Національний авіаційний університет
Міністерство освіти і науки України

Кваліфікаційна наукова праця
на правах рукопису

КОВАЛЕНКО Богдан Анатолійович

УДК 004.056.55

ДИСЕРТАЦІЯ

Методи побудови та оцінки стійкості клептографічних механізмів у гібридних криптосистемах

Спеціальність 05.13.21 – Системи захисту інформації
Технічні науки

Подається на здобуття наукового ступеня кандидата технічних наук

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело
_____ Коваленко Б.А.

Науковий керівник: Кудін Антон Михайлович, доктор технічних наук, старший науковий співробітник, професор кафедри

Київ 2020

Анотація

Коваленко Б.А. Методи побудови та оцінки стійкості клептографічних механізмів у гібридних криптосистемах. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук (доктора філософії) за спеціальністю 05.13.21 «Системи захисту інформації». – Національний авіаційний університет, Міністерство освіти і науки України, Київ, 2020.

У дисертації розв'язано актуальну наукову задачу розробки методу побудови криптосистем з доведеною відсутністю клептографічної модифікації, а також оцінці клептографічних ризиків криптопримітивів на етапі їх стандартизації і впровадження.

Метою робити є підвищення рівня захищеності гібридних криптосистем шляхом зменшення клептографічних ризиків через запропоновані методи побудови протоколів з доведеною відсутністю клептографічних каналів витоку секрету, методи оцінки ризиків наявності клептографічних лазівок у криптопримітивах та методів синтезу нових клептографічних механізмів з метою вивчення нових підходів до виявлення та протидії їм.

В роботі удосконалено класифікацію вразливостей прикладних криптосистем Б. Шнаєра в результаті чого отримана класифікація клептографічних атак за такими критеріями: за закритістю реалізації, за наслідками (для імплементації) аналізу, за рівнем побудови (лазівки в нових алгоритмах та модифікація реалізацій), за способом впровадження.

У дисертаційній роботі вперше було запропоновано математичну модель для протоколів типу «запит-відповідь» з клептографічним каналом витоку секрету, що дозволяє формалізувати певний клас криптографічних протоколів. Протокол подається у вигляді кортежу $\langle D_t, V, U, A_t, R_t^\omega, A_t^\omega \rangle$, де V та U – відповідно простори запитів та відповідей оракула, A_t – алгоритм оракула, D_t – алгоритм перевірки оракула, R_t^ω, A_t^ω – алгоритми відновлення та витоку секрету відповідно.

Базуючись на запропонованій моделі були вперше отримані достатні умови неможливості клептографічної модифікації протоколу, що дозволило побудувати модифікації криптографічних протоколів генерації nonce та

узгодження ключа Діффі-Хеллмана, що стійкі до атаки клептографічної модифікації SETUP. Ідея таких протоколів полягає у тому, що сторони комунікації не використовують власні джерела випадковості. Натомість, у випадку необхідності генерації випадкової послідовності, оракул обирає публічне унікальне значення з низькою ентропією (наприклад, лічильник або мітка часу) та перетворює його на випадкове значення використовуючи цифровий підпис без рандомізації (наприклад, BLS), а інша сторона перевіряє ці підписи. У випадку порушення автентичності, оракул підозрюється у зловмисній модифікації.

Також, вперше розроблено метод побудови геш функції з клептографічною лазівкою, що дозволяє розробнику, який задав стартовий вектор IV, ефективно обчислити невідому частину повідомлення за умови наявності геш коду та відомих інших частин даного повідомлення. Функція побудована на основі широкорозповсюдженої конструкції Меркла-Дамгарда та функції стиснення на базі блокового шифру. Отримана лазівка дозволяє Розробнику відновлювати до половини повідомлення, що може застосовуватися, наприклад, для відновлення коротких паролей із геш коду.

