УДК 004.056.55

МОДЕЛЬ БЕЗПЕКИ ПОСТКВАНТОВИХ ПРОТОКОЛІВ ІНКАПСУЛЯЦІЇ КЛЮЧІВ

М. В. ЄСІНА

У роботі розглядається модель безпеки постквантових протоколів інкапсуляції ключів Canetti-Krawczyk (CK). Наводяться основні положення стосовно протоколів. Досліджуються моделі неавтентифікованого та автентифікованого порушника. Досліджуються та наводяться приклади SK-безпечних протоколів.

Ключові слова: інкапсуляція ключів, модель безпеки, протокол.

ВСТУП

У критеріях відбору, які висуваються NIST США до кандидатів на постквантові стандарти криптографічного захисту інформації [12], визначено моделі безпеки, яким мають відповідати кандидати. Відповідно трьом кандидатам – асиметричний шифр (АСШ), цифровий підпис та протокол інкапсуляції ключів (ПІК), визначено три моделі безпеки. Стосовно АСШ – IND-CCA2 (IND-CPA, IND-CCA), для підпису – EUF-CMA модель та для ПІК – CK модель [12].

На наш погляд, проблемними сьогодні є питання, що стосуються узагальненого визначення та дослідження моделі безпеки постквантових ПІК, але з урахуванням основних положень та пропозицій, що викладені у [1–11].

Метою цієї статі є узагальнене визначення та дослідження моделей безпеки, зокрема визначення можливостей та умов застосування постквантових ПІК при протидії зі сторони класичного чи квантового порушника [1–11].

1. ВІДОМОСТІ ПРО ПРОТОКОЛИ

Взагалі під протоколами розумітимемо набори інтерактивних процедур, які одночасно виконуються взаємодіючими сторонами, які вказують на особливу обробку вхідних повідомлень та генерацію вихідних повідомлень. Протоколи можуть ініціювати підпротоколи або інші протоколи, і кілька копій таких протоколів можуть одночасно управлятися кожною стороною.

Протоколи обміну ключами (коротко KE) – це механізми, за допомогою яких дві сторони створюють секретний ключ, що в подальшому використовується для захисту інформації, що передається через мережу, до якої має доступ зловмисник (порушник). Протоколи KE є необхідними та виконуються для того, щоб дозволяти використання криптографії спільного ключа для захисту даних, що передаються через небезпечні мережі. Тому протоколи KE є важливими елементами в ході побудови безпечних зв'язків (наприклад, "захищених каналів"). Вони є одними з криптографічних протоколів, що найчастіше використовуються (наприклад, протоколи SSL, IPSec, SSH тощо [9]).

Згідно з [1, 8] протокол обміну ключами вважається безпечним, якщо зловмисник не може відрізнити значення ключа, що створюється KE протоколом, від незалежного випадкового значення. Протоколи KE, що перевірені та забезпечують доказову безпеку, називаються SK-безпечними протоколами. Їх можна використовувати в стандартних умовах надання "безпечних каналів".

PFS – perfect forward secrecy – досконала пряма безпека (секретність). PFS належить до властивості протоколів обміну ключами (KE), за допомогою якої розкриття довгострокового ключа, що використовується у протоколі для автентифікації та узгодження ключів сеансу, не ставить під загрозу секретність ключів сеансу, встановлених до розкриття. Найбільш поширеним способом досягнення PFS у протоколі обміну ключами є використання обміну Diffie-Hellman. Він виконується з тимчасовими показниками, щоб встановити значення сеансового ключа, обмежуючи використання довгострокових ключів для цілі автентифікації обміну.

2. МОДЕЛЬ БЕЗПЕКИ CANETTI-KRAWCZYK (CK)

Основною проблемою, що має бути вирішена для протидії загрозам в ході застосування класичних та особливо квантових комп'ютерів, є забезпечення безпеки встановлення (узгодження) ключів. У [9] такі криптопротоколи отримали назву протоколів інкапсуляції ключів. Як практична основа та нормативно-правової бази таких протоколів, певною мірою, рекомендується застосовувати стандарти ANSI NIST.

Проведений аналіз показав [1–8], що стосовно протоколів інкапсуляції ключів серед інших є важливим механізм, що грунтується на моделі безпеки Canetti-Krawczyk (CK). Модель CK включає в себе три основні компоненти: модель неавтентифікованого порушника (UM), модель автентифікованого порушника (AM) та механізм автентифікації (автентифікатор) (MT). Модель безпеки CK використовується для автентифікації обміну ключами (AKE) [1]. Розгляд вказаних моделей складає достатньо складну проблему. Вона розглядається нижче.

