DOI: 10.18372/2410-7840.21.13770
УДК: 681.3:519.72:003.26

ПОБУДОВА КЛЕПТОГРАФІЧНИХ МЕХАНІЗМІВ У ФУНКЦІЯХ ГЕШУВАННЯ

Антон Кудін, Богдан Коваленко

Робота присвячена клептографічним проблемам функцій гешування. Актуальність даної роботи витікає з ключової ролі функцій гешування у сучасних гібридних криптосистемах та з факту існування клептографічного вектору атак на такі системи. Наразі, попри те, що існує ряд робіт, у яких досліджено клептографічні можливості симетричних шифрів та асиметричних криптографічних протоколів, вкрай мало досліджень присвячено клептографічним проблемам функцій гешування. Недостатність досліджень клептографічних можливостей геш функцій зумовлює ризики наявності клептографічних механізмів у функціях гешування, вбудованих на етапі проектування та стандартизації. У даній роботі досліджуються можливості побудови функції гешування з клептографічним механізмом. Однією з неформальних вимог до таких функцій є вимога "подібності" її до відомих функцій гешування, тобто має базуватися на відомих загальних схемах геш функцій. У даній роботі для реалізації функції гешування з лазівкою пропонується використовувати схему Меркла-Дамгарда, що є основою багатьох відомих функцій гешування, а функцією стиснення обрано одну з загальновідомих конструкцій побудови функції стиснення на основі блокового шифру, що є доведено стійкими до побудови колізій. Замість блокового шифру в функції стиснення використовується перетворення спеціального виду, а також доводиться збереження стійкості до колізій з використанням даного перетворення. Результатом досліджень є геш функція з клептографічним механізмом, що дозволяє розробнику ефективно відновлювати частину (до 50%) повідомлення на основі геш коду та знання секрету у структурі клептографічного механізму. В той же час, функція залишається криптографічно стійкою для інших користувачів, що не володіють секретом.

Ключові слова: *геш функція, клептографія, клептографічний механізм, конструкція Меркла-Дамгарда, задача дискретного логарифмування.*

Вступ

Термін «клептографія» («kleptography») був уведений А. Яном та М. Юном у 1996 році [10]. Клептографія вивчає методи побудови в криптосистемах так званих витоків секрету (trapdoor) або ж «лазівок», що дозволяють розробникам отримувати доступ до певної секретної інформації користувачів криптосистем з метою стеження за потенційними злочинцями, збору маркетингової статистики клієнтів. Одним з найвідоміших прикладів є алгоритм генерації псевдовипадкових чисел Dual EC DRBG, що дозволятиме розробникам прогнозувати вихід генератора у випадку, якщо вони володіють секретним ключем каналу витоку [4]. Існують також роботи, присвячені побудові каналів витоків у розповсюджених асиметричних протоколах, проте наразі праць з клептографічних досліджень функцій гешування вкрай мало.

Актуальність даних досліджень випливає з актуальності досліджень стійкості функцій гешування та актуальності клептографічних аспектів побудови криптосистем. Функції гешування є одним з базових криптографічних примітивів, сфера використання яких включає:

1. Гарантування цілісності в симетричних системах шифрування (HMAC, [5]).

2. Основа для асиметричних криптосистем: систем електронного підпису, інфраструктури відкритих ключів, довірчих електронних послуг.

3. Гарантування конфіденційності в системах автентифікації: зберігання секретів, функції генерації ключа із пароля («Key Derivation Function», наприклад, [6]).

4. Основа генераторів псевдовипадкових чисел (наприклад, [1]).

5. Основа концепції Proof-of-Works в криптовалютах, а також технології децентралізованої обробки інформації Blockchain.

Щодо питань стійкості функції гешування, то з актуалізацією проблем побудови клептографічних механізмів, разом з класичними вимогами щодо складності пошуку прообразу, псевдопрообразу, сильної та слабкої колізії також розглядаються такі додаткові питання:

1. Клептографічна стійкість (КС) до пошуку прообразу/псевдопрообразу: функція є практично стійкою до пошуку прообразу/псевдопрообразу за умови відсутності інформації про «секрет» в її структурі. За умови знання «секрету» вона не є стійкою до пошуку прообразу/ псевдопрообразу.

2. КС до пошуку сильної/слабкої колізії: функція є практично стійкою до побудови сильної/слабкої колізії за умови відсутності інформації про «секрет» в її структурі. За умови знання «секрету» вона не є стійкою до побудови сильної/слабкої колізії.

Загальний аналіз клептографічної стійкості має певну специфіку:

1. З'являється поняття «секрету» в структурі функції, що знає лише розробник.

2. У випадку криптографічної стійкості, функції можна, грубо кажучи, розподілити на «стійкі» та «нестійкі» (не виконуються деякі з вимог криптографічної стійкості). У випадку ж клептографічного механізму також можна виділити класи криптопримітивів зі «стійким клептомеханізмом» (лише для розробника, що знає «секрет» не виконуються деякі з вимог криптографічної стійкості), із «нестійким клептомеханізмом» (користуватися лазівкою може зловмисник, не розробник), але ще з'являються класи криптопримітивів із «доведеною відсутністю клептомеханізмів» та «недоведеною відсутністю клептомеханізмів».