Одним з ключових результатів є вперше запропонована метрика «клептографічного потенціалу», яка дозволяє оцінити максимально можливий розмір секрету лазівки Розробника і, таким чином, оцінити клептографічні ризики криптографічного примітиву чи системи. Ідея підходу полягає у тому, що ми оцінюємо кількість інформації у структурі криптографічного примітиву, що і є верхньою межею розміру секрету лазівки Розробника. При цьому, у випадку, якщо клептографічний потенціал є високим, це означає не стільки ймовірність наявності лазівки, скільки відсутність аргументів проти її наявності. Але у випадку, коли потенціал низький можна стверджувати, що клептографічна лазівка відсутня. Проте проблема полягає в тому, що обчислити потенціал на практиці часто неможливо. Тому також вводиться поняття «клептографічної надлишковості», що є нижньою оцінкою клептографічного потенціалу. Для обчислення клептографічної надлишковості спершу задається клас еквівалентних примітивів і на основі його потужності обчислюється надлишковість.

Користуючись запропонованою метрикою, були отримані оцінки

клептографічної надлишковості для алгоритмів AES, SHA-256, ГОСТ-28147-89, російського шифру «Кузнєчік» та геш функції «Стрибог», українського шифру «Калина». Виявилося, що серед досліджених алгоритмів, найменшу надлишковість має AES – 32 біти, що говорить за малу ймовірність наявності лазівки на противагу російського стандарту гешування «Стрибог» – близько 12 кбіт надлишковості, яка означає його досить високу клептографічну ризиковість.

Отримано подальший розвиток методу Пренеля для побудови фейстелівських шифрів з диференційними шляхами високої ймовірності. А саме, був запропонований метод побудови геш функції на основі методу Пренеля, що дозволяє Розробнику отримати перевагу у побудові сильної колізії, що може використовуватися у специфічних режимах роботи геш функції, наприклад, протоколі консенсусу типу «Proof-of-Work» технології блокчейн. А саме, якщо Розробник може змінити послідовність бітів $\{i_0, i_1, ...i_n\}$ за допомогою набору диференційних шляхів повідомлення з середньою ймовірністю p для кожного, він отримує перевагу у пошуку повідомлення, геш код якого містить i_n нулів на початку.

Отримані наукові та практичні результати впроваджені при виконанні науково-дослідних проектів «Корифена» (номер держреєстрації 0118U001653) на замовлення Служби зовнішньої розвідки України, «Родоліт» (номер держреєстрації 0111U007473) на замовлення Служби безпеки України та Національного банку України, що підкріплено двома актами впровадження.

Ключові слова: клептографія, протокол типу «запит-відповідь», практична стійкість, SETUP, канал непомітного витоку секрету, обчислювальна стійкість.

Наукові праці, в яких опубліковано основні наукові результати дисертації:

1. Коваленко Б.А., Кудін А.М., Швидченко І.В., «Технологія блокчейн: питання аналізу та синтезу», Кібернетика та системний аналіз, т.55, №3, с.165-173, 2019.

2. Коваленко Б.А., Кудін А.М., «Диференційний аналіз функцій хешування та блокових шифрів: узагальнений підхід», Безпека інформації, т.21, №2, с.159–164,2015.

3. Коваленко Б.А., Кудін А.М., «Алгоритмічні аспекти пошуку прообразів геш-функцій на прикладі MD5», Захист інформації, т.17, №3, с.205-210, 2015.

4. Коваленко Б.А., Кудін А.М. «Побудова клептографічних механізмів у функціях гешування», Захист інформації, т.21, №2, с.121-128, 2019.

5. Кудін А.М., Ковальчук Л.В., Коваленко Б.А., «Теоретичні засади та застосування блокчейн-технологій: імплементація нових протоколів консенсусу та краудсорсінг обчислень», Математичне та комп'ютерне моделювання, №19, с.62-68, 2019.

6. Коваленко Б.А., «Побудова криптографічних протоколів вільних від клептографічних модифікацій», Безпека інформації, т.25, №2, с.122-126, 2019.