Сутність моделі CK. Модель безпеки CK стосується безпеки ключа сеансу, що використовується на сеансі зв'язку. В ході її оцінювання використовується формальна модель для протоколів обміну ключами та можливостей криптоаналітика (зловмисника). Понят-

тя безпеки, яке називається безпекою ключа сеансу (або SK-безпека), направлене на забезпечення безпеки окремих ключів сеансу. Її порушення є компрометацією сеансового ключа. У випадку безпечності ключа, зловмисник "нічого не дізнається про значення ключа", коли він перехоплює дані протоколу обміну ключами та здійснює атаки на інші сеанси та сторони, що взаємодіють. Такий підхід є стандартним для моделі семантичної безпеки, коли криптоаналітик не може відрізнити реальне значення ключа від незалежного випадкового значення. У даному випадку говорять про реалізацію криптографічного протоколу «нульових знань». Водночас таке визначення SK-безпеки необхідно використовувати обережно, тому що забезпечення необхідного рівня стійкості при встановленні та використанні протоколів обміну ключами для реалізації захищених каналів, може досягатись без складних вимог.

3. МОДЕЛЬ НЕАВТЕНТИФІКОВАНОГО ПОРУШНИКА (UM)

Для аналізу та оцінки безпеки протоколу, потрібно визначити можливості зловмисника, тобто можливі дії атакуючого. Необхідно, щоб ці можливості були максимально загальними (на відміну від, наприклад, просто подання списку можливих атак), але не має бути нереалістичних вимог. Визначимо формально зловмисника [1], але спеціалізуємо та розширимо його для випадку протоколів KE. Використовуватимемо термінологію [1], тобто модель, що називається моделлю неавтентифікованого порушника (UM).

Основні можливості атакуючого. Розглянемо імовірнісного атакуючого, що виконується за поліноміальний час (PPT), який має повний контроль над каналами зв'язку: він може прослуховувати всю передану інформацію, вирішувати, які повідомлення досягатимуть їх місця призначення, і коли змінювати ці повідомлення за бажанням або вставляти власні згенеровані повідомлення. Формалізм відображає цю здатність зловмисника, дозволяючи йому відповідати за передачу повідомлень від однієї сторони до іншої. Атакуючий також контролює планування всіх етапів протоколу, включаючи ініціювання протоколів та доставку повідомлень.

Отримання секретної інформації. На додаток до цих базових можливостей, зловмиснику дозволяється отримати секретну інформацію, що зберігається в пам'яті сторін шляхом явних атак. Розглядаємо всю секретну інформацію, що зберігається на стороні, як потенційно вразливу до вторгнення або інших видів витоку. Проте, під час визначення безпеки протоколу важливо гарантувати, що витік певної форми секретної інформації має, як мінімум, можливий вплив на безпеку інших секретів. Наприклад, необхідно гарантувати, що витік інформації, визначеної для одного сеансу (наприклад, витік сеансового ключа або інформації про тимчасовий стан), не матиме впливу на безпеку інших сеансів або, що навіть витік критичних довгострокових секретів (наприклад, особистих ключів), які використовуються під час кількох сеансів, не обов'язково скомпрометує секретну інформацію з усіх минулих сеансів. Також, щоб розрізняти різні вразливості та максимально забезпечити безпеку у разі розкриття інформації, класифікуємо атаки на три категорії залежно від типу інформації, яку хоче отримати зловмисник.

Розкриття стану сеансу. Атакуючий надає ім'я сторони та ідентифікатор сеансу ще неповного сеансу на цій стороні та отримує внутрішній стан цього сеансу (оскільки сеанси є процедури, що виконуються всередині сторони, то внутрішній стан сеансу добре визначений). Важливий момент полягає в тому, яка інформація міститься у локальному стані сеансу. Залишимо це для зазначення кожним протоколом KE. Тому визначення безпеки параметризується типом і кількістю інформації, що розкрита у цій атаці. Наприклад, виявлена таким чином інформація може бути показником x, який використовується стороною для обчислення значення g^x у протоколі обміну ключа Діффі-Гелмана або випадкових бітів, які використовуються для шифрування кількості у рамках імовірнісної схеми шифрування під час сеансу. Як правило, виявлена (розкрита) інформація включає в себе весь локальний стан сеансу та його підпрограми, за винятком локального стану підпрограм, які безпосередньо мають доступ до довгострокової секретної інформації, наприклад, локального ключа підпису/розшифрування криптосистеми з відкритим ключем або довгострокового спільного ключа.

Запит ключа сеансу. Атакуючий надає назву учасника (сторони) та сеансовий ідентифікатор завершеного сеансу на цій стороні, та отримує значення ключа, створеного названим сеансом. Ця атака передбачає формальне моделювання витоку інформації про певні сеансові ключі, що може виникнути внаслідок подібних порушень, криптоаналізу, необережного розпорядження ключами тощо. Воно також служитиме, опосередковано, для забезпечення того, що неминуче витікання інформації, виробленої шляхом використання ключів сеансу в додатку безпеки (наприклад, інформація, що витекла по ключу за допомогою його використання як ключа шифрування), не допоможе в отриманні додаткової інформації про цей та інші ключі.