Для уточнення ступеня актуальності завдання побудови геш-функції з відомою лазівкою або виявлення цієї клептографічної лазівки проаналізуємо основні класи криптографічних протоколів, стійкіть яких розраховується в припущенні «складності» рішення задач знаходження прообразу геш-функції та обчислення колізії геш-функцій (випадкової або за заданим прообразом).

1. Протоколи автентифікації (автентифікація абонента або джерела даних).

У протоколах слабкою захищеної автентифікації (приклад – парольна автентифікація) можливість обчислення як прообразу так і будь-який колізії веде до повного злому протоколу.

У протоколах сильної автентифікації (приклад – автентифікація користувача з використанням механізму цифрового підпису) можливість обчислення прообразу веде до часткового злому протоколу, можливість обчислення колізій - як правило, не веде до злому протоколу.

2. Протоколи цифрового нотаріату (автентифікація даних).

У протоколах цифрового нотаріату (приклад – протоколи електронного документообігу, цифрового підписання контрактів, електронних виборів) можливість обчислення прообразу веде до повного злому протоколів, можливість обчислення колізій – до часткового злому протоколу.

3. Протоколи асиметричного шифрування (для забезпечення стійкості до дешифрування асиметричної криптосистеми).

У протоколах імовірнісного шифрування асиметричних криптосистем (приклад - протокол OAEP) можливість обчислення прообразу і колізій веде до часткового злому протоколу.

4. Протоколи криптовалют.

Окремим питанням впливу клептографії геш-функцій на стійкість протоколу постає в технологіях блокчейн. Сьогодні ця технологія може бути основою для розподіленої децентралізованої технології забезпечення цілісності інформації для вирішення найширшого роду прикладних задач: від децентралізованого випуску та обігу електронної готівки («криптовалюта»), автентифікації і електронного нотаріату до розподіленого підписання контрактів і електронних виборів. При цьому завдання оцінки ступеня впливу стійкості геш-функції на технологію повинна розглядатися окремо для задачі генерації блоку транзакцій і окремо - для механізму «майнінгу». Так одна з поширених технологій роботи блокчейна - технології «доказу виконаної роботи (англ. - proof-of-works)» повністю заснована на складності обчислення прообразу геш-функції [12]. Недоліками існуючих алгоритмів «proof-of-works» є їх обмеження по швидкості і складності здійснення транзакцій, пов'язаних з формуванням «обчислювальних пулів» майнерів, високою ціною майнінгу і принципової неточністю опису складності обчислювальної здатності учасників блокчейна. Крім цього постійне зростання обчислювальної складності майнінгу починає обмежувати децентралізацію.

Аналіз оцінки ступеня впливу стійкості геш-функції на стійкість протоколів криптовалюта розглянемо на прикладі протоколу однієї з основних криптовалюта Bitcoin.

1. Для забезпечення цілісності транзакції в протоколі використовується цифровий підпис результату обчислення геш-коду за методом двійкового дерева Меркле [7]. Таким чином, для верифікації цілісності i-го блоку даних ($i \in \{0,1\}$) на n-му рівні ієрархії необхідна інформація про геш-коди відповідної гілки дерева на $n, n-1, .., 1$ рівнях ієрархії та «корінь геш-дерева», тобто кортеж $\langle (h_{1n}, \ldots, h_{kn}), (h_{1(n-1)}, \ldots, h_{k(n-1)}), \ldots, h_0 \rangle$.

Всього для верифікації блоку даних необхідно не більше $O(\log_2 n)$ операцій та геш-кодів.

2. Алгоритм додавання нового блоку в блокчейн при застосуванні протоколу угоди «proof-of-works».

(a) Нові транзакції отримують всі учасники протоколу угоди.

(b) Всі учасники протоколу угоди генерують і перевіряють правильність побудови дерева Меркля і цифрові підписи кожної транзакції.

(c) Всі учасники протоколу формують «цифрову пломбу» блоку шляхом обчислення геш-коду від блоку транзакцій, а саме: вирішують завдання знайти таке значення $nonce$, щоб значення гешш-коду $h\,(nonce||hcode_{i-1}||block_i) < t$, де $hcode_{i-1}$ – геш-код попереднього блоку, $block_i$ – данні поточного блоку, t – деякий поріг, однаковий для усіх учасників протоколу.

(d) Учасники протоколу перевіряють цифрову пломбу учасника, який сформував її раніше за всіх і при правильності перевірки додають цей блок в ланцюжок блоку транзакцій.

З наведеного опису видно, що завдання формування «цифрової пломби» блоку, від якого залежить стійкість протоколу узгодження транзакцій в Bitcoin, вирішується методом прямого перебору за всіма значеннями величини тільки для сильної геш-функції. Знання клептографічної лазівки дає можливість завжди «випереджати» інших майнерів, що порушує як децентралізацію емісії криптовалюти, так і легітимність будь-якої транзакції – тобто веде до повного злому протоколу. З іншого боку, цілісність блоку транзакцій може бути порушена тільки частково через застосування механізмів геш-покажчиків і цифрового підпису одночасно у великої кількості учасників протоколу.

