УДК 004.056.53; 004.051

В. Яковина, О. Одуха
Національний університет “Львівська політехніка”,
кафедра програмного забезпечення

ВДОСКОНАЛЕНИЙ ПРОТОКОЛ ВЗАЄМНОЇ АУТЕНТИФІКАЦІЇ НА ОСНОВІ СМАРТ-КАРТ ДЛЯ ПОБУДОВИ МОДУЛЯ ЗАХИСТУ РОЗПОДІЛЕНОЇ СИСТЕМИ ТЕПЛОВОГО ПРОЕКТУВАННЯ

Розроблено покращений протокол взаємної аутентифікації з використанням смарт-карт для отримання доступу до ресурсів серверної частини системи теплового проектування. Проаналізовано захищеність протоколу та показано його переваги над існуючими аналогами. Запропонована схема забезпечує більший захист від можливих атак за рахунок незначного зменшення швидкодії.

The improved smart card based mutual authentication protocol for client-server thermal design system has been developed in the present paper. The security analysis of the protocol has been carried out and the advantages comparing to existing protocols have been shown. The proposed scheme not only preserves all the advantages of previous schemes but also is more secured against some kinds of attacks while has slightly less efficiency.

Вступ

Аутентифікація користувача є особливо важливою в розподіленому комп’ютерному середовищі. Колись схеми аутентифікації були засновані переважно на паролях, а в останнє десятиліття починають відігравати значну роль смарт-карти [1–7].

Починаючи з 1981 р., відколи Л. Лампорт [1] запропонував схему аутентифікації віддаленого користувача при використанні незахищеного каналу передачі даних, більше десятка робіт (див., напр., [2–6]) було присвячено вдосконаленню його захищеності, функціональності та ефективності.

Безпека таких схем вдосконалювалась декількома шляхами. Традиційно, якщо користувач хоче зареєструватись в комп'ютерній системі, він повинен надати ідентифікатор та відповідний пароль системі. Надалі система порівнює отримані дані з даними, що зберігаються у файлі паролів. Оскільки паролі зберігаються у вигляді відкритого тексту, такий підхід, очевидно, є вразливим до викрадення паролю. Для того, щоб позбутися атаки на відкритий файл паролів, було запропоновано [8, 9] перетворити файл паролів на таблицю верифікації. Така таблиця містить значення односторонньої хеш-функції від паролів користувачів. Така схема забезпечує захист паролів навіть у випадку викрадення таблиці верифікації [8, 9]. Однак наявність таблиці верифікації (на диску та в пам'яті) залишає можливою модифікацію даних, атаку на основі бази значень хешу та збільшує вартість захисту та супроводу таблиці [10]. Відтак були запропоновані схеми, що не використовували таблиць верифікації [3, 4, 6, 7, 10]. У таких схемах поширені мітки часу чи порядкові номери для протидії атакам повтором (див. напр. [10]). На жаль, донедавна більшість таких схем забезпечували тільки односторонню аутентифікацію, тоді як на сучасному етапі розвитку комунікаційних технологій та Інтернету вимагаються двосторонні схеми аутентифікації (як користувача, так і сервера). Крім того, перевагу мають протоколи аутентифікації, що одночасно забезпечують обмін сеансовими ключами шифрування.

Завдяки своїй низькій вартості, компактності та криптографічним можливостям, смарт-карти знайшли широке застосування в багатьох додатках електронної комерції, протоколах мережевої безпеки та схемах віддаленої аутентифікації. Наступні критерії [7] повинні враховуватись для розроблення протоколу аутентифікації користувача з використанням смарт-карт:

• відсутність таблиць верифікації: на сервері не повинно зберігатись жодних таблиць паролів чи верифікації;

• вільний вибір пароля: користувачі можуть вільно вибирати свої власні паролі;

• низьке комунікаційне та обчислювальне навантаження: завдяки обмеженням обчислювальної потужності смарт-карт вони можуть не забезпечити високих обчислювальних можливостей при широкій смузі пропускання;

• взаємна аутентифікація: користувачі і сервери повинні взаємно аутентифікувати один одного.

