[4] Fridman A., Fridman O. Gradient Coordination Technique for Controlling Hierarchical and Network Systems. Systems Research Forum. 2010. Vol. 4. No. 2. Pp. 121-136. doi: https://doi.org/10.1142/S1793966610000223.
[5] Zaporozhtsev V. V., Novoseltsev V. I., Strukov A. Yu. Fuzzy parametric coordination in a multi-level hierarchical system. Control Systems and Information Technologies. 2012. Vol. 50. No. 4.1. P. 142145.
[6] Ladanyuk A.P., Zaєets N.A., Vlasenko L.O., Lutsk N.M. Coordination of the function of the technological plant of the cereal plant with the goal of forecasting tasks.
[7] Bayas M.M., Dubovoi V.M., Rovira R.H., Smailova S., Bissarinov B. Coordination of serial-parallel manufacturing processes of milk production Przeglad Elektrotechniczny, 2019. doi: https://doi.org/10.15199/48.2019.04.31.
[8] Wójcik W., Dubovoi V., Duda M., Romaniuk R., Yesmakhanova L., et al. Coordination in serial-parallel image processing Proc. SPIE 9816, Optical Fibers and Their Applications 2015, 981616 (December 18, 2015). doi: https://doi.org/10.1117/12.2229006
[9] Dubovoi V.. Bayas M., Shegebaeva J., Gromaszek K. Optimization of hierarchical management of technological processes Proc. SPIE 9816, Optical Fibers and Their Applications 2015, 981622 (December 18, 2015). doi: https://doi.org/10.1117/12.2229201
[10] Dubovoi V.M., Bayas M.M. Modeling the coordination of cleaning processes in a pasteurization line based on Petri Nets Automation - 2013: Materials of the XX International Conference of Automatic Control, 25-27 Veresnya 2013r. Mikolaev: NUK, 2013, P. 288.
[11] Bayas M.M., Dubovoi V.M. Model based in random walk for coordination of a dairy plant Fourth International Scientific Conference "Intelligence Systems in Industry and Education – 2013" Access mode: http://ispo.elit.sumdu.edu.ua/files/theses/Bayas%20M .M._Dubovoy% 20V.M.pdf
[12] Gorodetsky V. I. Self-organization and multi-agent systems. Part 1. Models of multi-agent self-organization // Bulletin of the Russian Academy of Sciences. Theory and control systems. 2012. No. 2. Pp. 92-120.
[13] Bayas M.M., Duvoboi V.M. Development of the structure of the multi-agent coordination in technological processes. System Analysis and Information Technologies. International Conference, SAIT 2014 Kyiv, Ukraine, May 26–30, 2014.
[14] Rovira R.H., Duvoboi V.M., Yukhimchuk M.S., Bayas M.M., Torres W.D. A Model of Self-oscillations in Relay Outputs Control Systems with Elements of Artificial Intelligence. In: Rocha Á., Guarda T. (eds) Proceedings of the International Conference on Information Technology & Systems (ICITS 2018). ICITS 2018. Advances in Intelligent Systems and Computing, vol 721. Springer, Cham. doi: https://doi.org/10.1007/978-3-319-73450-733
[15] Duvoboi V.M., Pylypenko I.V., Wójcik W., Sailarbek S. Synthesis of the control algorithm of cyclicity for branched technological process. Proc. SPIE 9816, Optical Fibers and Their Applications 2015, 981620 (December 18, 2015); doi: https://doi.org/10.1117/12.2229191
[16] *McCabe.* A Complexity Measure. IEEE Transactions on Software Engineering journal, 1976, December. Pp. 308-320. doi: https://doi.org/10.1109/TSE.1976.233837

УДК 004.91:004.056.55.347.135.224

АЛГОРИТМІЗАЦІЯ ШИФРУВАННЯ ЦИФРОГО ПІДПИСУ

Плотніков В. М.[1], Борцова Ю. В.[2]

[1,2] Одеська національна академія харчових технологій, Одеса, Україна
ORCID: [1] http://orcid.org/0000-0001-9000-2568, [2] http://orcid.org/0000-0001-6712-8357
E-mail: [2] bortsova.07@gmail.com

