

МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВНУТРІШНІХ СПРАВ
Кафедра інформаційних технологій факультету №4

ТЕКСТ ЛЕКЦІЇ

з навчальної дисципліни "Блокчейн технологія та криптовалюти"
вибіркових компонент
освітньої програми другого рівня вищої освіти

125 Кібербезпека (Безпека інформаційних та комунікаційних систем)

за темою – Криптографія в блокчейн технологіях (Шифрування і контроль цілісності даних)

Харків 2018

ЗАТВЕРДЖЕНО

Науково-методичною радою
Харківського національного
університету внутрішніх справ
Протокол від _____ № _____

СХВАЛЕНО

Вченою радою факультету №4
Протокол від _____ № _____

ПОГОДЖЕНО

Секцією науково-методичної ради
ХНУВС з технічних дисциплін
Протокол від _____ № _____

Розглянуто на засіданні кафедри інформаційних технологій
(протокол від _____ № _____)

Розробники:

доцент кафедри інформаційних технологій ХНУВС, к.т.н., доцент Носов В.В.

Рецензенти:

професор кафедри штучного інтелекту Харківського національного університету
радіоелектроніки, д.т.н., професор Петров К.Е.

План лекцій

1. Симетричне шифрування
2. Асиметричне шифрування
3. Контроль цілісності і авторства

Рекомендована література:

Основна:

1. Брюс Шнайер. Прикладная криптография. Протоколы, алгоритмы и исходные тексты на языке С. – М.: Издательство ТРИУМФ, 2003.

Текст лекцій

Вступ

Криптографія необхідна для реалізації, принаймні, трьох сервісів безпеки:

- шифрування;
- контроль цілісності;
- автентифікація.

Шифрування - найбільш потужний засіб забезпечення конфіденційності. У багатьох відношеннях воно займає центральне місце серед програмно-технічних регуляторів безпеки, будучи основою реалізації багатьох з них, і в той же час останнім (а часом і єдиним) захисним рубежем. Наприклад, для портативних комп'ютерів лише шифрування дозволяє забезпечити конфіденційність даних навіть в разі крадіжки.

В більшості випадків і шифрування, і контроль цілісності грають глибоко інфраструктурну роль, залишаючись прозорими і для додатків, і для користувачів.

1. Симетричне шифрування

Розрізняють два основні методи шифрування:

- симетричний
- асиметричний.

У першому з них один і той же ключ (що зберігається в секреті) використовується і для шифрування, і для дешифрування даних. Розроблені вельми ефективні (швидкі і надійні) методи симетричного шифрування.

Рис. 1 ілюструє використання *симетричного шифрування*. Для визначеності вестимемо мову про захист повідомлень, хоча події можуть розвиватися не лише в просторі, але і в часі, коли зашифровуються і розшифровуються файли, що нікуди не переміщуються.

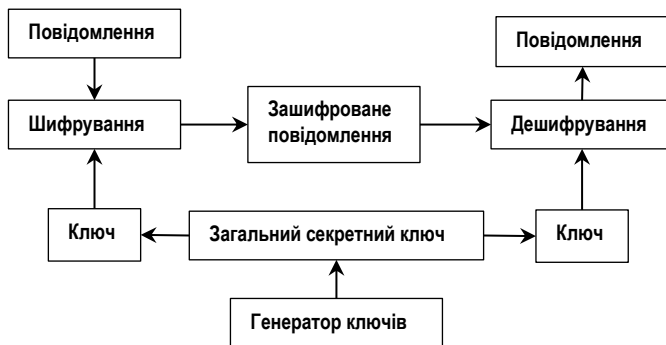


Рис. 1. Використання симетричного методу шифрування

Основним недоліком симетричного шифрування є те, що секретний ключ має бути відомий і відправникові, і одержувачеві. З одного боку, це створює нову проблему поширення ключів. З іншого боку, одержувач на підставі наявності зашифрованого і розшифрованого

повідомлення не може довести, що він отримав це повідомлення від конкретного відправника, оскільки таке ж повідомлення він міг згенерувати самостійно.