7. Kovalenko B. Kleptography trapdoor free cryptographic protocols [Електронний ресурс] / B. Kovalenko, A. Kudin // Cryptology ePrint Archive. – 2018. – Режим доступу до ресурсу: https://eprint.iacr.org/2018/989.

8. Коваленко, Б. А. Побудова клептографічних модифікацій функцій хешування / Б. А. Коваленко, А. М. Кудін // Матеріали XIII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики», м. Київ, 21-23 травня 2015. – Київ : НТУУ «КПІ». – 2015. – с. 122-124.

9. Коваленко Б.А., Кудін А.М., «Диференційний аналіз функцій хешування та блокових шифрів: узагальнений підхід»: матеріали п'ятої міжнар. наук.-практ. конф., 27-29 травня 2015 /М-во освіти і науки України, Прикарпатський національ. універ. ім. Василя Стефаника, Вінницький націон. тех. універ.– Івано-Франківськ: с.154-157.

10. Коваленко Б.А., «Побудова криптографічних примітивів без клептографічних лазівок»: зб. наук. праць за матеріалами міжнар. наук.-практ. конф., 29 липня 2019 р./ ГО «Європейська наукова платформа»'. – Лондон: т.3, с.97-102.

11. Коваленко Б.А., Кудін А.М. «Побудова криптографічних протоколів, захищених від побудови SETUP»: збір. тез наук. допов. конф. «Стан та удосконалення безпеки інформаційно-телекомунікаційних систем» 19-21 червня 2019 р./М-во освіти і науки України, ДССЗІ України [та ін.].–Миколаїв: с.23-24.

12. Kovalenko B., Kudin A., «Evaluation and minimization of kleptography risks in cryptographic algorithms»: збір. тез наук. допов. конф. «20th Central European Conference on Cryptology» 24-26 червня 2020 р. –Загреб: с.37-38.

Abstract

Kovalenko B. Methods of kleptographic mechanisms implementation and security estimation in hybrid cryptosystems. – Manuscript.

PhD dissertation, specialty 05.13.21 – Information Security Systems. – National Aviation University, Kyiv, 2020

There is actual problem solved in the dissertation, namely the problem of creation approaches and methods for provenkleptographic modification free cryptosystem and estimation of kleptographic risks during standardisation and deployment stages of algorithm's life cycle.

The goal of research is increasing of security level for hybrid cryptosystems via reducing of kleptographic risks using specific approaches for development of protocols with proven impossibility of kleptographic subliminal channels, methods for estimation of kleptographic risks belong to kleptographic trapdoors existence, and also kleptographic synthesis methods to investigate new countermeasures and detection approaches.

Is this research Shannon's practical vulnerabilities classification was improved. As a result, it was suggested new kleptographic trapdoors classification which is based on such criteria: by proprietary (open or proprietary implementations), by destructivity (destructive or non-destructive for target device or implementation), by implementation level (specially crafted algorithms or modification of implementations), type of deployment (modification of existent cryptosystems, public distribution of kleptographic modification, distribution of proprietary cryptosystems inside hardware modules, lobby in standardization processes).

In the dissertation, formal model for "challenge-response" protocols with kleptographic trapdoor has been first suggested. It allows to define a class of cryptographic protocols that may be represent as tuple $\langle D_t, V, U, A_t, R_t^\omega, A_t^\omega \rangle$, where V and U – spaces of requests and responses of oracle, A_t – oracle algorithm, D_t – oracle verifier's algorithm, R_t^ω, A_t^ω – algorithms for secret recovering and secret hidden transmission.

Based on the suggested model sufficient conditions for impossibility of kleptographic modification has been obtained, that allows to craft improvements for cryptographic protocols (nonce request and Diffie-Hellman key agreement), that are resistance against SETUP attacks. The idea of the approach – communication sides don't use their own sources of randomness. Instead, if it's necessary to generate random value, the oracle takes public unique value with low entropy and generates it's non-randomized digital signature (e.g., BLS), and another communicator verifies these signatures. If authenticity is violated, the oracle is suspected to be modified.