DOI: https://doi.org/10.15673/atbp.v12i1.1703

Анотація. *У криптосистемах на основі асиметричних ключів для шифрування й дешифрування використовується пара ключів - секретний і відкритий, унікальні для кожного користувача, та цифровий сертифікат. Цифровий сертифікат являє собою розширення відкритого ключа, та містить не тільки сам ключ, але й додаткову інформацію, що описує: приналежність ключа, час використання, доступні криптосистеми, назву центру засвідчення, та інше. Для реалізації такої взаємодії використовуються спеціальні структури - центри сертифікації. Їхня основна функція - поширення публічних і секретних ключів користувачів, а також верифікація сертифікатів. Центри сертифікації можуть поєднуватися: центр вищого рівня (кореневий) може видати сертифікат і права на видачу ключів центру, розташованому рівнем нижче. Той, у свою чергу, може видати права іншому центру ще нижчого рівня й так далі. Сертифікат, виданий одним із центрів, може бути верифікований кожним у такому ланцюгу. У такий спосіб існує можливість забезпечити наявність центру поширення секретних ключів у безпосередній близькості від користувача, що вирішує проблему дискредитації ключа при передачі у мережах зв'язку.*

Abstract. *Asymmetric key-based cryptosystems use a pair of keys - secret and public keys unique to each user - and a digital certificate for encryption and decryption. A digital certificate is an extension of the public key that includes not only the key itself but also additional information describing the key's ownership, usage time, available cryptosystems, name of the certifying center, etc. Special structures certifying centers are used to implement such interaction. Their main function is to distribute the public and private keys of users, as well as the verification of certificates. Certification centers can be joined into chains. A higher (root) certifying center may issue a certificate and key issuance rights to the lower center. The latter, in turn, may grant rights to another downstream center, and so on, whereby a certificate issued by one of the centers can be verified by each of the servers in the chain1. In this way, it is possible to set up a secret key distribution center in the immediate vicinity of the user, which solves the problem of discrediting the key during transmission over communication networks.*

Ключові слова: цифровий підпис, асиметричні ключі, шифрування, дешифрування
Keywords: digital signature, asymmetric keys, encryption, decryption

1. Теоретична складова. Сертифікація ключів

Для верифікації відкритого ключа застосовується сертифікат ключа – електронний документ, що зв'язує відкритий ключ із суб'єктом, що правомірно володіє відповідним закритим ключем. Без такої верифікації зловмисник може видати себе за будь-якого суб'єкта, підмінивши відкритий ключ. Для завірення сертифіката використовується ЕЦП установи, що видає сертифікати (засвідчуючий центр, CA - certificate authority). Засвідчуючий центр - основний компонент PKI. Маючи відкритий ключ засвідчуючого центру, будь-який суб'єкт може перевірити вірогідність виданого їм сертифіката. За вірогідність даних, що містяться в сертифікаті та ідентифікують правомірного власника, відповідає центр, що видав сертифікат

Для одержання сертифіката ключа суб'єкт повинен засобами PKI сформувати пару ключів (відкритий і закритий) та відправити відкритий ключ разом з ідентифікуючою себе інформацією в центр сертифікації, а закритий ключ зберегти в себе. Можлива також схема з формуванням ключової пари на прохання суб'єкта в самому засвідчуючому центрі. Закритий ключ може зберігатися в захищеній області на диску або в пам'яті спеціалізованого автономного носія, наприклад, USB-брелока або смарт-карти. Як правило, ключ додатково шифрується з використанням пароля або PIN-коду, відомих тільки правомірному власнику. Ключ може бути захищений і за допомогою інших методів, що ідентифікують власника.

Після необхідної перевірки (іноді потрібна особиста явка й пред'явлення підтверджувальних документів) центр, що засвідчує, видає й підписує сертифікат, у якому, крім відкритого ключа й ідентифікуючого власника інформації, вказується період його дії й атрибути сертифіката ключа видавця, необхідні для перевірки сертифіката.

Підробити сертифікат, не володіючи відповідним закритим ключем центра, що засвідчує, практично неможливо. Сертифікат може вільно поширюватися по мережі, однак той, хто не володіє відповідним закритим ключем, не зможе ним скористатися в злочинних цілях.

Центр, що засвідчує, володіє сертифікатом ключа ЕЦП, закритий ключ якого він використовує для завірення видаваних сертифікатів. Центр веде загальнодоступний реєстр виданих їм сертифікатів, кожний з яких ідентифікується унікальним реєстраційним номером. У функції центра, що засвідчує, входить також ведення списку сертифікатів, відкликаних по різних причинах (наприклад, при компрометації закритого ключа або втраті юридичної чинності документів, на підставі яких він виданий). Цей список підписується ЄЦП центра, що засвідчує, і відкрито публікується. Для кожного відкликаного сертифіката в списку вказуються реєстраційний номер, дата й причина відкликання.