Пошкодження сторони. Зловмисник може вирішити в будь-який момент пошкодити сторону, і в цьому випадку атакуючий вивчає всю внутрішню пам'ять цієї сторони, включаючи довгострокові секрети (наприклад, особисті ключі або спільні майстер-ключі, що використовуються для різних сеансів) та визначену інформацію про сеанси, що міститься в пам'яті сторони (наприклад, внутрішній стан неповних сеансів і ключів сеансу, відповідних завершеним сеансам). Оскільки, вивчаючи свої довгострокові секрети, атакуючий може видати себе за сторону у всіх

його діях, тоді сторона вважається повністю контрольованою зловмисником з моменту пошкодження і може, зокрема, відмовлятися від специфікацій протоколу.

Термінологія: якщо для сеансу застосовується будь-яка із зазначених вище трьох атак (тобто, розкриття стану сеансу, запит ключа сеансу або пошкодження сторони, яка проводить сеанс), то сеанс називається локально розкритим (locally exposed). Якщо сеанс або його відповідний сеанс локально розкритий, він називається розкритим (exposed) сеансом.

Закінчення терміну дії сеансу. Одним з важливих додаткових елементів моделі безпеки є поняття закінчення терміну дії сеансу. Це відбувається у формі дії протоколу, яка в ході активації призводить до стирання названого ключа сеансу (і будь-якого відповідного стану сеансу) з пам'яті цієї сторони. Дозволяється, щоб сеанс закінчився на одній стороні, не обов'язково закінчуючи відповідний сеанс. Ефект цієї дії у моделі безпеки полягає в тому, що значення сеансового ключа закінченого терміну дії неможливо знайти за допомогою будь-якої з наведених вище атак, якщо ці атаки виконуються після закінчення сеансу. Це має два важливі наслідки: це дозволяє моделювати загальну (та добру) практику безпеки обмеження терміну служби окремих ключів сеансу і дозволяє просте моделювання поняття досконалої прямої безпеки (секретності). Відзначається, що для того, щоб сеанс був локально розкритий (як зазначено вище), атака на сеанс має відбутися до закінчення сеансу.

Підтримка безпеки протоколів обміну ключами. Протоколи обміну ключами, як і інші криптографічні додатки, вимагають підтримки безпеки (особливо для автентифікації) за допомогою деяких передбачених засобів захисту. Приклади включають безпечне створення особистих ключів сторін, встановлення відкритих ключів інших сторін або встановлення спільних майстер-ключів. Тут також дотримуємось підходу [1], де підтримка функцій автентифікації абстрагується у функцію ініціалізації, яка виконується до початку будь-якого протоколу обміну ключами і яка забезпечує безпечний спосіб (тобто без участі зловмисника) необхідної (довгострокової) інформації. Абстрагуючись від цієї початкової фази, дозволяється комбінувати різні протоколи з різними функціями ініціалізації: зокрема, це дозволяє нашому аналізу протоколів (наприклад, Діффі-Гелмана) застосовуватись з двома поширеними параметрами автентифікації: симетрична та асиметрична автентифікація. Два зауваження: (1) специфікація функції ініціалізації є частиною визначення кожного протоколу KE; (2) секретна інформація, сформована цією функцією на заданій стороні, може бути виявлена атакуючим лише після порушення тієї сторони. Підкреслюється, що, хоча ця абстракція додає простоту та застосовність методів аналізу, підтримка безпеки в дійсних протоколах є елементом, який необхідно ретельно проаналізувати (наприклад, взаємодія з CA у випадку протоколів на основі відкритих ключів). Інтеграція цих явних елементів у модель може бути зроблена або безпосередньо, як зроблено у [7], або більш модульним способом через відповідний склад протоколу.

4. МОДЕЛЬ АВТЕНТИФІКОВАНОГО ПОРУШНИКА (AM), ЕМУЛЯЦІЯ ПРОТОКОЛУ ТА АВТЕНТИФІКАТОР

Модель порушника, яка називається моделлю автентифікованого порушника (AM), визначається таким чином, що є ідентичним для UM з однією принциповою різницею: атакуючий обмежується лише доставкою повідомлень, насправді породжених сторонами без будь-яких змін або доповнень до них. Потім вводять поняття "емуляція", щоб відобразити еквівалентність функціональних можливостей між протоколами в різних моделях порушника, зокрема між UM та AM. Протокол π' емулює протокол π в UM, якщо для будь-якого зловмисника, який взаємодіє з π' в UM, існує зловмисник, який взаємодіє з π в AM так, що обидві взаємодії "виглядають однаково" для зовнішнього спостерігача. Розробляються спеціальні алгоритми, які називаються автентифікаторами, з властивістю, що під час введення опису протоколу π автентифікатор виводить опис протоколу π' такого, що π' емулює протокол π в UM. Тобто автентифікатори виконують функцію автоматичного "компілятора", який перетворює протоколи в AM на еквівалентні (або "так само безпечні, як") протоколи в UM.

