



УКРАЇНА

(19) **UA** (11) **122327** (13) **C2**
(51) МПК
H04L 9/06 (2006.01)
H04L 9/14 (2006.01)

НАЦІОНАЛЬНИЙ ОРГАН
ІНТЕЛЕКТУАЛЬНОЇ
ВЛАСНОСТІ
ДЕРЖАВНЕ ПІДПРИЄМСТВО
"УКРАЇНСЬКИЙ ІНСТИТУТ
ІНТЕЛЕКТУАЛЬНОЇ
ВЛАСНОСТІ"

(12) ОПИС ДО ПАТЕНТУ НА ВІНАХІД

(21) Номер заявки:	а 2017 02158	(72) Винахідник(и):	Фіске Майкл (US)
(22) Дата подання заявки:	28.09.2015	(73) Володілець (володільці):	ФІСКЕ СОФТВАРЕ ЛЛС, 1449 Lake Street, San Francisco, CA 94118, United States of America (US)
(24) Дата, з якої є чинними права інтелектуальної власності:	27.10.2020	(74) Представник:	Ісаєва Світлана Геннадіївна, реєстр. №426
(31) Номер попередньої заявки відповідно до Парижської конвенції:	PCT/US2014/050462, 62/056,537, 14/843,999	(56) Перелік документів, взятих до уваги експертизою:	WO 2007/075156 A2, 05.07.2007 BOROWSKI MARIUSZ. The sponge construction as a source of secure cryptographic primitives. MILITARY COMMUNICATIONS AND INFORMATION SYSTEMS CONFERENCE, MILITARY UNIVERSITY OF TECHNOLOGY, 07.10.2013, pages 1 - 5, XP 032547287 PAWEL MORAWIECKI ET AL. A SAT-based preimage analysis of reduced KECCAK hash functions. INTERNATIONAL ASSOCIATION FOR CRYPTOLOGIC RESEARCH. 19.10.2010, vol. 20101019:140641, pages 1 - 12, XP 061004279
(32) Дата подання попередньої заявки відповідно до Парижської конвенції:	10.08.2014, 28.09.2014, 03.09.2015		
(33) Код держави-учасниці Парижської конвенції, до якої подано попередню заявку:	US, US, US		
(41) Публікація відомостей про заявку:	10.07.2017, Бюл.№ 13		
(46) Публікація відомостей про державну реєстрацію:	26.10.2020, Бюл.№ 20		
(86) Номер та дата подання міжнародної заявки, поданої відповідно до Договору РСТ	PCT/US2015/052734, 28.09.2015		

(54) NADO- КРИПТОГРАФІЯ З ГЕНЕРАТОРАМИ КЛЮЧІВ

(57) Реферат:

Представлена симетрична криптографія для шифрування і розшифровки інформації яка може бути ефективним чином реалізована апаратним або програмним способом. Симетрична криптографія використовує генератор ключів так, щоб криптографія не залежала від одного статичного криптографічного ключа. Генератор ключів являє собою величину або набір величин, з яких генерується перший ключ. Генератор ключів істотно збільшує обчислювальну складність диференційного криптоаналізу і інших криптографічних атак. В одному з прикладів здійснення винаходу генератор ключів оновлюється з використанням односторонніх функцій, що демонструють лавинний ефект, який створює непередбачувану послідовність ключів, які використовуються в процесі шифрування або розшифрування. В одному з прикладів здійснення винаходу динамічний ключ отримують з генератора ключів за допомогою односторонньої функції хешування. В одному з прикладів здійснення винаходу блоковий шифр використовує різні динамічні ключі для шифрування кожного блока звичайного тексту, де кожен ключ отримують з різного генератора ключів.

UA 122327 C2

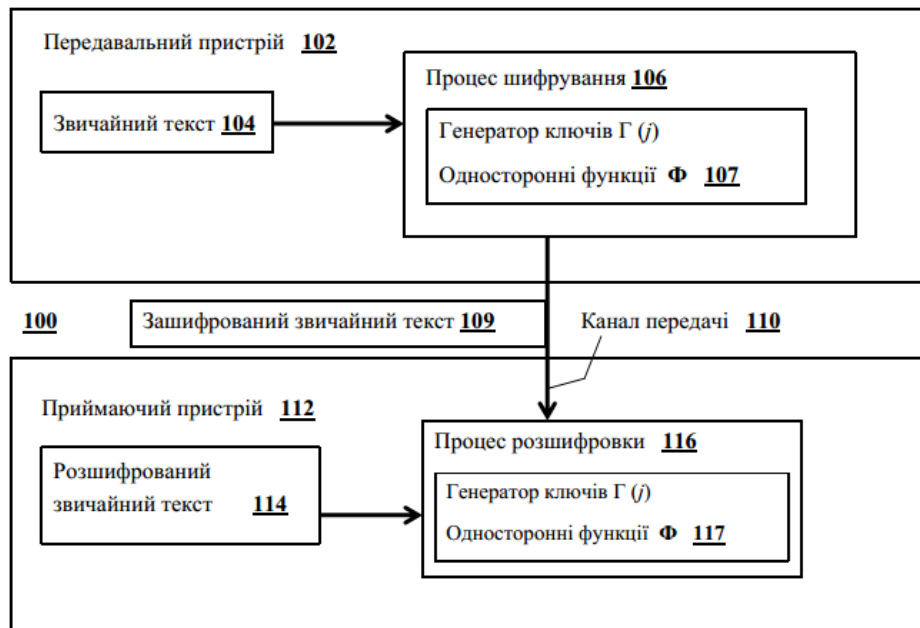


Fig. 1A

1. ПОВ'ЯЗАНІ ЗАЯВКИ

Дана заявка посилається на попередню заявку на патент США № 61 / 865,134 "NADO криптографія з використанням односторонніх функцій", занесену до реєстру 13 серпня 2013 року; дана заявка посилається на попередню заявку на патент США № 61/992,915 "NADO криптографія з генераторами ключів з використанням односторонніх функцій", занесену до реєстру 14 травня 2014 р; дана заявка посилається на попередню заявку на патент США № 62/004,852 "NADO криптографія з використанням односторонніх функцій", занесену до реєстру 29 травня 2014 р; дана заявка посилається на Міжнародну заявку на патент PCT/US14/50462 "NADO криптографія з використанням односторонніх функцій", занесену до реєстру 10 серпня 2014 р; дана заявка посилається на попередню заявку на патент США № 62/056,537 "Генератори ключів підсилюють симетричну криптографію", занесену до реєстру 28 вересня 2014 р; дана заявка посилається на заявку на патент США № 14/292,935 "NADO криптографія з використанням односторонніх функцій", занесену до реєстру 1 червня 2014 р; дана заявка посилається на заявку на патент США № 14 / 843,999 "NADO криптографія за допомогою генераторів ключів", занесену до реєстру 3 вересня 2015 року.

2. ОБЛАСТЬ ВИНАХОДУ

Даний винахід, загалом, відноситься до криптографічних методів і пристроїв. У деяких прикладах здійснення винахід відноситься до симетричних методів і пристроїв для криптографії. Криптографічні пристрої і методи в основному використовуються для шифрування і дешифрування інформації, яка передається за допомогою систем зв'язку і передачі даних. Наприклад, криптографічні методи можуть використовуватися для того, щоб шифрувати телефонний дзвінок; в деяких прикладах здійснення винаходу телефонний дзвінок може передаватися за допомогою голосу через IP (інтернет-протоколу), використовуючи мобільний телефон. Ці методи також можуть використовуватися для шифрування пасивних даних на комп'ютері або іншому фізичному пристрої, такому як накопичувач на магнітній стрічці. Зазвичай інформація, яка шифрується відправником, який іноді називається Боб, з використанням його унікального(их) ключа(ів), а зашифрована інформація, яка називається зашифроване повідомлення, передається одержувачу, який іноді називається Еліс. Одержувач, Еліс, використовує свій (ої) унікальний (і) ключ (і), щоб застосувати пристрій або метод шифрування до зашифрованого повідомлення. На виході з цього пристрою або методу шифрування виходить та сама інформація, яку відправник зібрав перед тим, як її зашифрувати і відправити. Єва - це назва агента, який намагається розшифрувати зашифроване повідомлення. Однією з первинних цілей Еліс або Боба є гарантія того, що Єва не зможе розшифрувати зашифроване повідомлення, яке передається між ними.

3. РІВЕНЬ ТЕХНІКИ

Об'єкт винаходу, який описується в даному розділі, не повинен вважатися таким, який є попереднім рівнем техніки тільки в результаті згадки про нього в даному розділі. Так само, задача, що зазначена в даному розділі, або пов'язана з об'єктом винаходу розділу, не повинна вважатися такою, яка раніше була визнана в попередньому рівні техніки. Об'єкт винаходу в розділі "Суть винаходу" представляє собою різні підходи, які самі по собі також можуть бути винаходами, а також різні проблеми, які можливо винахідник визнав першим.

Посилання [1] надає практичний і теоретичний опис криптографії та криптографічних методів. Посилання [2, 3, 4] також надають опис відомих криптографічних методів, які знаходяться у відкритому доступі. Криптографічний захист з відкритим ключем зазвичай використовується для управління ключами і величезного числа протоколів. Симетрична традиційна криптографія є корисною для шифрування даних і захисту приватних голосових і письмових передач даних.

У симетричній криптографії Еліс шифрує свій відкритий текст, а Боб розшифровує зашифроване повідомлення, отримане від Еліс, використовуючи той же закритий ключ. Ключ називається закритим, щоб продемонструвати те, що Еліс і Боб не хочуть, щоб Єва отримала цей ключ. Наприклад, алгоритм блочного шифрування $EA : \{0, 1\}^m \times \{0, 1\}^k \rightarrow \{0, 1\}^m$ використовує к-бітний ключ K в якості параметра і шифрує m -бітний блок відкритого тексту M , іменований як $EA(M, K)$. Ємність ключа блочного шифру $\{0, 1\}^k$ має розмір 2^k . Ємність ключа блочного шифру $\{0, 1\}^m$ має розмір 2^m . Алгоритм розшифровування $DA : \{0, 1\}^m \times \{0, 1\}^k \rightarrow \{0, 1\}^m$ має зворотню особливість, о $DA(EA(x, K), K) = x$ для кожного відкритого тексту $x \in \{0, 1\}^m$ і кожного ключа $K \in \{0, 1\}^k$.

Стандарт AES (вдосконалений стандарт шифрування) являє собою блоковий шифр з розміром блоку 16 байт (128 біт), який являє собою симетричний криптографічний алгоритм [5, 6]. Стандарт AES широко використовується в промисловості, рекомендований NIST (Національний інститут стандартів і технологій) і використовується Міністерством Оборони

Сполучених Штатів Америки. Стандарт AES є найбільш широко використовуваним ключем блочного шифрування на сьогоднішній день. Наприклад, стандарт AES-128, який використовує 128-бітові статичні ключі, в даний час використовується додатком FileVault на комп'ютерах Apple. FileVault шифрує жорсткий диск всередині комп'ютера Apple.

Протягом останніх років, різні атаки на шифр стандарту AES (відомий рівень техніки) продемонстрували слабкість в цьому шифрі. У деяких випадках практичні атаки доповненням на Oracle (oracle padded attacks) [7] були в змозі отримати звичайний текст із зашифрованого тексту, який шифрувався за стандартом AES. Принаймні, частиною слабкого місця є повільне перетворення відкритого тексту для порушення його статистичної структури в плануванні ключа [8, 9, 10]. Слабкі місця стандарту AES далі посилюються статичним блоком підстановки, і через свій статичний ключ стандарт AES перетворює два ідентичних блоку відкритого тексту в два ідентичних блоку зашифрованого тексту. Зокрема, ймовірну атаку на стандарт AES-256 демонструє менша на 1 кількість циклів - 13 циклів замість 14 - в таблиці ключів [11]. Взагалі, в останні роки були відкриті додаткові загальновідомі (не засекречені) атаки на стандарт шифрування AES [12, 13, 14, 15], які припускають, що стандарт AES не є таким самим крипостійким шифром, як вважало раніше криптографічне співтовариство.

Більш того, відомий рівень техніки [1, 2, 6, 16] не розкриває поняття послідовності генератора ключа, ані отримання нового ключа, виходячи з оновлення генератора ключа. Використання генераторів ключа в даному винаході виключає залежність криптографічної надійності від одного статичного криптографічного ключа. Зазвичай, криптографічні дані у відомому рівні техніки використовують статичний ключ K протягом всього виконання алгоритму шифрування. Використання статичного ключа у відомому рівні техніки далі впливає з деяких атак, як описано в попередньому абзаці, які намагаються отримати або відтворити статичний ключ. У відомому рівні техніки, якщо статичний ключ отримано, то криптографічна безпека повністю скомпрометована. Навпаки, в прикладах здійснення, описаних в даному винаході, якщо один з динамічних ключів отриманий Євою, то Єва все одно не може знайти попередні динамічні ключі, які використовують Еліс і Боб, а також Єва не може знайти або отримати майбутні динамічні ключі, які використовують Еліс і Боб.

Як приклад припущення статичного ключа у відомому рівні техніки, деякі атаки, пов'язані з доповненнями [7] покладаються на той факт, що у Єви є деяка інформація щодо останнього блоку. Це допомагає Єві працювати в зворотному напрямку для знаходження статичного ключа. В даному винаході використовується для того одного блоку відкритого тексту, оскільки для цього потрібна була б атака знаходження використання односторонньої функції для створення динамічних ключів перешкоджає Єві отримати навіть той ключ, який прообразу на односторонню функцію без будь-якої прямої інформації, що стосується профілю, і важко вирішуваний пошук по величезній послідовності непередбачуваних ключів. У відомому рівні техніки Єві доводиться здійснювати пошук тільки одного статичного ключа, який використовується для шифрування кожного блоку звичайного тексту.

4. СУТЬ ВИНАХОДУ

Винахід(и), описаний(і) в даному документі є процесом шифрування і розшифровки інформації, який використовується в системах зв'язку, передачі і зберігання даних. Метод називається Н-процес. Однією з цілей Н-процесу є використання односторонньої функції для шифрування звичайного тексту. Н-процес використовує генератор ключа, оновлюючи метод, який використовує односторонні функції.

Термін "генератор ключа" використовується в даному описі, щоб позначити значення або сукупність значень, за якими виконується одна або кілька операцій для створення іншого значення або набору значень, з якого отримують або можна отримати ключ. "Послідовність генераторів ключів" представляє собою послідовність генераторів ключів. У прикладі здійснення винаходу присвоювання позначення для "послідовності генераторів ключів" може бути автоматично представлено, як функція $\Gamma: N \rightarrow \{0, 1\}^n$, де N - це натуральні числа, а $\{0, 1\}^n$ - це набір всіх послідовностей бітів довжиною n . Наприклад, послідовність бітів 10101 являє собою елемент $\{0, 1\}^5$. Наприклад, набір всіх послідовностей бітів довжиною 3 являє собою $\{0, 1\}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$. Тому в описі k -тий генератор послідовності генератора ключів Γ буде позначатися, як $\Gamma(k)$.

У деяких прикладах здійснення винаходу фактичне отримання ключа є обов'язковим. Наприклад, частина k -того генератора ключів могла б використовуватися в якості k -того ключа. В даному описі в тих випадках, коли згадуються ключ і генератор ключів, в одному прикладі здійснення винаходу є окремий ключ і генератор ключів, де ключ отримують з генератора ключів; в іншому прикладі здійснення винаходу відсутній окремий ключ, який отримують фактично, а частина нинішнього генератора ключів може використовуватися в якості ключа. В

даному описі слово "ключ" і термін "криптографічний ключ" використовуються як взаємозамінні для позначення одного і того ж. Ключ являє собою одне або кілька значень, які описують, як конкретна функція шифрування буде шифрувати повідомлення. Наприклад, ключ може являти собою послідовність нулів і одиниць, які побитово виключають або виключаються біти/бітами, які складають повідомлення, щоб сформувати зашифроване повідомлення. Інші приклади використання ключів, як частини методів шифрування, наводяться в інших місцях в даному описі.

У деяких прикладах здійснення винаходу, Н-процес може бути реалізований за допомогою блочного шифру, де генератор ключів Γ оновлюється після того, як один або декілька блоків звичайного тексту були зашифровані за допомогою блочного шифру. Іноді послідовність всіх блоків простого тексту, які шифруються під час сесії шифрування, іменуються, як повідомлення звичайним текстом або повідомлення. У деяких прикладах здійснення винаходу, даний блоковий шифр може являти собою покращений AES-256 або покращений AES-128, або ж покращений Serpent. В даному описі, коли використовуються терміни "покращений" AES або "покращений" Serpent, це означає, що покращений AES і покращений Serpent більш не використовують статичний ключ під час шифрування або дешифрування. Далі, в іншому контексті покращений AES або покращений Serpent означають, що використовується динамічний ключ, який отриманий з генератора ключів. Ці поліпшення використовують послідовності генераторів ключів та динамічні ключі. В даному описі "стандартний" AES [5, 6] або "стандартний" Serpent [39] будуть використовуватися для опису відомого рівня техніки AES і Serpent, відповідно, в тих випадках, коли для кожного зашифрованого або дешифрованого блоку використовується статичний ключ.

Винахід вводить поняття послідовності генераторів ключів, поновлення генератора ключів та динамічних ключів. Це дає можливість кожному ключу, який використовується Н-процесом непередбачувано оновлюватися після того, як процес зашифрував один або більше блоків звичайного тексту. Більш того, в усьому цьому описі генератор ключів може бути значно більше, ніж той ключ, який використовується Н-процесом. Оновлення генератора ключів створює сприятливі криптографічні особливості і зміцнює криптографічні шифри, які вже існують і вже перевірені.

У прикладі здійснення винаходу j -тий генератор ключів $\Gamma(j)$ являє собою n біт за довжиною. $\Gamma(j)$ оновлюється до $\Gamma(j+1)$ шляхом застосування односторонньої функції хешування до q біт $\Gamma(j)$, де $q < n$, а профіль повідомлення з'єднується з рештою $n - q$ біт $\Gamma(j)$. Взагалі, $n - q$ біт $\Gamma(j)$ залишаються без змін, а інші q біт змінюються, завдяки односторонньої функції хешування. У прикладі здійснення винаходу динамічний ключ отримують з $\Gamma(j)$, і він використовується блоковим шифром для шифрування звичайного тексту. У прикладі здійснення винаходу дане шифрування може працювати як автономна симетрична криптографія. В альтернативному прикладі здійснення винаходу дане шифрування за допомогою динамічного ключа працює як Н-процес.

Даний метод оновлення генератора ключів використовує лавинний ефект односторонніх функцій хешування і призводить до того, що початковий генератор ключів $\Gamma(0)$ виконує перебір по величезній орбіті підстановок. На фіг. 1С показаний приклад лавинного ефекту для односторонньої функції хешування SHA-1 [18]. Якщо говорити більш конкретно, послідовність $\Gamma(0), \Gamma(1), \Gamma(2), \dots, \Gamma(n) \dots$ має накладень до тих пір, поки послідовність генераторів ключів нагадує довжину, яка передбачена парадоксом днів народжень, виходячи з рівномірного розподілу ймовірностей. На сторінці 77 джерела [1] представлено опис добре відомого парадоксу днів народжень.

Говорячи більш детально, кожен генератор ключів може бути представлений, як остаточно послідовність символів: наприклад, кожен генератор ключів може бути представлений J бітами. У зв'язку з цим, ефект дня народження може використовуватися в якості однієї статистичної перевірки непередбачуваності послідовності генераторів ключів, якщо ймовірність повторення (тобто накладення) будь-якого даного генератора ключів в послідовності має такий самий порядок, як і прогнозує ефект дня народження.

Припустимо, що $m \neq 2^j \in \mathbb{Z}^+$ є загальним числом можливих генераторів ключів. Тоді n , яка задовольняє рівнянню $\frac{n}{2} = m^{\lceil \frac{n}{m} \rceil}$ позначає розмір послідовності генераторів ключів, яка

має приблизно 50-відсоткову ймовірність виникнення накладення при допущенні рівномірного розподілу ймовірностей. Для кінцевої послідовності генераторів ключів, в якій початок послідовності генератора ключів коротший, ніж n має менш, ніж 50 % ймовірності виникнення накладення.

$1 - e^{\frac{-n^2}{2m}}$ це задовільне наближення для $\frac{m!}{m(m-n)!}$. В даному наближенні, рішення $\frac{1}{2} = 1$

$- e^{\frac{-n^2}{2m}}$ дає приблизно розмір зразка, при якому є 50-відсоткова ймовірність накладення. Рішення для n , число n приблизно є квадратним коренем з $(2m \ln 2)$, де $\ln x$ позначає логарифм натуральний дійсного числа x . В даному описі, принаймні, деяка корисність в зв'язку з лавинним ефектом може вимірюватися, як найбільше число генераторів ключів в послідовності, яка ймовірно не матиме повторення. У прикладі здійснення винаходу хороший лавинний ефект виникає, коли усереднені послідовності генераторів ключів, які менше або дорівнюють добутку 0,9 і квадратного кореня з $(2m \ln 2)$, ймовірно не матимуть накладення. В іншому прикладі здійснення винаходу, якщо кількість генераторів ключів в послідовності задається квадратним коренем з $(2m \ln 2)$, тоді послідовність має ймовірність накладання 55% або менше.

Більш того, взагалі, період орбіти Γ значно більший, ніж кількість можливих ключів, і зазвичай знаходиться в порядку $2^{\frac{|\Gamma|}{2}}$, де $|\Gamma|$ - це довжина генератора ключа Γ . Наприклад, якщо $|\Gamma| = 1024$ біта, а генератор ключів $\Gamma(n)$ використовується для отримання нового поліпшеного 256-бітного ключа AES для n -го блоку з 16 байт звичайного тексту, тоді очікувана довжина (період) цієї орбіти істотно більша, ніж 2^{256} , незважаючи на те, що для H -процесу отримання 256-бітного ключа від кожного генератора ключів являє собою орбіту $\{0,1\}^{256}$.

Зокрема, 50-відсоткова ймовірність накладання в послідовності генератора ключів $\Gamma(0)$, $\Gamma(1)$. . . передбачається для послідовності довжиною $n = \text{корінь квадратний з } (2^{1025} \ln 2) > 10^{154}$. Коли використовується даний метод оновлення генератора ключів - використання однієї або декількох односторонніх функцій з хорошим лавинним ефектом - де покращений AES-256 є блоковим шифром в H -процесі, то це істотно збільшує складність розрахунку, яку необхідно подолати для того, щоб порушити H -процес, в порівнянні зі стандартним шифром AES.