Also, methods for hash function with kleptographic trapdoor have been suggested. That allows Developer, that initializes initial vector IV, efficiently calculate unknown part of message if he knows hash digest and other message parts. The function is based on widely spread Merkle-Damgard construction and compression function that is based on symmetric cipher. The backdoor allows Developer to recover at most half of message that may be used for password recovering.

One of the key result of research is first suggested metric called "kleptographic potential", that allows to estimate maximal size of trapdoor's secret and thus to estimate kleptographic risks of cryptographic primitive or system. The idea is evaluation of information amount inside the structure of cryptographic primitive that is exactly higher limit of of Developer's secret size. Moreover, if kleptographic potential is high enough, it doesn't mean that the trapdoor is available, rather that there are no arguments against it. However, if potential is low, kleptographic trapdoor doesn't exist. But there is a problem – it's difficult enough to estimate kleptographic potential directly. That's why, it was suggested term "kleptographic redundancy" that is lower estimate of kleptographic potential. To calculate kleptographic redundancy first one needs to define class of equivalent algorithms

and using its cardinality one can calculate redundancy.

Using suggested metric, estimation for sequence of cryptographic primitives has been obtained: for symmetric encryption standard AES, hash function SHA-256, soviet encryption standard GOST-28147-89, russian symmetric cipher "Kuznechik" and hash function "Stribog", Ukrainian symmetric cipher "Kalyna". It was discovered, the lowest redundancy is for AES cipher is about 32 bits, which shows that the trapdoor existence risks are low. But, for example, russian hash function standard "Stribog" has the highest redundancy – 12 kbits so kleptographic risks are high.

Also, Preenel method of kleptographic trapdoor in Feistel ciphers was further developed. Namely, based on this method we suggested method of hash function crafting that allows Developer to search for strong collision efficiently. It gives him advantages in specific hashing modes, e.g. in Proof-of-Work blockchain consensus protocols, that requires member of protocol to search for a message that gives hash code with some amount of zeros in a prefix. If Developer can affect bit sequence $\{i_0, i_1, ...i_n\}$ in a hash code using is set of high-probability differentials with average probability p, he obtains advantage in search of message that gives hash digest with i_n prefix zeros. Thus, he may compete with member who have much higher computational resources.

Obtained theoretic and practical results are implemented during research projects "Korifena" (act #0118U001653) for External Intelligence Service of Ukraine, "Rodolith" (act #0118U001653) for Security Service of Ukraine and for National Bank of Ukraine, that is proved by two acts.

Key words: kleptography, "challenge-response" protocol, practical resistance, SETUP, subliminal channel, computational complexity

Scientific papers in which the basic thesis scientific results are published:

1. Коваленко Б.А., Кудін А.М., Швидченко І.В., «Технологія блокчейн: питання аналізу та синтезу», Кібернетика та системний аналіз, т.55, №3, с.165-173, 2019.

2. Коваленко Б.А., Кудін А.М., «Диференційний аналіз функцій хешування та блокових шифрів: узагальнений підхід», Безпека інформації, т.21, №2, с.159–164,2015.

3. Коваленко Б.А., Кудін А.М., «Алгоритмічні аспекти пошуку прообразів геш-функцій на прикладі MD5», Захист інформації, т.17, №3, с.205-210, 2015.

4. Коваленко Б.А., Кудін А.М. «Побудова клептографічних механізмів у функціях гешування», Захист інформації, т.21, №2, с.121-128, 2019.

5. Кудін А.М., Ковальчук Л.В., Коваленко Б.А., «Теоретичні засади та застосування блокчейн-технологій: імплементація нових протоколів консенсусу та краудсорсінг обчислень», Математичне та комп'ютерне моделювання, №19, с.62-68, 2019.

6. Коваленко Б.А., «Побудова криптографічних протоколів вільних від клептографічних модифікацій», Безпека інформації, т.25, №2, с.122-126, 2019.

