540-36178-2_24.

20. Dods C. Smart N., Stam M. Hash based digital signatures chemes. *Proc. 10th IMA International Conference "Cryptography and Coding"* (19-21 December 2005, Cirencester, UK). Cirencester, 2005. LNCS. 2005. Vol. 3796. P. 96–115. https://doi.org/10.1007/11586821_8.

21. Buchmann J., Dahmen E., Ereth S., Hulsing A., Ruckert M. On the security of the Winternitz one-time signature scheme. *Proc. 4th International Conference on Cryptology "Progress in Cryptology — Africacrypt 2011"* (5–7 July 2011, Dakar, Senegal). Dakar, 2011. LNCS. 2011. Vol. 6737. P. 363–378. https://doi.org/10.1007/978-3-642-21969-6_23.

22. Hulsing A. W-OTS+ shorter signatures for hash-based signature schemes. *Proc. 6th International Conference on Cryptology "Progress in Cryptology* — AFRICACRYPT 2013" (22–-24 June 2013, Cairo, Egypt). Cairo, 2013. LNCS. 2013. Vol. 7918. P. 173–188. https://doi.org/10.1007/978-3-642-38553-7_10.

A.V. Anisimov

DIGITAL AUTHENTICATION "FRIEND-OR-FOE"

Abstract. Based on a modified one-time Winternitz signature scheme, we develop a multi-time two-round group authentication protocol of the type "friend-or-foe." The main construction is as follows. At each authentication session, members of a group sign only designated *w*-blocks of a random message. The verifier checks the validity of the whole Winternitz signature. Session public keys are created and sent to the verifier at the previous session. This way, they form a hash-connected blockchain. Security of the Winternitz signature and blockchain structure of public keys imply the security of the suggested protocol. A trusted third party is needed for establishing first "genesis" keys. Also, the protocol has the property "honest verifier zero knowledge."

Keywords: authentication, coalition group, digital signature, the Winternitz signature, blockchain.

Надійшла до редакції 27.12.2023