Мотивацію для нового поняття генератора ключів, а також його конструкцію також можна зрозуміти з точки зору диференціального криптоаналізу [19]. У поліпшеному шифрі AES-256, кожен унікальний 256-бітний ключ K створює логічну функцію шифрування $E(K, \cdot)$, де $E: \{0, 1\}^{256} \times \{0, 1\}^{128} \rightarrow \{0, 1\}^{128}$. Іншими словами, ключ K діє як параметр, де кожне $E(K, \cdot)$ є функцією $f: \{0, 1\}^{128} \rightarrow \{0, 1\}^{128}$ при $f = (f_1, \dots, f_{128})$ і кожна $f_k: \{0, 1\}^{128} \rightarrow \{0, 1\}$. Як обговорювалося в [20], кожна f_k має ступінь ≤ 128 . З цієї точки зору послідовність динамічних ключів створює високу, тривимірну орбіту над простором функцій $\{f \mid f: \{0, 1\}^{128} \rightarrow \{0, 1\}^{128}\}$, яка істотно підвищує ефективну ступінь. Загальні, динамічні ключі, отримані при оновленні генератора ключів та на підставі односторонніх функцій з хорошим лавинним ефектом, створюють потужний криптографічний метод, який може покращувати криптографічну стійкість примітиву, такого як блоковий шифр AES-256, який вже аналізується протягом багатьох років.

Далі, в деяких прикладах здійснення винаходу властивість повноти і лавинний ефект хорошою(их) односторонньою(їх) функцією(ї) забезпечує можливість, щоб генератори послідовних ключів $\Gamma(n)$ і $\Gamma(n+1)$ мали хеммінговську відстань, яка становить приблизно $1/2$ від $|\Gamma(n)|$, яке означає, що $\Gamma(n) \oplus \Gamma(n+1)$ становить приблизно половину одиниць і половину нулів, а їх порядок є непередбачуваним. Дана вигідна властивість заважає атакам, пов'язаним з ключами. Зазвичай односторонні функції хешування використовуються для перевірки справжності інформації. Інформацію, справжність якої перевіряється, іноді називають повідомленням в криптографічній літературі, яка обговорює односторонні функції хешування. У відомому рівні техніки односторонні функції хешування не використовувалися безпосередньо в зашифруванні і дешифруванні, оскільки односторонні функції хешування не є однозначними.

Функція $\sigma: X \rightarrow X$ перетворює елементи X в елементи X . Функція σ є однозначною; це означає, що ніякі два різні елементи X не перетворюються за допомогою σ в однаковий елемент. Більш формально, якщо s_1, s_2 є двома різними елементами з X , іншими словами, якщо s_1 не дорівнює s_2 , то $\sigma(s_1)$ не дорівнює $\sigma(s_2)$.

У відомому рівні техніки зазвичай одностороння функція застосовується безпосередньо до звичайного тексту в процесі зашифрування, або ж одностороння функція безпосередньо застосовується до шифрованого тексту в процесі розшифровки. Для цього методу, в якому одностороння функція застосовується безпосередньо до тексту або шифрованому тексту, якщо одностороння функція, яка використовується у відомому рівні техніки, не є однозначною, тоді два або більше звичайних текстів можуть бути прив'язані за допомогою цієї односторонньої функції до того ж шифрованого тексту. У публікаціях FIPS 180-4, Стандарт з безпечного хешування, написаному Національним інститутом стандартів (NIST), в анотаціях [21] сказано:

В даному стандарті наводяться алгоритми хешування, які можуть використовуватися для

створення профілів повідомлень.

Профілі використовуються для того, щоб визначити, чи змінювалися повідомлення відтоді, як були створені профілі.

Даний опис наводить принципово нове використання односторонніх функцій для непередбачуваного поновлення генераторів ключів, а також спотворення Н-процесу, який використовується в криптографії. У прикладах здійснення винаходу Н-процес використовує односторонні функції. Лавинна властивість односторонніх функцій допомагає зміцненню NADO-криптографії проти атак за допомогою диференціального криптоаналізу і інших видів атак.

NADO може бути ефективно реалізовано в апаратному або програмному забезпеченні. У деяких прикладах здійснення винаходу процес Н є блоковим шифром. Іншим поліпшенням є складність злому цього методу шифрування, як функція його швидкості виконання. Для виконаного коду, який реалізує приклад здійснення винаходу NADO, потрібна невелика кількість комп'ютерної пам'яті, менш, ніж 20К оперативної пам'яті для навіть відносно великих генераторів ключів Г і менш, ніж 5К в інших прикладах здійснення винаходу. Приклад здійснення винаходу може виконуватися з процесором 150 МГц на комп'ютері з архітектурою скороченого набору команд (Reduced Instruction Set Computer (RISC)) [22]: даний приклад здійснення винаходу захищає конфіденційність бесіди по мобільному телефону в режимі реального часу. Для даного прикладу здійснення винаходу генератор ключа Г для Н-процесу має розмір, як мінімум, 512 біт; Далі в даному прикладі здійснення винаходу в мобільному телефоні в режимі реального часу, кожні з цих генераторів ключів є незалежними від інших двох і оновлюються за допомогою односторонньої функції хешування SHA-512 [23] або іншої односторонньої функції хешування, такої як Кессак, Blake, Skein або $Gr^{\circ} stl$. Деякі приклади здійснення винаходу NADO є досить швидкими для того, щоб дозволити програмам, таким як шифрування в режимі реального часу бездротової передачі даних, вбудовані системи реального часу, безпечний зв'язок між супутниками і безпечної маршрутизації і передачі Інтернет-трафіку.

У наступних фіг., хоча вони і можуть відображати різні приклади винаходу, винаходи не обмежуються прикладами, які наведені на рисунках.

Фіг. 1А демонструє приклад здійснення інформаційної системи для відправки та отримання зашифрованої інформації.

Фіг. 1В демонструє приклад здійснення процесу для кодування інформації, який може використовуватися в прикладі здійснення винаходу на Фіг. 1А.

Фіг. 1С демонструє приклад лавинного ефекту після 16 циклів односторонньої функції хешування SHA-1 на перших 46 бітах вихідних даних SHA-1, які можуть використовуватися в прикладі здійснення винаходу на Рисунку 1А.

Фіг. 1D демонструє схему прикладу здійснення напівпровідникового мікропроцесора, який визначає фотони і генерує недетермінований процес, який може використовуватися в прикладі здійснення винаходу на Фіг. 1А.

На Фіг. 1Е показана схема прикладу здійснення одного етапу оновлюваного генератора ключів за допомогою односторонньої функції хешування Ф. Розмір генератора ключа становить n біт. Вихідний розмір функції хешування Ф становить q біт. В даному прикладі здійснення винаходу на етапі поновлення генератора ключа одностороння функція Ф застосовується до перших q бітів генератора ключів, а останні $n - q$ біт генератора ключів залишаються без змін. Як описано в методі 1, фіг. 1Е демонструє етап Серія $(\Gamma_{i+1,0} \ \Gamma_{i+1,1} \ \dots \ \Gamma_{i+1,q-1}) = \Phi(\Gamma_{i,0} \ \Gamma_{i,1} \ \dots \ \Gamma_{i,q-1})$, виражений як $\Phi(b_1 \ \dots \ b_q) = (c_1 \ \dots \ c_q)$. На фіг. 1Е також показаний наступний етап Серія $\Gamma_{i+1,j} = \Gamma_{i,j}$ для кожної j , що задовольняє $q \leq j \leq n-1$, виражена як $(a_1 \ a_2 \ \dots \ a_{n-q}) = (a_1 \ a_2 \ \dots \ a_{n-q})$ на цій фігурі.

На Фіг. 1F показана схема альтернативного прикладу здійснення одного етапу оновлюваного генератора ключів за допомогою односторонньої функції хешування Ф. Розмір генератора ключа становить n біт. Вихідний розмір функції хешування Ф становить q біт. В даному альтернативному прикладі здійснення винаходу на етапі поновлення генератора ключа одностороння функція Ф застосовується до останніх q бітів генератора ключів, а перші $n - q$ біт генератора ключів залишаються без змін.

На Фіг. 1G показана схема ключа з k битами, який отриманий з генератора ключів. Одностороння функція Ψ застосовується до генератора ключів, а перші k біт цих вихідних даних вибираються в якості динамічного ключа. Фіг. 1G являє собою етап отримання ключа, який відповідає етапу оновлення генератора ключа, що показаний на Фіг. 1Е.

На Фіг. 1H показана схема ключа з k битами, який отриманий з генератора ключів. Одностороння функція Ψ застосовується до генератора ключів, а перші k біт цих вихідних даних вибираються в якості динамічного ключа. Фіг. 1H являє собою етап отримання ключа, який відповідає етапу оновлення генератора ключа, що показаний на Фіг. 1F.

Фіг. 1I демонструє приклад здійснення винаходу в формі процесу для кодування інформації, який може використовуватися в прикладі здійснення винаходу на Фіг. 1A.

На Фіг. 2A показаний приклад здійснення винаходу в формі комп'ютерної мережі, що передає зашифрований звичайний текст, яка в деяких прикладах здійснення винаходу може бути мережею Інтернет або частиною мережі, яка підтримує інфраструктуру, таку як електрична мережа, фінансова біржа, або електростанція, яка може використовуватися з прикладом здійснення винаходу на Рисунку 1A.

На фіг. 2B показаний приклад здійснення винаходу в формі захищеної обчислювальної зони для шифрування інформації, яка містить у собі процесор, пам'ять і систему вводу/виводу, які можуть бути пристрої, передають і/або приймають на Фіг. 1A.

На Рисунку 3A показаний приклад здійснення винаходу в формі USB-диска, який може діяти в якості відправляючого пристрою і приймаючого пристрою для зберігання і захисту даних користувача шляхом шифрування даних.

На Фіг. 3B показаний приклад здійснення винаходу в формі аутентифікаційного маркера, який може включати в себе пристрої для відправлення і/або приймання з Фіг. 1A, який містить комп'ютерний процесор, який може зашифрувати звичайний текст, що представляє собою дані аутентифікації.

На Фіг. 4 показаний приклад здійснення винаходу у вигляді мобільного телефону 400, який зашифровує бездротові голосові дані, які можуть включати в себе пристрої для відправлення і/або приймання з Фіг. 1A. Мобільний телефон 500 являє собою приклад здійснення винаходу, який відправляє бездротовий зашифрований звичайний текст в автомобіль, який може містити передаючі/приймаючі пристрої на Фіг. 1A.

На Фіг. 5A показаний приклад здійснення винаходу в Н-процесі, реалізованому за допомогою покращеного блочного шифру AES-256 [5], який може використовуватися в передаючих/приймаючих пристроях на Фіг. 1A.

На Фіг. 5B показаний ще один приклад здійснення винаходу в Н-процесі, реалізованому за допомогою покращеного блочного шифру Serpent [39], який може використовуватися в передаючих/приймаючих пристроях на Фіг. 1A.

6. ДОКЛАДНИЙ ОПИС

Хоча різні приклади здійснення винаходів, можливо, і були продиктовані різними вадами у відомому рівні техніки, які можуть обговорюватися або згадуватися в одному або декількох місцях в описі, приклади здійснення винаходу не обов'язково стосуються будь-яких з цих недоліків. Іншими словами, різні приклади здійснення винаходу можуть стосуватись різних недоліків, які можуть обговорюватися в описі. Деякі приклади здійснення винаходу можуть тільки частково згадувати деякі недоліки або тільки один недолік, які можуть обговорюватися в описі, а деякі можуть не згадувати будь-які з цих недоліків.

Розділ 6.1 описує інформаційні системи, які використовують криптографічний процес. Розділ 6.2 описує лавинний ефект і односторонні функції. Розділ 6.4 описує способи шифрування за допомогою блочного шифру, який використовує динамічні ключі, отримані з генераторів ключів та оновлень генерації ключів за допомогою односторонніх функцій хешування. Розділ 6.8 пояснює, як використання динамічних ключів зупиняє просту атаку на блоковий шифр. Розділи 6.5, 6.6, 6.7, 6.9, 6.10 і 6.11 описують принципово нові алгоритми, концепції, апаратне забезпечення, інфраструктуру, пристрої, математику, способи, методику і системи, які роблять внесок у деякі приклади здійснення криптографічного процесу.

6.1 ІНФОРМАЦІЙНА СИСТЕМА

На Фіг. 1A показана інформаційна система 100 для шифрування інформації способом, який, як передбачається, є надійним. Інформаційна система 100 включає в себе звичайний текст (незашифрована інформація), процеси шифрування 106, генератори ключів 107 і одностороннє хешування 107, передавальний пристрій 102, зашифрований звичайний текст (зашифрована інформація) 109 і шлях передачі 110, приймаючий пристрій 112, процес розшифровки 116, розшифрований звичайний текст 114, генератори ключів 117 і одностороннє хешування 117. В інших прикладах здійснення винаходу інформаційна система 100 може не мати всіх компонентів, які перелічені вище, або може мати інші компоненти замість і/або на додаток до перелічених вище.

Інформаційна система 100 може використовуватися для передачі зашифрованого звичайного тексту. Звичайний текст 104 відноситься до інформації, яка ще не була зашифрована, яку передбачається доставити в інше місце, програмний блок, пристрій, іншій особі або об'єкту. Хоча термін "звичайний текст" і містить в собі слово "текст", значення звичайного тексту в даному описі ширше і стосується будь-якого типу інформації, яка не була зашифрована. Наприклад, звичайним текстом можуть бути голосові дані, які ще не були

зашифровані. У прикладі здійснення винаходу звичайний текст може бути розшифрованою інформацією, яка передається по бездротовому зв'язку між супутниками. Звичайний текст може бути представлений в деяких прикладах здійснення винаходу в аналоговій формі, а також може бути представлений в цифровій формі. У прикладі здійснення винаходу звукові хвилі, які передаються з рота того, хто говорить в мікрофон мобільного телефону, є звичайним текстом. Подання інформації в цьому звичайному тексті перед тим, як вона потрапляє в мікрофон, відбувається в аналоговій формі. Згодом, інформація в формі звичайного тексту може бути дискретизованою цифровим способом, тому вона представлена в цифровій формі після того, як її приймає мікрофон мобільного телефону. Взагалі, звичайний текст в даному документі ставиться до будь-якого роду інформації, яка не була зашифрована.

В даному описі, термін "місце розташування" може відноситися до географічних розташувань і/або місць зберігання. Конкретне місце зберігання може являти собою сукупність суміжних і/або несуміжних місць на одному або декількох машинозчитувальних носіях. Два різні місця зберігання можуть позначати два різні набори місць розташування на одному або декількох машинозчитувальних носіях, в яких розташування одного комплекту може переплітатися з місцями розташування іншого комплекту. В даному описі термін "машинозчитуваний носій" використовується для позначення будь-якого носія, який може нести в собі інформацію, яка зчитується пристроєм. Одним із прикладів машинозчитуваного носія є носій, що зчитується комп'ютером. Іншим прикладом машинозчитуваного носія є папір з отворами, які визначаються, як такі, які запускають різні механічні, електричні та/або логічні відгуки. Термін "машинозчитуваний носій" також включає в себе носії, які несуть в собі інформацію, поки інформація знаходиться в процесі передачі від одного місця розташування до іншого, такі як мідний дріт і/або оптоволокно, і/або атмосферу, і/або космічний простір. Можливо, бажано зберігати зміст звичайного тексту 104 в таємниці. Згодом, може бути бажано, зашифрувати звичайний текст 104 таким чином, щоб передбачалося, що передана інформація не була зрозумілою незаконному одержувачу в разі, якщо незаконний одержувач намагається переглянути та/або розшифрувати переданий зашифрований звичайний текст. Звичайний текст 104 може являти собою сукупність декількох незашифрованих інформаційних блоків, весь звичайний текст, сегмент звичайного тексту (інформацію) або будь-яку іншу частину звичайного тексту.

Процес розшифровування 106 може являти собою серію етапів, які виконуються зі звичайним текстом 104. У даному описі термін "процес" позначає серію з однієї або декількох операцій. В одному прикладі здійснення винаходу термін "процес" позначає одну або кілька інструкцій для пристрою шифрування 102 для виконання серії операцій, які можуть зберігатися на машинозчитуваному носії. В іншому випадку, процес може виконуватися обладнанням, і, таким чином, позначати обладнання (наприклад, логічні схеми), або може являти собою комбінацію з інструкцій, які зберігаються на машинозчитуваному носії, і обладнання, які призводять до виконання операцій передавальним пристроєм 102 або приймаючим пристроєм 112. Звичайний текст 104 може бути вхідними даними для процесу шифрування 106. Кроки, які входять до складу процесу шифрування 106, можуть включати в себе одну або кілька математичних операцій і/або одну або кілька інших операцій. У прикладі здійснення винаходу "процес" може також включати в себе операції або дії, які найкраще описуються, як "недетерміновані". У прикладі здійснення винаходу "процес" може включати деякі операції, які можуть виконуватися цифровою комп'ютерною програмою і деякими фізичними операціями, які є недетермінованими.

У цьому документі термін "процес" позначає і висловлює більш широке поняття, ніж "алгоритм". Формальне поняття "алгоритм" було представлено в документі Тьюринга [24] і позначає кінцевий пристрій, яке виконує кінцеве число інструкцій за допомогою обмеженої пам'яті. "Алгоритм" являє собою детермінований процес в наступному сенсі: якщо кінцевий пристрій повністю відомий, і відомі вхідні дані для пристрою, тоді можна визначити майбутню поведінку пристрою. Однак, є обладнання для квантового генератора випадкових чисел (QRNG) [25, 26] та інші приклади здійснення винаходу, які вимірюють квантові дії фотонів (або інших фізично недетермінованих процесів), фізичний процес яких є недетермінованим. Визнання недетермінованості, яке спостерігається квантовими генераторами випадкових чисел і іншими квантовими прикладами здійснення винаходу, засноване на експериментальному доведенні і багатьох роках статистичних випробувань. Більш того, квантова теорія - виведена з теореми Кохена-Шпекера і її узагальнень [27, 28, 29] - має на увазі, що вихідні дані квантового вимірювання не можуть бути заздалегідь відомі і не можуть бути згенеровані машиною Тьюринга (цифровою комп'ютерною програмою). В результаті, фізично недетермінований процес не може генеруватися алгоритмом: конкретно, послідовністю операцій, що виконуються

цифровою комп'ютерною програмою. На Фіг. 1D показаний приклад здійснення винаходу недетермінованого процесу, який виникає з квантових подій, тобто вхідних потоків фотонів.

Деякі приклади фізично недетермінованих процесів наведені нижче. У деяких прикладах здійснення винаходу, які використовують недетермінованість, може використовуватися напівпрозоре дзеркало, де фотони, які вдаряються об дзеркало, можуть приймати два або більше шляхів в просторі. В одному прикладі здійснення винаходу, якщо фотон відбивається, тоді він приймає одне бітове значення $b \in \{0, 1\}$; якщо фотон передається, тоді він приймає інше бітове значення $1 - b$. В іншому прикладі здійснення винаходу можна відібрати напрямок рух електрона, щоб згенерувати наступний недетермінований біт. У ще одному прикладі здійснення винаходу, білок, який складається з амінокислот, що заповнюють мембрану клітини або штучну мембрану, яка має дві або більше конформацій, може використовуватися для того, щоб визначити недетермінованість: відібрана конформація білка може використовуватися для того, щоб згенерувати недетерміноване значення в $\{0, \dots, n - 1\}$, де білок має n неповторних конформацій. В іншому прикладі здійснення винаходу один або кілька білків родопсина могли б використовуватися для визначення часу надходження фотонів, а різниця в часі надходження могла б використовуватися для створення недетермінованих бітів. У деяких прикладах здійснення винаходу для вибору недетермінованості може використовуватися лічильник Гейгера. У деяких прикладах здійснення винаходу недетермінована величина заснована на похибці округлення в найменш значимому біті при обчисленні в зв'язку з обмеженнями в апаратному забезпеченні. Нарешті, будь-яка з односторонніх функцій даного опису може бути заснована на випадковій події, такий як квантова подія (недетермінована), сгенероване квантовим генератором випадкових чисел з Рисунка 1D, який описується далі в розділі 6.3.

На Фіг. 1A генератори ключів 107 можуть включати в себе один або кілька генераторів ключів. Генератори ключів 107 можуть використовуватися процесом розшифровки 106, щоб допомогти отримати один або кілька ключів, які використовуються для шифрування, як мінімум, частини звичайного тексту 104. Генератори ключів 117 можуть використовуватися процесом шифрування 116, щоб допомогти отримати один або кілька ключів, які використовуються для розшифрування, як мінімум, частини шифрованого звичайного тексту 109. У прикладі здійснення винаходу один або кілька генераторів ключів 107 і генератор ключів 117 отримані з недетермінованого генератора 136 на Фіг. 1B. В іншому прикладі здійснення винаходу при використанні генератора ключів 107 дві сторони можуть використовувати один і той же процес шифрування, але все одно не передбачається, що вони зможуть розшифрувати зашифровану інформацію один одного, якщо вони не використовують однакові генератори ключів 107 в тому ж порядку в криптографічному процесі. Генератори ключів 107 можуть мати великий діапазон розмірів. Наприклад, якщо розмір генератора ключів 107 вимірюється в бітах, один або кілька генераторів ключів можуть бути 256-бітними, 512-бітними, 1000-бітними, 1024-бітними, 4096-бітними або більше. У прикладі здійснення винаходу дві сторони (Еліс і Боб) можуть призначати однакові генератори ключів 107, спочатку створюючи генератори закритих ключів зі своїх відповідних недетермінованих генераторів 136, а потім обмінюючись генераторами ключів. У прикладі здійснення винаходу апаратний пристрій, показаний на Фіг. 1D, може являти собою частину недетермінованого генератора 136.

Передавальний пристрій 102 може являти собою інформаційну машину, яка передає інформацію в перше місце розташування, програмний блок, пристрій, особі, відправнику або іншому об'єкту, або пов'язана з ними. Передавальним пристроєм 102 може бути комп'ютер, телефон, мобільний телефон, телеграф, супутник або інший тип електронного пристрою, механічний пристрій або інший вид пристрою, який передає інформацію. Передавальний пристрій 102 може включати в себе один або кілька процесорів і / або може включати в себе спеціалізоване компонування схем для роботи з інформацією. Передавальний пристрій 102 може отримувати звичайний текст 104 з іншого джерела (наприклад, перетворювача, такого як мікрофон), може виробляти весь або частину звичайного тексту 104, може реалізовувати процес шифрування 106 і/або може передавати вихідні дані іншому об'єкту. В іншому прикладі здійснення винаходу передавальний пристрій 102 приймає звичайний текст 104 з іншого джерела, тоді як процес шифрування 106 і доставка результатів процесу шифрування 106 реалізуються вручну. В іншому прикладі здійснення винаходу передавальний пристрій 102 реалізує процес шифрування 106 при введенні звичайного тексту 104 за допомогою клавіатури (наприклад) або за допомогою мікрофона мобільного телефону в передавальний пристрій 102. В іншому прикладі здійснення винаходу передавальний пристрій 102 приймає результат процесу шифрування 106 і відправляє його іншому об'єкту. В одному з прикладів здійснення винаходу передавальний пристрій 102 може генерувати нові генератори ключів 107 для інших інформаційних машин.