Розрізняють підлеглий центр, що засвідчує, сертифікат якого виданий іншим центром, що засвідчує, і кореневий центр, що засвідчує, сертифікат якого виданий їм самим. Кореневих центрів, що засвідчують, незалежних друг від друга, може бути кілька. Тим самим вся безліч центрів, що засвідчують, утворить сукупність ієрархічних дерев у змісті теорії графів. Сертифікати всіх центрів, що засвідчують (кореневих і підлеглих), яким довіряє суб'єкт, повинні бути йому відомі й зберігатися в захищеному сховищі. Щоб перевірити дійсність деякого сертифіката, треба пройти по «ланцюжку довіри» від сертифіката його видавця до сертифіката центра, що засвідчує, якому довіряє суб'єкт.

Суб'єкт при перевірці сертифіката, виданого деяким центром, що засвідчує, повинен перевірити, чи не значиться цей сертифікат у числі відкликаних.

Як же виробляється властиво шифрування за допомогою цих чисел :

Відправник розбиває своє повідомлення на блоки, рівні k=[$\log_2(n)$] біт, де квадратні дужки позначають узяття цілої частини від дробового числа.

Подібний блок може бути інтерпретований як число з діапазону (0;2 k-1). Для кожного такого числа (назвемо його m_i) обчислюється вираження $c_i=((m_i)e)\text{mod}n$. Блоки c_i і є зашифроване повідомлення, і їх можна спокійно передавати по відкритому каналі, оскільки операція зведення в ступінь по модулі простого числа, є необоротним математичним завданням. Зворотна їй завдання зветься "логарифмування в кінцевому полі" і є на кілька порядків більше складним завданням. Тобто навіть якщо зловмисник знає числа e і n, те по c_i прочитати вихідні повідомлення mi він не може ніяк, крім як повним перебором m_i.

А от на прийомній стороні процес дешифрування все-таки можливий, і допоможе нам у цьому збережене в секреті число d. Досить давно була доведена теорема Ейлера, окремий випадок якої затверджує, що якщо число n представимо у вигляді двох простих чисел p і q, те для будь-якого x має місце рівність $(x(p-1)(q-1))\text{mod } n = 1$. Для дешифрування RSA-Повідомлень скористаємося цією формулою.

Зведемо обидві її частини в ступінь (-y) : $(x^{(-y)}(p-1)(q-1))\text{mod } n = 1^{(-y)} = 1$.

Тепер помножимо обидві її частини на x : $(x^{(-y)}(p-1)(q-1)+1)\text{mod } n = 1*x = x$.

А тепер згадаємо як ми створювали відкритий і закритий ключі. Ми підбирали за допомогою алгоритму Евклида d таке, що $e*d+(p-1)(q-1)*y=1$, тобто $e*d=(-y)(p-1)(q-1)+1$. А отже в останнім вираженні попереднього абзацу ми можемо замінити показник ступеня на число (e*d). Одержуємо $(xe*d)\text{mod } n = x$. Тобто для того щоб прочитати повідомлення $c_i=((m_i)e)\text{mod } n$ досить звести його в ступінь d по модулі m :

$$((c_i)d)\text{mod } n = ((m_i)e*d)\text{mod } n = m_i. \quad (1)$$

Насправді операції зведення в ступінь більших чисел досить трудомісткі для сучасних процесорів, навіть якщо вони виробляються по оптимізованим за часом алгоритмам. Тому звичайно весь текст повідомлення кодується звичайним блоковим шифром (набагато більше швидким), але з використанням ключа сеансу, а от сам ключ сеансу шифрується саме асиметричним алгоритмом за допомогою відкритого ключа одержувача й міститься в початок файлу.

2.Односпрямовані геш-функції

Геш-Функція призначена для стиску документа, що підписується, M до декількох десятків або сотень біт. Геш-Функція h( ) приймає як аргумент повідомлення (документ) M довільної довжини й повертає геш-значення h(M)=H фіксованої довжини. Звичайно гешированная інформація є стислим двійковим поданням основного повідомлення довільної довжини. Слід зазначити, що значення геш-функції h(M) складним образом залежить від документа M и не дозволяє відновити сам документ M.