У загальному випадку симетричне шифрування швидко і легко реалізується за допомогою апаратних або програмних засобів.

Алгоритми симетричного шифрування

У системах безпеки використовуються різні симетричні алгоритми шифрування. З них можна виділити наступні.

DES (Data Encryption Standard). Розроблений компанією IBM на початку 1970-х рр. Національний інститут стандартів і технологій США (NIST) прийняв на озброєння алгоритм (публікація FIPS 46) для DES в 1977 р. після вивчення, модифікації і затвердження алгоритму в NSA. Алгоритм піддавався подальшій модифікації в 1983, 1988, 1993 і 1999 рр. DES використовує ключ завдовжки 56 біт.

Потрійний DES (TDES). У 1992 році дослідження показали, що DES можна використовувати тричі для забезпечення потужнішого шифрування. Використовуваний при цьому ключ забезпечує велику потужність потрійного DES порівняно із звичайним DES.

AES (Advanced Encryption Standard), також відомий як **Rijndael** - симетричний алгоритм блокового шифрування (розмір блоку 128 біт, ключ 128/192/256 біт), прийнятий як стандарт шифрування урядом США.

IDEA (International Data Encryption Algorithm). Розроблений в Швейцарії. У IDEA використовується 128-бітовий ключ; окрім цього, IDEA також використовується в Pretty Good Privacy (PGP).

RC5. Розроблений Роном Рівестом в інституті MIT. Цей алгоритм дозволяє використовувати ключі із змінною довжиною.

Blowfish. Дозволяє використовувати змінні ключі завдовжки до 448 біт; алгоритм оптимізований для роботи на 32-бітових процесорах.

Twofish. Використовує 128-бітові блоки, а також ключі завдовжки 128, 192 або 256 біт.

CAST-128. Використовує 128-бітовий ключ. Застосовується в нових версіях PGP.

Алгоритм ГОСТ (ГОСТ 28147-89). Російський стандарт шифрування, розроблений у відповідь на DES. У нім використовується ключ завдовжки 256 біт.

Загальний рівень безпеки системи визначає не лише сам алгоритм як такий, але і реалізація і метод використання самої системи.

2. Асиметричне шифрування

У *асиметричних методах* використовуються два ключі. Один з них, несекретний (він може публікуватися разом з іншими відкритими відомостями про користувача), застосовується для шифрування, інший (секретний, відомий лише одержувачеві) - для розшифрування. Найпопулярнішим з асиметричних є метод RSA (Райвест, Шамір, Адлеман), заснований на операціях з великими (наприклад, 100-значними) простими числами і їх творами.

Проілюструємо використання асиметричного шифрування (див. рис. 2).

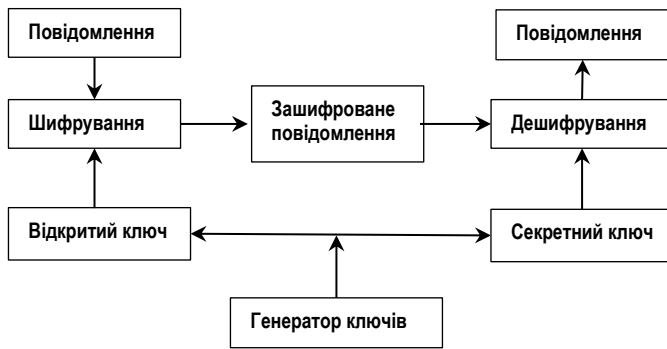


Рис. 2. Використання асиметричного методу шифрування.