Передавальний пристрій 102 може реалізовувати будь-який з методів шифрування, описаних в даному описі. Процес шифрування 106 може включати в себе будь-який із способів шифрування, описаних в даному описі (наприклад, процес шифрування 106 може реалізувати будь-який приклад здійснення винаходу Н-процесу). Зашифрований звичайний текст 109

5 включає в себе, як мінімум, деякий звичайний текст 104, який шифрується процесом шифрування 106.

Канал передачі 110 являє собою шлях, який займає зашифрований звичайний текст 109, щоб дійти до місця призначення, в яке був відправлений зашифрований звичайний текст 109. Канал передачі 110 може включати в себе одну або кілька мереж. Наприклад, канал передачі

10 110 може являти собою Інтернет; наприклад, канал передачі 110 може бути бездротовим з використанням голосу по Інтернет-протоколу. Канал передачі 110 може включати в себе будь-яку комбінацію будь-якого прямого підключення, ручної доставки, голосової передачі, однієї або декількох локальних обчислювальних мереж (LAN), однієї або декількох глобальних мереж (WAN), однієї або декількох телефонних мереж, включаючи підземні траси через оптоволоконні

15 кабелі і/або одну чи кілька бездротових мереж, і/або бездротову передачу даних в атмосфері Землі або за її межами.

Приймаючий пристрій 112 може являти собою інформаційну машину, яка обробляє інформацію в місці призначення зашифрованого звичайного тексту 109. Приймальним пристроєм 112 може бути комп'ютер, телефон, телеграф, маршрутизатор, супутник або інший

20 тип електронного пристрою, механічний пристрій або інший вид пристрою, який приймає інформацію. Приймаючий пристрій 112 може включати в себе один або кілька процесорів і/або спеціалізовану компоновку схем, сконфігуровану для обробки такої інформації, як зашифрований звичайний текст 109. Приймаючий пристрій 112 може приймати зашифрований звичайний текст 109 з іншого джерела і/або відтворити (наприклад, розшифрувати) весь

25 зашифрований звичайний текст 109 або його частину. Приймаючий пристрій 112 може реалізувати будь-який зі способів шифрування, описаних в даному описі, і може розшифрувати будь-яке повідомлення, зашифроване передавальним пристроєм 102 і процесом шифрування 106.

В одному з прикладів здійснення винаходу приймаючий пристрій 112 тільки приймає зашифрований звичайний текст 109 з каналу передачі 110, тоді як процес розшифровки 106

30 реалізується вручну і/або за допомогою іншої інформаційної машини. В іншому прикладі здійснення винаходу приймаючий пристрій 112 реалізує процес розшифровки 116, який відтворює весь звичайний текст 104 або його частину, що називається розшифрований звичайний текст 114. В іншому прикладі здійснення винаходу приймаючий пристрій 112 приймає

35 зашифрований звичайний текст 109 з каналу передачі 110 і відтворює увесь розшифрований текст 114 або його частину за допомогою процесу розшифровки 116.

Процес розшифровки 116 може зберігати будь-який з процесів розшифровки інформації, описаних в цьому описі. Процес розшифровки 116 може включати в себе будь-який із способів розшифровки, описаних в даному описі (наприклад, процес розшифровки 106 може реалізувати

40 будь-який з методів для розшифровки будь-якого прикладу здійснення винаходу Н-процесу).

Приймаючий пристрій 112 може бути ідентичним передавальному пристрою 102. У прикладі здійснення винаходу, в якому приймаючий пристрій і передавальний пристрій однакові, як передавальний, так і приймаючий пристрій, кожне включає в себе звичайний текст 104 (незашифровану інформацію), процес шифрування 106, генератори ключів 107 (які можуть

45 включати в себе одностороннє хешування), зашифрований звичайний текст (зашифрована інформація) 109, процеси дешифрування 116, розшифрований звичайний текст 114 і генератори ключів 117 (які можуть включати в себе одностороннє хешування), і вони обидва можуть реалізовувати будь-які процеси шифрування, процес розшифровки і методи обміну генераторами ключів, представлені в даному описі.

Наприклад, приймаючий пристрій 112 може приймати звичайний текст 104 з іншого джерела, формувати весь звичайний текст або його частину 104 і/або реалізовувати процес шифрування 106. Подібно передавальному пристрою, приймаючий пристрій 112 може

50 створювати генератори ключів 117. Приймаючий пристрій 112 може передавати результат процесу розшифровки 116 по каналу передачі 110 в інший об'єкт і/або приймати зашифрований звичайний текст 109 (по каналу передачі 110) від іншого об'єкту. Приймаючий пристрій 112 може надавати зашифрований звичайний текст 109 для використання в якості вхідних даних для процесу розшифровки 116.

6.2. ЛАВИННИЙ ЕФЕКТ І ОДНОСТОРОННІ ФУНКЦІЇ

Одностороння функція 107 на Фіг. 1А і одностороння функція 126 на Фіг. 1В можуть

60 включати в себе одну або кілька односторонніх функцій. Одностороння функція Ф має таку

властивість, що маючи вихідне значення z , вона є складною для обчислення, щоб визначити інформаційний елемент m_z такий, щоб $\Phi(m_z) = z$. Іншими словами, одностороння функція Φ являє собою функцію, яку можна легко обчислити, але її зворотна величина Φ^{-1} є складною для обчислення. Обчислення, для якого потрібно 10^{101} кроків обчислення вважається таким, що має обчислювальне нерозкриття 10^{101} . Більше інформації представлено в розділі про обчислювальне нерозкриття. В одному з прикладів здійснення винаходу існує кількість часу T , протягом якого зашифрована інформація повинна залишатися секретною. Якщо у зашифрованої інформації немає економічної цінності або стратегічної цінності після часу T , то "обчислювальне нерозкриття" означає, що число обчислювальних кроків, які необхідні для всієї обчислювальної потужності світу, займе більше часу для обчислення, ніж час T . Припустимо, $C(t)$ позначає всю обчислювальну потужність світу при часі t , вираженому в роках.

Розглянемо банківську онлайн-транзакцію, яка зашифровує відомості по даній транзакції. В такому випадку, в більшості прикладів здійснення винаходу кількість обчислювальних кроків, які можуть бути зроблені всіма комп'ютерами світу протягом наступних 30 років у багатьох прикладах здійснення винаходу ймовірно є такими, що обчислювально не розкриваються, оскільки такий конкретний банківський рахунок, ймовірно, не буде більше існувати через 30 років, або буде мати інший інтерфейс для аутентифікації.

Щоб зробити числа більш конкретними, китайський суперкомп'ютер 2013 року, який побив світовий рекорд швидкості обчислення, виробляє приблизно 33 000 трильйонів обчислень в секунду [30]. Якщо $T = 1$ один рік, і ми можемо припустити, що існує максимум 1 мільярд цих суперкомп'ютерів. (Це можна логічно вивести з економічних міркувань, виходячи з дуже заниженої ціни в 1 мільйон доларів за кожен суперкомп'ютер. Тоді ці 1 мільярд суперкомп'ютерів коштували б 1 000 трильйонів доларів). Таким чином, $C(2014 \text{ року}) \times 1 \text{ рік}$ менше, ніж $10^9 \times 33 \times 10^{15} \times 3600 \times 24 \times 365 = 1.04 \times 10^{33}$ обчислювальних кроків. Для отримання деякого сприйняття щодо криптографії, криптографія на основі 25519 еліптичних кривих [31] припускає складність [32] в 2^{128} обчислювальних кроків. Також, $2^{128} > 10^{38}$, тому в рамках цього виміру обчислювального нерозкриття, криптографія на основі 25519 еліптичних кривих має обчислювальне нерозкриття, як мінімум, 10^{38} обчислювальних кроків.

Як зазначалось вище, в деяких прикладах здійснення винаходу і застосуваннях, обчислювальне нерозкриття може бути виміряне в рамках того, скільки коштує зашифрована інформація в економічній вартості, і яка нинішня вартість обчислювальної потужності, яка необхідна для розшифровки такої зашифрованої інформації. В інших прикладах здійснення винаходу економічне обчислювальне нерозкриття може бути марним. Наприклад, припустимо, сім'я хоче зберегти місце розташування своєї дитини невідомим для жорстоких викрадачів. Припустимо, що $T = 100$ років, оскільки це приблизно в два рази більше, ніж їх передбачуваний термін життя. Тоді $100 \text{ років} \times C(2064)$ є найкращим виміром обчислювального нерозкриття для даного використання. Іншими словами, для ключових способів застосування, які виходять за межі економічної вартості, необхідно прагнути до гарної оцінки світової обчислювальної потужності.

Односторонні функції, які демонструють повноту і належний лавинний ефект або строгий лавинний критерій [33], являються собою бажані приклади здійснення винаходу: ці властивості є сприятливими для поновлення генератора ключів. На Фіг. 1С показаний лавинний ефект через 16 циклів SHA-1 на перші 46 бітів вихідних даних SHA-1. Розмір профілю SHA-1 становить 160 біт (тобто довжина його вихідних даних). Тільки один біт був переключений з b на $1-b$ у вхідних даних. Перемкнутий біт у вхідних даних відображається невеликим білим прямокутником біля верхньої частини Фіг. 1С. Білі квадрати показують біти, які переключилися з 0 на 1 або з 1 на 0 в результаті перемикання одного біта вхідних даних. На 16ом циклі є більше білих бітів, ніж чорних бітів. Строгий лавинний критерій говорить, що існує 50-процентна вірогідність того, що з'явиться перемикання бітів. Передбачається, що 80 циклів SHA-1 забезпечують достатнє перетворення звичайного тексту для порушення його статистичної структури.

Визначення повноти і належний лавинний ефект цитуються безпосередньо з [33]:

Якщо криптографічна трансформація є повною, тоді кожен біт зашифрованого тексту повинен залежати від усіх бітів звичайного тексту. Таким чином, якби було можливо знайти найпростіший булевий вираз для кожного біта зашифрованого тексту в перекладі на біти звичайного тексту, кожне з цих виразів мало б містити всі біти звичайного тексту, якби функція була повною. В іншому випадку, якщо є, як мінімум, одна пара n -бітових векторів звичайного тексту X і X_i , які відрізняються тільки бітом i , а $f(X)$ і $f(X_i)$ відрізняються, як мінімум, бітом j для всього $\{(i, j): 1 \leq i, j \leq n\}$, то функція f має бути повною.

Для цієї трансформації, щоб продемонструвати лавинний ефект, середнє значення половини вихідних бітів повинно змінюватися в тих випадках, коли буде доповнюватися один

вихідний біт. Для того щоб визначити, чи задовольняє цій вимозі $m \times n$ (m вхідних бітів і n вихідних бітів) функція f , вектори звичайного тексту 2^m повинні розділятися на 2^{m-1} пар, X і X_j , такі як X і X_j відрізняються тільки бітом i . Тоді необхідно розрахувати суми $V_i = f(X) \oplus f(X_i)$ суворої диз'юнкції 2^{m-1} . Ці суми суворої диз'юнкції будуть вказуватися в якості лавинних векторів, кожен з яких містить n біт, або лавинні змінні.

Якщо процедура повторюється для всіх i , таких як $1 \leq i \leq m$, а половина лавинних змінних дорівнює 1 для кожної i , то функція f має хороший лавинний ефект. Звичайно цим методом можна слідувати, тільки якщо m є вельми маленької величиною; в іншому випадку кількість векторів звичайного тексту стає занадто великою. Якщо ситуація саме така, тоді краще, що можна зробити - це взяти випадковий зразок векторів звичайного тексту X і для кожного значення i розрахувати всі лавинні вектори V_i . Якщо приблизно половина одержаних лавинних змінних дорівнює 1 для величин i , тоді ми можемо зробити висновок, що функція має хороший лавинний ефект.

Функція хешування, також позначена, як Φ , являє собою функцію, яка приймає в якості свого вхідного аргументу достатньо довгий рядок біт (або байтів) і створює вихідну інформацію з фіксованим розміром. Інформація на виході зазвичай називається профілем повідомлення або цифровим відбитком пальця. Іншими словами, функція хешування перетворює змінну довжину m вхідної інформації у вихідні дані з фіксованим розміром, $\Phi(m)$, які представляють собою профіль повідомлення або профіль інформації. Стандартний діапазон розмірів вихідних даних становить від 160 до 512 біт, але також може бути і більше. Ідеальна функція хешування являє собою функцію Φ , вихідні дані якої рівномірно розподіляються наступним чином: Припустімо, що вихідний розмір Φ становить n біт. Якщо повідомлення m вибирається випадковим чином, тоді для кожного з 2^n можливих виходів z , ймовірність того, що $\Phi(m) = z$ становить 2^{-n} . У прикладі здійснення винаходу, функції хешування, які використовуються, є односторонніми.

Хороша одностороння функція хешування також є стійкою до накладення. Накладення виникає, коли два різних елементи інформації перетворюються за допомогою односторонньої функції хешування Φ в такий самий профіль. Стійкість до накладення означає, що противник не може зробити обчислення, щоб виявити накладення: точніше, неможливо зробити обчислення, щоб виявити два різних елементи інформації m_1, m_2 , де m_1 не дорівнює m_2 , і так щоб $\Phi(m_1) = \Phi(m_2)$. Може використовуватися ряд функцій одностороннього хешування. SHA-512 являє собою односторонню функцію хешування, розроблену АНБ і стандартизовану NIST [23]. Розмір профілю повідомлення SHA-512 становить 512 біт. Іншими альтернативними функціями хешування є той тип, який відповідає стандарту SHA-384, який створює розмір профілю повідомлення 384 біт. SHA-1 має розмір профілю повідомлення 160 біт. Прикладом здійснення винаходу односторонньої функції хешування є Кесак [34]. Прикладом здійснення винаходу односторонньої функції хешування є BLAKE [35]. Прикладом здійснення винаходу односторонньої функції хешування є Grøstl [36]. Прикладом здійснення винаходу односторонньої функції хешування є JH [37]. Ще одним прикладом здійснення винаходу односторонньої функції хешування є Skein [38]. В інших прикладах здійснення винаходу, замість односторонньої функції хешування можуть використовуватися інші односторонні функції. Наприклад, еліптична крива по полю з кінцевим числом елементів може використовуватися в якості односторонньої функції. Для цих альтернативних односторонніх функцій повнота і хороший лавинний ефект є сприятливими властивостями, які демонструють ці функції. Суворий лавинний критерій також є сприятливою властивістю для цих альтернативних односторонніх функцій.

В одному з прикладів здійснення винаходу, одностороння функція 126 на Фіг. 1В може бути реалізована, як виконувані інструкції пристрою у внутрішніх інструкціях пристрою мікропроцесора. У ще одному прикладі здійснення винаходу, одностороння функція 126 на Фіг. 1В може бути реалізована в такому обладнанні, як програмована користувачем матриця логічних елементів (FPGA), яка може забезпечити надійну зону для виконання розрахунків. Надійна зона показана на Фіг. 2В. У деяких прикладах здійснення винаходу безпечна зона знаходиться всередині пластикової картки з мікропроцесором.

6.3. ОБЛАДНАННЯ ТА ІНФРАСТРУКТУРА ДЛЯ КРИПТОГРАФІЇ

На Фіг. 1D показаний приклад здійснення винаходу недетермінованого процесу, який визначає час надходження фотонів. Час надходження фотонів вважається квантовою подією. На Фіг. 1D наводиться приклад здійснення винаходу недетермінованого генератору 136. $h\nu$ означає енергію надходжуючого фотона, де h - це постійна Планка, а ν - це частота. В одному з прикладів здійснення винаходу можна порівнювати 3 послідовних часи прибуття $t_1 < t_2 < t_3$ трьох послідовних фотонів. Якщо $t_2 - t_1 > t_3 - t_2$, то недетермінований генератор 142 формує біт 1.

Якщо $t_2 - t_1 < t_3 - t_2$, то недетермінований генератор 142 формує біт 0. Якщо $t_2 - t_1 = t_3 - t_2$, то не формується недетермінована інформація, і цей недетермінований процес вибирає ще три часи прибуття.

На фіг. 2A інформаційна система 200 демонструє деякі з варіацій реалізації інформаційної системи 100. Передавальний пристрій 202 являє собою один із прикладів здійснення винаходу передавального пристрою 101. Передавальний пристрій 202 може бути захищеним запам'ятовуючим пристроєм USB, як показано на Фіг. 3A. Передавальний пристрій 202 може являти собою аутентифікаційний маркер, як показано на Фіг. 3B. Приклад здійснення передавального пристрою 202 в формі мобільного телефону, як показано на кресленні, передавального пристрою 202 або передавального пристрою 400 може сполучатися бездротовим способом з комп'ютером 204. В одному з прикладів здійснення винаходу комп'ютер 204 може представлятися собою станцію виклику для прийому зашифрованого звичайного тексту 109 від передавального пристрою 400. Користувач може використовувати систему вводу 254 і систему виводу 252 передавального пристрою (мобільний телефон) 400 для передачі зашифрованих голосових даних в приймальний пристрій, яке є мобільним телефоном. В одному з прикладів здійснення винаходу система вводу 254 на Фіг. 2B включає в себе мікрофон, який інтегрований з передавальним пристроєм (мобільним телефоном) 400. В одному з прикладів здійснення винаходу система виводу 252 на Фіг. 2B включає в себе динамік, який інтегрований з передавальним пристроєм (мобільним телефоном) 400. В іншому прикладі здійснення винаходу передавальний пристрій 202 має можливість підключення та обміну даними з комп'ютером 204 або з іншими системами за допомогою комп'ютера 204.

Комп'ютер 204 підключається до системи 210 і підключається через мережу 212 до системи 214, системи 216 і системи 218, яка підключена до системи 220. Мережа 212 може бути однією з або комбінацією з однієї або кількох локальних обчислювальних мереж (LAN), глобальних мереж (WAN), бездротових мереж, телефонних мереж і/або інших мереж. Система 218 може безпосередньо підключатися до системи 220 або підключатися по LAN до системи 220. Мережа 212 і система 214, 216, 218 і 220 можуть являти собою Інтернет-сервери або вузли, які маршрутизують зашифрований звичайний текст (голосові дані), отримані від передавального пристрою 400, показаного на Фіг. 4. На Фіг. 2A система 214, 216, 218 і система 220, а також мережа 220 можуть спільно виступати як канал передавання 110 для зашифрованого звичайного тексту 109. В одному з прикладів здійснення винаходу система 214, 216, 218 і система 220, а також мережа 212 можуть виконувати Інтернет-протокол спільно виступати в якості каналу передавання 110 для зашифрованого звичайного тексту 109. В одному з прикладів здійснення винаходу зашифрований звичайний текст 102 може являти собою голосові дані. В одному з прикладів здійснення винаходу зашифрований звичайний текст 109 може являти собою дані маршрутизації. В одному з прикладів здійснення винаходу зашифрований звичайний текст 109 може являти собою повідомлення електронної пошти. В одному з прикладів здійснення винаходу зашифрований звичайний текст 109 може являти собою текстові дані, відправлені з передавального пристрою 400.

На фіг. 1B процес дешифрування 122 може бути реалізований за допомогою будь-якої з, будь-якої, або будь-якої комбінації будь-якої системи 210, мережі 212, системи 214, системи 216, системи 218 та/або системи 220. Наприклад, інформація маршрутизації каналу передачі 110 може бути зашифрована за допомогою процесу шифрування 122, який виконується в системному комп'ютері 210, мережевих комп'ютерах 212, системному комп'ютері 214, системному комп'ютері 216, системному комп'ютері 218 і/або системному комп'ютері 220. Процес шифрування 106 може виконуватися в передавальному пристрої 400, а процес розшифровки 116 може виконуватися в прийальному пристрої 400 на Фіг. 4.

В одному з прикладів здійснення винаходу Н-процес NADO виконується в безпечній зоні системи процесора 258 Фіг. 2B. В одному з прикладів здійснення винаходу в системі процесора 258 може використовуватися спеціалізоване обладнання для прискорення обчислення односторонніх функцій 126 на Фіг. 1B, який використовується в Н-процесі. В одному з прикладів здійснення винаходу дане спеціалізоване апаратне забезпечення в системі процесора 258 може бути реалізовано у вигляді ASIC (замовна спеціалізована мікросхема), яка обчислює SHA-1 і / або SHA-512, і / або Кессак, і / або BLAKE, і / або JH, і / або Skein. Мікросхема ASIC може збільшувати швидкість виконання обчислень Н-процесу. В одному з прикладів здійснення винаходу система вводу даних 256 приймає голосові дані і відправляє їх у систему процесора 258, де голосові дані розшифровуються. Система вихідних даних 252 відправляє зашифровані голосові дані 109 в телекомунікаційну мережу 212. В одному з прикладів здійснення винаходу система пам'яті 256 зберігає генератори ключів 124 і обробляє інструкції по Н-блоковому шифруванню 130.

Стан відноситься до конкретного значення або набору значень з будь-якого набору однієї або декількох внутрішніх змінних, при якому спосіб, яким виконуються операції порушується при виборі значення або набору значень, які складають стан. Генератор станів виконує одну або кілька операцій для поновлення стану.

В одному з прикладів здійснення винаходу інструкції Н-процесу 130 виконуються в надійній зоні системи процесора 258, яка знаходиться всередині автономного USB-диска, представленого на Фіг. 3А. В одному з прикладів здійснення винаходу процес шифрування 122 зашифровує дані, які зберігаються на USB-диску, для захисту конфіденційності даних.

В одному з прикладів здійснення винаходу інструкції Н-процесу 130 шифрують голосову бесіду в надійній зоні системи процесора 258, яка знаходиться в мобільному телефоні 400, який являє собою приклад здійснення винаходу передавального пристрою 102 і приймального пристрою 112.

В одному з прикладів здійснення винаходу на фіг. 1В інструкції Н-процесу 130 виконуються в надійній зоні кожної системи процесора 258 (Фіг. 2В), яка міститься в системних комп'ютерах 210, 214, 216, 218 і 220 і всередині мережі 212, показаної на Фіг. 2А.

6.4. ШИФРУВАННЯ ЗА ДОПОМОГОЮ ДИНАМІЧНИХ КЛЮЧІВ

Створення і використання динамічних ключів залежить від узгодження Еліс і Бобом елемента наступного генератора ключів $G(i+1)$ від попереднього генератора ключів $G(i)$, і саме так побудована послідовність генератора ключів $G(0), G(1), \dots, G(i), G(i+1), \dots$. Незліченна кількість послідовностей генераторів ключів є необчислювальними за Тьюрингом; в даному документі наші приклади здійснення винаходу описують обчислювані за Тьюрингом послідовності генераторів ключів, оскільки обчислюваність допомагає спростити координацію оновлення генератора ключів між Еліс і Бобом.