Насьогодні відомі числені приклади як теоретичних так і практичних клептографічних механізмів, серед яких ГПВЧ з лазівкою Dual EC DRBG [4], модифікації алгоритмів El-Gamal [11], RSA [10]. Проте, наразі мало прикладів клептомеханізмів у функціях гешування, і при цьому до деяких стандартизованих функцій є обгрунтовані питання щодо наявності в них таких механізмів (наприклад, ГОСТ Р34-11-2012 [9]). Актуальною є як задача побудови клептографічної лазівки для задач контролю шифрованого трафіку, так і зворотної задачі виявлення лазівок у наявних геш-функціях. Метою даної роботи є створення методу побудови функції гешування з вбудованим клептографічним механізмом (закладкою) зі збереженням її криптографічних властивостей.

Задачі клептографічних механізмів у функціях гешування

Клептографічні механізми для функцій гешування виглядають дещо різноманітнішими за канали витоку секрету в інших криптопримітивах. Задачі клептомеханізмів у функціях гешування:

– Побудова каналу витоку секрету.

– Створення можливості ефективного пошуку прообразу або другого прообразу для розробника.

– Створення можливості ефективного пошуку слабкої колізії для розробника.

– Можливість ефективного пошуку розробником повідомлень, геш код яких матиме певну структуру.

При цьому є два принципових способи приховати механізм: будувати його на основі асиметричного криптопримітиву (можливість побудови такої геш-функції розглядатиметься далі) або ж розглядати його у моделі Розробника з переважаючими обчислювальними потужностями (як це ймовірно відбулося при розробці шифру DES). Якщо ми припускаємо, що в основі побудови S-блоків DES лежить метод диференційного аналізу, відомий лише розробнику-АНБ, то логічним є припущення, що схожа ситуація може бути і в перших геш функціях. Проте метод диференційного аналізу для геш-функцій має дещо інший сенс. У рамках даної роботи були проведені дослідження, а саме:

1. У роботі [13] було проведено узагальнення підходів диференційного аналізу блокових шифрів та геш функцій, показані відмінності у способах їх побудови. Для геш функцій диференційний шлях є не ймовірнісний розрізнювач частини ключової інформації, як у блокових шифрів, а є набором умов для пари повідомлень, що з високою ймовірністю дозволяють будувати сильну колізію.

2. Далі, у роботі [14] продемонстровано власне метод автоматизованої побудови диференційних шляхів для алгоритмів родини MD4. Розробника, що володіє методом автоматичної побудови диференційних шляхів та має значні обчислювальні ресурси, можливо значно підвищити ефективність побудови сильної колізії та прообразу (користуючись підходом Аокі) відносно екстенсивного перебору.

Схеми побудови функцій гешування

Найпоширенішою схемою побудови функцій гешування є конструкція Меркла-Дамгарда [8], що лягла в основу зокрема таких алгоритмів як MD5, SHA1, SHA256. Вона складається з ланцюжка функцій стиснення, що мають вигляд:

$$F: \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^n, \quad (1)$$

$$\Phi = F(F(F(F(IV, M_0), M_1), \dots), M_k), \quad (2)$$

де $\{M_i\}_{i=0..k}$ – блоки повідомлень бітової довжини t Авторами конструкції [8] було доведено, що при стійкості функції стиснення до колізій геш функція також буде стійкою до колізій. Тож основна увага щодо стійкості надається саме функції стиснення. Основні вразливості цієї схеми:

1. Задача пошуку другого прообразу (сильна колізія) простіша від задачі пошуку першого прообразу для довгих повідомлень.

2. У випадку можливості ефективного пошуку псевдопрообразу, пошук прообразу також спрощується.

3. Атаки доповнення повідомлення: знаючи лише геш повідомлення можна отримати геш даного повідомлення з вибраним доповненням.

Перша та третя атаки послаблюються доданням блоку падінгу з контрольною сумою або лічильником блоків.

В свою чергу, функції стиснення можуть базуватися на різних схемах. Розповсюдженою практикою є використання блокових симетричних шифрів у режимах Девіса-Мейера, Матіса-Мейера-Осеаса, Міягучі-Пренеля [2], блоки повідомлення в такому випадку грають роль ключів шифру. Це дає змогу зводити стійкість пошуку прообразу функції гешування до стійкості відновлення секретного ключа блокового шифру. У роботі [3] розглядаються 64 конструкцій, серед яких 20 є стійкими до пошуку колізій:

Таблиця 1

Конструкції функцій стиснення стійкі до колізій

$f_1(v,m) \equiv E_v(m) \oplus m$	$f_{11}(v,m) \equiv E_{m\oplus v}(m) \oplus v$
$f_2(v,m) \equiv E_v(m \oplus v) \oplus m \oplus v$	$f_{12}(v,m) \equiv E_{m\oplus v}(v) \oplus m$
$f_3(v,m) \equiv E_v(m) \oplus m \oplus v$	$f_{13}(v,m) \equiv E_{m\oplus v}(m) \oplus const$
$f_4(v,m) \equiv E_v(m \oplus v) \oplus m$	$f_{14}(v,m) \equiv E_{m\oplus v}(m) \oplus m \oplus v$
$f_5(v,m) \equiv E_m(v) \oplus v$	$f_{15}(v,m) \equiv E_m(v) \oplus const$
$f_6(v,m) \equiv E_m(m \oplus v) \oplus m \oplus v$	$f_{16}(v,m) \equiv E_{m\oplus v}(v) \oplus const$
$f_7(v,m) \equiv E_m(v) \oplus m \oplus v$	$f_{17}(v,m) \equiv E_m(v) \oplus m$
$f_8(v,m) \equiv E_m(m \oplus v) \oplus v$	$f_{18}(v,m) \equiv E_{m\oplus v}(v) \oplus m \oplus v$
$f_9(v,m) \equiv E_{m\oplus v}(m) \oplus mi$	$f_{19}(v,m) \equiv E_m(m \oplus v) \oplus const$
$f_{10}(v,m) \equiv E_{m\oplus v}(v) \oplus v$	$f_{20}(v,m) \equiv E_m(m \oplus v) \oplus m$

Альтернативою схеми Меркла-Дамгарда для побудови функцій гешування є Sponge-конструкція, що зокрема використовується у алгоритмі гешування SHA-3(Keccak). Основою даної конструкції є бієктивна нелінійна функція:

$$F: V^n \to V^n, \quad (3)$$

де $n = r + c$, r – параметр, що визначає продуктивність обчислення (bitrate), c – параметр, що визначає стійкість функції гешування (capacity).

Перед процесом обчислення гешу повідомлення M розбивається на блоки $m_1, m_2, \dots m_k$ по r бітів кожен. Результатом обчислення буде:

$$H(M) = G(F(F(F(m_1 || \underbrace{0 \dots 0}_{c}) \oplus m_2) \oplus m_3 \dots)), \quad (4)$$

де $G: V^n \to V^r$ – функція редукції, наприклад, просте відкидання молодших c бітів.

Конструкція покликана замінити старішу схему Меркла-Дамгарда оскільки вона позбавлена принципових її вразливостей.

Схеми побудови функцій гешування з вбудованими каналами витоку

Розглянемо функції хешування з каналом витоку, тобто такі функції, в яких розробник за певних умов може практично шукати прообраз геш коду.

Така функція має задовольняти таким неформальним критеріям:

1. Практична стійкість до пошуку прообразу.

2. Практична стійкість до колізій 1-го та 2-го роду.

3. Розробник, знаючи секрет функції, може у деяких випадках шукати прообраз геш коду за практичний час.

4. Функція має опиратися на розповсюджені схеми побудови, тобто бути «схожими» на стандартні.

Важливий той факт, що криптографічні вимоги до стійкості такої функції є значно зниженими порівняно з вимогами на сильні геш-функції, а саме, складність пошуку прообразу 2^n і складність пошуку сильної колізії $2^{n/2}$.

Визначення 1 *(t-практична стійкість до побудови прообразу) Функцію $g: V^m \to V^n$ називатимемо t-практично стійкою до побудови прообразу, якщо $max_{A_t} P\{g(A_t(v)) = v\} < \varepsilon(t)$, де $A_t: V^n \to V^m$ – ймовірнісний алгоритм обмежений часом роботи t, $v \overset{rand}{\in} V^n$, $\varepsilon(t)$ – порогове значення "незначної" ймовірності.*

Визначення 2 *(t-практична стійкість до побудови слабкої колізії) Функцію $g: V^m \to V^n$ називатимемо t-практично стійкою до побудови прообразу, якщо $max_{A_t} P\{g(A_t(u)) = g(u)\} < \varepsilon(t)$, де $A_t: V^m \to$*

$V^m, A_t(u) \neq u$ – *ймовірнісний алгоритм обмежений часом роботи* t, $u \overset{rand}{\in} V^m$, $\varepsilon(t)$ – *порогове значення «незначної» ймовірності.*

Визначення 3 *(t-практична стійкість до побудови сильної колізії) Функцію* $g: V^m \to V^n$ *називатимемо* t*-практично стійкою до побудови прообразу, якщо* $max_{A_t^0, A_t^1} P\{g(A_t^0(r)) = g(A_t^1(r))\} < \varepsilon(t)$, *де* $A_t^i: R \to V^m, \forall r \in R: A_t^0(r) \neq A_t^1(r), A_t^i(r) = A_t^i(r') \Leftrightarrow r = r'$ – *ймовірнісні алгоритми обмежений часом роботи* t, $r \overset{rand}{\in} R$ – *рандомізатор,* $\varepsilon(t)$ – *порогове значення «незначної» ймовірності.*

Інтуїтивно зрозуміло, що одним з варіантів організації каналу витоку є асиметрична криптосистема направленого шифрування. Отже, шукатимемо схему такої функції у вигляді конструкції Меркла-Дамгарда з функцією стиснення, що базується на асиметричному не рандомізованому шифрі, що практично стійка до колізій та пошуку прообразу.