Нещодавно [4, 6] було представлено декілька подібних протоколів ефективного обміну ключами, заснованому на парольній аутентифікації з використанням смарт-карт. Це протоколи аутентифікації користувача на основі смарт-карт, в яких після успішної аутентифікації користувачі отримують спільний таємний ключ для подальшого забезпечення конфіденційності передачі даних. Подібними ці протоколи є тим, що вони включають етапи реєстрації, входу в систему і аутентифікації; використовують схожим чином паролі та смарт-карти та є еволюцією протоколу аутентифікації, описаного в [1].

Однак навіть найновіші запропоновані протоколи [4, 6] не позбавлені недоліків і можуть бути скомпрометовані різними способами. Зокрема, криптоаналіз протоколу [4] показав, що можливі такі атаки на цей протокол [5]:

• підбір пароля;

• атака з компрометацією ключа;

• використання сценарію зв'язаних ключів;

• колізії в хеш-функціях,

тоді як протокол, запропонований в [6], використовує мітки часу (хоча і не вимагає синхронізації годинників), що не завжди є припустимим в реальних системах [5, 11].

Отже, метою цієї роботи є розроблення вдосконаленого протоколу віддаленої аутентифікації на основі смарт-карт для авторизації користувачів у клієнт-серверній системі теплового проектування, який позбавлений деяких недоліків, притаманних існуючим протоколам.

Огляд та аналіз існуючих підходів

Схема Жуанга [4]. На етапі реєстрації користувач U_i надає свій ідентифікатор ID_i та пароль PW_i серверу S для реєстрації. Якщо дані прийняті, S здійснює такі кроки:

1. Обчислює таємну інформацію користувача U_i у такій формі $v_i = h(ID_i, x)$ та значення $w_i = v_i \oplus PW_i$, де x – довготривале таємне значення сервера.

2. Зберігає значення ID_i та w_i в пам'яті смарт-карти користувача U_i.

Коли користувач хоче отримати доступ до ресурсів сервера, він здійснює етап входу в систему та обміну ключів, під час якого вставляє смарт-карту в пристрій читання, вводить свій ідентифікатор ID_i та пароль PW_i. Під час j-го входу в систему користувача виконуються такі кроки:

1. $U_i \to S$: $N_1, ID_i, E_{vi}\left(ru_j, h(ID_i \| N_1)\right)$
2. $S \to U_i$: $E_{vi}\left(rs_j, N_1 + 1, N_2\right)$
3. $U_i \to S$: $E_{kj}\left(N_2 + 1\right)$

тут $E_k(m)$ означає симетричне шифрування повідомлення m секретним ключем k, h – одностороння функція хешування, $\oplus$ – побітове виключне АБО, $\|$ – конкатенація рядків, ru_j та rs_j – j-ті випадкові значення, генеровані користувачем U_i та сервером S відповідно, а N_1 та N_2 – оказії.

На першому кроці смарт-карта користувача U_i спочатку обчислює $v_i = w_i \oplus PW_i$ та надсилає свій ідентифікатор ID_i, оказію N_1 та шифроване повідомлення $E_{vi}\left(ru_j, h(ID_i \| N_1)\right)$ сер-

веру S. Після отримання цих даних на другому кроці сервер S обчислює $v_i = h(ID_i, x)$ і розшифровує $E_{vi}(ru_j, h(ID_i \| N_1))$, щоб перевірити, чи повідомлення містить дійсне значення $h(ID_i \| N_1)$ і чи оказія N_1 є новою (неповторюваною), в іншому випадку вхід в систему відхиляється. Якщо умови виконуються, сервер S повертає шифроване повідомлення $E_{vi}(rs_j, N_1 + 1, N_2)$ користувачеві U_i. Після отримання відповіді, на третьому кроці, смарт-карта користувача U_i розшифровує повідомлення і перевіряє наявність в ньому правильного значення $N_1 + 1$, після чого обчислює j-й сеансовий ключ як $k_j = h(rs_j, ru_j, v_i)$ і повертає шифроване повідомлення $E_{kj}(N_2 + 1)$ серверу S. Оскільки сервер S також знає значення, необхідні для обчислення k_j, він може розшифрувати повідомлення для перевірки, чи містить воно коректне значення $N_2 + 1$.