Наші односторонні функції прообразу мають принципово нове використання при типовому застосуванні аутентифікації повідомлень, яке виконується односторонніми функціями хешування у відомому рівні техніки. У криптографічному методі 1 Ф являє собою односторонню функцію прообразу з розміром профілю q . Послідовність генераторів ключів $G: N \times \{0,1\}^n$ задовольняє $q < n$. Символ $G_{i,j}$ є j -тим бітом i -го генератора ключів $G(i)$. Перший етап використовує підписаний обмін генераторами ключів, коли в деяких прикладах здійснення винаходу приватні секрети Еліс і Боба створюються з недетермінованого генератора 172, як показано на Фіг. 1І. У деяких прикладах здійснення винаходу інструкції з обміну генераторами ключів 170 визначають перший генератор ключів, який створюється з недетермінованого процесу. Обмін генераторами ключів обговорюється далі в розділах 6.9, 6.10 і 6.11.

Криптографічний метод 1. Оновлення генератора ключів за допомогою односторонньої функції прообразу

Еліс і Боб здійснюють підписаний обмін генератором ключів, щоб створити спільно використовуваний генератор ключів

$$G(0) = G_{0,0} \dots G_{0,n-1}$$

Задати початкове значення $i = 0$ тоді як (Еліс і Боб вимагають наступний генератор ключів $G(i+1)$)

{
Задати, що $(G_{i+1,0} \ G_{i+1,1} \dots G_{i+1,q-1}) = \Phi(G_{i,0} \ G_{i,1} \dots G_{i,q-1})$
Задати, що $G_{i+1,j} = G_{i,j}$ для кожного j задовольняє $q \leq j \leq n-1$
Збільшити i

}
Метод 1 призначений для створення багатовимірної орбіти на перших q бітах $G_{i,0} \ G_{i,1} \dots G_{i,q-1}$, під впливом лавинних властивостей [33] функції Φ ; для збереження тих, що залишилися $n - q$ біт постійними для всіх i ; і для забезпечення того, щоб ніяка інформація з останніх $n - q$ не впливала на орбіту перших q біт.

На Фіг. 1Е показаний етап оновлення генератора ключів: Задати, що $(G_{i+1,0} \ G_{i+1,1} \dots G_{i+1,q-1}) = \Phi(G_{i,0} \ G_{i,1} \dots G_{i,q-1})$. На Фіг. 1Е перші q біт $(G_{i,0} \ G_{i,1} \dots G_{i,q-1})$ i -го генератора ключів виражаються, як $b_1 \ b_2 \dots b_q$. На Фіг. 1Е останні $n-q$ біт i -го генератора ключів $(G_{i,q} \ G_{i,q+1} \dots G_{i,n-1})$ виражені, як $a_1 \ a_2 \dots a_{n-q-1}$. На Фіг. 1Е вираз $(c_1 \ c_2 \dots c_q) = \Phi(b_1 \ b_2 \dots b_q)$ представляє перші q біт $(G_{i+1,0} \ G_{i+1,1} \dots G_{i+1,q-1})$ $i+1$ -го генератора ключів. Далі, на Фіг. 1Е біти $a_1 \ a_2 \dots a_{n-q-1}$ представляють останні $n - q$ біт $(G_{i+1,q} \ G_{i+1,q+1} \dots G_{i+1,n-1})$, оскільки ці біти залишаються без змін. В одному з прикладів здійснення винаходу $n = 768$, а $q = 512$. В іншому прикладі здійснення винаходу $n = 1024$, а $q = 512$. У ще одному прикладі здійснення винаходу $n = 20000$, а $q = 1000$.

У разі звичайного використання противник - Єва - ніколи не отримає доступу до жодних бітів генератора ключів Еліс $G(i)$. Це аналогічно тому, що Єва не має доступу до будь-яких бітів

статичного ключа Еліс, який використовується в реалізаціях симетричної криптографії у відомому рівні техніки.

В іншому прикладі здійснення винаходу одностороння функція Φ може застосовуватися до останніх q бітів, ті, що залишилися $n-q$ біт залишаються незмінними для всіх i . Даний приклад здійснення винаходу показано на Фіг. 1F, де останні q біт генератора ключів оновлюються, як $(c_1 c_2 \dots c_q) = \Phi(b_1 b_2 \dots b_q)$. В інших прикладах здійснення винаходу постійні біти $a_{k1} a_{k2} \dots a_{kn-q}$ можуть чергуватися між бітами генератора ключів, які оновлюються, як $(c_{j1} \dots c_{jq}) = \Phi(b_{j1} b_{j2} \dots b_{jq})$, де набір місць розташування бітів $\{k_1, k_2, \dots, k_{n-q}\}$ для постійних бітів відділений від набору місць розташування бітів $\{j_1, j_2, \dots, j_q\}$, що вказує на біти $(b_{j1} b_{j2} \dots b_{jq})_{jq}$, які оновлюються. Тобто, $\{k_1, k_2, \dots, k_{n-q}\} \cap \{j_1, j_2, \dots, j_q\} = \emptyset$.

У деяких прикладах здійснення винаходу різні односторонні функції можуть застосовуватися на різних етапах оновлення генератора ключів. Наприклад, для обчислень першого генератора ключів $\Gamma(1)$ з генератора ключів $\Gamma(0)$ може використовуватися SHA-512; SHA-384 може використовуватися для обчислення другого генератора ключів $\Gamma(2)$ з генератора ключів $\Gamma(1)$; Кессак може використовуватися для обчислення третього генератора ключів $\Gamma(3)$ з генератора ключів $\Gamma(2)$; і т. д. У цих прикладах здійснення винаходу є інструкції для генераторів ключів 162 (фіг. 1I), які викликають різні інструкції односторонніх функцій 164 в залежності від j -го генератора ключів. Інструкції односторонніх функцій 164 можуть реалізовувати SHA-384, Кессак, SHA-512 та інші односторонні функції.

Метод 2 отримує динамічний ключ K_i для блочного шифру A з i -го генератора ключів $\Gamma(i)$ послідовності генераторів ключів, як показано на Фіг. 1G і 1H. Символ Ψ позначає односторонню функцію, вихідний розмір якої становить r біт, де $k \leq r$. Як показано на Фіг. 1G і 1H, Ψ застосовується до конкатенації динамічної частини $\Gamma_{i,0} \Gamma_{i,1} \dots \Gamma_{i,q-1}$ із $\Gamma(i)$ і постійної частини $\Gamma_{i,q} \dots \Gamma_{i,n-1}$ для того, щоб отримати різний ключ K_i для кожного блоку, який шифрується. На Фіг. 1G перші q біт $(b_1 b_2 \dots b_q)$ є частиною генератора ключів, які змінюються після кожного етапу оновлення генератора ключів, показаного на Фіг. 1E; на Фіг. 1G, останні $n - q$ біт $(a_1 a_2 \dots a_{n-q})$ залишаються незмінними. На Фіг. 1H перші $n - q$ біт $(a_1 a_2 \dots a_{n-q})$ залишаються незмінними; на Фіг. 1H останні q біт $(b_1 b_2 \dots b_q)$ є частиною генератора ключів, які змінюються після кожного етапу оновлення генератора ключів, показаного на Фіг. 1F.

У деяких прикладах здійснення винаходу Ψ є відмінною від Φ односторонньою функцією. Наприклад, в деяких прикладах здійснення винаходу Ψ може бути реалізована за допомогою Кессак, а Φ може бути реалізована за допомогою SHA-512. У деяких прикладах здійснення винаходу Ψ може використовуватися для отримання першого динамічного ключа, а інша одностороння функція Ψ' може використовуватися для отримання другого динамічного ключа, і т.д.

Вираз $E_A(M, K)$ являє собою блоковий шифр A , що зашифровує блок звичайного тексту M за допомогою ключа K , а $D_A(C, K)$ являє собою блоковий шифр A , що розшифровує шифрований текст C за допомогою ключа K . Розмір ключа $|K|$ блочного шифру становить k біт і задовольняє $k \leq r$. Визначимо проєкційну карту $\text{пк} : \{0, 1\}^r \rightarrow \{0, 1\}^k$, где $\text{пк}(x_1 x_2 \dots x_r) = (x_1 x_2 \dots x_k)$.

В інших частинах опису Н-процес буде вказаний в деяких прикладах здійснення винаходу, як такий, що реалізується за допомогою блочного шифру, який використовує динамічні ключі, отримані з оновлення генератора ключів. У зв'язку з цим, криптографічні методи 1, 2, 3, 4, 5 можуть реалізовувати Н-процес.

Криптографічний метод 2.

Блоковий шифр A виконує шифрування за допомогою Динамічних Ключів, отриманих з Генератора Ключів

Еліс обчислює використовуваний спільно генератор $\Gamma(0)$ за допомогою 1-го етапу методу 1.

Задати початкове значення $i = 0$ тоді як (додатковий звичайний текст M_i , щоб Еліс його зашифрувала)

```
{
    Отримати динамічний ключ  $K_i = \text{пк} \circ \Psi(\Gamma_{i,0} \Gamma_{i,1} \dots \Gamma_{i,n-1})$ 
    Обчислити  $C_i = E_A(M_i, K_i)$ , який зашифровує звичайний текст  $M_i$  за допомогою ключа  $K_i$ 
    Метод 1 обчислює елемент генератора ключів  $\Gamma(i+1)$  на підставі  $\Gamma(i)$ 
    Збільшити  $i$ 
}
```

Криптографічний метод 3.

Блоковий шифр A виконує розшифровку за допомогою Динамічних Ключів, отриманих з Генератора Ключів

Боб обчислює використовуваний спільно генератор $\Gamma(0)$ за допомогою 1-го етапу методу 1.

Задати початкове значення $i = 0$

тоді як (додатковий шифрований текст C_i , щоб Боб його розшифрував)

{

5 Отримати динамічний ключ $K_i = \text{пк} \circ \Psi(\Gamma_i, 0 \dots \Gamma_i, n-1)$
 Обчислити $M_i = D_A(C_i, K_i)$, яке розшифровує зашифрований текст C_i за допомогою ключа

K_i

 Метод 1 обчислює елемент генератора ключів $\Gamma(i+1)$ на підставі $\Gamma(i)$

 Збільшити i

10 }

Реалізація стандартного Serpent являє собою 16-байтний блоковий шифр з 256-бітовим ключем [39]. Зразок оновлення генератора ключів та отримання динамічного ключа для поліпшеного Serpent описується нижче і показаний на Фіг. 5В. "Фотони є ключами" представляє собою 16-байтний блок звичайного тексту, який з'єднується разом 4 рази для створення 64-байтного звичайного тексту. Іншими словами, $V_1 = V_2 = V_3 = V_4 = \text{"Фотони є ключами"}$.

15 У наведеному нижче описі кожен байт (8 біт) виражений, як число між 0 і 255 включно. Як

показано на Фіг. 5В, 16-байтний блок V_1 звичайного тексту "Фотони є ключами" – це

80 104 111 116 111 110 115 32 97 114 101 32 107 101 121 115

Генератор ключів $\Gamma(1)$ - це 768 біт (96 байт) і показаний нижче.

20 112 132 168 213 84 252 132 50 143 235 140 71 123 248 243 160 240 248 237 200 113 43

65 95 208 97 175 125 184 234 154 227 130 187 104 4 131 204 239 92 44 187 34 166 71

146 186 241 108 149 70 166 66 35 108 195 13 235 58 218 85 229 180 66 109 55 178 123

110 135 57 238 177 21 29 225 222 84 215 155 21 179 210 201 122 202 210 208 51 104

213 58 247 238 139 116

25 Перший 256-бітний ключ K_1 , отриманий з генератора ключів $\Gamma(1)$ - це 32 23 248 49 234 86

223 193 83 37 87 247 191 22 112 130 34 177 54 67 56 186 154 188 149 130 23 146 220 118 192 55

Після шифрування перший 16-байтний блок звичайного тексту $V_1 = \text{"Фотони є ключі"}$ з поліпшеним Serpent і динамічним ключем K_1 , зашифрований текст - 33 175 244 28 210 147 63 101 221 74 197 89 195 30 31 228, який виражений, як $\Delta(V_1, \Gamma(1))$ на Фіг. 5В.

30 Генератор ключів $\Gamma(2)$ це 768 біт (96 байт) і показаний нижче.

219 14 199 128 227 62 241 230 111 13 179 127 82 52 211 235 216 220 52 233 191 255 22

121 103 165 109 90 168 10 36 23 172 246 97 184 183 134 6 195 84 171 123 50 184 60

112 217 7 249 224 23 186 238 174 199 235 214 22 152 211 212 186 202 240 109 55

178 123 110 135 57 238 177 21 29 225 222 84 215 155 21 179 210 201 122 202 210

35 208 51 104 213 58 247 238 139 116

Другий 256-бітний ключ K_2 , отриманий з генератора ключів $\Gamma(2)$ - це 86 35 129 230 137 79 46

48 202 130 242 209 127 25 84 159 185 250 154 249 12 245 176 61 12 242 200 226 63 90 62 44

Після шифрування другий 16-байтний блок звичайного тексту "Фотони є ключі" з покращеним Serpent і ключем K_2 , зашифрований текст - 79 101 31 159 181 228 83 121 166 170 215 94 99 67 100 139, який виражений, як $\Delta(V_2, \Gamma(2))$ на Фіг. 5В.

40 Генератор ключів $\Gamma(3)$ це 768 біт (96 байт) і показаний нижче.

106 83 207 235 94 38 238 182 252 52 145 130 208 170 9 222 90 70 48 182 140 87 211

89 241 135 27 217 27 4 83 65 122 137 153 188 253 116 162 45 70 34 57 162 77 45

116 126 190 163 194 142 206 195 184 102 154 112 164 53 38 215 50 187 109 55 178

45 123 110 135 57 238 177 21 29 225 222 84 215 155 21 179 210 201 122 202 210 208 51

104 213 58 247 238 139 116

Третій 256-бітний ключ K_3 , отриманий з генератора ключів $\Gamma(3)$ - це 137 141 95 102 34 68 172

9 169 183 22 154 200 144 84 232 251 87 33 62 155 72 214 82 81 119 46 198 52 253 120 133

Після шифрування третій 16-байтний блок звичайного тексту "Фотони є ключі" з покращеним Serpent і ключем K_3 , зашифрований текст - 138 83 40 138 141 153 198 180 164 108 233 135 99 130 205 34, який виражений, як $\Delta(V_3, \Gamma(3))$ на Фіг. 5В.

50 Генератор ключів $\Gamma(4)$ це 768 біт (96 байт) і показаний нижче.

22 228 65 144 60 200 76 27 17 148 227 251 74 182 41 167 6 215 249 33 9 219 36 170

139 106 189 109 42 190 115 21 220 162 232 214 66 167 48 226 230 77 73 198 147 180

55 29 41 103 238 224 24 103 225 181 252 103 103 194 164 76 132 242 207 109 55 178

123 110 135 57 238 177 21 29 225 222 84 215 155 21 179 210 201 122 202 210 208 51

104 213 58 247 238 139 116

Четвертий 256-бітний ключ K_4 , отриманий з генератора ключів $\Gamma(4)$ - це 184 244 102 78 50

249 102 189 46 27 147 31 37 96 37 36 50 13 62 209 109 30 126 93 248 239 161 157 195 223 108 48

60 Після шифрування четвертий 16-байтний блок звичайного тексту "Фотони є ключі" з

покращеним Serpent і ключем K4, зашифрований текст - 248 255 208 238 140 14 26 6 121 1 52 78 22 48 168 112, який виражений, як $\Delta(B_4, \Gamma(4))$ на Фіг. 5В.

У деяких прикладах здійснення винаходу оновлення генератора ключів Γ виникає після кожного шифрування блоку: оновлення відбувається після блоків $B_2, B_4, B_6 \dots$, але не після блоків $B_1, B_3, B_5 \dots$. В інших прикладах здійснення винаходу оновлення генератора ключів виникає тільки після блоків $B_1, B_3, B_5 \dots$, але не після блоків $B_2, B_4, B_6 \dots$. У деяких прикладах здійснення винаходу оновлення генератора ключів Γ виникає тільки після четвертих блоків $B_4, B_8, B_{12} \dots$ шифрування. В інших прикладах здійснення винаходу оновлення генератора ключів виконується аперіодично; наприклад, оновлення генератора ключів відбувається тільки після блоків $B_2, B_3, B_5, B_7, B_{11}, B_{13}, B_{19}$, і так далі.

Використання генератора ключів, який оновлюється в методах 2 і 3, не слід плутати з існуючими режимами роботи блочного шифру, таким як CBC або CTR. По-перше, кожен з цих режимів все ще спирається на статичний ключ. Навіть $K_i = E_A(\text{випадкове число} \parallel i, K)$ і i -ий блок зашифрованого тексту представляє собою $C_i = M_i \oplus K$, спирається на статичний ключ K . По-друге, оновлення генератора ключів використовує значення n для генератора ключів, який може бути істотно більше, ніж розмір блочного і статичного ключа. Тобто, звичайно $n \gg |M_i|$ і $n \gg \kappa$, де \gg означає "набагато більше, ніж". У деяких прикладах здійснення винаходу $n = 1024$, тоді як розмір ключа $\kappa = 128$, а розмір блоку $|M_i| = 128$; це приклад, де $n \gg \kappa$. Як пояснюється в розділі 6.7, періодичність орбіти динамічних ключів, створених генератором ключів, може бути значно більше, ніж 2^κ .

Кожен з цих режимів встановлює верхню межу значення для суми збільшення ентропії, виходячи з розмірів блоку або розміру ключа. У разі ECB не виникає збільшення ентропії. У разі CBC збільшення ентропії обмежене зверху розміром області повідомлення. У разі CTR випадкове число, яке з'єднується з лічильником i , обмежене зверху розміром області повідомлення, а отримана орбіта ключа обмежена зверху розміром області повідомлення. Оскільки n може бути значно більше, ніж розмір ключа або блоку, більш значне збільшення ентропії може статися при оновленні генератора ключів.

Більш того, ніщо не перешкоджає комбінації оновлення генератора ключів з режимом CBC або режимом CTR. В альтернативних прикладах здійснення винаходу криптографічний режим, такий як режим зчеплення блоків шифрованого тексту (CBC), може додаватися до криптографічних методів 2 і 3. Методи 4 і 5 демонструють оновлення генератора ключів в комбінації з режимом CBC.

Криптографічний метод 4. Блочний шифр А виконує шифрування за допомогою динамічних ключів і режиму CBC

Еліс обчислює секрети $\Gamma(0)$, C-1 за допомогою 1-го етапу криптографічного методу 1.

Задати початкове значення $i = 0$, тоді як (додатковий звичайний текст M_i , щоб Еліс його зашифрувала)

```
{
    Отримати динамічний ключ  $K_i = \text{пк} \circ \Psi(\Gamma_i, 0 \dots \Gamma_i, n-1)$ 
    Обчислити  $C_i = E_A(M_i \oplus C_{i-1}, K_i)$ , який шифрує  $M_i \oplus C_{i-1}$  за допомогою ключа  $K_i$ 
    Метод 1 обчислює елемент генератора ключів  $\Gamma$  ( $i + 1$ ) на підставі  $\Gamma$  ( $i$ )
    Збільшити  $i$ 
}
```

У методах 4 і 5 символ C-1 являє собою вектор ініціалізації, встановлений між Еліс і Бобом в процесі обміну генераторами ключів.

Криптографічний метод 5. Блочний шифр А виконує розшифровку за допомогою динамічних ключів і режиму CBC

Боб обчислює секрети $\Gamma(0)$, C-1 за допомогою 1-го етапу криптографічного методу 1.

Задати початкове значення $i = 0$

тоді як (додатковий шифрований текст C_i , щоб Боб його розшифрував)

```
{
    Отримати динамічний ключ  $K_i = \text{пк} \circ \Psi(\Gamma_i, 0 \dots \Gamma_i, n-1)$ 
    Обчислити  $M_i = C_{i-1} \oplus D_A(C_i, K_i)$ , яка розшифровує  $C_i$  за допомогою ключа  $K_i$ 
    Метод 1 обчислює елемент генератора ключів  $\Gamma$  ( $i + 1$ ) на підставі  $\Gamma$  ( $i$ )
    Збільшити  $i$ 
}
```

В інших прикладах здійснення винаходу можуть використовуватися інші криптографічні режими, такі як CTR або OFB. В одному з прикладів здійснення винаходу метод 1 виконується в передавальному пристрої 102, а також приймальному пристрої 112, як показано на Фіг. 1А. В одному з прикладів здійснення винаходу методи 2 і 3 виконуються в передавальному пристрої

102, а також приймальному пристрої 112, як показано на Фіг. 1А. В іншому прикладі здійснення винаходу методи 4 і 5 виконуються в передавальному пристрої 102, а також приймальному пристрої 112, як показано на Фіг. 1А. У деяких прикладах здійснення винаходу недетермінований генератор 172 на Фіг. 1І, який використовується в першому етапі методу 1, може використовувати фотони, як показано на Фіг. 1D, або інші види квантових ефектів для створення недетермінованості.

У деяких прикладах здійснення винаходу, які показані на Фіг. 1І, інструкції з оновлення генератора ключів 162 і інструкції з отримання ключів 168, описаних в методах 1, 2, 3, 4 і 5 є частиною процесу шифрування 160. У деяких прикладах здійснення винаходу, які показані на Фіг. 1І, 1Е і 1F, генератор ключів при оновленні за методами 1, 2, 3, 4 і 5 може бути реалізований, як виконувати інструкції пристрою у власних інструкціях до мікропроцесору пристрою. В інших прикладах здійснення винаходу генератор ключів при оновленні за методами 1, 2, 3, 4 і 5 може бути реалізований в обладнанні, такому як FPGA (програмована користувачем матриця логічних елементів) або ASIC (замовна спеціалізована мікросхема). В інших прикладах здійснення винаходу інструкції по оновленню генератора ключів 162 за методами 1, 2, 3, 4 і 5 можуть бути реалізовані, як вихідний код С і скомпільовані з внутрішніми інструкціями для ASIC, мікропроцесора або FPGA.