Функція стискання конструкції Меркла-Дамгарда є відображенням $F: \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^n$, $t \geq n$. Таку функцію зручно реалізувати за допомогою блокового шифру, для чого використовуються схеми Девіса-Мейера, Матіса-Мейера-Осеаса, Міягучі-Пренеля та інші. При цьому, в базових шифрах довжина входу та виходу співпадає. Асиметричне шифрування може також бути кандидатом на функцію стискання, оскільки стійке в теоретико-обчислювальному сенсі до пошуку прообразу та до колізій. Проте асиметричні шифри не можна напряму використовувати у схемі Меркла-Дамгарда, оскільки, наприклад, в системі направленого шифрування Ель-Гамаля вихід шифру вдвічі довший за вхід, а для криптосистеми NtruEncrypt – в 4 рази і більше разів.

Розглянемо одну з можливих схем побудови функції стиснення на основі асиметричного шифрування.

Визначення 4 *(базове перетворення на основі DLP) Нехай задана циклічна група* $\langle G, + \rangle$ *з генератором порядку* $g: ord(g) = n$. *Для групи можна визначити асоціативну операцію* $w \cdot a = \underbrace{a + a + \ldots + a}_{w}$.

Базовим перетворенням називатимемо функцію на $T_k: G^k \times Z_n^k \to Z_n^k$:

$$T_k(\vec{v}, \vec{x}) = \vec{\eta} \circ \begin{bmatrix} x_0 & x_1 & \ldots & x_{k-1} \\ x_1 & x_2 & \ldots & x_0 \\ \ldots & \ldots & \ldots & \ldots \\ x_{k-1} & x_1 & \ldots & x_{k-2} \end{bmatrix} \times \begin{bmatrix} v_0 \\ v_1 \\ \ldots \\ v_{k-1} \end{bmatrix},$$

де $\vec{v} \in G^k, \vec{x} = Z_n^k, \vec{\eta} \circ = \langle \underbrace{\eta \circ, \ldots, \eta \circ}_{k} \rangle$, $\eta: G \to Z_n$ – бієктивне відображення, причому η та η^{-1} можуть бути реалізовані ефективним алгоритмом.

Лема 1 *(умови ін'єктивності функції* $T_k(\vec{v}, \cdot)$*)*

Якщо для заданого $\vec{v} = \langle c_0 \cdot g, c_1 \cdot g, \ldots, c_{k-1} \cdot g \rangle \in G^k, T_k(\vec{v}, \vec{x}) = \vec{h}$ і $rank(a_{i,j}) = k$ над Z_n, $a_{i,j} = c_{j-imodk}$, то $\nexists \vec{x}' \neq \vec{x}: T_k(\vec{v}, \vec{x}') = \vec{h}$.

Доведення.

$$T_k(\vec{v}, \vec{x}) = \begin{bmatrix} \eta(\sum_{i=0..k-1} \{x_i * c_i\} \cdot g) \\ \eta(\sum_{i=0..k-1} \{x_{i+1modk} * c_i\} \cdot g) \\ \ldots \\ \eta(\sum_{i=0..k-1} \{x_{i+k-1modk} * c_i\} \cdot g) \end{bmatrix} = \begin{bmatrix} h_0 \\ h_1 \\ \ldots \\ h_{k-1} \end{bmatrix},$$

де операції $\sum$ та $*$ визначені над Z_n.

Звідси випливає:

$$\begin{bmatrix} c_0 & c_1 & \ldots & c_{k-1} \\ c_{k-1} & c_0 & \ldots & c_{k-2} \\ \ldots & \ldots & \ldots & \ldots \\ c_1 & c_2 & \ldots & c_0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \ldots \\ x_{k-1} \end{bmatrix} = \begin{bmatrix} \tilde{h}_0 \\ \tilde{h}_1 \\ \ldots \\ \tilde{h}_{k-1} \end{bmatrix}, \quad (5)$$

де $\tilde{h}_i: \tilde{h}_i \cdot g = \eta^{-1}(h_i)$ (для $\tilde{h}_i \in Z_n$ таке представлення однозначне).

Оскільки за умовою ранг матриці дорівнює k, отримана система лінійних рівнянь або має один розв'язок або несумісна, і оскільки заданий вектор $\vec{x}$ задовольняє систему за побудовою, то він і є її єдиним розв'язком.

Доведемо еквівалентність задачі отримання вектору Z_n^k задачі дискретного логарифмування в моделі теоретичної складності.

Теорема 1 *(стійкість функції* T_k*)*

Задача пошуку довільної компоненти x_r вектора $\vec{x} \in Z_n^k$ для $\vec{v} \in G^k$, що задовольняють умові з леми 1, та $\vec{h} = T_k(\vec{v}, \vec{h})$ поліноміально зводиться до задачі дискретного логарифмування в групі $\langle G, + \rangle$ відносно асоціативної операції "$\cdot$".

Доведення. Нехай ми маємо два алгоритми:

1. A_k^r такий, що $\forall \vec{x} \in Z_n^k, \forall \vec{v} \in G^k, \vec{h} = T_k(\vec{v}, \vec{x}): A_k^r(\vec{v}, \vec{h}) = x_r$.

2. A такий, що $\forall x \in Z_n, \forall v \in G, h = x \cdot v: A(v, h) = x$.

Доведемо, що алгоритми A та A_k^r є асимптотично еквівалентні.