Геш-Функція повинна задовольняти цілому ряду умов:

1. геш-функція повинна бути чутлива до всіляких змін у тексті M, таким як вставки, викиди, перестановки й т.п.;
2. геш-функція повинна мати властивість необоротності, тобто завдання підбора документа M', що мав би необхідне значення геш-функції, повинна бути вичислительно нерозв'язна;
3. імовірність того, що значення геш-функцій двох різних документів (поза залежністю від їхніх довжин) збіжаться, повинна бути мізерно мала.

Більшість геш-функцій будується на основі односпрямованої функції f(•), що утворить вихідне значення довжиною n при завданні двох вхідних значень довжиною n. Цими входами є блок вихідного тексту M, і геш-значення H_{i-1} попереднього блоку тексту:

$H_i = f(M_i, H_{i-1})$.

Геш-Значення, що обчислюється при уведенні останнього блоку тексту, стає Геш-значенням усього повідомлення M.

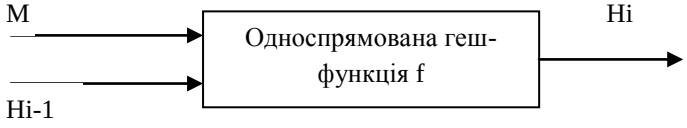

Рис. 1 – Загальна схема побудови односпрямованої геш-функції

У результаті односпрямована Геш-функція завжди формує вихід фіксованої довжини n (незалежно від довжини вхідного тексту).

3. Односпрямовані геш-функції на основі симетричних блокових алгоритмів

Односпрямовану геш-функцію можна побудувати, використовуючи симетричний блоковий алгоритм. Найбільш очевидний підхід полягає в тому, щоб шифрувати повідомлення M за допомогою блокового алгоритму в режимі CBC

або CFB за допомогою фіксованого ключа й деякого вектора ініціалізації IV. Останній блок шифртекста можна розглядати в якості геш-значення повідомлення M. При такому підході не завжди можливо побудувати безпечну односпрямовану геш-функцію, але завжди можна одержати код аутентифікації сообщения MAC (Message Authentication Code). Більше безпечний варіант геш-функції можна одержати, використовуючи блок повідомлення як ключ, що предыдет геш-значення - як вхід, а поточне геш-значення - як вихід. Реальні геш-функції проектуються ще більш складними. Довжина блоку звичайно визначається довжиною ключа, а довжина геш-значення збігається сдлиной блоку.

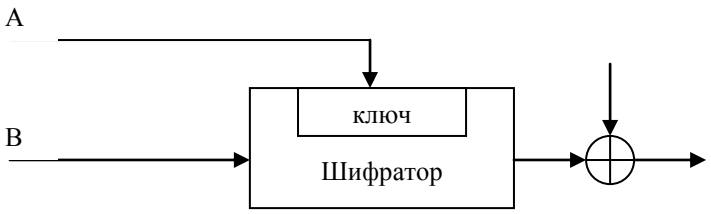

Рис. 2 – Узагальнена схема формування геш-функції

Оскільки більшість блокових алгоритмів є 64-бітовими, деякі схеми гешированія проектують так, щоб геш-значення мало довжину, рівну подвійній довжині блоку.

Якщо прийняти, що одержувана геш-функція коректна, безпека схеми гешированія базується на безпеці лежачі в її основі блокового алгоритму. Схема гешированія, у якої довжина геш-значення дорівнює довжині блоку. Її робота описується вираженнями:

$$H_0 = I_н,\ H_i = E_A(B),$$

де $I_н$ - деяке випадкове початкове значення; A, У и C можуть приймати значення M_i, H_{i-1}, (M_i, H_{i-1}) або бути константами.

Таблиця 1 – Схеми безпечного гешированія, у яких довжина геш-значення дорівнює довжині блоку

Номер схеми	Функція гешированія
1	$H_i = E_{Hi-1}(M_i) \oplus M_i$
2	$H_i = E_{Hi-1}(M_i \text{ A } H_{i-1}) \oplus M_i \oplus H_{i-1}$
3	$H_i = EHi\text{-}1\ (M_i) \oplus H_{i-1} \oplus M_i$
4	$H_i = E_{Hi-1}(M_i \oplus {}_{Hi-1}) \oplus M_i$
5	$H_i = E_{Mi}(H_{i-1}) \oplus H_{i-1}$
6	$H_i = E_{Mi}(M_i \text{ A } H_{i-1})\ M_i \text{ A } H_{i-1}$
7	$H_i = E_{Mi}(H_{i-1})\ M_i \oplus H_{i-1}$
8	$H_i = E_{Mi}(M_i \oplus H_{i-1}) \oplus H_{i-1}$
9	$H_i = E_{Mi\oplus Hı-1}(M_i) \oplus M_i$
10	$H_i = E_{Mi\oplus Hı-1}(H_{i-1}) \oplus H_{i-1}$
11	$H_i = E_{Mi\oplus Hı-1}(M_i) \oplus \subseteq\geq -i$
12	$H_i = E_{Mi\oplus Hı-1}(H_{i-1}) \oplus M_i$