6.5. ОНОВЛЕННЯ ГЕНЕРАТОРА КЛЮЧІВ І ДИНАМІЧНІ КЛЮЧІ З ОДНОСТОРОННЬОЇ ФУНКЦІЄЮ ХЕШУВАННЯ

В даному розділі описується один із прикладів здійснення винаходу, який є варіацією методу 2: поліпшений AES-128 є блоковим шифром, який використовує 128-бітові динамічні ключі. Кессак діє, як одностороння функція хешування Φ , яка виконує оновлення генератора ключів. SHA-512 [23] реалізує односторонню функцію хешування Ψ , яка допомагає виконувати отримання динамічних ключів. Два кроки для зміни методу 2:

SHA-512 має розмір профілю в 512 біт, тому обчислення одиночного профілю може створити чотири різних 128-бітних ключа. Таким чином, швидкості виконання шифрування і розшифровки збільшуються при виконанні цих двох кроків, тільки коли $i \bmod 4 = 0$. Визначити функцію $\Pi : \{0,1,2,3\} \times \{0,1\}^{512} \rightarrow \{0,1\}^{128}$, так як $\Pi(a, (x_0, x_1, \dots, x_{511})) = (x_{128a}, x_{128a+1}, \dots, x_{128a+127})$, де $a \in \{0,1,2,3\}$ і $(x_0, x_1, \dots, x_{511}) \in \{0,1\}^{512}$. Задати, що $n = 1024$, тому для всіх i і для кожного $j \in \{512, \dots, 1023\}$, тоді $G_{i,j} = G_{i+1,j}$. Функція $\Phi = \text{Кессак}$, а $\Psi = \text{SHA-512}$.

Криптографічний метод 6. Оновлення генератора ключів за допомогою Кессак і отримання динамічних ключів за допомогою SHA-512

Метод шифровки Еліс

Перший крок методу 1 надає Еліс загальний секрет $\Gamma(0)$.

Задати початкове значення $i = 0$, тоді як (додатковий звичайний текст M_i , щоб Еліс його зашифрувала)

{

Задати, що $a = i \bmod 4$

Якщо $(a == 0)$, тоді обчислити $\beta_{i/4} = \Psi(\Gamma_{i,0} \ \Gamma_{i,1} \ \dots \ \Gamma_{i,1023})$

Задати, що динамічний ключ $K_i = \Pi(a, \beta_{i/4})$

Обчислити $C_i = E_{\text{AES}}(M_i, K_i)$, який зашифровує M_i ключем K_i

Якщо $(a == 0)$, то криптографічний метод 1 обчислює елемент $\Gamma(i+1)$ на підставі $\Gamma(i)$

Збільшити i

}

Метод розшифровки Боба

Перший крок методу 1 надає Бобу загальний секрет $\Gamma(0)$.

Задати початкове значення $i = 0$, тоді як (додатковий звичайний текст M_i , щоб Боб його зашифрував)

{

Задати, що $a = i \bmod 4$

Якщо $(a == 0)$, тоді обчислити $\beta_{i/4} = \Psi(\Gamma_{i,0} \ \Gamma_{i,1} \ \dots \ \Gamma_{i,1023})$

Задати, що динамічний ключ $K_i = \Pi(a, \beta_{i/4})$

Обчислити $M_i = D_{\text{AES}}(C_i, K_i)$, який розшифровує C_i ключем K_i

Якщо $(a == 0)$, то криптографічний метод 1 обчислює елемент $\Gamma(i+1)$ на підставі $\Gamma(i)$

Збільшити i

}

6.6. ОБЧИСЛЮВАЛЬНА СКЛАДНІСТЬ І ОДНОСТОРОННІ ФУНКЦІЇ ПРООБРАЗУ

На основі машин Тьюринга даний розділ вводить поняття обчислювальної складності, а потім визначає односторонню функцію хешування прообразу. Першою метою наших визначень

є виключити складність того, що асимптотичні визначення складності не можуть моделювати односторонні функції хешування, які використовуються на практиці. Другою більш довгостроковою метою є подальший розвиток належної структури для опису односторонності, шляхом застосування потужних інструментів на основі динамічних систем до машини Тьюринга.

5 Якщо говорити коротко, то машина Тьюринга являє собою потрійне (Q, Σ, η) , де Q являє собою кінцевий набір станів, який не містить унікального стану припинення h . Коли починається машинне виконання, то машина знаходиться в початковому стані $s \in Q$. Σ - це кінцевий алфавіт, символи якого читаються зі стрічки T і записуються на неї: $Z \rightarrow \Sigma$. Символ алфавіту в k -ої клітці стрічки - це $T(k)$. -1 і $+1$ представляють собою подачу головки стрічки в ліву чи праву

10 клітку стрічки відповідно. η - це програмна функція, де $\eta: Q \times \Sigma \rightarrow Q \cup \{h\} \times \Sigma \times \{-1, +1\}$.
Для кожного q в Q і α в Σ , інструкція $\eta(q, \alpha) = (r, \beta, x)$ описує те, як машина виконує один крок розрахунку. Перебуваючи в стані q і читаючи символ алфавіту α на стрічці: машина переходить в стан r . На стрічці машина заміняє символ алфавіту α на символ β . Якщо $x = -1$ або $x = +1$, тоді машина пересуває свою голівку стрічки на одну клітку вліво або вправо відповідно, а

15 потім читає символ в цій новій клітці. Якщо $r = h$, то машина досягає стану припинення і припиняє виконання.
Визначення 1. Обчислювальна складність для величини $u \in \Sigma^*$, яка вводиться в машину, припустимо, $|u|$ буде довгою u . Припустимо, $g: N \rightarrow N$ буде функцією $|u|$. Машина $M = (Q, \Sigma, \eta)$ має обчислювальну складність $C(g, \sigma, \rho, |u|)$, якщо дотримуються три наступних умови: При введенні u , машині M потрібно, як мінімум, $g(|u|)$ кроків обчислення для припинення. Алфавіт M задовольняє умові $|\Sigma| \leq \sigma$. Стану M задовольняють умові $|Q| \leq \rho$.

Примітка 1.

25 Параметри σ і ρ вводять обмеження на розмір програми η машини Тьюринга для того, щоб виключити попередні обчислення (табличні підстановки). Передбачається, що попередні обчислення будуть закодовані в η або / вхідні дані u .

Примітка 2.

Відзначимо, що попередні визначення складності залежать від значення алгоритму. Для будь-якого даного алгоритму може бути невизначена кількість машин Тьюринга, яка реалізує алгоритм, в якому кожна з цих машин має Складність Шеннона Стан \times Символ [40], таку що $|Q| |\Sigma| > \rho \sigma$. Різниця між реалізацією машиною алгоритму і абстрактним алгоритмом може призвести до глибоких тонкощів [41, 42, 43]. В [44], чорний ящик створений за допомогою самозмінюваної, паралельної машини, яка використовує квантову випадковість; даний необчислюваний метод піднімає більше питань про відмінності між алгоритмом і "машиною",

35 яка виконує його. Також дивіться [45].

Примітка 3.

З точки зору практики, атаки по побічним каналам зазвичай використовують реалізацію алгоритму конкретно машиною. (Наприклад, дивіться [46].) Це тим більш підтримує нашу тезу про те, що визначення складності повинно бути засноване на машині, а не на алгоритмі. Довільно, $h: \{0, 1\}^N \rightarrow \{0, 1\}^q$ є (N, σ, ρ, r) односторонньою функцією прообразу, якщо A і B відповідають умовам:

А. Машина Тьюринга M приймає, що на вхідному значенні x виводиться $h(x)$ в допустимій кількості кроків обчислення.

45 В. Будь-яке ймовірне P машини Тьюринга, якому присвоюється $y \in \{0, 1\}^q$ в якості вхідних даних, і яке шукає точку повного прообразу $x \in h^{-1}(\{y\})$, є успішним тільки при експоненціально низької можливості при наступних трьох умовах: (1) Машина Тьюринга P має максимум σ алфавітних символів. (2) Машина Тьюринга P має максимум ρ станів. (3) Є деяка фіксована ступінь r і кожна є успішною, як мінімум $|x| \geq r$ кроків.

У нашому формальному визначенні ніяких припущень не робиться щодо стійкості до накладення.

50 Визначення 2. Одностороння функція прообразу. Припустимо, $\sigma, \rho, r \in N$. Припустимо, $N \in N \cup \{\omega_0\}$. Функція $h: \{0, 1\}^N \rightarrow \{0, 1\}^q$ називається (N, σ, ρ, r) одностороння функція прообразу з розміром профілю q , якщо дотримуються дві наступні умови:

А. Легко оцінити: Існує машина Тьюринга для полиноміального часу A , така щоб $M(x) = h(x)$ для кожного $x \in \{0, 1\}^N$.

В. Обчислювальна складна для інверсії: Для кожного ймовірного P машини Тьюринга і кожного n , такого щоб $q \leq n < N$, ймовірність множини $\{x \in \{0, 1\}^n : P(h(x) \upharpoonright n) \in h^{-1}(\{h(x)\})\}$ і

Рвлодіє обчислювальною складністю $C(n, \sigma, \rho, |h(x) \upharpoonright n|) \leq 2^{-\frac{n}{2}}$.

Наступні примітки допомагають більш пояснити визначення 2.

Примітка 4.

ω_0 - це перший лічильно-нескінченний порядковий числівник. Коли $N = \omega_0$, це має на увазі, що область визначення h становить $\{0,1\}^*$, і в цьому випадку визначення 2 є асимптотичним.

Примітка 5.

5 Пристрій противника P отримує $h(x)$ в якості вхідних даних і допоміжні вхідні дані $1n$, які представляють собою двійкову довжину x . Метою допоміжних вхідних даних $1n$ є можливість того, що функція хибним чином вважається односторонньою, оскільки пристрій P не має достатньо часу, щоб вивести на друк її вихідні дані. Наприклад, функція $h(x) = y$, де $y = \log n$ найменш значущих бітів x з $|x| = n$. Жоден пристрій не може знайти точку $h^{-1}(\{y\})$ в часі

10 поліноміальному в $|h(x)|$; однак, існує пристрій, який знаходить точку $h^{-1}(y)$ у часі поліноміальному в $|x|$.

Примітка 6.

Для наших цілей тільки $n \geq q$ потрібно в алгоритмах 1 і 2. Є деяке $k < q$, таке яке противник може з використанням грубої сили обчислити $h(x)$ для кожного $x \in \{0, 1\}^j$ у всіх випадках, коли $j \leq k$. Число k залежить від обчислювальних ресурсів противника.

15

Примітка 7.

Одностороння концепція є ймовірнісною. Визначення не вказує, що для пристрою противника P є неможливим знайти точку в повному прообразі $h^{-1}(\{h(x)\})$; воно стверджує, що P має ймовірність $\leq 2^{-n}$ в знаходженні точки в повному прообразі, де пристрою потрібно, як мінімум ng кроків обчислення для того, щоб знайти її. Тут $g(|u|) = (|u| - q)r$, де $u = h(x) 1n$. Для "успіху" пристрій противника P має тільки визначити деяку точку в $h^{-1}(\{h(x)\})$. P не потрібно визначати x , яку використовувала машина M . Більш того, розподіл ймовірності є рівномірним щодо вхідних даних x , і можливого підкидання монет машини противника P .

20

Примітка 8.

Інтуїція для верхньої межі 2^{-n} за ймовірносними відгалуженнями на основі парадокса дня народження: за умови, що випадково обраний профіль y , для Єви має бути набагато складніше з обчислювальної точки зору знайти точку прообразу $x \in h^{-1}(\{y\}) \cap \{0, 1\}^n$, ніж для випадкового вибору Євою точок прообразів x_1, x_2, \dots, x_m обчислити $h(x_1), h(x_2), \dots, h(x_m)$, і знайти накладення в $\{h(x_1), h(x_2), \dots, h(x_m)\}$.

25

Приклад 1.

Припустимо, $\Phi_{512} : \{0,1\}^* \rightarrow \{0,1\}^{512}$ позначає SHA-512. Для Φ_{512} , $N = 2^{128}$, а $q = 512$. В даний час, не існує ніякого математичного доказу того, що SHA-512 є односторонньою функцією прообразу деяких значень r, σ і ρ . У зв'язку з цим корисно буде згадати про недавню атаку знаходження прообразу по повному дводольному графу [47] на скорочених 50 циклах Φ_{512} : їх оцінка складності прообразу в $2^{511.5}$ все ж підтримує таку можливість і набагато перевищує сьогодишню обчислювальну потужність. На практиці вхідні рядки $\geq 2^{128}$ біт не з'являються. Однак, на підставі визначення(ь) односторонності нинішньої практики SHA-512 не задовольняє їх математичному визначенню односторонньої функції хешування, оскільки область визначення SHA-512 не складає $\{0,1\}^*$ і в подальшому не може задовольняти асимптотичним вимогам визначення.

30

35

40

6.7. ДЕЯКИЙ АНАЛІЗ КРИПТОГРАФІЧНИХ МЕТОДІВ 1, 2, 3, 4 І 5

Припустимо, $f : X \rightarrow X$ є функцією на деякому топологічному просторі X . Орбітою точки $p \in X$ є $O(p, f) = \{p, f(p), f^2(p), \dots, f^n(p), \dots\}$. Взагалі, орбіта може являти собою нескінченну множину. У криптографічних методах 1, 2, 3, 4 і 5, простір $X = \{0, 1\}^m$ для деякого $m \in \mathbb{N}$, тому наші орбіти ключів і орбіти генератора ключів є кінцевими. Точка $p \in X$ являє собою періодичну точку, якщо є $j \in \mathbb{N}$, при якому $f^j(p) = p$. Точка $x \in X$ є евентуально періодичною, якщо є $k \in \mathbb{N}$, при якому $f^k(x) = p$, і p є періодичною точкою.

45

Припустимо, $f : \{0, 1\}^m \rightarrow \{0, 1\}^m$ є функцією. Принцип Діріхле передбачає, що кожна точка $x \in \{0, 1\}^m$ є евентуально періодичною з періодом максимум 2^m . Кожна функція $f : \{0,1\}^m \rightarrow \{0,1\}^m$ виводить відношення еквівалентності за множиною $\{0,1\}^m$ наступним чином. Якщо x і y є евентуально періодичними в тій самій орбіті відносно f , тоді x і y називаються евентуально періодичним еквівалентом, вираженням як $x \sim_f y$. Припустимо, $[x]$ позначає клас еквівалентності $\{y \in \{0,1\}^m : x \sim_f y\}$.

50

Орбіта генератора ключів $O(\Gamma, \Phi, A_1) = \{\pi_q \circ \Gamma(i) \in \{0,1\}^q : \Gamma(i) \text{ обчислюється за Методом 1}\}$. Розмір орбіти генератора ключів являє собою кількість точок в $O(\Gamma, \Phi, A_1)$. Також, A_2 і A_4 позначають криптографічні методи 2 і 4 відповідно.

55

Визначення 3. Припустимо $\phi : \{0, 1\}^{<N} \rightarrow \{0, 1\}^q$ є функцією з розміром профілю q . (Ніяких

припущень не робиться щодо однобічності ϕ .) ϕ має періодичну точку $p \in \{0, 1\}^q$ з періодом m , якщо m є найдрібнішим позитивним числом, при якому $\phi^m(p) = p$.

Періодична орбіта, яка укладена в $O(\Gamma, \Phi, A_1)$ має період $\leq |O(\Gamma, \Phi, A_1)|$. Один з наших інструментів використовує теорему 1, щоб надати метод для визначення атаки знаходження прообразу на Φ на підставі евентуально періодичних класів еквівалентності.

Коли $q > k$, де $k = |K_j|$, необхідно згадати важливі тонкощі. З першого погляду, можна припустити, що послідовність динамічних ключів K_1, K_2, \dots повинна завжди мати період $\leq 2^k$, оскільки множина $\{K_1, K_2, \dots, K_{2^{k+1}}\}$ повинна мати накладення.

Воно ще більше збільшується парадоксом дня народження, який для послідовності $K_1, K_2, \dots, K_{2^{\lceil k/2 \rceil}}$ можливо містить два ідентичних динамічних ключа.

Якби ця послідовність динамічних ключів була створена дискретною, автономною динамічною системою $f : \{0, 1\}^k \rightarrow \{0, 1\}^k$, то перше накладення визначило б періодичність послідовності ключів. Замість цього орбіта $O(\Gamma, \phi, A_1) \subseteq \{0, 1\}^q$ використовується для отримання динамічних ключів K_1, K_2, \dots . Таким чином, розмір $O(\Gamma, \phi, A_1)$ може бути значно більше, ніж 2^k , зокрема, коли q істотно більше, ніж k . Дане спостереження приводить до теореми 1.

Теорема 1. Припустимо, що $z \in \{0, 1\}^q$ має період m щодо ϕ . Потім на z відбувається атака знаходження прообразу шляхом обчислення $m - 1$ ітерацій ϕ .

ДОВЕДЕННЯ. Обчислити $x = \phi^{m-1}(z)$. Потім $\phi(x) = \phi^m(z) = z$.

Наступне визначення допомагає проаналізувати криптографічні методи 2 і 4.

Визначення 4. Функція $\phi : \{0, 1\}^{<N} \rightarrow \{0, 1\}^q$ є регулярною в своїй підобласті $\{0, 1\}^k$ при $k \geq q$, якщо для кожного $y \in \{0, 1\}^q$, то перетин повного прообразу $\phi^{-1}(\{y\}) \cap \{0, 1\}^k$ мають однакову кількість точок. Це означає, що для кожного $y \in \{0, 1\}^q$, тоді $|\phi^{-1}(\{y\}) \cap \{0, 1\}^k| = 2^{k-q}$.

Теорема 2. Припустимо, що функція $\phi : \{0, 1\}^{<N} \rightarrow \{0, 1\}^q$ є регулярною в підобласті $\{0, 1\}^q$. Тоді кожна точка в $\{0, 1\}^q$ є періодичною точкою і знаходиться в унікальній періодичній орбіті щодо ϕ .

ДОВЕДЕННЯ. При приведенні до абсурду, припустимо, що $x \in \{0, 1\}^q$ не є періодичною точкою. Припустимо, k є найменшим позитивним натуральним числом $y = \phi^k(x)$ є періодичним числом.

Припустимо, m буде періодом y . Тоді $\phi^{-1}(\{y\})$ містить, як мінімум, дві точки $\phi^{m-1}(y)$ і $\phi^{k-1}(x)$. Ці дві точки суперечать умові регулярності ϕ . Унікальність періодичної орбіти x безпосередньо впливає з відношення еквівалентності \sim_ϕ , яке ϕ обумовлює на $\{0, 1\}^q$.

Коли ϕ задовольняє умові регулярності на підобласті $\{0, 1\}^q$, корисними є теореми 1 і 2, оскільки немає необхідності шукати розумні атаки знаходження прообразу. Замість цього, можна проаналізувати розмір і кількість періодичних орбіт ϕ на $\{0, 1\}^q$. Слідство 3 говорить про те, що 2^q дорівнює сумі періодів кожної періодичної орбіти щодо ϕ .

Слідство 3. Припустимо, що функція $\phi : \{0, 1\}^{<N} \rightarrow \{0, 1\}^q$ є регулярною в підобласті $\{0, 1\}^q$. Тоді сума всіх $|[x]| = 2^q$, де сума знаходиться в діапазоні кожного класу еквівалентності $[x]$, обумовлених \sim_ϕ і $|[x]|$, є число точок в $[x]$. Тобто, $|[x]|$ являє собою період x щодо ϕ .

ДОВЕДЕННЯ. \sim_ϕ є відношенням еквівалентності за $\{0, 1\}^q$. Використовувати теорему 2. Слідство 3 створює інструмент обчислення для визначення ймовірності того, що точка знаходиться в періодичній орбіті з періодом m . Як простий приклад припустимо, що $S : \{0, 1\}^8 \rightarrow \{0, 1\}^8$ позначає блок заміни, який використовується в AES. Потім \sim_S обумовлює п'ять класів еквівалентності $[0], [1], [4], [11], [115]$ за $\{0, 1\}^8$. Клас еквівалентності $[0]$ має 59 елементів. Це передбачає, що $S^{59}(0) = 0$ оскільки S є взаємно-однозначною відповідністю. Відзначимо, що $[11] = \{43, 241, 161, 50, 35, 38, 247, 104, 69, 110, 159, 219, 185, 86, 177, 200, 232, 155, 20, 250, 45, 216, 97, 239, 223, 158\}$. Також, $|[1]| = 81, |[4]| = 87, |[11]| = 27, |[115]| = 2, |[0]| + |[1]| + |[4]| + |[11]| + |[115]| = 2^8$.

В процесі одноразового виконання криптографічного методу 2 присутній низька ймовірність шифрування двох різних блоків з ідентичними ключами. Іншими словами, коли i не дорівнює j , то подія $K_i = K_j$ має низьку ймовірність. Наступна лема допомагає посилити вираз "низька ймовірність".

Лема 4. Припустимо, $\Phi : \{0, 1\}^{<N} \rightarrow \{0, 1\}^q$ є односторонньою функцією прообразу $(N, \sigma, \rho, r + m + 2)$, яка задовольняє умові регулярності по подобласті $\{0, 1\}^q$, де $r, m \geq 1, N = n + 1$, а $\sigma = q$ і $\rho = q^2$. Припустимо, що машина M обчислює Φ з яких-небудь вхідних даних $x \in \{0, 1\}^q$ за максимум qm кроків обчислення. Припустимо, що Еліс випадково вибирає $x \in \{0, 1\}^q$ і обчислює $\Phi(x) = y$. Припустимо, що Єва бачить тільки y . Задати, що $S = \{x \in \{0, 1\}^q : |O(\Gamma, \Phi, A_1)| < q^r \text{ і } \pi_q \circ \Gamma(0) = x\}$. Тоді $|S| \leq 2^{-q/2}$.

СХЕМА ДОКАЗУ. Використовуючи машину M , Єва обчислює орбіту $[y, \Phi(y), \Phi^2(y), \dots]$ з максимум q^r ітерацій. Після виконання обчислення кожної ітерації $\Phi^k(y)$, Єва здійснює пошук на предмет накладення в $[y, \Phi(y), \Phi^2(y), \dots, \Phi^k(y)]$. Якщо накладення знайдено, то пристрій Єви призупиняється. Якщо пристрій Єви досягає $\Phi^{q^r}(y)$ і не виявляє накладення, тоді пристрій Єви

призупиняється.

Там, де присутнє накладення в $\{y, \Phi(y), \dots, \Phi^k(y)\}$, за теоремою 2, умова регулярності передбачає, що y знаходиться в даній періодичній орбіті (клас еквівалентності). Припустимо, $a = |y|$. Тоді теорема 1 передбачає, що $x = \Phi^{a^{-1}}(y)$ є точкою прообразу, яку шукає Єва. Якщо $|S| > 2^{-q/2}$, тоді пристрій Єви знайде точку прообразу x за менш ніж $q^{r+m-1} \log q$ кроків обчислення з імовірністю більше, ніж $2^{-q/2}$, суперечачи твердженням, що Φ є односторонньою функцією прообразу $(N, \sigma, \rho, r + m + 2)$.

Приклад 2.