Для спрощення побудови автентифікаторів [1] пропонується наступна методологія. Спочатку розглянемо дуже простий однопотоковий (одношаровий) протокол в AM, що називається MT, єдиним функціоналом якого є передача одного повідомлення від відправника до одержувача. Тепер створимо автентифікатор з обмеженим типом, який називається MT-автентифікатором, для забезпечення емуляції тільки для цього конкретного протоколу MT. Нарешті, для будь-якого такого MT-автентифікатора λ, зіставляється один алгоритм (або компілятор) $C\lambda$, який перетворює будь-який вхідний протокол π на інший протокол π' таким чином: до кожного з повідомлень, визначених у протоколі π, застосовується MT-автентифікатор λ. У [1] доведено, що $C\lambda$ є автентифікатором (тобто, результуючий протокол π' емулює π в UM). Особливі реалізації MT-автентифікаторів подані у [1] на основі криптографічних функцій різних типів (наприклад, ЕП, шифрування з відкритим ключем, MAC та ін.).

5. ВИЗНАЧЕННЯ SK-БЕЗПЕКИ

Спочатку подано визначення для UM. Формалізація в AM є аналогічною. Почнемо з визначення "експерименту", де зловмисник U вибирає сеанс, в якому потрібно "перевіряти" інформацію, яку він дізнався з ключа сеансу; зокрема, попросимо зловмисни-

ка відрізнити реальне значення вибраного ключа сеансу від випадкового значення.

Для цього експерименту розширюємо звичайні можливості зловмисника, *U*, в UM, дозволяючи йому виконувати запит на тестовий сеанс. Тобто, на додаток до звичайних дій *U* проти протоколу обміну ключами π, *U* дозволяється у будь-який час під час його виконання тестувати сеанс серед сеансів, що були завершені, нереалізовані або нерозкриті на той час. Нехай к – значення відповідного ключа сеансу. Кидаємо монету *b*, $b \xleftarrow{R} \{0,1\}$. Якщо *b*=0, ми забезпечуємо *U* значенням к. В іншому випадку надаємо *U* значення *r*, випадковим чином обране з розподілу ймовірності ключів, створених протоколом π. Зловмиснику *U* тепер дозволено продовжувати регулярні дії UM-зловмисника, але не дозволяється розкривати тестовий сеанс (а саме, не дозволено розкривати стан сеансу, запити на ключі сеансу або пошкоджувати партнера при тестуванні сеансу, або його відповідний сеанс.) В кінці його запуску, *U* виводить біт *b*' (як його припущення для *b*).

Посилатимемося на зловмисника, який дозволяє запити тестового сеансу як KE-зловмисник.

Визначення 1. KE-протокол π називається SK-безпечним, якщо для будь-якого KE-зловмисника *U* в UM існують такі властивості:

1) протокол π задовольняє властивість, що, якщо дві непошкоджені сторони виконують відповідні сеанси, то вони обидві виводять однаковий ключ, та

2) імовірність того, що *U* правильно вгадає біт *b* (тобто виходи *b*'=*b*), не перевищує 1/2 плюс незначну частку в параметрі безпеки.

Якщо вищезазначені властивості задовольняються для всіх KE-зловмисників в AM, то π є SK-безпечним в AM.

Перша умова – це "послідовність" вимоги до сеансів, виконаних двома непошкодженими сторонами. Немає жодних вимог щодо значення сеансового ключа сеансу, де один із партнерів був пошкоджений до завершення сеансу – фактично більшість протоколів KE дозволяють пошкодженій стороні сильно вплинути на обмін ключами. Друга умова – "основна властивість" для SK-безпеки. Зазначимо, що термін "незначний", як звичайно, належить до будь-якої функції (в параметрі безпеки), яка асимптотично зменшується швидше, ніж будь-яка частка поліному. (Це формулювання, за бажанням, дозволяє кількісно оцінювати безпеку за допомогою конкретної процедури безпеки. У цьому випадку кількісно визначається потужність атакуючого через певні межі часу обчислення, кількість пошкоджень тощо, тоді як її перевага обмежується через певний параметр ε.)

Виділяється наступні три аспекти Визначення 1:

– атакуючий може продовжувати працювати і атакувати протокол навіть після отримання відповіді (реальної чи випадкової) на його запит тестового сеансу. Ця здатність (яка є суттєвим посиленням безпеки відносно [2, 3]) є важливою для підтвердження основної властивості SK-безпеки;

– зловмисник не може пошкодити учасників тестового сеансу або випускати будь-яку іншу команду викриття проти цього сеансу, поки він не існує. Це відображає той факт, що неможливо гаран-тувати безпечне використання ключа сеансу, який було виставлено шляхом втручання зловмисника (або криптоаналізу). Зокрема, це обмеження має важливе значення для підтвердження безпеки окремих важливих протоколів (наприклад, обміну ключами Diffie-Hellman);

– наведене вище обмеження для зловмисника, за допомогою якого він не може пошкодити партнера для тестового сеансу, скасовується, як тільки закінчується сеанс цього партнера. У цьому випадку зловмисник повинен залишатися нездатним розрізняти дійсне значення ключа та випадкове значення. Це є основою для гарантії досконалої прямої безпеки (секретності).

6. ПРЯМА БЕЗПЕКА (СЕКРЕТНІСТЬ)

Неформально поняття "досконала пряма безпека (секретність)" (PFS) [4, 5] зазначається як властивість, що "компрометування довгострокових ключів не компрометує минулі ключі сеансу".