Повідомлення M розбивається на блоки M_i прийнятої довжини, які обробляються по черзі.

Три різні змінні A, У и C можуть приймати одне із чотирьох можливих значень, тому в принципі можна одержати 64 варіанта загальної схеми цього типу. З них 52 варіанта є або тривіально слабкими, або небезпечними. Інші 12 безпечних схем гешированія перераховані в табл.1.

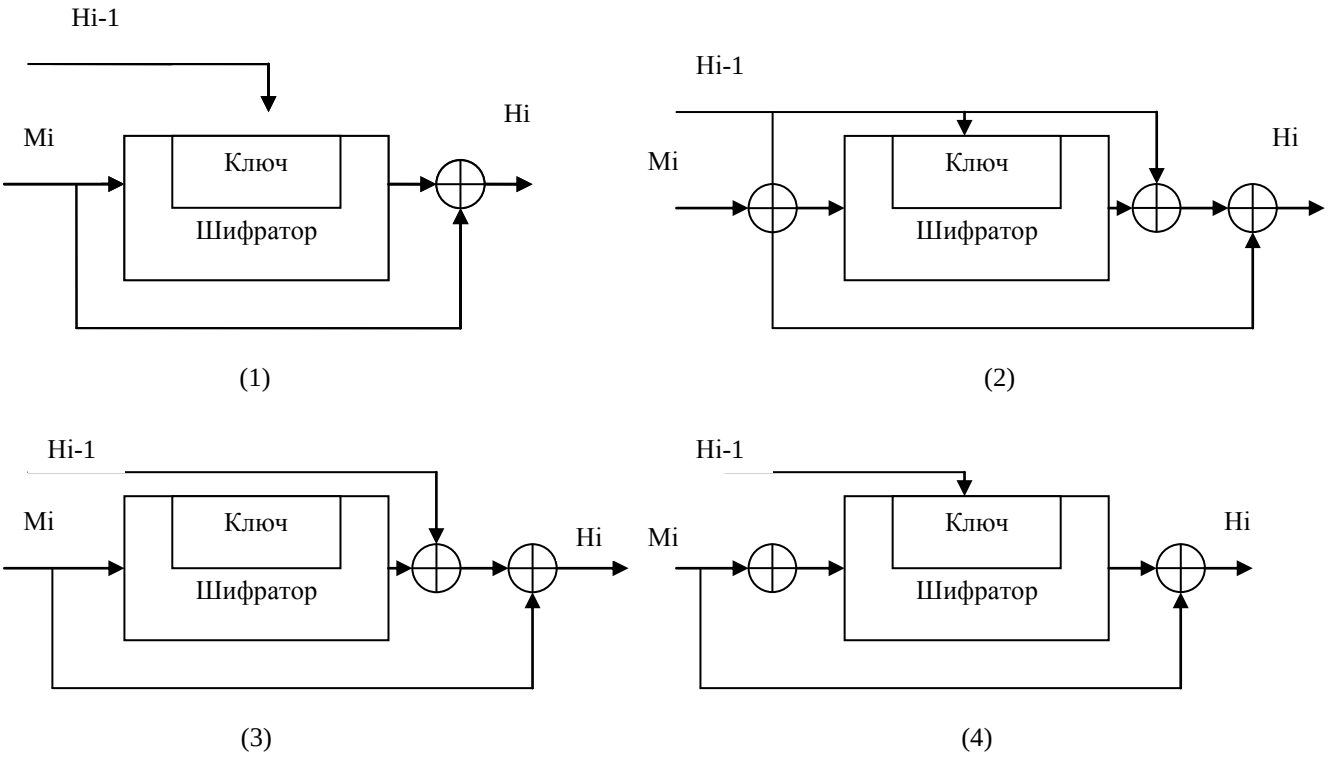

Рис. 3 – Чотири схеми безпечного гешированія

4. Один з можливих стандартів геш-функції

Російський стандарт ДЕРЖСТАНДАРТ Р 34.11-95 визначає алгоритм і процедуру обчислення геш-функції для будь-яких послідовностей двійкових символів, застосовуваних у криптографічних методах обробки й захисту інформації. Цей стандарт базується на блоковому алгоритмі шифрування ДЕРЖСТАНДАРТ 28147-89, хоча в принципі можна було б використовувати й другои блоковий алгоритм шифрування з 64-бітовим блоком і 256-бітовим ключем.