Розглянемо Φ_{512} , де $q = 512$. Припустимо, що $m = 3$, тому що 512^3 кроки є більш консервативною верхньою межею для обчислення машиною Тьюринга Φ_{512} за $x \in \{0,1\}^{512}$, ніж 512^2 . Якщо Φ_{512} задовольняє умові регулярності по підобласті $\{0, 1\}^{512}$, а Φ_{512} є функцією прообразу $(2^{128}, q, q^2, 9)$, тоді ймовірність становить $\leq 2^{-256}$, що генератор ключів за криптографічним методом 2 має орбіту, яка задовольняє $|O(\Gamma, \Phi_{512}, A_2)| < q^4$; з ймовірністю, як мінімум, $1 - 2^{-256}$, в тих випадках, коли j не рівняється k , тоді $\Gamma(j)$ не рівняється $\Gamma(k)$ для довжини шифрування до 8,5 мільярда байт. Бачення двох ідентичних ключів, які шифрують різні блоки, вимагає накладення SHA-512 тільки після 134 217 728 ітерацій SHA-512. В даний час не існує ніяких математичних доказів однобічності Φ_{512} ; проте, $(2^{128}, q, q^2, 9)$ здається консервативним, виходячи з атаки знаходження прообразу по повному дводольному графу [47], яка залежить від скорочених 50 циклів замість стандартних 80 циклів.

Примітка 9.

У відомому рівні техніки стандартні алгоритми блокового шифру, такі як AES, Serpent або DES не повинні показувати статичний ключ Єви; у відомому рівні техніки, якщо статичний ключ скомпрометований, то і криптографічна безпека фатально скомпрометована. Приклади здійснення винаходу, які наведені в даному описі, є кращими: Якщо динамічний ключ, який використаний в методах 2, 3, 4 і 5, зламаний Євою, то такий злом не відчиняє попередніх динамічних ключів і майбутніх динамічних ключів, які використовуються блоковим шифром A .

Для створення майбутніх динамічних ключів K_k , при яких $k > j$, Єва повинна знайти точку прообразу $\Gamma(j)$, j -ий генератор ключів. У криптографічному методі 2, припустимо, що блоковий шифр A - це AES-256, $q = 512$, а $n = 1024$ і припустимо, що з шляху обходу системи захисту процесора Єва отримує 256-бітний ключ K_{leak} . Навіть після витоку, щоб побудувати майбутні ключі, Єві необхідно знати $\Gamma(j)$. Для алгоритму 2 побудова майбутніх ключів вимагає від Єви значно більше обчислювальних кроків, ніж визначення однієї точки прообразу $x \in \{0,1\}^{1024}$, при якій $\Phi_{512}(x) = K_{leak}$. Якщо Φ_{512} є регулярною на субобласті $\{0,1\}^{1024}$, тоді $|\Phi_{512}^{-1}(K_{leak})| = 2^{512}$. Умова регулярності передбачає, що Єва повинна вгадати $\Gamma(j)$ з 2^{512} можливих точок прообразів. Коли Єва намагається з'ясувати динамічні ключі, які передують K_j , тоді у неї є навіть менше інформації, ніж коли вона намагається побудувати майбутні ключі. Тоді як останні $n - q$ біт з $\Gamma(j)$ є постійними, навіть якщо Єві відомий генератор ключів $\Gamma(j)$, то знання Єви не дозволяють їй негайно отримати $\Gamma(j - 1)$, оскільки $\Phi_{512}(\Gamma_{j-1,0} \dots \Gamma_{j-1,q-1}) = \Gamma_{j,0} \dots \Gamma_{j,q-1}$, а Φ_{512} витримує атаки знаходження прообразу.

Примітка 10.

Логічна функція $f: \{0,1\}^n \rightarrow \{0,1\}$ може бути виражена, як $f(x_1, \dots, x_n) = \sum_{a \in \{0,1\}^n} c_a x_1^{a_1} \dots x_n^{a_n}$ по $c_a = \sum_{x \leq a} f(x_1, \dots, x_n)$, де $F_2[x_1, \dots, x_n] / (x_1^2 - x_1, \dots, x_n^2 - x_n)$, а $x \leq a$, якщо $f(x_i) \leq a_i$ для кожного i .

Алгебраїчна ступінь f визначається, як $\deg f = \max\{wt(a) : a \in \{0,1\}^n \text{ и } c_a \neq 0\}$, де $wt(a)$ - це вага Хеммінга a . Розглянемо функції $f_1, f_2, \dots, f_n: \{0, 1\}^n \rightarrow \{0, 1\}$ і функцію $F: \{0, 1\}^n \rightarrow \{0, 1\}^n$, які визначаються, як $F(x) = (f_1(x), f_2(x), \dots, f_n(x))$. Алгебраїчна ступінь $F = \max\{\deg f_1, \deg f_2, \dots, \deg f_n\}$. Для статичного ключа AES K , функція шифрування AES $E_K: \{0,1\}^{128} \rightarrow \{0,1\}^{128}$ має алгебраїчну ступінь ≤ 128 , і E_K є функцією з 128 булевих логічних змінних. Добре відомо, що опір булевої функції диференціального криптоаналізу і диференціалам високого порядку залежить від її алгебраїчного ступеня і від того, наскільки швидко її ступінь може бути зменшена при взятті дискретних похідних [48, 49, 50].

Задати, що $M = |O(\Gamma, \Phi, A_4)|$. Для кожного динамічного ключа K_i , припустимо, що $E_{K_i}: \{0,1\}^{128} \rightarrow \{0,1\}^{128}$, позначають функцію кодування блокового шифру; таким блоковим шифром може бути AES, Serpent або інший блоковий шифр, розмір блоку якого становить 16 байт. В процесі виконання криптографічного методу 4 є $4M$ різні функції $E_{K_0}, \dots, E_{K_{4M-1}}$, де функція шифрування E_{K_0} застосовується до блоку звичайного тексту M_0 , функція шифрування E_{K_1} застосовується до блоку M_1 , і так далі. Така послідовність шифрування зумовлює функцію $f_\Gamma: \{0, 1\}^{512M} \rightarrow \{0, 1\}^{512M}$, где $f_\Gamma = (f_1, f_2, \dots, f_{512M})$. Як обговорювалося в прикладі 2, навіть для винятково малої ймовірності події, такого як накладення після всього 134217728 ітерацій SHA-512 (якщо така

орбіта існує), обумовлена f_r буде являти собою функцію з 68719476736 булевих змінних в порівнянні з 128 булевими змінними для E_k . Зчеплення блоків шифрованого тексту і орбіта генератора ключів створює компоновку функцій шифрування блоків шифрованого тексту E_{k_0} , E_{k_1} , ...; наприклад, $C_2 = E_{k_2}(M_2 \oplus E_{k_1}(M_1 \oplus E_{k_0}(M_0 \oplus C_{-1})))$. Таким чином, функції f_{1+128k} , ... $f_{128(k+1)}$ є функцією з $128(k+1)$ змінних $x_1, \dots, x_{128(k+1)}$ для $0 \leq k < 4M$. Більш того, дана примітка і факт того, що 68 719 476 736 набагато більше, ніж 128, пояснює те, як метод 4 істотно збільшує алгебраїчну ступінь в порівнянні з максимальним алгебраїчним ступенем фонового блочного шифрування, яке використовує статичний ключ у відомому рівні техніки.

6.8. ДИНАМІЧНІ КЛЮЧІ ЗУПИНЯЮТЬ ПРОСТУ АТАКУ НА БЛОКОВИЙ ШИФР

Динамічні ключі, отримані при криптографічних методах 2, 3, 4 і 5 допомагають зупинити просту атаку Хуанга і Лаї на блоковий шифр [51], яка описана нижче в алгоритмі Хуанга і Лаї 7. Наступний перелік описує символи, які використовуються в їх алгоритмі атаки 7.

P - звичайний текст

C - шифрований текст

n - розмір блоку

K - основний ключ

k - розмір основного ключа R - число циклів S - нелінійний шар L - лінійний шар

K^r - підключ, який використовується в циклі rX^r - вхідний блок в цикл r , де $X^0 = P$

Y^r - вихідний блок ключа, який змішується в циклі r

Z^r - вихідний блок нелінійного шару в циклі r

Z_i^r - i -ий підблок в Z^r

S_1 являє собою внутрішній стан, яке може розраховуватися на підставі P тільки за допомогою k_1 біт підключів, де k_1 є максимально меншою, ніж k , яка може бути отримана. Точно також S_2 - це внутрішній стан, який може бути отримано тільки з C за допомогою (інших) підключів k_1 біт. Для будь-якого блочного шифру можна визначити стан S_1 і S_2 . Алгоритм атаки має два етапи:

1. Стадія "зустріч посередині" створює перелік кандидатів, який містить 2^{k-M} ключів, де M - це дотримуваний проміжний розмір.

2. Стадія перевірки, яка досліджує ключі в переліку кандидатів.

Лінійні числа були додані в цей алгоритм атаки [51], щоб допомогти пояснити як криптографічні методи 2 і 4 заважають цій атаці.

Проста атака на блоковий шифр

Дані: Пари найменших чисел, які більше, ніж $(k / n) + 1$ (звичайний текст, зашифрований текст)

Результат: вихідний ключ K

1 для кожного значення в бітах 1ого ключа k_1 {

2 обчислити S_1 на підставі P при цих k_1 біт

3 для кожного значення в останніх бітах ключів $k - k_1$ {

4 обчислити $Z_0^{R/2}$ на підставі S_1

5 зберегти $Z_0^{R/2}$ в таблиці, відповідної вгаданому ключу

6}

7}

8 для кожного значення в останніх бітах ключів k_1 {

9 обчислити S_2 на підставі C при цих k_1 біт

10 для кожного значення в останніх бітах ключів $k - k_1$ {

11 обчислити $Z_0^{R/2}$ на підставі S_2

12 якщо $Z_0^{R/2}$ відповідає вгаданому ключу в таблиці {

13 додати вгаданий ключ до переліку кандидатів

14 перейти до наступної відгадки

15 }

16 або перейти до наступної відгадки

17 }

18}

19 Звірити ключі в переліку кандидатів з іншими парами (звичайний текст, зашифрований текст)

Метод алгоритму 7, який використовує перелік ключів-кандидатів для визначення статичного ключа блочного шифру не є ефективним проти криптографічних методів 2 і 4. Щоб це проілюструвати, розглянемо криптографічний метод 2, наприклад, з використанням 16-байтного блочного шифру, такого як Serpent, при $q = 512$ і $n = 768$. Після того, як кожен 16-байтний блок зашифрований, перелік кандидатів-ключів змінюється, тому що наступний 256-бітний ключ виходить з оновленого генератора ключів $\Gamma_{j,0} \dots \Gamma_{j,767}$, а середня відстань

Хеммінга між $\Gamma_{j,0} \dots \Gamma_{j,511}$ та $\Gamma_{j-1,0} \dots \Gamma_{j-1,511}$ становить 256. Розглянемо криптографічний метод 2, який шифрує 256000 байт голосових даних в секунду. При такій швидкості для одногодинної телефонної бесіди потрібна орбіта генератора ключів ($\Gamma_{0,0} \dots \Gamma_{0,511}$), $\Phi_{512}(\Gamma_{0,0} \dots \Gamma_{0,511})$, $\dots \Phi_{512}^{1440000}(\Gamma_{0,0} \dots \Gamma_{0,511})$ з розміром 1440001. Якщо в цій орбіті виникає накладення в процесі одногодинного телефонного дзвінка, тоді теорема 1 забезпечує переважну атаку знаходження прообразу на SHA-512, як мінімум, з 1440000 ітерацій SHA-512. Виходячи з надзвичайно низької ймовірності цієї рідкісної події (такі орбіти можуть навіть не існувати), накладення також призвело б до того, що SHA-512 не задовольняло б будь-які обґрунтовані значення складності прообразу (2^{128} , σ , ρ , r). "Обґрунтовані" означає, що вони не обмежують пристрій Єви Р настільки сильно, що вона не може обчислити, наприклад, SHA-512. Припустимо, $\rho = 1$, тому пристрій Р може мати тільки один стан.

Нагадуємо, що атака знаходження прообразу по повному дводольному графу [47] - на скорочені 50 циклів SHA-512 замість повних 80 - має розрахункову складність прообразу в $2^{511.5}$. Для типової орбіти надзвичайно ймовірно, що $O(\Gamma, \Phi_{512}, A_2)$ має розмір набагато більший, ніж число ітерацій SHA-512, щоб забезпечити повне шифрування для будь-якого прогнозованого застосування. В такому випадку допущення, що є пари \bar{n} (звичайний текст, зашифрований текст), не підходить для методу 2. Більш того, відсутність пар \bar{n} (звичайний текст,

зашифрований текст) робить недійсним ефективність контуру, який складається з ліній від 1 до 7 включно, і контуру, який складається з ліній від 8 до 18 включно.

6.9. РОЗПОДІЛ ГЕНЕРАТОРІВ КЛЮЧІВ NADO

NADO використовує симетричний генератор закритих ключів. Це означає, що початковий закритий ключ, який використовується шифратором для кожного процесу, є таким самим, як і генератор закритих ключів, який використовується дешифратором для відповідного процесу. Існують різні методи для поширення генератора закритих ключів NADO.

1. Може використовуватися електронний обмін генераторами ключів між сторонами. 2. Кур'єр може передати безпосередньо генератори ключів двом або більше сторонам. Метод 1 є кращим, коли кількість потенційних одержувачів зашифрованої передачі даних є великим, а потенційні одержувачі невідомі. Дані способи використання включають в себе: захищені бездротові способи використання, такі як розмови по мобільному телефону, бездротові передачі повідомлень електронної пошти, транзакції по бездротовій мережі, електронна торгівля по бездротовій мережі і супутникова передача даних. Захищені програмні способи використання, такі як додатки для роботи з повідомленнями електронної пошти, корпоративні обчислення, електронна торгівля в режимі реального часу, відправлення повідомлень в режимі реального часу, корпоративні порталні програми та інші інтернет-додатки.

У способах використання, де кількість потенційних одержувачів зашифрованої передачі даних є невеликим, а одержувачі відомі заздалегідь, можна використовувати метод 2, при якому відправники та одержувачі можуть домовитися про передачу їх генераторів закритих ключів безпечним чином. Цей метод може використовуватися, коли є побоювання щодо атаки із застосуванням технології "незаконний посередник" при обміні ключами.

6.10. ОБМІН ГЕНЕРАТОРАМИ КЛЮЧІВ

У наявному рівні техніки обмін ключами за методом Діффі-Хеллмана-Меркле є методом обміну ключами, при якому дві сторони (Еліс і Боб), які попередньо не знають один одного, встановлюють загальний секретний ключ на небезпечний канал зв'язку. В даному описі додаток до цього методу обміну використовують дві сторони (Еліс і Боб), щоб створити початковий генератор загального ключа $\Gamma(0)$ для Н-процесу; Обмін генераторами ключів залежить від характеристик комутативних груп. Група G являє собою множину з операцією з двійковими числами $*$, (g^2 означає $g * g$, а g^5 означає $g * g * g * g * g$), при якому витримуються чотири наступні властивості:

Операція з двійковими числами $*$ замикається на G . Іншими словами, $a * b$ знаходиться в G для всіх елементів a і b в G .

Операція з двійковими числами $*$ є асоціативною за G . $a * (b * c) = (a * b) * c$ для всіх елементів a , b і c в G .

У G . $a * e = e * a = a$ присутній унікальний елемент тотожності e .

Кожен елемент a в G має унікальну зворотну величину, яка позначається, як a^{-1} . $a * a^{-1} = a^{-1} * a = e$.

В абстрактному сенсі, іноді оператором нехтують, тому $a * b$ пишеться, як ab . Іноді тотожність групи представлена, як 1, коли групова операція являє собою деяку форму

множення. Іноді тотожність групи представлена, як 0, коли групова операція являє собою деяку форму складання. Числа {..., -2, -1, 0, 1, 2, ...} щодо їх операції з двійковими числами + є прикладом нескінченної групи. 0 є елементом тотожності. Наприклад, зворотна величина від 5 - це 5, а зворотна величина від 107 - це 107. Множина перестановок n елементів $\{1, 2, \dots, n\}$, позначене, як S_n , є прикладом кінцевої групи з $n!$ елементами, де операція з двійковими числами є композицією функцією. Кожен елемент S_n є функцією $\sigma: \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$, тобто від 1 до 1 і далі. У цьому контексті, σ називається перестановкою. Тотожна перестановка є являє собою елемент тотожності S_n , при якому $e(k) = k$ для кожної k в $\{1, 2, \dots, n\}$.

Якщо H є непустою підмножиною групи G , а H є групою щодо групової операції з двійковими числами $*$ з G , тоді H називається підгрупою G . H є власною підгрупою G , якщо H не дорівнює G (тобто, H є власною підмножиною G). G є циклічною групою, якщо G не має власних підгруп. Числа по модулю n (тобто, $Z_n = \{[0], [1], \dots, [n-1]\}$) є прикладом кінцевої групи щодо додавання по модулю n . Якщо $n = 5$, $[4] + [4] = [3]$ в Z_5 , тому що 5 ділить $(4 + 4)$. Так само, $[3] + [4] = [3]$ в Z_5 . Z_5 є циклічною групою, оскільки 5 є простим числом. Коли p є простим числом, Z_p є циклічною групою, що містить p елементів $\{[0], [1], \dots, [p-1]\}$. $[1]$ називається генеруючим елементом для циклічної групи Z_p , оскільки $[1]^m = [m]$, де m є натуральним числом, при якому $0 < m \leq p-1$, а $[1]^p = [0]$. Ця мультиплікативна концепція працює наступним образом: $[1]^2 = [1] + [1]$; $[1]^3 = [1] + [1] + [1]$; і так далі. Ця мультиплікативна концепція (тобто з використанням верхніх індексів) використовується в описі обміну генераторами ключів, описаного нижче.

Є нескінченне число циклічних груп, і нескінченне число цих циклічних груп є надзвичайно великим. Концепція надзвичайно великого числа означає наступне: якщо вважати, що 2^{1024} є надзвичайно великим числом, виходячи з обчислювальної потужності нинішніх комп'ютерів, то тоді все ще є нескінченне число кінцевих циклічних груп, і при цьому кожна циклічна група містить більше, ніж 2^{1024} елементів.

Кроки 1, 2, 3, 4 і 5 описують обмін генераторами ключів.

1. Еліс і Боб погоджуються на надзвичайно велику, кінцеву, циклічну групу G і генеруючий елемент g в G . Ця група G пишеться мультиплікативно, як пояснювалося раніше.

2. Еліс вибирає випадкове натуральне число a і відправляє g^a Бобу.

3. Боб вибирає випадкове натуральне число b і відправляє g^b Еліс.

4. Еліс обчислює $(g^b)^a$.

5. Боб обчислює $(g^a)^b$.

Еліс і Боб іноді домовляються про кінцеву циклічну групу G і про елемент g задовго до іншої частини протоколу обміну ключами; передбачається, що g відома всім атакуючим. Математична концепція циклічної групи, елемент, що генерує, і кінцеве поле представлені в [55].

Як Еліс, так і Боб тепер мають груповий елемент g^{ab} , який може служити в якості загального секретного ключа. Величини $(g^b)^a$ і $(g^a)^b$ є однаковими, тому що g є елементом групи G . Еліс може зашифрувати інформацію m , як $m * (g^{ab})^a$, і направляє $m * (g^{ab})^a$ Бобу. Боб знає $|G|$, b і g^a . Для кінцевих груп теорема Лагранжа передбачає, що порядок кожного елемента групи розділяє число елементів в групі, позначених як $|G|$. Це означає $x^{|G|} = 1$ для

всіх x в G , де 1 є елементом тотожності в групі G . Боб обчислює $(g^a)^{|G|b} = (g^{|G|})^a * g^{ab} = (g^{ab})^{-1}$. Після того, як Боб отримує зашифровану інформацію $m * (g^{ab})^a$ від Еліс, тоді Боб застосовує $(g^{ab})^{-1}$ і розшифровує зашифровану інформацію шляхом обчислення $m * (g^{ab})^a * (g^{ab})^{-1} = m$.

6.11. ОБМІН ГЕНЕРАТОРАМИ КЛЮЧІВ НА ОСНОВІ ЕЛІПТИЧНИХ КРИВИХ

Даний розділ описує криптографію на основі асиметричних ключів, яка називається криптографією на основі еліптичних кривих, яка в деяких прикладах здійснення винаходу може використовуватися для того, щоб реалізувати обмін генераторами ключів. Концепція $\text{Enc}(E, m)$ використовується для того, щоб представити результат шифрування звичайного тексту m з використанням еліптичної кривої E . З чого випливає, що концепція $\text{Dec}(E, c)$ використовується для представлення результату розшифровки шифрованого тексту c , який включений, як точка на еліптичній кривій E . В одному з прикладів здійснення винаходу криптографія на основі еліптичної кривої представляє собою метод асиметричної криптографії, який використовується

для створення генераторів ключів загальних для Еліс і Боба.

В одному з прикладів здійснення винаходу передбачається, що Е являє собою еліптичну криву за кінцевим полем F_p , де p - це просте число, а H - це циклічна підгрупа $E(F_p)$, що генерується точкою P , яка знаходиться в $E(F_p)$. Еліс хоче безпечно відправляти інформацію Бобу, загальноновідомим ключем якого є (E, P, aP) , і закритим ключем якого є натуральне число $a < p-1$.

Еліс виконує наступний Етап Кодування. Вибрати випадкове натуральне число $b < p-1$. Взяти до уваги, що інформація звичайного тексту вставлена, як точки m на E . Обчислити $\beta = bP$ та $\gamma = m + b(aP)$. Направити зашифрований текст $Enc(E, m) = c = (\beta, \gamma)$ Бобу.

Боб виконує наступний Етап Розшифровки після отримання зашифрованого тексту $c = (\beta, \gamma)$. Звичайний текст m витягується за допомогою закритого ключа, як $Dec(E, c) = m = \gamma - a\beta$.

Обчислення еліптичної кривої за кінцевим полем, також дають можливість Еліс і Бобу створити загальні генератори закритих ключів перед тим, як почнеться симетрична криптографія. Нижче наводиться простий приклад, описаний тут для ілюстрації, а не для цілей безпеки. Взяти до уваги, що еліптична крива E задана $y^2 = x^3 + 4x + 4$ по F_{13} . Можна показати, що $E(F_{13})$ має 15 елементів, яка обов'язково є циклічною. Також $P = (1, 3)$ являє собою генератор E . Припускаючи, що загальноновідомим ключем Боба є $(E, P, 4P)$, де $a = 4$ є закритим ключем, а $m = (10, 2)$ - це інформація, яку Еліс хоче направити Бобу, то Еліс виконує наступне. Еліс вибирає $b = 7$ випадковим чином. Потім Еліс обчислює $Enc(E, m) = Enc(E, (10, 2)) = (bP, m + b(aP)) = (7P, (10, 2) + 7(4P)) = ((0, 2), (10, 2) + 7(6, 6)) = ((0, 2), (10, 2) + (12, 5)) = ((0, 2), (3, 2)) = (\beta, \gamma) = c$. Потім Еліс направляє зашифрований текст $c = (\beta, \gamma) = ((0, 2), (3, 2))$ Бобу, який використовує свій закритий ключ для розшифровки зашифрованого тексту і отримання інформації $m = (10, 2)$ наступним чином: $Dec(E, c) = (3, 2) - 4(0, 2) = (3, 2) - (12, 8) = (3, 2) + (12, 8) = (10, 2)$. Додаткова інформація за еліптичними кривими наводиться в [56, 57].