Як було зазначено вище, в роботі [5] проведено аналіз цього протоколу та показано такі вразливі місця:

- підбір пароля (противник намагається вгадати пароль та ввести його у смарт-карту);
- атака з компрометацією ключа (ця атака відповідає випадку, коли довготривале таємне значення сервера x виявляється скомпрометованим, що дає змогу противнику обчислити попередні сеансові ключі; для протидії цій загрозі в [5] пропонується обчислення сеансового ключа за схемою типу Діффі–Хеллмана, наприклад, $k_j = h(rs_j^Y, v_i) = h(ru_j^X, v_i)$, де X та Y – випадкові числа, обрані користувачем U_i та сервером S відповідно);
- використання сценарію зв'язаних ключів (зловмисник може отримати результати шифрування з використанням невідомих, однак зв'язаних функціональною залежністю ключів, і використати їх для полегшення атаки на шифровану схему; так, якщо в протоколі [4] зловмисник введе інший пароль $PW_i' = PW_i \oplus \Delta$ з відомим Δ, тоді секретний ключ, обчислений смарт-картою, буде $v_i' = w_i \oplus PW_i' = w_i \oplus PW_i \oplus \Delta = v_i \oplus \Delta$, тобто пов'язаним з v_i відомим співвідношенням, хоча як v_i, так і v_i' залишаються невідомими);
- колізії в хеш-функціях (існують пари ID_i, N_1 та ID_i', N_1' – ці значення можуть мати іншу бітову довжину – такі, що $h(ID_i \| N_1) = h(ID_i' \| N_1')$).

Крім цих зауважень, у [6] піддається критиці відсутність в протоколі Жуанга можливості перевірки цілісності відправлених повідомлень, що може бути використане зловмисником, наприклад, для модифікації значення ru_j, яка не буде виявлена, поки значення оказій залишаються узгодженими. Крім того, другий блок шифрованого тексту повідомлення $E_{vi}(ru_j, h(ID_i \| N_1))$ та відповідний відкритий текст $h(ID_i \| N_1)$ відомі зловмиснику. Отже, секретний ключ піддається атаці з відомим відкритим текстом. Крім того, достатньо пізнє відтворення першого повідомлення схеми [4] одразу не розпізнається сервером як атака повтором, сервер повинен очікувати третього повідомлення схеми для розпізнавання атаки повтором. Тому сервер не повинен дозволяти входу користувача в систему до отримання коректного третього повідомлення [6].

Схема Ші–Ванга [6]. У цій схемі для усунення недоліків попереднього протоколу пропонується замість оказій використовувати мітки часу, однак без необхідності синхронізації годинників. Відтак схема Ші–Ванга виглядає так. Фаза реєстрації повторює схему Чена та ін. [7]: користувач U_i через захищений канал надає свій ідентифікатор ID_i та пароль PW_i серверу S для реєстрації. Якщо дані прийняті, сервер S обчислює $R_i = h(ID_i \oplus x) \oplus PW_i$ (умовні позначення відповідають описаним вище позначенням протоколу Жуанга) та видає користувачеві U_i смарт-карту, що містить значення R_i та реалізацію алгоритму хешування $h(\ )$.

Фаза входу в систему та обміну ключем містить обмін трьома повідомленнями. Коли користувач хоче отримати доступ до ресурсів сервера, він вставляє смарт-карту в пристрій читання, вводить свій ідентифікатор ID_i та пароль PW_i. Смарт-карта здійснює такі кроки, щоб розпочати сеанс доступу:

1. Обчислює $a_i = R_i \oplus PW_i$.

2. Отримує значення поточної мітки часу T_u, зберігає це значення до кінця сеансу та обчислює значення коду автентичності повідомлення $MAC_u = h(T_u \| a_i)$.

3. Надсилає перше повідомлення (ID_i, T_u, MAC_u) серверу та очікує його відповіді. Якщо відповідь не отримано в заданий проміжок часу або ж відповідь є некоректною, користувачеві видається повідомлення про помилку входу в систему і сеанс закінчується.

Після отримання повідомлення (ID_i, T_u, MAC_u) від U_i сервер S здійснює такі кроки, щоб переконатись у цілісності повідомлення, дати відповідь користувачеві U_i та надати запит користувачеві для запобігання атакам повтором:

1. Перевіряє новизну значення T_u. Якщо отримане значення вже використовувалось в поточному сеансі користувача U_i, сервер відхиляє вхід цього користувача в систему і завершує сеанс. В іншому випадку значення T_u є новим.