Істотним недоліком асиметричних методів шифрування є їх низька швидкість, тому дані методи доводиться поєднувати з симетричними (асиметричні методи на 3 - 4 порядки повільніші). Так, для вирішення завдання ефективного шифрування з передачею секретного ключа, використаного відправником, повідомлення спочатку симетрично зашифровують випадковим ключем, потім цей ключ зашифровують відкритим асиметричним ключем одержувача, після чого повідомлення і ключ вирушають по мережі.

Асиметричні алгоритми шифрування

З найбільш відомих асиметричних алгоритмів шифрування можна виділити наступні.

Алгоритм обміну ключами Діффі-Хеллмана. Уїтфілд Діффі (Whitfield Diffie) і Мартін Хеллман (Martin Hellman) розробили свою систему шифрування з відкритим ключем в 1976 р. Система Діффі-Хеллмана (Diffie-Hellman) розроблялася для вирішення проблеми поширення ключів при використанні систем шифрування з секретними ключами. Ідея полягала в тому, аби застосовувати безпечний метод узгодження секретного ключа без передачі ключа яким-небудь іншим способом. Отже, необхідно було знайти безпечний спосіб здобуття секретного ключа за допомогою того ж методу зв'язку, для якого розроблявся захист. Алгоритм Діффі-Хеллмана не можна використовувати для шифрування або дешифровки інформації.

Алгоритм RSA. У 1978 р. Рон Рівест, Аді Шамір і Лен Адлеман розробили алгоритм шифрування Rivest-Shamir-Adleman (RSA) з відкритим ключем. На відміну від алгоритму Діффі-Хеллмана RSA може використовуватися для шифрування і дешифровки. Також, на відміну від алгоритму Діффі-Хеллмана, безпека алгоритму RSA базується на факторизації великих чисел. Завдання факторизації великих чисел прийнято вважати дуже складним, якщо числа дуже великі (1024 біт або більше).

Алгоритм Ель-Гамаль. Ель-Гамаль (Taher Elgamal) розробив варіант системи Діффі-Хеллмана. Він удосконалив алгоритм Діффі-Хеллмана і отримав один алгоритм для шифрування і один для забезпечення аутентифікації. Алгоритм Ель-Гамала не був запатентований (на відміну від RSA) і, таким чином, став дешевшою альтернативою, оскільки не була потрібна сплата ліцензійних внесків. Оскільки цей алгоритм базувався на системі Діффі-Хеллмана, безпека інформації при його використанні забезпечується складністю рішення задачі дискретного логарифмування.

Алгоритм цифрового підпису DSA. Алгоритм Digital Signature Algorithm (DSA) був розроблений урядом США як стандартний алгоритм для цифрових підписів. Даний алгоритм базується на системі Ель-Гамала, але дозволяє здійснювати лише аутентифікацію. Конфіденційність цим алгоритмом не забезпечується.

Шифрування з використанням еліптичних кривих ECC. Еліптичні криві були запропоновані для використання в системах шифрування в 1985 р. Системи шифрування з використанням еліптичних кривих (ECC) базуються на іншому складному математичному завданні, ніж факторизація або дискретне логарифмування. Дане завдання полягає в наступному: маючи дві точки А і В на еліптичній кривій, такі, що $A = kB$, дуже важко визначити

ціле число k . Існує ряд переваг використання ECC перед алгоритмом RSA або Діффі-Хеллмана. Найбільшою перевагою є те, що ключі мають меншу довжину (внаслідок складності завдання), внаслідок чого обчислення виробляються швидшим із збереженням рівня безпеки. Наприклад, безпека, що забезпечується 1024-бітовим ключем RSA може бути забезпечена 160-бітовим ключем ECC.

Рис. 3 ілюструє комбіноване шифрування, реалізоване шляхом поєднання симетричного і асиметричного методів.