В одному з прикладів здійснення винаходу, еліптична крива з формою $y^2 = x^3 + Ax + x$, яка запропонована Монтгомері [58], може використовуватися для виконання обміну генераторами ключів на основі еліптичних кривих. В одному з прикладів здійснення винаходу закритий ключ може бути створений за допомогою послідовності з 96 байт (768 біт). В іншому прикладі здійснення винаходу закритий ключ може бути з 1024 байт (8192 біт). У прикладі здійснення винаходу, який показано під номером 136 на Фіг. 1B і під номером 140 на Фіг. 1D недетермінований генератор створює ці біти шляхом вимірювання часу настання події для фотонів, як описано в розділі 6.3, який називається КРИПТОГРАФІЧНЕ ОБЛАДНАННЯ ТА ІНФРАСТРУКТУРА. В одному з прикладів здійснення винаходу 768 різних точок часу настання події для фотонів $(t_{(1,1)}, t_{(1,2)}, t_{(1,3)}), (t_{(2,1)}, t_{(2,2)}, t_{(2,3)}) \dots (t_{(k,1)}, t_{(k,2)}, t_{(k,3)}), \dots (t_{(256,1)}, t_{(256,2)}, t_{(256,3)})$, які для кожного k задовольняють умові $t_{(k,1)} < t_{(k,2)} < t_{(k,3)}$ и $t_{(k,2)} - t_{(k,1)}$ не дорівнює $t_{(k,3)} - t_{(k,2)}$, спостерігаються при використанні недетермінованого генератора. Кожна трійка створює "1" або "0" в залежності від того чи буде $t_{(k,2)} - t_{(k,1)} > t_{(k,3)} - t_{(k,2)}$ АБО $t_{(k,2)} - t_{(k,1)} < t_{(k,3)} - t_{(k,2)}$.

Кожна відповідна загальнодоступна точка на еліптичній кривій обчислюється за допомогою базової точки на еліптичній кривій $y^2 = x^3 + Ax + x$ і 96-байтного закритого ключа, який був отриманий на підставі недетермінованого генератора 136 на Фіг. 1B. В одному з прикладів здійснення винаходу Еліс створює свою загальнодоступну точку на еліптичній кривій на підставі свого закритого ключа і базової точки; Боб обчислює свою загальнодоступну точку на еліптичній кривій за допомогою тієї ж кривої $y^2 = x^3 + Ax + x$ і тієї самої базової точки на цій кривій. У деяких прикладах здійснення винаходу Еліс і Боб можуть виконувати обмін генераторами ключів кілька разів.

В результаті виконання цього обміну один раз Еліс і Боб можуть створити загальні 96 байт для генератора ключів (0). В даному прикладі здійснення винаходу та інших прикладах здійснення винаходу, новий загальний генератор ключів може бути створений незалежно від попередніх генераторів ключів. В одному з прикладів здійснення винаходу еліптична крива з формою кривої $y^2 = x^3 + Ax + x$ описана нижче. В одному з прикладів здійснення винаходу, число A вибирається таким чином, щоб $(A-2)/4$ було невеликим числом. Це допомагає прискорити множення.

На підставі недетермінованого генератора (наприклад, обладнання на Рисунку 1D) Еліс створює наступний 96-байтний закритий ключ.

56 118 47 136 134 34 224 68 132 171 49 19 207 117 184 159 218 125 120 98 138 117 103 249
109 235 127 143 11 246 210 126 88 145 54 249 200 62 228 161 5 98 23 4 203 144 24 232 148
196 78 37 85 58 13 45 188 136 220 254 9 32 236 76 192 39 121 233 214 163 41 88 26 164 6
203 186 154 41 146 254 13 249 232 109 68 146 91 167 54 202 82 235 81 230 83

Еліс використовує свій закритий ключ і базову точку для обчислення на кривій $y^2 = x^3 + Ax + x$

х наступної загальнодоступної точки еліптичної кривої.

236 131 148 127 55 62 150 40 81 11 71 122 94 145 91 37 181 245 124 3 254 164 47 75 129
171 56 128 8 237 202 24 56 227 93 5 194 105 0 228 9 29 86 101 156 203 48 118 152 142
190 28 179 174 106 86 122 89 54 193 192 234 83 50 40 146 144 54 136 94 200 4 144 123
190 191 246 167 128 76 97 219 54 233 114 37 145 207 155 98 195 89 137 56 225 80

Еліс відправляє свою точку еліптичної кривої Бобу. На підставі недетермінованого обладнання, показаного на Фіг. 1D, Боб створює наступний 96-байтний закритий ключ.

176 99 149 32 146 54 112 229 54 146 107 179 39 0 64 13 38 201 241 89 203 87 41 227 95
183 205 206 133 201 117 92 40 134 199 246 72 205 82 245 119 149 67 30 151 73 154 238
25 96 96 106 175 247 80 69 174 13 220 219 115 243 152 71 168 237 59 17 227 94 148 27
190 90 48 139 16 141 20 89 71 51 159 23 63 70 175 239 0 144 130 90 208 84 150 64

Боб використовує свій закритий ключ і базову точку для обчислення на кривій $y^2 = x^3 + Ax + x$ наступної загальнодоступною точки еліптичної кривої.

78 33 138 252 54 163 232 126 202 160 84 232 216 188 31 8 131 133 101 195 178 208 241 223
127 35 129 40 42 137 40 95 25 113 231 248 138 236 54 147 113 119 226 59 141 6 27 192 175
179 56 187 151 93 151 106 228 52 58 96 87 26 249 121 164 164 41 244 19 183 74 32 189
249 124 109 246 68 254 173 186 233 187 57 118 242 197 140 135 199 90 254 200 227 14 33

Боб відправляє свою точку еліптичної кривої Еліс. Далі Еліс використовує свій закритий ключ і загальнодоступну точку еліптичної кривої Боба, щоб обчислити на кривій $y^2 = x^3 + Ax + x$ загальну точку, показану нижче, яка представляє собою генератор кривої $\Gamma(0)$.

235 126 229 251 14 146 63 250 203 150 6 199 175 116 50 153 21 37 23 64 172 71 113 143
125 226 131 23 41 73 63 72 227 109 78 221 181 3 6 204 128 84 141 146 60 218 238 121
98 245 57 203 26 192 215 170 218 148 42 176 14 142 193 8 29 157 107 135 231 5 222 171
5 227 97 236 110 144 214 165 76 246 193 22 84 220 92 202 3 219 10 253 231 45 13 114

Даний обмін дає можливість Бобу і Еліс створювати 96 байт загального генератора ключів $\Gamma(0)$.

Хоча винахід і було описано у відношенні конкретних прикладів здійснення винаходу, ті, хто має спеціальні знання, розуміють, що можуть бути внесені різні зміни, а еквіваленти можуть бути замінені на відповідні елементи без виходу за межі справжнього духу і обсягу винаходу. До того ж, зміни можуть бути внесені, не відхиляючись від фундаментальних ідей винаходу.

Список літератури

[1] Mihir Bellare and Phillip Rogaway. Introduction to Modern Cryptography. 2005.

<http://www.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf>

[2] Oded Goldreich. Foundations of Cryptography. Volumes I Basic Tools. Cambridge University Press 2001.

[3] Oded Goldreich. Foundations of Cryptography. Volume II Basic Applications. Cambridge University Press. 2004.

[4] T.W. Cusick and P. Stanica. Cryptographic Boolean Functions and Applications. Academic Press, Elsevier, 2009.

[5] NIST. Advanced Encryption Standard (AES), FIPS 197. Nov. 2001.

<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

[6] R. A. Mollin. Codes: The Guide to Secrecy From Ancient to Modern Times. Chapman & Hall. 527–530, 2005.

[7] Juliano Rizzo and Thai Duong. Practical Padding Oracle Attacks. Black Hat Conference. 201

https://www.usenix.org/legacy/event/woot10/tech/full_papers/Rizzo.pdf

http://en.wikipedia.org/wiki/Padding_oracle_attack

<http://people.cs.kuleuven.be/~andre.marien/security/playing%20with%20cbc.pdf>

[8] Alex Biryukov and Khovratovich, D.: Related-Key Cryptanalysis of the Full AES-192 and AES-256. In Matsui, M., ed.: Asiacrypt. LNCS 5912, Springer, 1–18, 2009.

[9] Alex Biryukov, Khovratovich, D., Nikolic, I. Distinguisher and Related-Key Attack on the Full AES-256. Advances in Cryptology - Crypto 2009. LNCS 5677. Springer, 231–249, 2009.

[10] Patrick Derbez, Pierre-Alain Fouque and Jeremy Jean. Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting. Advances in Cryptology - Eurocrypt 2011. LNCS 7881. Springer, 371–387, 2011.

[11] Alex Biryukov and Dmitry Khovratovich. Feasible Attack on the 13-round AES-256. 2010.

[12] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique Cryptanalysis of the Full AES. Advances in Cryptology - Asiacrypt 2011. LNCS 7073, Springer, 344–371, 2011.

[13] Daniel Bernstein and Tanja Lange. Non-uniform cracks in the concrete: the power of free precomputation. Advances in Cryptology - Asiacrypt. LNCS 8270. Springer, 321–340, 2013.

[14] Orr Dunkelman, Nathan Keller, Adi Shamir. Improved Single-Key Attacks on 8-round AES.

Cryptology ePrint Archive, Report 2010:322, 2010.

<http://eprint.iacr.org/2010/322.pdf>

[15] Jon Passki and Tom Ritter. An Adaptive-Ciphertext Attack against $I \oplus C$ Block Cipher Modes with Oracle. IACR Cryptology ePrint Archive 2012:292, 2012.

5 <http://eprint.iacr.org/2012/292.pdf>http://ritter.vg/blog-separator_oracle.html

[16] Horst Feistel. Cryptography and Computer Privacy. Scientific American. 228, No. 5, 15–23, 1973.

[17] Clark Robinson. Dynamical Systems Stability, Symbolic Dynamics, and Chaos. CRC Press. 1995.

10 [18] Dake. Image of SHA-1 avalanche effect.

http://commons.wikimedia.org/wiki/File:Sha1_avalanche_effect.png#

[19] Xuejia Lai. Higher Order Derivatives and Differential Cryptanalysis. In Communications and Cryptography: Two Sides of One Tapestry, R.E. Blahut et al., eds., Kluwer Academic Publishers, 227–233, 1994.

15 [20] Ming Duan, Xuejia Lai, Mohan Yang, Xiaorui Sun and Bo Zhu. Distinguishing Properties of Higher Order Derivatives of Boolean Functions. IACR Cryptology ePrint. 2010.

<https://eprint.iacr.org/2010/417.pdf>.

[21] NIST. FIPS-180-4. Secure Hash Standard, March 2012.

<http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>

20 [22] John Hennessy, David Patterson. Computer Architecture. 5th Edition, Morgan Kaufmann, 2012.

[23] NIST. FIPS-180-2: Secure Hash Standard, August 2002.

<http://www.itl.nist.gov/fipspubs/>.

25 [24] Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem. Proc. London Math. Soc. Series 2 42 (Parts 3 and 4), 230–265, 1936.

[25] Andre Stefanov, Nicolas Gisin, Olivier Guinnard, Laurent Guinnard, and Hugo Zbinden. Optical quantum random number generator. Journal of Modern Optics, 47(4):595–598, 2000.

[26] Mario Stipcevic and B. Medved Rogina. Quantum random number generator based on photonic emission in semiconductors. Review of Scientific Instruments. 78, 045104: 1–7, 2007.

30 [27] A. A. Abbott, C. S. Calude, J. Conder & K. Svozil. Strong Kochen-Specker theorem and incomputability of quantum randomness. Physical Review A. 86 062109, 1–11, 2012.

[28] John Conway and Simon Kochen. The Strong Free Will Theorem. Notices of the American Mathematical Society. 56(2), 226–232, February 2009.

35 [29] Simon Kochen and Ernst P. Specker. The Problem of Hidden Variables in Quantum Mechanics. Journal of Mathematics and Mechanics (now Indiana Univ. Math Journal) 17 No. 1, 59–87, 1967.

[30] Klint Finley. Chinese Supercomputer Is Still the Worlds Most Powerful. Wired Magazine. Nov. 18, 2013.

40 [31] Daniel Bernstein. Curve25519: new Diffie–Hellman speed records. Public Key Cryptography. LNCS 3958. New York, Springer. 207–228, 2006.

<http://cr.yp.to/ecdh/curve25519-20060209.pdf>

[32] Stephen Cook. The P VS NP Problem. <http://www.claymath.org/sites/default/files/pvsnp.pdf>

[33] A.F. Webster and S.E. Tavares. On the Design of S-Boxes. Advances in Cryptology. CRYPTO 85 Proceedings. LNCS 218. Springer, 523–534, 1986.

45 [34] Guido Bertoni, Joan Daemen, Michael Peeters, Gilles Van Assche. Keccak Reference 3.0 2011.

<http://keccak.noekeon.org/> <http://en.wikipedia.org/wiki/Keccak>

[35] Jean-Philippe Aumasson, Samuel Neves, Zooko Wilcox-O’Hearn, Christian Winnerlein. BLAKE.

50 <https://131002.net/blake/> [http://en.wikipedia.org/wiki/BLAKE_\(hash_function\)](http://en.wikipedia.org/wiki/BLAKE_(hash_function))

[36] Praveen Gauravaram, Lars Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schllfer, and Sren S. Thomsen. Grstl a SHA-3 candidate.

<http://www.groestl.info> <http://www.groestl.info/Groestl.pdf>

[37] Hongjun Wu. The Hash Function JH. 2011.

55 <http://ehash.iaik.tugraz.at/wiki/JH> http://www3.ntu.edu.sg/home/wuhj/research/jh/jh_round3.pdf

[38] Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, Jesse Walker. The Skein Hash Function Family. 2010.

<https://www.schneier.com/skein1.3.pdf> [http://en.wikipedia.org/wiki/Skein_\(hash_function\)](http://en.wikipedia.org/wiki/Skein_(hash_function))

[39] Ross Anderson, Eli Biham, Lars Knudsen. A Proposal for the Advanced Encryption Standard. <http://www.cl.cam.ac.uk/~rja14/Papers/serpent.pdf> <http://www.cl.cam.ac.uk/~rja14/serpent.html>

60

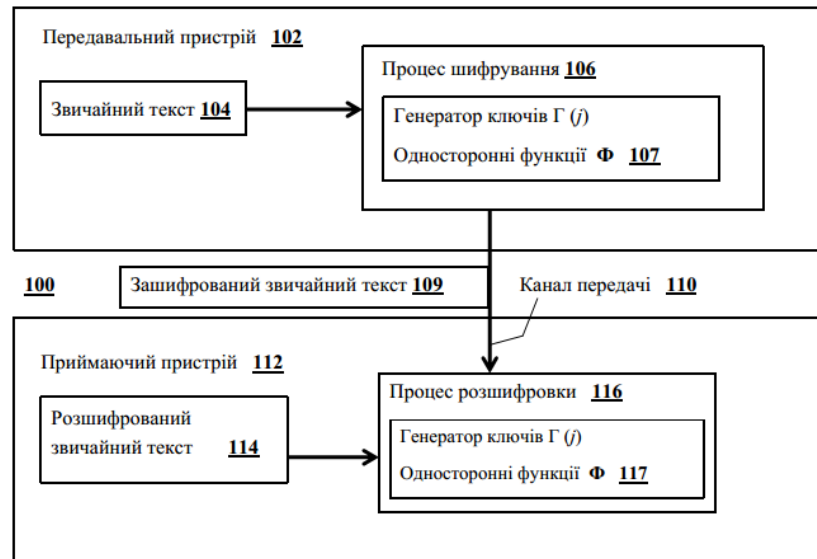
- [40] Claude Shannon. A universal Turing machine with two internal states. Automata Studies, C.E. Shannon and J. McCarthy (eds.). Princeton University Press, 129–153, 1956.
- [41] Yiannis N. Moschovakis. What is an algorithm? In Mathematics Unlimited 2001 and beyond (eds. B. Engquist and W. Schmid), Springer. 919–936, 2001.
- 5 [42] Yiannis N. Moschovakis. Algorithms and Implementations. Tarski Lecture 1, March 3, 2008. <http://www.math.ucla.edu/~ynm/lectures/tlect1.pdf>
- [43] Yuri Gurevich. What is an algorithm? In SOFSEM: Theory and Practice of Computer Science (eds. M. Bielikova et al.), LNCS 7147. Springer. 31–42, 2012. <http://research.microsoft.com/pubs/155608/209-3.pdf>
- 10 [44] Michael S. Fiske. Turing Incomputable Computation. Turing-100 Proceedings. Alan Turing Centenary. EasyChair 10, 69–91, 2012. <http://www.aemea.org/Turing100>.
- [45] Michael S. Fiske. Quantum Random Active Element Machine. UCNC 2013 Proceedings. LNCS 7956, 252–254, Springer, 2013. <http://www.aemea.org/UCNC2013>.
- 15 [46] Daniel Bernstein. Cache-timing attack on AES. 2005. <http://cr.yp.to/antiforgery/cachetiming-20050414.pdf>
- [47] Dmitry Khovratovich, Christian Rechberger and Alexandra Savelieva. Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 family. FSE, 244–263, 2012.
- 20 [48] E. Biham and Adi Shamir. Differential Cryptanalysis of DES-like Cryptosystems. Advances in Cryptology. CRYPTO 90 Proceedings. LNCS 537. Springer, 2–21, 1990.
- [49] Xuejia Lai. Higher Order Derivatives and Differential Cryptanalysis. In Communications and Cryptography: Two Sides of One Tapestry, R.E. Blahut et al., eds., Kluwer Academic Publishers, 227–233, 1994.
- 25 [50] L. Knudsen. Truncated and higher order differentials. In Fast Software Encryption. Springer, 196–211, 1995.
- [51] Jialin Huang and Xuejia Lai. What is the Effective Key Length for a Block Cipher: an Attack on Every Block Cipher. Science China Information Sciences. 57, Issue 7, Springer, 1–11, 2014.
- 30 [52] Fouz Sattar and Muid Mufti. Spectral Characterization and Analysis of Avalanche in Cryptographic Substitution Boxes using Walsh-Hadamard Transformations. International Journal of Computer Applications. 28 No.6. August 2011.
- [53] Michael Stephen Fiske. Non-autonomous Dynamical Systems Applicable to Neural Computation. Northwestern University. 1996.
- 35 [54] Mike Spivak. Differential Geometry. Volume I. Publish or Perish, Inc. 1979.
- [55] Nathan Jacobson. Basic Algebra I. W.H. Freeman and Company. 1985.
- [56] Neil Koblitz. Introduction to Elliptic Curves and Modular Forms. Springer-Verlag 1984.
- [57] Joseph Silverman and John Tate. Rational Points on Elliptic Curves. Springer-Verlag 1992.
- [58] Peter L. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. Mathematics of Computation 48, 243–264, 1987.
- 40 <http://cr.yp.to/bib/entries.html#1987/montgomery>.

ФОРМУЛА ВІНАХОДУ

- 45 1. Спосіб шифрування інформації, що включає в себе шифрування (106, 122, 160) одного або декількох блоків інформації повідомлення за допомогою блокового шифру на підставі першого ключа, який **відрізняється** тим, що включає етапи, на яких:
- отримують (155, 158, 168) перший ключ (156, 159) з генератора першого ключа (107, 117, 124, 154, 157, 162); причому
- 50 генератор першого ключа відрізняється від першого ключа;
- оновлюють зазначений генератор першого ключа (107, 117, 124, 150, 151, 152, 153, 162), на підставі функції (107, 117, 126, 164) для обчислення генератора другого ключа (107, 117, 124, 150, 151, 152, 153, 162);
- при якому оновлення не змінює щонайменше частину генератора ключів (151, 153);
- 55 отримують (155, 158, 168) другий ключ (156, 159) з генератора другого ключа (107, 117, 124, 154, 157, 162); причому
- генератор другого ключа відрізняється від першого ключа і від генератора першого ключа;
- другий ключ відрізняється від генератора другого ключа, першого ключа і від генератора першого ключа;