Дана геш-функція формує 256-бітове геш-значення.

Функція стиску $H_i = f(M_i, H_{i-1})$ (обоє операнда M_i й H_{i-1} є 256-бітовими величинами) визначається в такий спосіб:

1) генеруються 4 ключі шифрування K_j, j = 1...4, шляхом лінійного змішування M_i, H_{i-1} і деяких констант C_j;

2) кожний ключ K_j, використовують для шифрування 64-бітових подслов h_i слова H_{i-1} у режимі простої заміни: $Si = E_{Kj}(h_j)$. Результуюча послідовність S_4, S_3, S_2, S_1 довжиною 256 біт запам'ятовується в тимчасовий змінної S.

3) значення H_i є складної, хоча й лінійній функції змішування S, M_i, і H_{i-1}.

При обчисленні остаточного геш-значення повідомлення M ураховуються значення трьох зв'язаних між собою змінних:

H_n - геш-значення останнього блоку повідомлення;

Z - значення контрольної суми, одержуваної при додаванні по модулі 2 всіх блоків повідомлення;

L - довжина повідомлення.

Ці три змінні й доповнений останній блоки M' повідомлення поєднуються в остаточне геш-значення в такий спосіб:

$$H = f(Z, M', f(L, f(M', H_n))).$$

Дана геш-функція визначена стандартом ДЕРЖСТАНДАРТ Р 34.11-95 для використання разом з російським стандартом електронного цифрового підпису.

5. Алгоритм цифрового підпису RSA

Першої й найбільш відомої в усьому світі конкретною системою ЕЦП стала система RSA, математична схема якої була розроблена в 1977 р. у Массачуссетском технологічному інституті США.

Спочатку необхідно обчислити пару ключів (секретний ключ і відкритий ключ). Для цього відправник (автор) електронних документів обчислює два більших простих числа P и Q, потім знаходить їхній добуток N = P * Q і значення функції (N) = (P-1)(Q-1).

Далі відправник обчислює число E з умов:

$$E (N), НОД (E, (N)) =1$$

і число D з умов:

$$D < N, E * D \equiv 1 (mod (N)).$$

Пари чисел (E, N) є відкритим ключем. Цю пару чисел автор передає партнерам по переписці для перевірки його цифрових підписів. Число D зберігається автором як секретний ключ для підписування.