Спершу зведемо алгоритм A до A_k^r.

Нехай задані $v' \in G$ та $h' = x \cdot v$ для деякого $x \in Z_n$. Визначимо вектор $\vec{v}$ так, що $v_i = c_i \cdot v', i = 0..k-1$, коефіцієнти обираються таким чином, щоб матриця $(a_{i,j} = c_{j-imodk})$ мала ранг k. Також визначимо вектор $\vec{h}$, де $h_i = \eta(c_{r+i} \cdot h' + \sum_{j=0..k-1, j\neq r+i} \{c_{j+i}\} \cdot v')$ Згідно з визначенням 4, одне з можливих виходів алгоритму A_k^r буде $A_k^r(\vec{v}, \vec{h}) = x_r : x_r \cdot v' = h'$, причому, якщо існує розв'язок, то він єдиний, згідно з лемою 1, тож $A(v, h) = A_k^r(\vec{v}, \vec{h})$.

Тепер зведемо алгоритм A_k до A.

Нехай задані $\vec{v} \in G^k$ та $\vec{h} = T_k(\vec{v}, \vec{x})$ для деякого $\vec{x} \in Z_n^k$. З доведення леми 1 видно, що для отримання $\vec{x}$ достатньо розв'язати систему лінійних рівнянь 5 над Z_n, для побудови якої необхідно отримати значення $\{c_i\}_i = 0..k-1$ та $\{\tilde{h}_i\}_i = 0..k-1$. Ці значеня можна обчислити як $c_i = A(g, v_i)$, $\tilde{h}_i = A(g, h_i)$, тобто необхідно $2k$ разів запустити алгоритм A. Отже час роботи алгоритма A_k буде $\tau_k \leq 2k \cdot \tau + \xi$, де τ – час роботи алгоритма A, ξ – час розв'язання системи рівнянь 5 над Z_n. Якщо вважати, що час виконання операцій $(*, +)$ над Z_n^k нехтовно малий відносно τ, а параметр k – константний, то $\tau_k = O(\tau)$.

Тепер перейдемо власне до побудови функції стиснення з лазівкою.

Визначення 5 *(Функція стиснення на базі перетворення T_k)*

Функцією стиснення на базі перетворення T_k називатимемо функцію:

$$F(\vec{v}, \vec{m}) = T_k(\vec{v}, \vec{m}) \boxplus \vec{m}, \qquad (6)$$

де $\vec{v} \in G^k, \vec{m} \in Z_n^k$, операція '$\boxplus$' – покомпонентне додавання у Z_n.

Теорема 2 *Для $\forall k \geq 2, G, Z_n$ конструкція 5 з базовим перетворенням $T_k : G \times Z_n \to Z_n$: дозволяє непомітно передати розробнику, який обрав та зафіксував $\vec{v}$, фрагменту m_r блоку повідомлення $\vec{m}$ за умови знання $\{m_i\}, i = 0..k-1, i \neq r$ та розкладу $\vec{v} = \langle s_0 \cdot g, \ldots, s_{k-1} \cdot g \rangle$. При цьому складність задачі відновлення m_r зводиться до задачі дискретного логарифмування над $\langle G, + \rangle$.*

Доведення. Розглянемо випадок відомого гешу $\vec{h} = T_k(\vec{v}, \vec{M}) \boxplus \vec{M}$ для одноблокового повідомлення $\vec{M} = \langle m_0, \ldots, m_k \rangle, m_i \in Z_n$, при чому компонент m_r невідома розробнику, а решта компонентів – відомі. Для відновлення m_r розробник виконує такі кроки:

1. На етапі впровадження алгоритму розробник задає $\vec{v}$ у вигляді $v_i = s_i \cdot g, i = 0..k-1$, де $s_i \in Z_n$ – секретні (відомі тільки розробнику) параметри, вибрані випадково з рівномірного над Z_n розподілу. Розробник також може визначити коефіцієнти $s_{i,j} : v_i = s_{i,j} \cdot v_j, s_{i,j} = s_i * s_j^{-1} \in Z_n$.

2. Обирає довільне $w \neq r$ та відновлює m_r з системи з 2-х рівнянь, отриманої з визначень 4 та 5:

$$\begin{bmatrix} \eta(\sum_{i=0..k-1} \{x_{i+rmodk} * c_i\} \cdot g) \boxplus m_r \\ \eta(\sum_{i=0..k-1} \{x_{i+wmodk} * c_i\} \cdot g) \boxplus m_w \end{bmatrix} = \begin{bmatrix} h_r \\ h_w \end{bmatrix}.$$

3. Для цього обчислює $\alpha = \eta^{-1}(h_w \boxminus m_w) - \sum_{i=0..k-1, i\neq r} \{x_{i+wmodk} * c_i\} \cdot g = m_r \cdot v_w$.

4. Обчислює $\beta = s_{r,w} \cdot \alpha = m_r \cdot (s_{r,w} \cdot v_w) = m_r \cdot v_r$.

5. Відновлює частину повідомлення m_r: $m_r = h_r \boxminus \eta(\beta + \sum_{i=0..k-1, i\neq r} \{x_{i+rmodk} * c_i\} \cdot g)$.