2. Обчислює значення $a_i' = h(ID_i \oplus x)$, $MAC_u' = h(T_u \| a_i')$ та перевіряє рівність обчисленого значення MAC_u' та отриманого значення MAC_u. Якщо вони не дорівнюють один одному, вхід користувача в систему відхиляється, і сеанс завершується.

3. Отримує поточне значення мітки часу T_s. Для перевірки новизни зберігає до кінця сеансу пару міток часу (T_u, T_s) та ID_i. Обчислює $MAC_s = h(T_u \| T_s \| a_i')$ і сеансовий ключ $K_s = h((T_u \| T_s) \oplus a_i')$, після чого надсилає друге повідомлення (T_u, T_s, MAC_s) користувачеві U_i та очікує його відповіді. Якщо відповідь не отримано в заданий проміжок часу або ж відповідь є некоректною, вхід користувача в систему відхиляється, і сеанс завершується.

Після отримання повідомлення (T_u, T_s, MAC_s) від сервера, смарт-карта здійснює такі кроки для забезпечення аутентифікації сервера, обміну сеансовим ключем та відповіді серверу:

1. Перевіряє відповідність отриманого значення T_u збереженому значенню для забезпечення неповторюваності отриманого повідомлення. Якщо значення не збігаються, користувачеві видається повідомлення про помилку входу в систему і сеанс закінчується.

2. Обчислює значення $MAC_s' = h(T_u \| T_s \| a_i)$ та перевіряє його відповідність отриманому значенню MAC_s. Якщо значення не збігаються, користувачеві видається повідомлення про помилку входу в систему, і сеанс закінчується. В іншому випадку робиться висновок, що відповідь прийшла від чинного сервера.

3. Обчислює значення $MAC_u'' = h(T_s \| (a_i + 1))$ і сеансовий ключ $K_s = h((T_u \| T_s) \oplus a_i)$, після чого надсилає третє повідомлення (T_s, MAC_u'') серверу S, яке є відповіддю на запит сервера, здійснений в другому повідомленні протоколу.

Після отримання третього повідомлення протоколу від користувача U_i, сервер S здійснює такі кроки для аутентифікації користувача та обміну сеансовим ключем:

1. Перевіряє відповідність отриманого значення T_s збереженому значенню. Якщо значення не збігаються, вхід користувача в систему відхиляється, і сеанс завершується.

2. Обчислює значення $MAC_u''' = h\left(T_s \| (a_i'+1)\right)$ та перевіряє його відповідність отриманому значенню MAC_u''. Якщо значення не збігаються, вхід користувача в систему відхиляється, і сеанс завершується. В іншому випадку робиться висновок, що користувач U_i є легальним користувачем, і йому надається дозвіл на вхід в систему. В цей момент завершується взаємна аутентифікація користувача U_i і сервера S та обмін сеансовим ключем K_s.

Слід зазначити, що ця схема без принципових змін може використовуватись у випадку синхронізації годинників користувача і сервера [6], при цьому кількість повідомлень протоколу зменшиться до двох, оскільки не буде потреби створювати пару запит–відповідь для запобігання атаки повтором.

До переваг цього протоколу слід віднести [6] захищеність секретного ключа x сервера односторонньою функцією хешування $h(\ )$; використання в якості оказій для запиту-відповіді міток часу, що забезпечує неповторюваність оказій; взаємну аутентифікацію користувача і сервера шляхом використання кодів автентичності повідомлень MAC_s і MAC_u'' відповідно; захищеність від атак повтором шляхом використання схеми запит-відповідь; захищеність від атак методом паралельних сеансів через асиметричну структуру кодів автентичності повідомлень MAC_u і MAC_u''.

Разом з тим, хоча синхронізації годинників користувача і сервера не вимагається, занадто велике вікно коректних міток часу може спричинити атаку на протокол, наприклад, шляхом затримки повідомлень [11]; цей протокол також не забезпечує захисту від атак з використанням сценарію зв'язаних ключів та на основі колізій функцій хешування, як і протокол Жуанга [5].

Опис запропонованого протоколу

На етапі реєстрації користувач U_i надає свій ідентифікатор ID_i та пароль PW_i серверу S для реєстрації. Якщо дані прийняті, S здійснює такі кроки:

1. Обчислює таємну інформацію користувача U_i у такій формі $v_i = h\left(h\left(ID_i, x\right)\right)$, та значення $w_i = v_i \oplus h\left(PW_i\right)$, де x – довготривале таємне значення сервера.