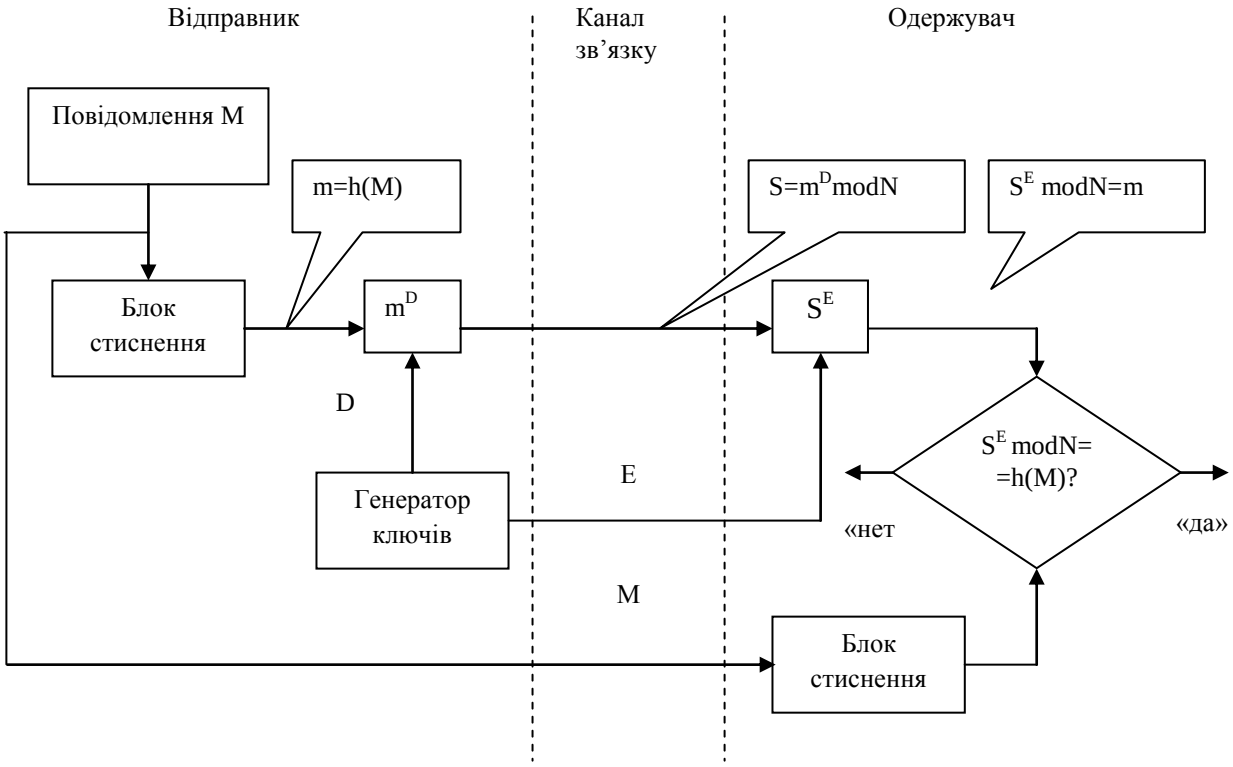

Рис. 4 – Узагальнена схема цифрового підпису RSA

Допустимо, що відправник хоче підписати повідомлення M перед його відправленням. Спочатку повідомлення M (блок інформації, файл, таблиця) стискають за допомогою геш-функції h(M) у ціле число m:

$$m = h(M).$$

Потім обчислюють цифровий підпис S під електронним документом M, використовуючи геш-значення m і секретний ключ D:

$$S = m (mod N).$$

Пари (M, S) передається партнерові-одержувачеві як електронний документ M, підписаний цифровим підписом S, причому підпис S сформований власником секретного ключа D.

Після прийому пари (M, S) одержувач обчислює геш-значення сообщения M двома різними способами. Насамперед він відновлює геш-значення m', застосовуючи криптографічне перетворення підпису S з використанням відкритого ключа E:

$$m' = S^E (mod N).$$

Крім того, він знаходить результат гешировання прийнятого повідомлення M с допомогою такої ж геш-функції h():

$$m = h(M).$$

Якщо дотримується рівність обчислених значень, тобто

$$S^E (mod N) = h(M),$$

те одержувач визнає пару (M,S) справжньої. Доведено, що тільки власник секретного ключа D може сформувати цифровий підпис S по документі M, а визначити секретне число D по відкритому числу E не легше, ніж розкласти модуль N на множники.

Крім того, можна строго математично довести, що результат перевірки цифрового підпису S буде позитивним тільки в тому випадку, якщо при обчисленні S був використаний секретний ключ D. відповідному відкритому ключу E. Тому відкритий ключ E іноді називають "ідентифікатором" що підписав.

Недоліки цифрового підпису RSA.

1. При обчисленні модуля N. ключів E и D для системи цифрового підпису RSA необхідно перевіряти велику кількість дополнительних умов, що зробити практично важко. Невиконання кожного із цих умов уможливлює фальсифікацію цифрового підпису з боку того, хто виявить таке невиконання. Під час підписання важливих документів не можна допускати таку можливість навіть теоретично.

2. Для забезпечення криптостойкости цифрового підпису RSA стосовно спроб фальсифікації на рівні, наприклад, національного стандарту США на шифрування інформації (алгоритм DES), тобто 10^{18}, необхідно використовувати при обчисленнях N, D і E цілі числа не менш 2^{512} (або близько 10^{154}) кожне, що вимагає більших обчислювальних витрат, що перевищують на 20...30% обчислювальні витрати інших алгоритмів цифрового підпису при збереженні того ж рівня криптостойкости.

3. Цифровий підпис RSA уразливий до так званої мультиплікативної атаки. Інакше кажучи, алгоритм цифрового підпису RSA дозволяє зловмисникові без знання секретного кпюча D сформувати підписи під тими документами, у яких результат гешировання можна обчислити як добуток результатів гешировання вже підписаних документів.