При цьому, для всіх інших учасників протоколу, за відсутності інформації про секретні параметри $s_0, \ldots, s_{k-1}$, складність відновлення повідомлення m_r зводиться до задачі дискретного логарифмування над $\langle G, + \rangle$ відносно асоціативної операції «·» згідно з теоремою 1.

В результаті дослідження можливості побудови геш-функції з лазівкою розроблений метод побудови такої функції стиснення, яка дозволяє Розробнику ефективно відновлювати повідомлення за умови, що воно частково відоме (наприклад, як у випадку гешування короткого паролю – решта блоку відома та нульова).

Вихідні дані: циклічна група $\langle G, + \rangle$ з генератором g, параметр розміру блоку k, секрет розробника $\vec{s} \in Z_n^k$.

Результат: функція стиснення $F(\vec{v}, \vec{m})$, що дозволяє, знаючи секрет $\vec{s} \in Z_n$ та частини блоку повідомлення $\{m_i\}_{i\neq r}$ відновити частину m_r.

Кроки методу:

1. Формування стартового вектора геш функції: $\vec{v} = \langle s_0 \cdot g, .., s_{k-1} \cdot g \rangle$.

2. Побудова перетворення T_k (визначення 4) на основі параметру k.

3. Побудова функції стиснення F (визначення 5) на основі перетворення T_k та стартового вектора $\vec{v}$.

Згідно з теоремою 2 дана конструкція дозволяє непомітно передати розробнику, знаючи секрет $\vec{s} \in Z_n$ та частини блоку повідомлення $\{m_i\}_{i\neq r}$ відновити частину m_r.

Висновки

В статті аналізуються особливості побудови клептографічних механізмів для функцій гешування. На відміну від симетричних шифрів, задачі клептографії для функцій гешування є дещо ширшими, оскільки додатково включають можливості побудови лазівок, що спрощують для Розробника пошук колізій першого та другого роду, а також лазівки, що змінюють криптографічні властивості функції гешування в специфічних моделях використання, наприклад, у протоколах консенсусу технології блокчейн.

Однією з проблем побудови клептографічних механізмів загалом є вимога «подібності» примітиву із закладкою до примітивів, що зазвичай використовуються. Поняття «подібності» є не надто формальним і може трактуватися різними дослідниками по-різному. У даній роботі, підставою такої «подібності» є побудова геш функції з використанням широко розповсюдженої схеми Меркла-Дамгарда, а також використання у функції стиснення однієї із загальноприйнятих схем побудови безколізійного перетворення на основі симетричного шифру.

В результаті досліджень був розроблений метод побудови геш функції з такими особливостями:

1. Геш функція базується на розповсюдженій конструкції Меркла-Дамгарда.

2. Функція стиснення використовує одну із загальноприйнятих схем побудови функцій стиснення без колізій на основі симетричного шифру.

3. Симетричним шифром обрано одностороннє перетворення, сформульована та доведена теорема про стійкість даного перетворення.

4. Для геш коду $h \in Z_n^k$ повідомлення $\vec{x} \in Z_n^k$, умови знання секрету Розробника та компонентів повідомлення $\{x_i\}_{i=0..k-1} \backslash \{x_j\}$, можна ефективно відновити невідомий компонент x_j.

5. Сформульована та доведена теорема про неможливість відновлення будь-якої невідомої частини повідомлення за геш кодом без знання секрету Розробника.

ЛІТЕРАТУРА

[1]. E. Barker, J. Kelsey, "Sp 800-90a. recommendation for random number generation using deterministic random bit generators", *Technical report*, Gaithersburg, MD, United States, 2012.

[2]. J. Black, P. Rogaway, T. Shrimpton, "Black-box analysis of the block-cipher-based hash-function constructions from pgv", *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, vol. 2442, *Lecture Notes in Computer Science*, pp. 320-335, 2002.

[3]. J. Black, P. Rogaway, T. Shrimpton, "Black-box analysis of the block-cipher-based hash-function constructions from pgv", Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, pages 320–335, Berlin, Heidelberg, 2002.

[4]. R. Daniel, L. Brown, G. Kristian, "A security analysis of the nist sp 800-90 elliptic curve random number generator", Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pp. 466-481, 2007.

[5]. J. Turner, C. Gutierrez, *The Keyed-Hash Message Authentication Code (HMAC)*, National Institute of Standards and Technology, Dec 2010.

[6]. W. Burr Meltem Turan, E. Barker, *Recommendation for Password-Based Key Derivation*, National Institute of Standards and Technology, July 2008.

[7]. R. Charles Merkle, *Secrecy, Authentication, and Public Key Systems,* PhD thesis, Stanford, CA, USA, 1979. AAI8001972.

[8]. A. Degtyarev V. Dolmatov, *GOST R 34.11-2012: Hash Function*, *RFC 6986 (Informational)*, August 2013.

[9]. A. Young, M. Yung, *The Dark Side of "Black-Box" Cryptography or: Should We Trust Capstone*, pp. 89-103, 1996.

[10]. A. Young, M. Yung, *Kleptography: Using Cryptography Against Cryptography*, pp. 62–74, 1997.