2. Зберігає значення ID_i та w_i в пам’яті смарт-карти користувача U_i.

Коли користувач хоче отримати доступ до ресурсів сервера, він здійснює етап входу в систему та обміну ключів, під час якого вставляє смарт-карту в пристрій читання, вводить свій ідентифікатор ID_i та пароль PW_i. Під час j-го входу в систему користувача виконуються такі кроки:

1. $U_i \to S$: $ID_i, E_{vi}\left(N_1, ru_j \oplus w_i, h\left(ID_i \| N_1\right)\right)$
2. $S \to U_i$: $E_{vi}\left(rs_j \oplus w_i, N_1 + 1, N_2\right)$
3. $U_i \to S$: $E_{kj}\left(N_2 + 1\right)$

тут $E_k(m)$ означає симетричне шифрування повідомлення m секретним ключем k, h – одностороння функція хешування, $\oplus$ – побітове виключне АБО, $\|$ – конкатенація рядків, ru_j та rs_j – j-ті випадкові значення, генеровані користувачем U_i та сервером S відповідно, а N_1 та N_2 – оказії, які використовуються для запобігання атакам повтором [11].

На першому кроці смарт-карта користувача U_i спочатку обчислює значення $v_i = w_i \oplus h\left(PW_i\right)$ та надсилає серверу S свій ідентифікатор ID_i, оказію N_1, та зашифроване повідомлення $E_{vi}\left(N_1, ru_j \oplus w_i, h\left(ID_i \| N_1\right)\right)$. Після отримання цієї інформації на другому кроці

сервер S обчислює $v_i = h(h(ID_i, x))$, а потім розшифровує повідомлення $E_{vi}(N_1, ru_j \oplus w_i, h(ID_i \| N_1))$ для перевірки, чи воно містить дійсне значення $h(ID_i \| N_1)$, та чи є значення N_1 новим (унікальним), в іншому випадку спроба входу відхиляється. Якщо отримана інформація є прийнятною, сервер S надсилає у відповідь користувачеві U_i повідомлення $E_{vi}(rs_j \oplus w_i, N_1 + 1, N_2)$. Після отримання цього повідомлення на третьому кроці смарт-карта користувача U_i розшифровує його та перевіряє наявність значення $N_1 + 1$, після чого обчислює j-й сеансовий ключ у вигляді $k_j = h(rs_j, ru_j, v_i)$ та відповідає серверу S зашифрованим повідомленням $E_{kj}(N_2 + 1)$. Оскільки сервер S також має всі значення для обчислення k_j, він розшифровує повідомлення для перевірки наявності значення $N_2 + 1$.

Аналіз запропонованого протоколу

Проаналізуємо запропонований протокол на вразливість до атак, описаних у [5].

1. Атака з підбором пароля. Запобігти цьому виду атаки на рівні протоколу аутентифікації практично неможливо. Для зменшення її імовірності при реалізації протоколу слід ввести лічильник хибних входів в систему та блокувати обліковий запис користувача на деякий час.

2. Для запобігання атаці з компрометацією ключа (якщо зловмиснику стало відомим значення x, а відповідно і v_i) в запропонованому протоколі значення ru_j та rs_j не передаються в явному вигляді, а у вигляді комбінації $ru_j \oplus w_i$ та $rs_j \oplus w_i$ відповідно, крім того, при створенні повідомлень $E_{vi}(N_1, ru_j \oplus w_i, h(ID_i \| N_1))$ та $E_{vi}(rs_j \oplus w_i, N_1 + 1, N_2)$ слід використати режим шифрування зі зв'язаними блоками, що утруднить криптоаналіз.

3. Використання сценарію зв'язаних ключів утруднюється завдяки використанню при створенні значення w_i хеш-функції паролю $h(PW_i)$ замість паролю у відкритому вигляді PW_i, як це було запропоновано в [4], а оскільки при використанні значення зв'язаного пароля $PW_i' = PW_i \oplus \Delta$, значення хеш-функції $h(PW_i') \neq h(PW_i \oplus \Delta) \oplus \Delta$, а отже, і v_i' не буде пов'язане з v_i відомим співвідношенням Δ, що практично унеможливить реалізацію цієї атаки.