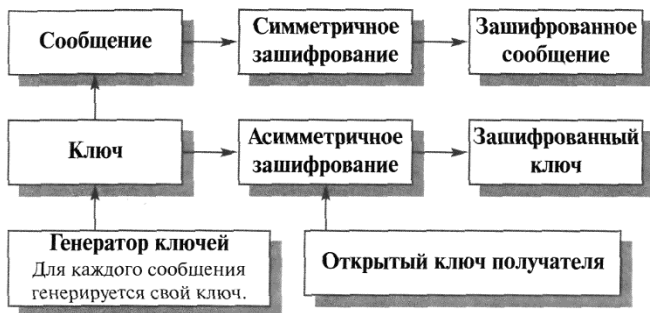


Рис. 3. Комбіноване шифрування повідомлення

На рис. 4 показана дешифровка комбіновано зашифрованого повідомлення.

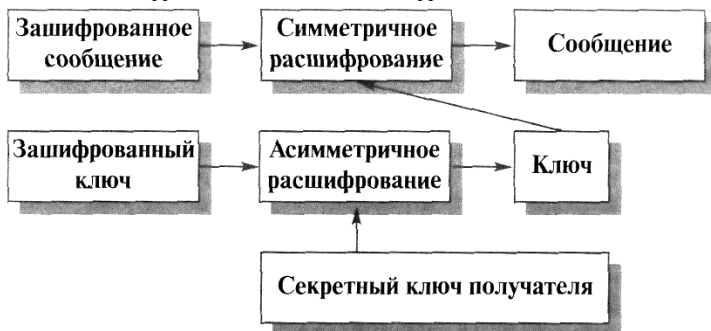


Рис. 4. Дешифрування комбіновано зашифрованого повідомлення

Асиметричні методи дозволили вирішити важливе завдання спільного вироблення секретних ключів (це істотно, якщо сторони не довіряють один одному), обслуговуючих сеанс взаємодії, при початковій відсутності загальних секретів. Для цього використовується алгоритм Діффі-Хеллмана.

Певного поширення набув різновид симетричного шифрування, заснований на використанні складених ключів. Ідея полягає в тому, що секретний ключ ділиться на дві частини, що зберігаються окремо. Кожна частина сама по собі не дозволяє виконати розшифрування. Якщо в правоохоронних органах з'являється підозріння відносно особи, що використовує деякий ключ, вони можуть в установленому порядку отримати половинки ключа і далі діяти звичайним для симетричного розшифрування чином.

Порядок роботи із складеними ключами - хороший приклад дотримання принципу розділення обов'язків. Він дозволяє поєднувати права різного роду таємниці (персональну, комерційну) з можливістю ефективно стежити за порушниками закону, хоча, звичайно, тут дуже багато тонкощів і технічного, і юридичного плану.

Багато криптографічних алгоритмів як один з параметрів вимагають псевдовипадкове значення, в разі передбаченості якого в алгоритмі з'являється вразливість. Генерація псевдовипадкових послідовностей - важливий аспект криптографії.

3. Контроль цілісності і авторства

Криптографічні методи дозволяють надійно контролювати цілісність, як окремих порцій даних, так і їх наборів (таких як потік повідомлень); визначати достовірність джерела даних; гарантувати неможливість відмовитися від зроблених дій ("невідмовність").

У основі криптографічного контролю цілісності лежать два поняття:

- хеш-функція;
- електронний цифровий підпис (ЕЦП).

Хеш-функція — це важко оборотне перетворення даних (однобічна функція), що реалізується, як правило, засобами симетричного шифрування із зчепленням блоків. Результат шифрування останнього блоку (залежний від всіх попередніх) і служить результатом хеш-функції.

Хай ϵ дані, цілісність яких потрібно перевірити, та хеш-функція - раніше обчислений результат її застосування до вихідних даних (так званий дайджест). Позначимо хеш-функцію через h , вихідні дані - через T , дані, що перевіряються, - через T^* . Контроль цілісності даних зводиться до перевірки рівності $h(T^*) = h(T)$. Якщо воно виконане, вважається, що $T^* = T$. Рівність дайджестів для різних даних називається **колізією**. В принципі, колізії, звичайно, можливі, оскільки потужність безлічі дайджестів менша, ніж потужність безлічі хешуємих даних, проте те, що h є функція однобічна, означає, що за прийнятний час спеціально організувати колізію неможливо.