Коли доводять, що протокол має бути SK-безпечним за допомогою Визначення 1, автоматично отримується доказ того, що цей протокол гарантує PFS.

Визначення 2. Говорять, що протокол KE задовольняє SK-безпеку без PFS, якщо він використовує SK-безпеку відносно будь-якого KE-зловмисника в UM, що не допускає припинення дії ключів. (Аналогічно, якщо вищесказане виконується для будь-яких таких зловмисників у AM, то говорять, що π є SK-безпечним без PFS в AM).

7. SK-БЕЗПЕЧНІ ПРОТОКОЛИ

7.1. Двопрохідний протокол Diffie-Hellman

Протокол 2DH

Загальна інформація: Прості числа *p*, *q*, *q*|*p*–1, та *g* порядку *q* в *Z*p*.

Крок 1: Ініціатор P_i, на вході (P_i, P_j, *s*), вибирає $x \xleftarrow{R} Z_q$ і посилає (P_i, *s*, $\alpha=g^x$) до P_j.

Крок 2: При отриманні (P_i, *s*, α) відповідач P_j вибирає $y \xleftarrow{R} Z_q$, надсилає (P_j, *s*, $\beta=g^y$) до P_i, стирає *y* і виводить ключ сеансу $\gamma'=\alpha^y$ під ідентифікатором (id) сеансу *s*.

Крок 3: При отриманні (P_j, *s*, β) сторона P_i обчислює $\gamma'=\beta^x$, стирає *x* та виводить ключ сеансу γ' під ідентифікатором (id) сеансу *s*.

Рис. 1. Двопрохідний протокол Diffie-Hellman у AM

7.2 SK-безпечний протокол Diffie-Hellman в UM

Зауваження по протоколу SIG-DH. Протокол є результатом застосування автентифікатора на основі підпису [1] до двопрохідного протоколу Diffie-Hellman, наведеного на рис. 1, де значення α та β (експоненти DH) служать викликами, що вимагаються автентифікатором на основі підпису. Це припускає (як зазначено у протоколі 2DH), що ці експоненти вибираються знов для кожного нового обміну (інакше кожна сторона може додати явне виключення до повідомлень, які також включені у підпис). Зауважимо, що ідентифікатор сторони-отримувача, що входить до складу підписів, є частиною специфікації автентифікатора на основі підпису [1] і є основою для забезпечення безпеки протоколу.

8. ПРОТОКОЛ SIG-DH

Опис SIG-DH на рис. 2 передбачає, як формалізовано в моделі, що значення s ідентифікатора (id) сеансу надано сторонам. На практиці, як правило, генерується ідентифікатор сеансу s як пара (s_1, s_2), де s_1 – це значення, обране P_i, та інше (з дуже великою ймовірністю) з усіх інших таких значень, обраних P_i у його інших сеансах з P_j. Аналогічно, s_2 вибирається P_j з аналогічною властивістю єдиності. Ці значення s_1, s_2 можуть бути обмінені сторонами як попередня частина до вищевказаного протоколу (це може бути у випадку протоколів, які реалізують таку попередню частину для обміну деякою іншою системною інформацією та для обговорення пара-метрів обміну [6]). Крім того, s_1 може бути включено P_i у перше повідомлення SIG-DH, а s_2 може бути включено P_j у друге повідомлення. У будь-якому випадку, важливо, щоб ці значення були включені під підписи сторін.

Крок 1: Ініціатор P_i, на вході (P_i, P_j, s), вибирає $x \xleftarrow{R} Z_q$ і посилає (P_i, s, α=g^x) до P_j.

Крок 2: При отриманні (P_i, s, α) відповідач P_j вибирає $y \xleftarrow{R} Z_q$ та надсилає до P_i повідомлення (P_j, s, β=g^y) разом з його підписом $SIG_j(P_j, s, \beta, \alpha, P_i)$; він також обчислює ключ сеансу γ'=α^y та стирає y.

Крок 3: При отриманні (P_j, s, β) та підпису P_j, сторона P_i перевіряє підпис та правильність значень, що входять до підпису (такі як ідентифікатори гравців, ідентифікатор (id) сеансу, значення показників тощо). Якщо перевірка була успішною, тоді P_i надсилає P_j повідомлення (P_i, s, $SIG_j(P_i, s, \alpha, \beta, P_j)$), обчислює γ'=β^x, стирає x та виводить ключ сеансу γ' під ідентифікатором (id) сеансу s.

Крок 4: Після отримання кортежу (P_i, s, sig), P_j перевіряє підпис P_i sig і значення, що він включає. Якщо перевірка буде успішною, вона виведе ключ сеансу γ під ідентифікатором (id) сеансу s.

Рис. 2. Протокол Diffie-Hellman у UM: автентифікація за допомогою підписів

Загальна інформація: Прості числа p, q, $q|p-1$, та g порядку q в Z^*p. Кожен гравець має особистий ключ для алгоритму підпису SIG, і всі мають від-криті ключі перевірки інших гравців.