4. Імовірність використання колізій в значеннях хеш-функцій у запропонованому протоколі зменшується в результаті подвійного використання хеш-функції [12].

До переваг запропонованого протоколу також можна віднести захищеність секретного ключа x сервера подвійним використанням функції хешування $h(\ )$, що завдяки атакам на основі парадоксу днів народження [11] збільшує сильну опірність колізіям до 2^n проти $2^{n/2}$, де n – довжина хеш-коду; взаємну аутентифікацію користувача і сервера шляхом використання оказій N_1 і N_2 відповідно; захищеність від атак повтором шляхом використання схеми запит-відповідь; захищеність від атак методом паралельних сеансів через асиметричну структуру запиту і відповіді (повідомлення 2 і 3) [6]. Крім того, значення оказії N_1 у першому повідомленні протоколу передається в зашифрованому вигляді, що унеможливлює атаку з відомим відкритим текстом на ключ шифрування v_i, як це описано в [6], а відсутність використання пароля у відкритому вигляді, на противагу існуючим схемам, забезпечує дієвий захист від вибору користувачами простих чи коротких паролів [11, 12].

З метою довготривалого (а не тільки в межах поточного сеансу) захисту від атак повтором, як оказії N_1 та N_2 можна використати мітки часу і вимагати від обох сторін зберігати значення останньої оказії та перевіряти умову, щоб значення кожної нової оказії наступного сеансу було більшим за значення збереженої останньої оказії. Крім того, для посилення захисту від атак з

компрометацією ключа можна, як це запропоновано в [5], використовувати схему Діффі–Хеллмана, однак це збільшить обчислювальну складність протоколу, також слід мати на увазі, що базова версія схеми Діффі–Хеллмана не захищена від атак типу "посередник" [11].

Оцінка та порівняння продуктивності запропонованого протоколу

В цій частині оцінимо характеристики продуктивності запропонованого протоколу та порівняємо з результатами протоколів Жуанга та Ші–Ванга [4, 6]. Порівняння з деякими іншими спорідненими протоколами зведено в таблиці наприкінці розділу.

Усі три протоколи (запропонований, Жуанга та Ші–Ванга) засновані на використанні односторонніх функцій хешування, крім того, запропонована схема і протокол Жуанга використовують симетричне шифрування. На етапі реєстрації протоколи Жаунга і Ші–Ванга використовують одну операцію хешування, тоді як запропонований протокол – дві. На етапі верифікації протокол Ші–Ванга здійснює сім операцій хешування для забезпечення взаємної аутентифікації. Протокол Жуанга вимагає три операції симетричного шифрування, три операції симетричного дешифрування та три операції хешування. Натомість запропонований протокол для досягнення взаємної аутентифікації використовує так само три операції симетричного шифрування і три – дешифрування, а також п'ять операцій хешування. Таким чином запропонована схема є дещо повільнішою (на три операції хешування), ніж схема Жуанга [4]. Слід зазначити, що, оскільки швидкодія програмної реалізації функцій хешування є вищою, ніж симетричного шифрування [11], схема Ші–Ванга є продуктивнішою [6]. Натомість запропонована схема забезпечує більший захист від можливих атак за рахунок незначного зменшення швидкодії.

Окрім продуктивності смарт-карти, інші характеристики, такі як відсутність таблиць верифікації, вільний вибір пароля користувачем, взаємна аутентифікація, відсутність необхідності синхронізації та обмін сеансовим ключем, є важливими для протоколів аутентифікації [4, 6]. Ці характеристики включені в порівняння протоколів аутентифікації (таблиці). Позначення в таблиці означають відповідно: C_1 – вартість обчислень і комунікацій, C_2 – відсутність таблиці верифікації, C_3 – вільний вибір пароля користувачем, C_4 – взаємна аутентифікація, C_5 – відсутність необхідності синхронізації, C_6 – обмін сеансовим ключем.