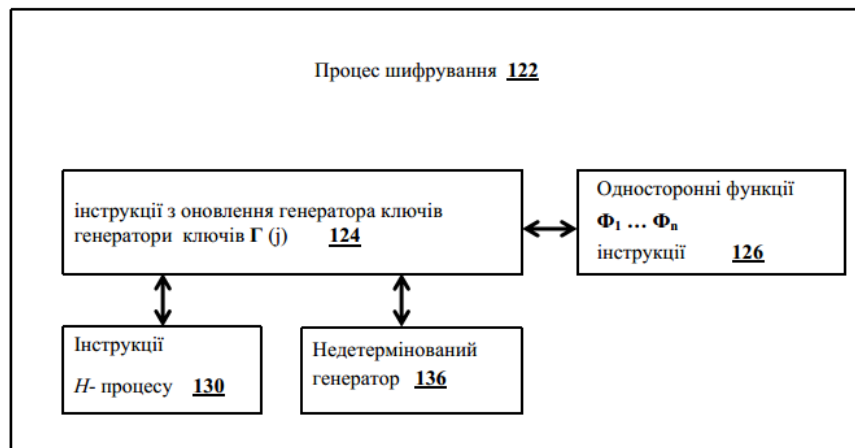
- шифрують (106, 122, 160) один або декілька блоків інформації повідомлення за допомогою блокового шифру на підставі другого ключа;
при якому в процесі отримання першого ключа одностороння функція (107, 117, 126, 155, 158, 164) застосовується як до перших, так і до других частин (150, 151, 152, 153, 154, 157) генератора першого ключа; і
при якому розмір першого ключа менше, ніж розмір генератора першого ключа.
2. Спосіб за п. 1, який **відрізняється** тим, що в ході зазначеного оновлення решта генератора першого ключа (107, 117, 124, 150, 152, 162) змінюється щонайменше частково при застосуванні односторонньої функції хешування (107, 117, 126, 155, 158, 164) або односторонньої функції прообразу (107, 117, 126, 155, 158, 164).
3. Спосіб за будь-яким з пп. 2-3, який **відрізняється** тим, що зазначені односторонні функції (107, 117, 126, 155, 158, 164) вимагають як мінімум 2^{64} кроків для обчислення точки прообразу або як мінімум 2^{64} кроків для обчислення, щоб визначити накладення.
4. Спосіб за будь-яким з пп. 1-3, який **відрізняється** тим, що в процесі отримання другого ключа одностороння функція (107, 117, 126, 155, 158, 164) застосовується як до перших, так і до других частин (150, 151, 152, 153, 154, 157) генератора другого ключа; і
при якому розмір другого ключа менше, ніж розмір генератора другого ключа.
5. Спосіб за п. 1, який **відрізняється** тим, що зазначений генератор першого ключа (107, 117, 124, 162) створюється на підставі недетермінованого процесу (142, 172), який використовує фотони.
6. Спосіб за п. 1, який **відрізняється** тим, що зазначена одностороння функція (107, 117, 126, 155, 158, 164), яка використовується при отриманні першого ключа, вимагає як мінімум 2^{64} кроків для визначення точки прообразу.
7. Криптографічна система (210, 214, 216, 218, 220, 250) шифрування інформації, яка містить процесор шифрування (106, 122, 160) одного або декількох блоків інформації повідомлення за допомогою блокового шифру на підставі першого ключа, яка **відрізняється** тим, що включає етапи, на яких:
отримують (155, 158, 168) перший ключ (156, 159) з генератора першого ключа (107, 117, 124, 154, 157, 162);
при якому генератор першого ключа відрізняється від першого ключа;
оновлюють зазначений генератора першого ключа (107, 117, 124, 150, 151, 152, 153, 162), на підставі функції (107, 117, 126, 164) для обчислення генератора другого ключа (107, 117, 124, 150, 151, 152, 153, 162);
при якому оновлення не змінює щонайменше частину генератора ключів (151, 153);
отримують (155, 158, 168) другий ключ (156, 159) з генератора другого ключа (107, 117, 124, 154, 157, 162); причому
генератор другого ключа відрізняється від першого ключа і від генератора першого ключа;
другий ключ відрізняється від генератора другого ключа, першого ключа і від генератора першого ключа;
шифрують (106, 122, 160) один або декілька блоків інформації повідомлення за допомогою блокового шифру на підставі другого ключа;
при якому в процесі отримання першого ключа одностороння функція (107, 117, 126, 155, 158, 164) застосовується як до перших, так і до других частин (150, 151, 152, 153, 154, 157) генератора першого ключа; і
при якому розмір першого ключа менше, ніж розмір генератора першого ключа.
8. Криптографічна система (210, 214, 216, 218, 220, 250) за п. 7, яка **відрізняється** тим, що в ході зазначеного оновлення решта генератора першого ключа (107, 117, 124, 150, 152, 162) змінюється щонайменше частково при застосуванні односторонньої функції хешування (107, 117, 126, 155, 158, 164) або односторонньої функції прообразу (107, 117, 126, 155, 158, 164).
9. Криптографічна система (210, 214, 216, 218, 220, 250) за будь-яким з пп. 7-8, яка **відрізняється** тим, що зазначені односторонні функції (107, 117, 126, 155, 158, 164) вимагають як мінімум 2^{64} кроків для обчислення точки прообразу або як мінімум 2^{64} кроків для обчислення, щоб визначити накладення.
10. Спосіб оновлення генератора ключів, реалізований за допомогою криптографічної системи, який **відрізняється** тим, що включає етапи, на яких:
оновлюють генератор першого ключа (107, 117, 124, 154, 157, 162), в процесі якого в результаті виходить генератор другого ключа (107, 117, 124, 154, 157, 162);
генератор першого ключа (107, 117, 124, 154, 157, 162) являє собою перший набір значень;
генератор другого ключа (107, 117, 124, 154, 157, 162) являє собою другий набір значень;

- при якому в процесі оновлення щонайменше частина генератора першого ключа (107, 117, 124, 151, 153, 162) не змінюється;
 при якому в процесі оновлення до другої частини генератора першого ключа застосовується
 одностороння функція (107, 117, 124, 150, 152, 162) отримання першого ключа (156, 159) з
 5 генератора першого ключа (107, 117, 124, 154, 157, 162);
 при якому перший ключ являє собою третє значення або третій набір значень;
 отримують другий ключ (156, 159) з генератора другого ключа (107, 117, 124, 154, 157, 162);
 при якому другий ключ являє собою четверте значення або четвертий набір значень;
 при якому в процесі отримання першого ключа одностороння функція (107, 117, 126, 155, 158,
 10 164) застосовується як до перших, так і до других частин генератора першого ключа; і
 при якому кількість значень першого ключа менше, ніж кількість значень в генераторі першого
 ключа.
 11. Спосіб за п. 10, який **відрізняється** тим, що оновлення використовує щонайменше функцію
 оновлення (107, 117, 126, 150, 152);
 15 функція (107, 117, 126, 150, 152) приймає комбінацію значень як вхідні дані;
 комбінація значень вхідних даних може мати різні довжини;
 функція (107, 117, 126, 150, 152) обчислює як вихідний результат комбінацію значень;
 комбінація значень вихідного результату має фіксовану, заздалегідь визначену довжину;
 функція (107, 117, 126, 150, 152) може бути обчислена за допомогою машини Тьюрінга, з
 20 кількістю кроків обчислення не більше вказаної;
 вказана кількість кроків визначається за допомогою поліноміальної функції з довжиною в
 комбінацію вхідних даних;
 12. Спосіб за будь-яким з пп. 10 або 12, який **відрізняється** тим, що для будь-якої машини
 Тьюрінга, яка має не більше першого зазначеного числа станів машини, і не більше другого
 25 зазначеного числа символів на стрічці, ймовірність при максимальному $2^{\frac{-n}{2}}$ того, що машина
 Тьюрінга може визначити поліноміальну кількість кроків обчислення, як функцію від n ,
 комбінацію вхідних даних, яка призвела в результаті до комбінації результатів функції, що
 оновлюється (107, 117, 126, 150, 152), не маючи комбінації вхідних даних, де n - це розмір
 комбінації вхідних даних, яку намагається знайти машина Тьюрінга.
 30 13. Спосіб за будь-яким з пп. 10-12, який **відрізняється** тим, що в процесі отримання (155, 158,
 168) другого ключа одностороння функція (107, 117, 126, 155, 158, 164) застосовується як до
 перших, так і до других частин генератора другого ключа; і
 при якому кількість значень другого ключа менша, ніж кількість значень в генераторі другого
 ключа.
 35 14. Криптографічна система (210, 214, 216, 218, 220, 250), яка включає в себе процесор для
 оновлення, яка **відрізняється** тим, що включає етапи:
 оновлення генератора першого ключа (107, 117, 124, 154, 157, 162), в процесі якого в результаті
 виходить генератор другого ключа (107, 117, 124, 154, 157, 162);
 генератор першого ключа (107, 117, 124, 154, 157, 162) являє собою перший набір значень;
 40 генератор другого ключа (107, 117, 124, 154, 157, 162) являє собою другий набір значень;
 при якому в процесі оновлення щонайменше частина генератора першого ключа (107, 117, 124,
 151, 153, 162) не змінюється;
 при якому в процесі оновлення до другої частини генератора першого ключа застосовується
 одностороння функція (107, 117, 124, 150, 152, 162) отримання першого ключа (156, 159) з
 45 генератора першого ключа (107, 117, 124, 154, 157, 162);
 при якому перший ключ являє собою третє значення або третій набір значень;
 отримання другого ключа (156, 159) з генератора другого ключа (107, 117, 124, 154, 157, 162);
 при якому другий ключ являє собою четверте значення або четвертий набір значень;
 при якому в процесі отримання першого ключа одностороння функція (107, 117, 126, 155, 158,
 50 164) застосовується як до перших, так і до других частин генератора першого ключа; і
 при якому кількість значень першого ключа менша, ніж кількість значень в генераторі першого
 ключа.



Фіг. 1А

120



Фіг. 1В

SHA-1 Лавинний Ефект



Fig. 1C

140

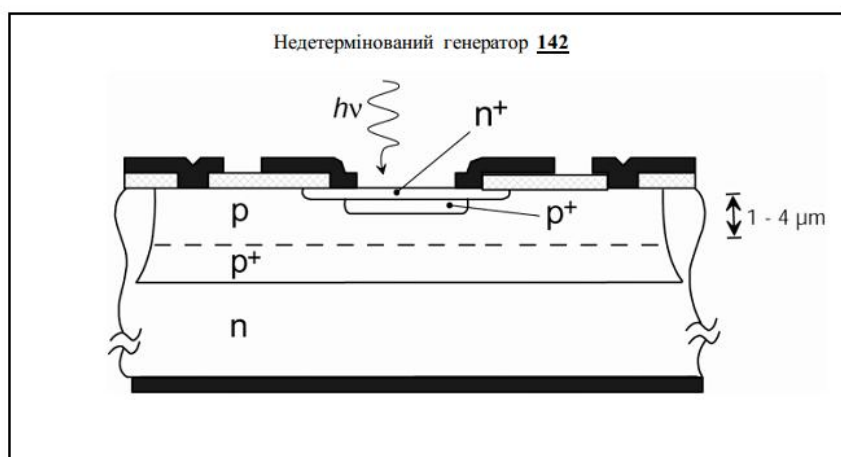
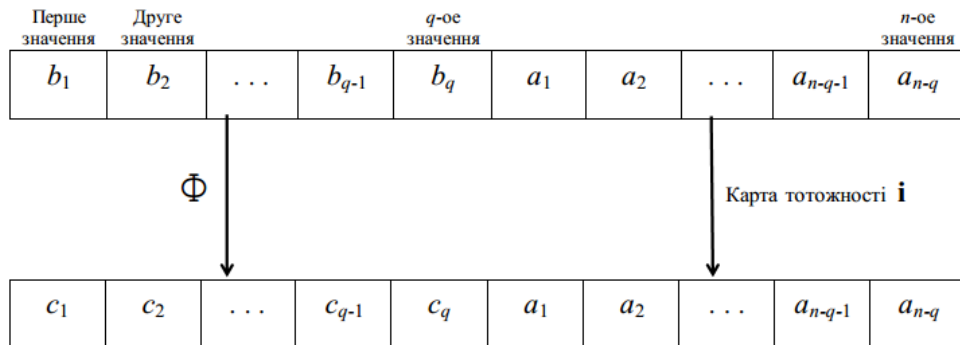


Fig. 1D

Крок оновлення генератора ключів



150 змінна частина генератора ключів

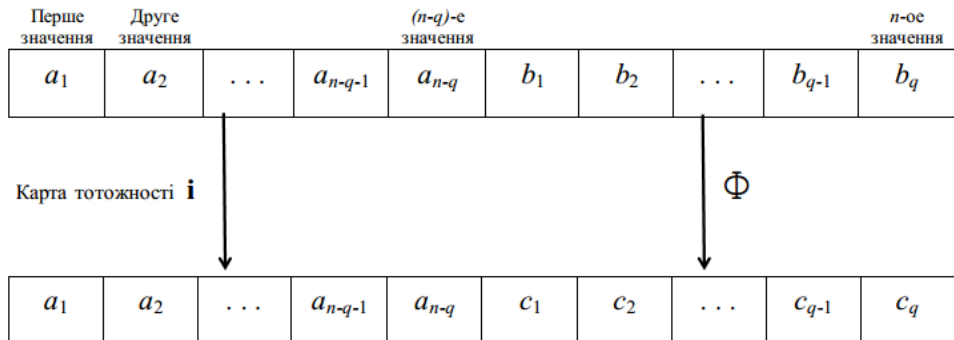
незмінна частина генератора ключів **151**

$$\Phi(b_1 \ b_2 \ \dots \ b_q) = (c_1 \ c_2 \ \dots \ c_q)$$

$$\mathbf{i}(a_1 \ a_2 \ \dots \ a_{n-q}) = (a_1 \ a_2 \ \dots \ a_{n-q})$$

Фіг. 1Е

Крок оновлення генератора ключів



153 незмінна частина генератора ключів

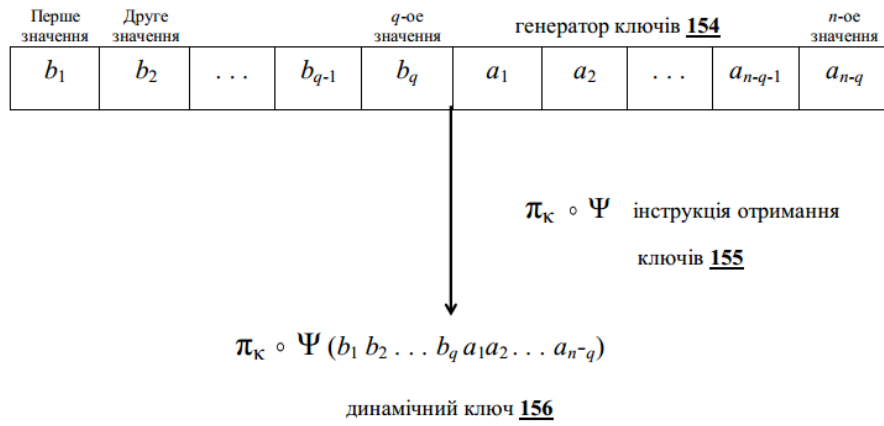
змінна частина генератора ключів **152**

$$\mathbf{i}(a_1 \ a_2 \ \dots \ a_{n-q}) = (a_1 \ a_2 \ \dots \ a_{n-q})$$

$$\Phi(b_1 \ b_2 \ \dots \ b_q) = (c_1 \ c_2 \ \dots \ c_q)$$

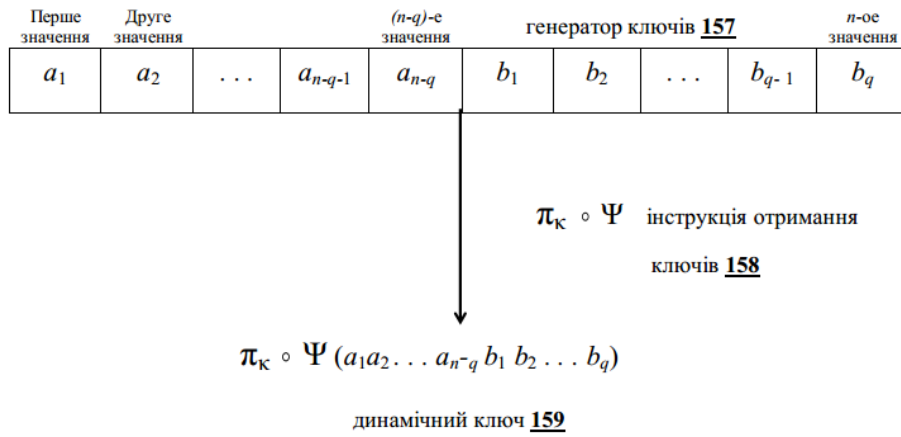
Фіг. 1F

Отримання динамічного ключа

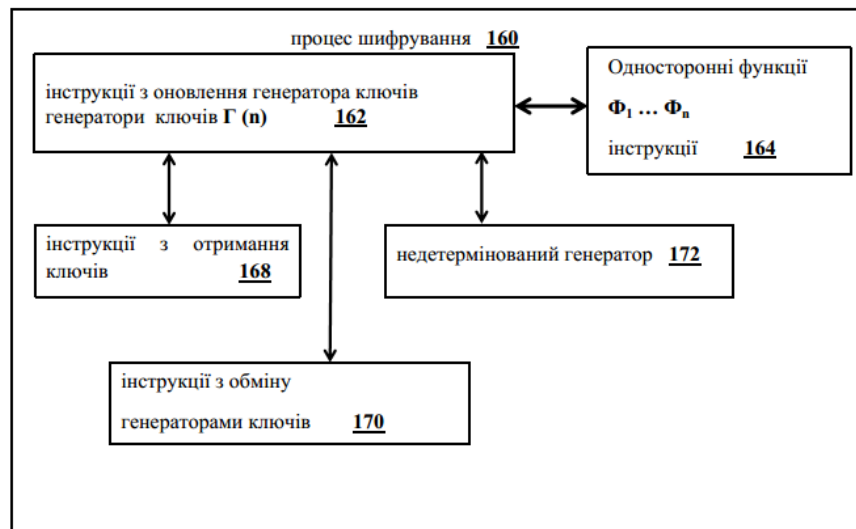


Фіг. 1G

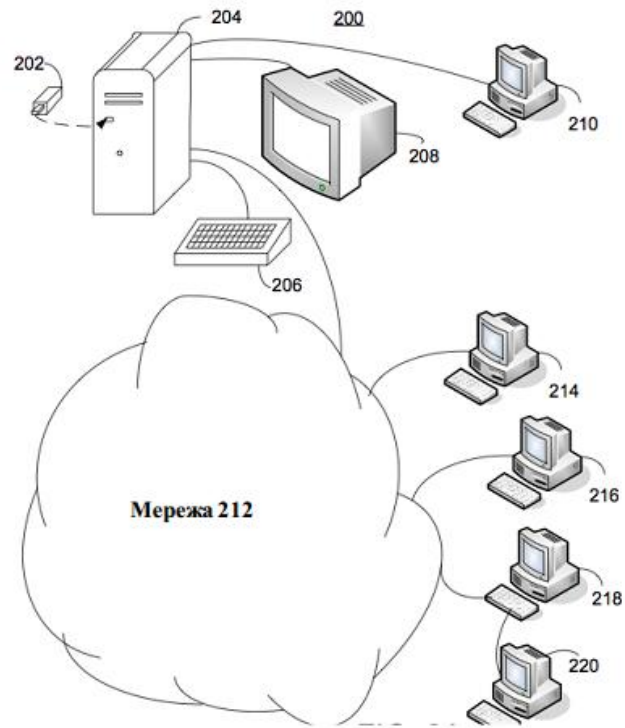
Отримання динамічного ключа



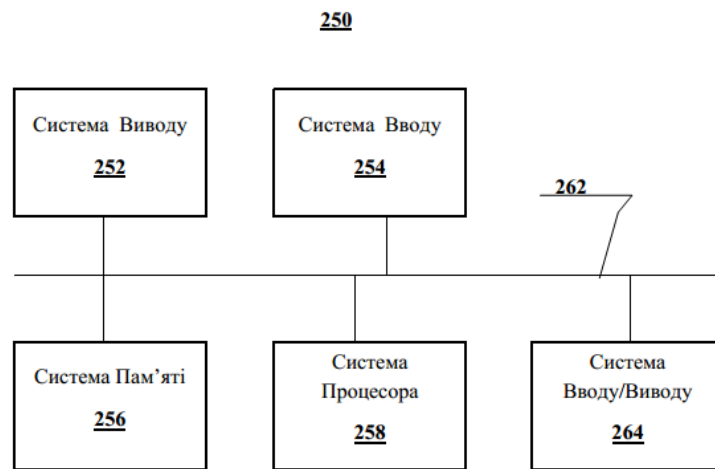
Фіг. 1H



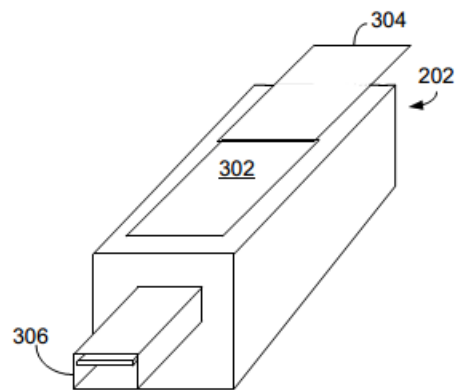
Фіг. 11



Фіг. 2А



Фіг. 2В



Фіг. 3А

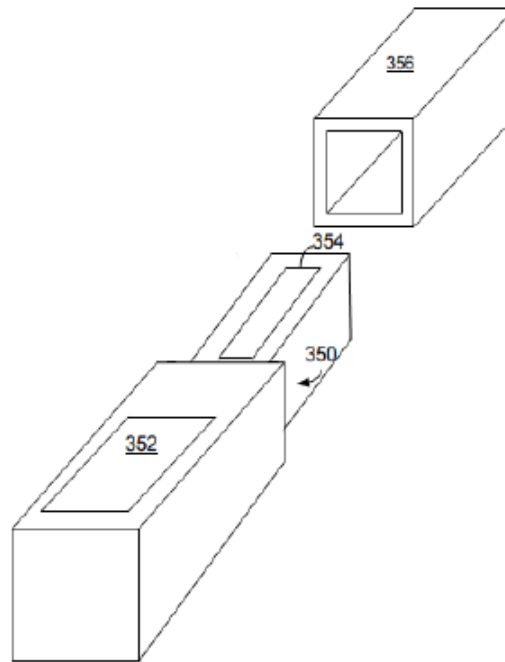


Fig. 3B

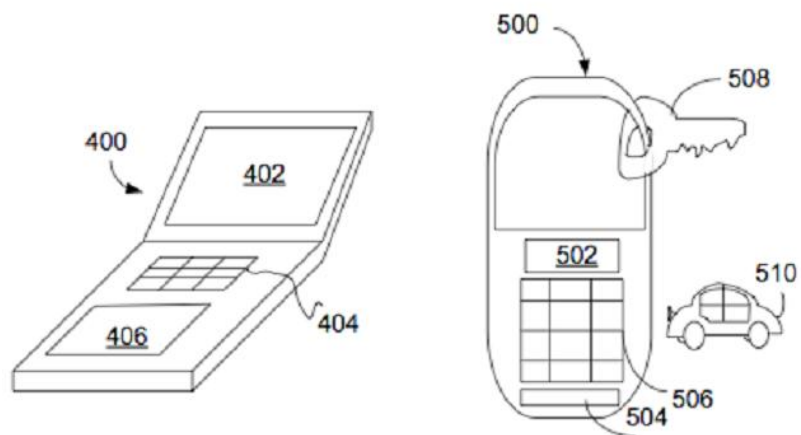
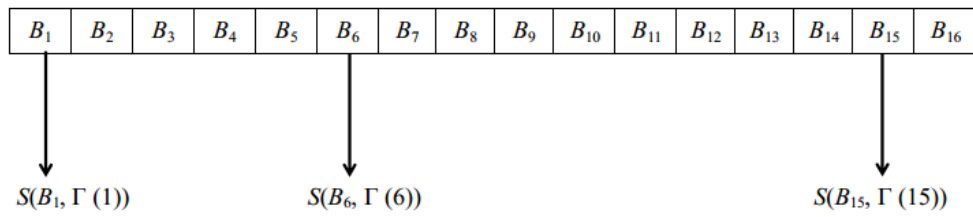


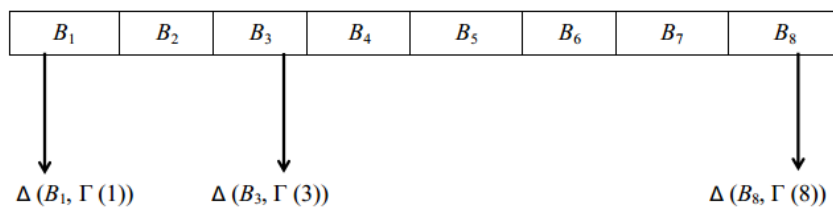
Fig. 4

Покращений блоковий шифр AES



Фіг. 5А

Покращений блоковий шифр Serpent



Фіг. 5В