9. ЗАСТОСУВАННЯ ДЛЯ ЗАХИЩЕНИХ КАНАЛІВ

Шаблонний протокол: Мережний Канал. Шаблонний протокол, який називається NetChan, застосовується до моделі неавтентифікованого порушника UM, а також до моделі автентифікованого порушника AM. Далі розглядаються конкретні реалізації цього шаблонного протоколу, де загальні примітиви 'відправити' та 'прийняти', визначені там, є екземплярами з дійсними функціями (наприклад, для забезпечення автентифікації та/або шифрування). Також визначається, що означає, що така реалізація буде "безпечною".

Протокол мережного каналу (канал сеансу) NetChan(π, snd, rcv), визначається на основі KE-протоколу π, а також двох загальних функцій snd та rcv. Обидві snd та rcv – це імовірнісні функції, які в якості аргументів використовують ключ сеансу (позначається цей ключ як індексний символ до функції) та повідомлення m. Ці функції також можуть залежати від інших даних сеансу, таких як ідентифікатор сеансу та ідентифікатори партнерів (сторін). Вихідні дані snd є єдиним значенням m', тоді як виходом rcv є пара (v; ok), де ok – біт, і v – довільне значення. (Біт ok буде використано, щоб повернути значення перевірки, наприклад, результат перевірки тегу автентифікації.) На основі таких функцій визначаємо NetChan(π, snd, rcv) на рис. 3 [2–5].

Мережна автентифікація. На підставі вищезгаданого формального визначення розглядається випадок мережних каналів, що забезпечують автентифікацію інформації через канали, що контролюються зловмисником. А саме, ми зацікавлені в протоколі NetChan, який працює в моделі неавтентифікованого порушника UM, та забезпечує автентичність переданих повідомлень. Ця реалізація NetChan (яка називається NetAut) буде спрямована на захоплення практики, за допомогою якої взаємодіючі сторони використовують протокол обміну ключами, щоб створити загальний ключ сеансу, і використовувати цей ключ для автентифікації (за допомогою функції автентифікації повідомлень, MAC) інформації, якою обмінюються під час цього сеансу. А саме, якщо P_i та P_j поділяють відповідний сеанс s і P_i хоче відправити повідомлення m до P_j протягом цього сеансу, то P_i передає m разом з MACк(m), де к – відповідний ключ сеансу. Таким чином, в цьому випадку ілюструватимемо прикладами (підтверджувати) snd та rcv функції NetChan за допомогою функції MAC, як показано нижче [5].

10. ПРОТОКОЛ NETAUT

Нехай π є протоколом KE і нехай f – функція MAC. Протокол NetAut(π, f) – це протокол NetChan(π, snd, rcv), як визначено на рис. 3, де функції snd та rcv визначаються як [2–5]:

– на вході m, $\text{snd}_\kappa(m)$ виробляє вихід $m'=(m, t)=(m, f_k(m))$.

– на вході m', $\text{rcv}(m')$ виводить (v, ok) наступним чином. Якщо m' має форму (m, t), та пара (m, t) проходить функцію перевірки f за ключем к, то ok=1 та $v=m$. Інакше ok=0 та v=null.

Протокол NetChan(π, snd, rcv)

NetChan(π, snd, rcv) ініціалізується з тією ж функцією ініціалізації I протоколу KE π. Потім він може бути викликаний в межах сторони P_i за наступними діями:

1. establish-session(P_i, P_j, s, role): запускає (викликає) KE-сеанс з π всередині P_i з партнером P_j, ідентифікатором сеансу s та role ∈ {initiator; responder} ({ініціатор; відповідач}). Якщо KE-сеанс завершує записи P_i у своєму локальному виводі "встановлено сеанс s з P_j " і зберігає створений сеансовий ключ.

2. expire-session(P_i, P_j, s): P_i позначає сеанс (P_i, P_j, s) (якщо він існує у P_i), як такий, що закінчився, і ключ сеансу стирається. P_i записи в локальному виводі "сеанс s з P_j закінчився".

3. send(P_i, P_j, s, m): P_i перевіряє, що сеанс (P_i, P_j, s) був завершений, і не закінчився, якщо так, він обчислює $m'=\text{snd}_\kappa(m)$, використовуючи відповідний ключ сеансу к, відправляє (P_i, s, m') у P_j, і записує "відправлено повідомлення m до P_j протягом сеансу s" у локальному виводі.

4. На вхідному повідомленні (P_j, s, m') P_i перевіряє, чи сеанс (P_i, P_j, s) був завершений і не закінчився, якщо так, він обчислює $(m, \text{ok})=\text{rcv}_\kappa(m')$ з відповідним сеансовим ключем к. Якщо ok=1, то P_i записує "отримане повідомлення m від P_j протягом сеансу s". Якщо ok=0, то подальших дій не виконується.

Рис. 3. Загальний протокол мережних каналів

11. ПРОТОКОЛ SMT

Розширюється протокол MT з [1], щоб відповідати налаштуванню на основі сеансу, в якому передані повідомлення групуються у різні сеанси. Розширений протокол називається протоколом передачі повідомлень на основі сеансу (SMT) та визначається на рис. 4. Зверніть увагу на структурну схожість між SMT і NetChan – відмінності полягають у тому, що в SMT фактичний обмін ключами не виконується, а функції snd та rcv є екземплярами простих "ідентифікаційних функцій".