Порівняння протоколів аутентифікації на основі смарт-карт

Протокол	C_1	C_2	C_3	C_4	C_5	C_6
Запропонована схема	низька	+	+	+	+	+
Ші–Ванга [6]	дуже низька	+	+	+	+	+
Жуанга [4]	дуже низька	+	+	+	+	+
Чена та ін. [7]	дуже низька	+	+	+	–	–
Ванга–Чанга [2]	середня	+	+	–	–	–
Янга–Ші [3]	середня	+	+	–	–	–

Висновки

У цій роботі проведено аналіз і показано недоліки існуючих протоколів віддаленої аутентифікації користувачів з використанням смарт-карт та розроблено покращений протокол взаємної аутентифікації для отримання доступу до ресурсів серверної частини системи теплового проектування. Порівняно з існуючими протоколами [2–4, 6, 7] запропонована схема володіє наступними перевагами: (1) покращений захист від атак з компрометацією ключа, (2) покращений захист від атаки зі зв'язаними ключами, (3) більша стійкість до використання колізій функцій хешування, (4) захист від криптоаналізу з відомим відкритим текстом, (5) покращена можливість використання користувачами простих чи коротких паролів, (6) як і у протоколі [6] не вимагається синхронізації годинників та (7) забезпечується обмін сеансовим ключем.

Запропонована схема забезпечує більший захист від можливих атак за рахунок незначного зменшення швидкодії.

Робота виконувалась у межах держбюджетної теми "Розробка методів та засобів розподілення обчислень в задачах теплового проектування електронних пристроїв нового покоління" ДБ Діаграма.

1. Leslie Lamport Password Authentication with Insecure Communication // Communications of the ACM, Vol. 24 (1981), No 11, pp. 770–772. 2. Shiuh-Jeng Wang, Jin-Fu Chang Smart card based secure password authentication scheme // Computers & Security, Vol. 15 (1996), No. 3, pp. 231–237. 3. Wen-Her Yang, Shiuh-Pyng Shieh Password Authentication Schemes with Smart Cards // Computers & Security, Vol. 18 (1999), No.8, pp.727–733. 4. Wen-Shenq Juang Efficient password authenticated key agreement using smart cards // Computers & Security, Vol. 23 (2004), pp. 167–173. 5. Raphael C.-W. Phan Cryptanalysis of two password-based authentication schemes using smart cards // Computers & Security, Vol. 25 (2006), pp. 52–54.6. Wen-Gong Shieh, Jian-Min Wang Efficient remote mutual authentication and key agreement // Computers & Security, Vol. 25 (2006), pp. 72–77. 7. Chien H.Y., Jan J.K., Tseng Y.H. An efficient and practical solution to remote authentication: smart card. // Computers & Security, Vol. 21 (2002), No. 4, pp. 372–375. 8. A. Jr Evans, W. Kantrowitz, and E. Weiss A user authentication system not requiring secrecy in the computer // Communications of the ACM, Vol. 17 (1974), pp. 437–442. 9. R.E. Lennon, S.M. Matyas, and C.H. Mayer Cryptographic authentication of time-invariant quantities // IEEE Trans. on Communications, Vol. COM-29 (1981), No. 6, pp. 773–777. 10. K. Tan, and H. Zhu Remote password authentication scheme based on cross-product // Computer communications, Vol. 18 (1999), pp. 390–393. 11. Столлингс В. Криптография и защита сетей: принципы и практика. – М.: Вильямс, 2001. – 672 с. 12. Нильс Фергюсон, Брюс Шнайер Практическая криптография. – М.: Диалектика, 2004. – 432 с.

УДК 621.382

Р. Базилевич, А. Ждан
Національний університет "Львівська політехніка",
кафедра програмного забезпечення

АЛГОРИТМИ ПОСЛІДОВНОГО ПАКУВАННЯ СИЛЬНОЗВ'ЯЗНИХ ЧАСТИН СХЕМ ІЗ ЗАДАНИМИ ОБМЕЖЕННЯМИ

Розглянуто декілька стратегій та алгоритмів послідовного пакування схем із заданими обмеженнями. Розкрито особливості різних стратегій пакування схем із заданими обмеженнями.

A few strategies and algorithms line packing of charts with the set limitations. The features of different strategies of packing of charts are exposed with the set limitations.

Вступ

Пакування складних схем мінімальною кількістю ПЛМ (програмованих логічних матриць) є однією з важливих задач, які виникають при проектуванні сучасних засобів комп'ютерної техніки. Кількість елементів та зовнішніх зв'язків кожного утвореного модуля є найважливішими обмеженнями, які необхідно задовольняти при розбитті схем на декілька частин. Задача з математичного погляду належить до важкорозв'язуваних неполіноміальних комбінаторних