[11]. F. Zhang, I. Eyal, R. Escriva, A. Juels, R. Renesse, "Resource-efficient mining for blockchains" *Cryptology ePrint Archive*, Report 2017/179, 2017. http://eprint.iacr.org/2017/179.

[12]. Б. Коваленко, А. Кудін, "Диференційний аналіз функцій хешування та блокових шифрів: узагальнений підхід", *Безпека інформації*, № 21(2). С. 159-164, 2015.

[13]. Б. Коваленко, А. Кудін. "Алгоритмічні аспекти пошуку прообразів геш-функцій на прикладі md5", *Захист інформації*, № 17(3). С. 205-210, 2015.

DEVELOPMENT OF KLEPTOGRAPHIC MECHANISMS INTO HASH FUNCTIONS

This research belongs to kleptographic problems of hash functions. Relevance of the research follows from importance of hash functions in hybrid cryptosystem and also from existence of kleptographic attack vectors on such systems. Currently, there are numerous results at kleptography in symmetric ciphers and asymmetric crypto protocols which demonstrate different aspects of kleptographic trapdoor implementation, however, a few of them highlight kleptographic problems of hash functions. Insufficiency of researches in hash kleptography problems leads

to kleptography related risks in hash function at designing and standardization stage. In this article, we analyse ways to develop hash functions with kleptographic trapdoor. One of informal requirements for such functions is "proximity" to famous and common used constructions, i.e. it must be based on common schemes, that are used for development of well known hash functions. In current paper, it's suggested to build trapdoored hash function based on Merkle-Damgard scheme, which is the base of numerous of wide spread hash function. As compression function we choose one of the well known compression function schemes which are based on block ciphers and are proved to be collision resistant (like as Davice-Mayer or Miyaguchi-Preneel constructions). Instead of block ciphers in compression function we use special transformation based of Discrete Logarithm Problem and prove collision resistance preserving. The final result of the research is hash function with kleptographic trapdoor which allows developer effectively recover part of message (till 50\%) using knowledge of hash digest and secret in the kleptographi trapdoor design. In the same time, this function is still secure for other users who don't own design's secret.
Keywords: hash function, kleptography, subliminal channel, Merkle-Damgard scheme, discrete logarithm problem.

ПОСТРОЕНИЕ КЛЕПТОГРАФИЧЕСКИХ МЕХАНИЗМОВ В ФУНКЦИЯХ ХЕШИРОВАНИЯ

Работа посвящена клептографическим проблемам функций хеширования. Актуальность данной работы вытекает из ключевой роли функций хеширования в современных гибридных криптосистемах и из факта существования клептографического вектора атак на такие системы. Сейчас, несмотря на то, что существует ряд работ, в которых исследованы клептографические возможности симметричных шифров и асимметричных криптографических протоколов, крайне мало исследований посвящено клептографическим проблемам функций хеширования. Недостаточность исследований клептографических возможностей хеш функций приводит к ризкам наличия клептографических механизмов в функциях хеширования, встроенных на этапе проектирования и стандартизации. В данной работе исследуются возможности построения функции хеширования с клептографическим механизмом. Одной из неформальных требований к таким функциям является требование «сходства» ее к известным функций хеширования, то есть должна базироваться на известных общих схемах хеш функций. В данной работе для реализации функции хеширования с лазейкой предлагается использовать схему Меркла-Дамгарда, что является основой многих известных функций хеширования, а функцией сжатия выбрана одна из общеизвестных конструкций построения функции сжатия на основе блочного шифра, что доказано устойчивыми к построению коллизий. Вместо блочного шифра в функции сжатия используется преобразование специального вида, а также приходится сохранение устойчивости к коллизиям с использованием данного преобразования. Результатом исследований является хеш-функция с клептографическим механизмом, что позволяет разработчику эффективно восстанавливать часть (до 50%) сообщения на основе хеш-кода и знания секрета в структуре клептографического механизма. В то же время, функция остается криптографически стойкой для других пользователей, не владеющих секретом.
Ключевые слова: хеш-функция, клептография, клептографический механизм, конструкция Меркла-Дамгарда, задача дискретного логарифмирования.

Кудін Антон Михайлович, д.т.н., с.н.с., професор кафедри математичних засобів захисту інформації, Національний технічний університет України «КПІ».
E-mail: pplayshner@gmail.com.
Orcid ID: 0000-0002-3966-6489.
Кудин Антон Михайлович, д.т.н., с.н.с., профессор кафедры математических средств защиты информации, Национальный технический университет Украины «КПИ».
Kudin Anton, Dr. Eng (Information security), Senior Researcher, professor at Mathematical Methods of Information Security department, National Technical Institute of Ukraine "KPI".

Коваленко Богдан Анатолійович, ТОВ "Globallogic Ukraine", інженер інформаційної безпеки.
E-mail: animantbk@gmail.com.
Orcid ID: 0000-0001-7802-0587.
Коваленко Богдан Анатольевич, ООО "Globallogic Ukraine", инженер информационной безопасности.
Kovalenko Bohdan, Globallogic Ukraine LLC, information security engineer.