Протокол SMT забезпечує цілком автентичний обмін повідомленнями. Реалізація протоколу NetChan є безпечним протоколом мережної автентифікації, якщо він емулює протокол SMT в UM.

Протокол SMT може бути викликаний у стороні P_i за такими діями:

1. establish-session(P_i, P_j, s): у цьому випадку P_i записує у своєму локальному виводі "встановлені сеанси s з P_j".

2. expire-session(P_i, P_j, s): в цьому випадку P_i записує у своєму локальному виводі "сеанс s з P_j закінчився".

3. send(P_i, P_j, s, m): у цьому випадку P_i перевіряє, чи сеанс (P_i, P_j, s) був встановлений і не закінчився, якщо так, то він посилає повідомлення m до P_j разом з ідентифікатором сеансу (тобто значення m та s надсилаються по ідеально-автентифікованому зв'язку між P_i і P_j); P_i записує у своєму локальному виводі "відправлено повідомлення m до P_j протягом сеансу s".

4. На вхідному повідомленні (m, s), отриманому за його посиланням від P_j, P_i перевіряє, чи сеанс (P_i, P_j, s) встановлений, і не закінчився, якщо так, він записує у локальному виводі "отримано повідомлення m від P_j протягом сеансу s".

Рис. 4. SMT: Протокол MT на основі сеансу в AM

ВИСНОВКИ

1. Згідно з аналізом визначено, що стосовно ПІК постквантового періоду, перспективною є модель безпеки Canetti-Krawczyk (CK). Модель CK включає в себе три основні складові компоненти: модель неавтентифікованого порушника (UM), модель автентифікованого порушника (AM) та механізм автентифікації (автентифікатор) (MT). Як правило модель безпеки CK використовується для автентифікації обміну ключами (AKE).

2. Модель безпеки CK стосується безпеки ключа сеансу, що використовується на сеансі зв'язку. В ході оцінки протоколів обміну ключами та можливостей криптоаналітика (зловмисника) використовується формальна модель. Поняття безпеки, яке називається безпекою ключа сеансу (або SK-безпека), направлене на забезпечення безпеки окремих ключів сеансу. Її порушення може призвести до компрометації ключа сеансу. У випадку безпечності ключа зловмисник "нічого не дізнається про значення ключа", коли він перехвачує дані протоколу обміну ключами та здійснює атаки на інші сеанси та сторони, що взаємодіють. Такий підхід є стандартним для моделі семантичної безпеки, коли криптоаналітик не може відрізнити реальне значення ключа від незалежного випадкового значення.

3. Протоколи обміну ключами (KE) – це механізми, за допомогою яких дві сторони створюють

секретний ключ, що в подальшому використовується для захисту інформації, що передається через мережу, до якої має доступ зловмисник (порушник). Протоколи KE, що перевірені та забезпечують доказову безпеку, називаються SK-безпечними протоколами. Їх можна використовувати в стандартних умовах надання "безпечних каналів".

4. Протокол KE задовольняє SK-безпеку без PFS, якщо він використовує SK-безпеку відносно будь-якого KE-зловмисника в UM, що не допускає припинення дії ключів. KE-протокол π називається SK-безпечним, якщо для будь-якого KE-зловмисника U в UM існують такі властивості:

– протокол π задовольняє властивість, що, якщо дві непошкоджені сторони виконують відповідні сеанси, то вони обидві виводять однаковий ключ;

– імовірність того, що U правильно вгадає біт b (тобто виходи $b'=b$), не перевищує 1/2 плюс незначну частку в параметрі безпеки.

5. SK-безпечним протоколом є двопрохідний Diffie-Hellman, протокол SIG-DH та протокол NetAut. Протокол SMT забезпечує цілком автентичний обмін повідомленнями. Реалізація протоколу NetChan є безпечним протоколом мережної автентифікації, якщо він емулює протокол SMT в UM.

6. PFS – це досконала пряма безпека (секретність). PFS належить до властивості протоколів обміну ключами (KE), за допомогою якої розкриття довгострокових ключових даних, що використовується у протоколі для автентифікації та узгодження ключів сеансу, не ставить під загрозу секретність ключів сеансу, встановлених до розкриття.

7. Модель порушника, яка називається моделлю автентифікованого порушника (AM), визначається так, що є ідентичним для UM з однією принциповою різницею: атакуючий обмежується лише доставкою повідомлень, насправді породжених сторонами без будь-яких змін або доповнень до них. Потім вводять поняття "емуляція", щоб відобразити еквівалентність функціональних можливостей між протоколами в різних моделях порушника, зокрема між UM та AM. Протокол π' емулює протокол π в UM, якщо для будь-якого зловмисника, який взаємодіє з π' в UM, існує зловмисник, який взаємодіє з π в AM так, що обидві взаємодії "виглядають однаково" для зовнішнього спостерігача.