Розглянемо тепер використання асиметричного шифрування для вироблення і перевірки електронного цифрового підпису. Хай $E(T)$ позначає результат шифрування тексту T за допомогою відкритого ключа, а $D(T)$ - результат розшифрування тексту T (як правило, шифрованого) за допомогою секретного ключа. Аби асиметричний метод міг застосовуватися для реалізації ЕЦП, необхідне виконання тотожності

$$E(D(T)) = D(E(T)) = T$$

На рис. 5 показана процедура вироблення електронного цифрового підпису, що полягає в шифруванні перетворенням D дайджеста $h(T)$.



Рис. 5. Вироблення електронного цифрового підпису
Перевірка ЕЦП може бути реалізована так, як показано на рис. 6.

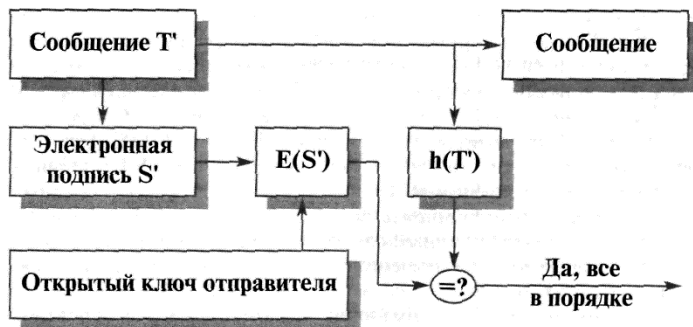


Рис. 6. Проверка электронного цифрового подписи

З рівності $E(S') = h(T')$ витікає, що $S' = D(h(T))$ (для доказу досить застосувати до обох частин перетворення D і викреслити в лівій частині тотожне перетворення $D(E())$). Таким чином, електронний цифровий підпис захищає цілісність повідомлення і засвідчує особу відправника, тобто захищає цілісність джерела даних і служить основою невідомості.

Для контролю цілісності послідовності повідомлень (тобто для захисту від крадіжки, дублювання і зміни порядку повідомлень) застосовують часові мітки і нумерацію елементів послідовності, при цьому мітки і номери включають в підписуваний текст.

Висновки

У симетричному шифруванні один і той же ключ (що зберігається в секреті) використовується і для шифрування, і для дешифрування даних.

Основним недоліком симетричного шифрування є те, що секретний ключ має бути відомий і відправникові, і одержувачеві.

Одним із надійних на сьогодні алгоритмів симетричного шифрування є AES (Advanced Encryption Standard), також відомий як Rijndael - алгоритм блокового шифрування (розмір блоку 128 біт, ключ 128/192/256 біт), прийнятий як стандарт шифрування урядом США.

У асиметричних методах використовуються два ключі. Один з них, несекретний (він може публікуватися разом з іншими відкритими відомостями про користувача), застосовується для шифрування, інший (секретний, відомий лише одержувачеві) - для розшифрування.

Істотним недоліком асиметричних алгоритмів шифрування є їх низька швидкодія, тому дані методи доводиться поєднувати з симетричними.

Одним із надійних на сьогодні алгоритмів асиметричного шифрування, що використовується для ЕЦП в bitcoin, є ECDSA – алгоритм шифрування з використанням еліптичних кривих ECC.

Хеш-функція (дайджест геш-функції) — це важко оборотне перетворення даних (однобічна функція), що реалізовується, як правило, засобами симетричного шифрування із зчепленням блоків. Результат шифрування останнього блоку (залежний від всіх попередніх) і служить результатом хеш-функції.

ЕЦП у bitcoin використовує дайджест геш-функції sha256 і асиметричний алгоритм ECDSA.