Висновок

Електронний цифровий підпис - ефективне рішення для всіх, хто хоче йти в ногу з новими вимогою часу. Якщо ви Не маєте часу чекати приходу фельд'єгерської або кур'єрської пошти за багато сотень кілометрів, щоб перевірити підтвердити висновок угоди або дійсність отриманої інформації. Переваги ЕЦП очевидні - документи, підписані електронним цифровим підписом, можуть бути передані до місця призначення протягом декількох секунд. Всі учасники електронного обміну документами одержують рівні можливості незалежно від їхньої віддаленості друг від друга. Границі завдяки новим технологіям стираються в 21 столітті.

Підробити ЕЦП неможливо – для цього потрібно величезної кількості обчислень, які не можуть бути реалізовані при сучасному рівні обчислювальної техніки й математики за прийнятний час, тобто поки інформація, що втримується в підписаному документі, зберігає актуальність.

Додаткова защита від підробки забезпечується сертифікацією центром, Що Засвідчує, відкритого ключа підпису. Крім того за бажанням клієнта центр, Що Засвідчує, може застрахувати ЕЦП клієнта.

З використанням ЕЦП міняється мислення схема роботи "розробка проекту в електронному виді - створення паперової копії для підпису - пересилання паперової копії з підписом - розгляд паперової копії - перенос її в електронному виді на комп'ютер" іде в минуле.

Список використаних джерел

[1] Молдовян Н. А. Практикум по криптосистемам с открытым ключом. - СПб.: БХВ -Петербург, 2007. - 304 с: ил.
[2] Математические и компьютерные основы криптологии: Учеб пособие / Ю.С. Харин В.И. Берник, Г.В. Матвеев, С.В. Агиевич. -Мн.: Новое знание, 2003. - 382 с.
[3] Фергюсон Н., Шнайер Б. Практическая криптография: Пер. с англ. - М.: Изд. Дом «Вильямс», 2005. - 424е.
[4] Осипян В. О., Осипян К. В. Криптография в задачах и упражнениях.-М.: Гелиос АРВ, 2004.-144с.
[5] Криптографические методи защити информации. Совершенные шифры: Учеб. пособие/ А.Ю. Зубов. -М.: Гелиос АРВ, 2005. - 192с.
[6] Мао, Венбо. Современная криптография: теория и практика: Пер. с англ. - М: Изд. Дом «Вильямс», 2005. -768с.
[7] Коблиц Н. Курс теории чисел и криптографии.-М., 2001.
[8] Лекции по дискретной математике / Ю.В.Капитонова, С.Л. Кривой, А.А. Летичевский, Г.М. Луцкий / СПб.: БХВ - Петербург, 2004. - 624с.
[9] Шаньгин В.Ф. Информационная безопасность компьютерных систем и сетей. - Изд - во: Форум, 2О07.-416с.
[10] Столлингс В. Криптография и защита сетей. Принципи и практика. Изд-во Диалектика -2001. ~ 672с.

References

[1] Moldovyan N. A. *Workshop on public key cryptosystems.* SPb.: BHV-Petersburg, 2007. 304 p .
[2] Kharin Yu. S., Bernik V. I., Matveev G. V., Agievich S. V. *Mathematical and computer foundations of cryptology: Textbook*. Moskow, New knowledge, 2003. 382 p.
[3] Ferguson N., Schneier B. *Practical cryptograph,* (transl. from eng.) Moskow, Publishing. The Williams House, 2005. 424 p.
[4] Osipyan V. O., Osipyan K. V. *Cryptography in tasks and exercises*. Moskow, Helios ARV, 2004. 144 p.
[5] Zubov A. Yu. *Cryptographic methods of information protection. Perfect ciphers: Textbook. allowance*. Moskow, Helios ARV, 2005. 192 p.
[6] Mao, Wenbo. *Modern cryptography: theory and practice*: (transl. from eng.). Moskow, Publ. Williams House, 2005. 768 p.
[7] Koblitz N. *The course of number theory and cryptography*. Moskow, 2001.
[8] Kapitonova Yu. V., Krivoy S. L., Letichevsky A .A., Lutsky G.. M. *Lectures on discrete mathematics*. SPb.: BHV-Petersburg, 2004. 624 p.
[9] Shangin V. F. *Information security of computer systems and networks*. Publisher: Forum, 2007. 416 p.
[10] Stollings W. *Cryptography and network protection. Principles and practice*. Publishing House of Dialectics, 2001. 672 p.