8. В моделі UM можливості противника суттєво розширюються, тобто U противник в UM моделі може виконувати запит на тестовий сеанс. На додаток до звичайних дій U у будь-який час під час його виконання може за вибором тестувати певний сеанс, який чи які були завершені, нереалізовані або нерозкриті на той час. Нехай k – значення відповідного ключа сеансу. За цих умов, по суті, підкидається монета b, $b \xleftarrow{R} \{0,1\}$.

9. Окремою особливістю щодо ключів є нерозрізнюваність ключів, модель безпеки, щодо якої безпека визначена як IK-CPA/CCA2. Сутність її в тому, що конфіденційність ключа має забезпечуватися при атаках на основі підібраного(вибраного) відкритого тексту та підібраного(вибраного) шифр-тексту. При таких атаках зловмисник протидіє в два етапи. На етапі find він приймає два відкриті ключі pk_0 і pk_1 (що відповідають секретним ключам sk_0 та sk_1, відповідно) і виводить повідомлення x разом з деякою інформацією про стан s секретних ключів. На етапі guess він викликає шифртекст y, який утворюється шляхом випадкового зашифрування повідомлень з одним із двох ключів, і повинен визначити, який ключ був вибраний.

Література

[1] *Ran Canetti*, *Hugo Krawczyk* Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. – Режим доступу: http://iacr.org/archive/eurocrypt2001/20450451.pdf.

[2] *V. Shoup* On Formal Models for Secure Key Exchange, Theory of Cryptography Library, 1999. – Режим доступу: http://philby.ucsd.edu/cryptolib/1999/9912.html.

[3] *M. Bellare, R. Canetti, H. Krawczyk* A modular approach to the design and analysis of authentication and key-exchange protocols. – 30th STOC. – 1998.

[4] *M. Bellare, E. Petrank, C. Rackoff, P. Rogaway* Authenticated key exchange in the public key model, manuscript. – 1995-96.

[5] *M. Bellare, P. Rogaway* Entity authentication and key distribution, Advances in Cryptology, – CRYPTO'93, Lecture Notes in Computer Science Vol. 773, D. Stinson ed, Springer-Verlag, 1994. – pp. 232-249.

[6] *W. Diffie, P. van Oorschot, M. Wiener* Authentication and authenticated key exchanges, Designs, Codes and Cryptography, 2, 1992. – pp. 107-125.

[7] *C.G. Gunther* An identity-based key-exchange protocol, Advances in Cryptology – EUROCRYPT'89, Lecture Notes in Computer Science Vol. 434, Springer-Verlag, 1990. – pp. 29-37.

[8] *D. Harkins, D. Carrel, ed.* The Internet Key Exchange (IKE), RFC 2409, November 1998.

[9] *Горбенко І.Д., Горбенко Ю.І.* Прикладна криптологія. Підручник. Х. : «Форт», 2013. – 878 с.

[10] *Yoshida Y., Morozov K., Tanaka K.* CCA2 Key-Privacy for Code-Based Encryption in the Standard Model. In: Lange T., Takagi T. (eds) Post-Quantum Cryptography. PQCrypto 2017. Lecture Notes in Computer Science, vol 10346. Springer, Cham.

[11] *M. Bellare, A. Boldyreva, A. Desai, D. Pointcheval* Key-privacy in public-key encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248. – pp. 566–582. Springer, Heidelberg (2001). doi:10.1007/3-540-45682-1 33.

[12] Post-Quantum Cryptography. – Electronic resource. – Access mode: https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Надійшла до редколегії 25.12.2018

Єсіна Марина Віталіївна, канд. техн. наук, старший викладач кафедри безпеки інформаційних систем і технологій, факультету комп'ютерних наук, Харківського національного університету імені В. Н. Каразіна. Галузь наукових інтересів – захист інформації, постквантова криптографія.

УДК 004.056.55

Есина М. В. **Модель безопасности постквантовых протоколов инкапсуляции ключей** / М.В. Есина // Прикладная радиоэлектроника: науч.-техн. журнал. – 2018. – Том 17. № 3,4. – С. – 160–167.

В работе рассматривается модель безопасности постквантовых протоколов инкапсуляции ключей Canetti-Krawczyk (CK). Приводятся основные положения относительно протоколов. Исследуются модели неаутентифицированного и аутентифицированного нарушителя. Исследуются и приводятся примеры SK-безопасных протоколов.

Ключевые слова: инкапсуляция ключей, модель безопасности, протокол.

Ил.: 04. Библиогр.: 12 назв.

UDC 004.056.55

Yesina M. V. **Security model of post-quantum key incapsulation protocols** / M. V. Yesina // Applied Radio Electronics: Sci. Joum. – 2018. – Vol. 17. № 3, 4. – P. 160–167.

The paper considers the security model of the post-quantum Canetti-Krawczyk (CK) key encapsulation protocols. The main positions of the protocols are given. Unauthenticated and authenticated attacker models are explored. Examples of SK-secure protocols are investigated and provided.

Key words: key encapsulation, security model, protocol.

Fig. 04. Ref.: 12 items.