7. Kovalenko B. Kleptography trapdoor free cryptographic protocols [Електронний ресурс] / B. Kovalenko, A. Kudin // Cryptology ePrint Archive. – 2018. – Режим доступу до ресурсу: https://eprint.iacr.org/2018/989.

8. Коваленко, Б. А. Побудова клептографічних модифікацій функцій хешування / Б. А. Коваленко, А. М. Кудін // Матеріали XIII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики», м. Київ, 21-23 травня 2015. – Київ : НТУУ «КПІ». – 2015. – с. 122-124.

9. Коваленко Б.А., Кудін А.М., «Диференційний аналіз функцій хешування та блокових шифрів: узагальнений підхід»: матеріали п'ятої міжнар. наук.-практ. конф., 27-29 травня 2015 /М-во освіти і науки України, Прикарпатський національ. універ. ім. Василя Стефаника, Вінницький націон. тех. універ.– Івано-Франківськ: с.154-157.

10. Коваленко Б.А., «Побудова криптографічних примітивів без клептографічних лазівок»: зб. наук. праць за матеріалами міжнар. наук.-практ. конф., 29 липня 2019 р./ ГО «Європейська наукова платформа»'. – Лондон: т.3, с.97-102.

11. Коваленко Б.А., Кудін А.М. «Побудова криптографічних протоколів, захищених від побудови SETUP»: збір. тез наук. допов. конф. «Стан та удосконалення безпеки інформаційно-телекомунікаційних систем» 19-21 червня 2019 р./М-во освіти і науки України, ДССЗІ України [та ін.].–Миколаїв: с.23-24.

12. Kovalenko B., Kudin A., «Evaluation and minimization of kleptography risks in cryptographic algorithms»: збір. тез наук. допов. конф. «20th Central European Conference on Cryptology» 24-26 червня 2020 р. –Загреб: с.37-38.

Зміст

Перелік умовних позначень

АНФ – алгебраїчна нормальна форма

ГПВЧ – Генератор Псевдовипадкових Чисел

КМ – клаптографічний механізм

КСЗІ – Комплексна Система Захисту Інформації

AES – Advanced Encryption Standard

BEAST – Browser Exploit Against SSL/TLS

CBC – Cipher-Block Chaining [mode]

DLP – Discrete Logarithm Problem

HMAC – Hash-based message authentication code

IoT – Internet of Things

KDF – Key Derivation Function

PoW – Proof of Work

SETUP – Secretly Embedded Trapdoor with Universal Protection

SSL – Secure Sockets Layer

XSS – Cross Site Scripting

Вступ

Актуальність теми З огляду на широке розповсюдження використання гібридних криптосистем у інформаційно телекомунікаційних системах, особливої гостроти набувають задачі захисту даних криптосистем на всіх рівнях життєвого циклу: на етапі проектування, реалізації, розгортання та використання. Однією з характеристик сучасних криптосистем є розповсюдження їх використання в тому числі і в пристроях, які не відповідають стандартам з безпеки реалізації, що породжує нові вектори атак, зокрема клептографічних атак, наприклад, з модифікацією реалізації криптосистеми на кінцевому пристрої. Такі типи атак є особливо небезпечними, враховуючи той факт, що жертва зловмисника, будучи частиною певної захищеної системи (електронного документообігу, платіжної системи, секретного зв'язку тощо), може нести загрозу також для не скомпрометованих учасників системи (наприклад, витік спільних секретних даних). Можливими напрямками вирішення цієї проблеми є побудова криптосистем, стійких до різних типів клептографічних атак; розробка критеріїв наявності (відсутності) клептографічних закладок у примітивах; синтез криптографічних систем та криптопримітивів з закладками з метою розширення множини шаблонів проектування закладок для дослідження методів виявлення та протидії.

Клептографія вивчає методи синтезу та аналізу каналів прихованого витоку секрету (embedded trapdoor, subliminal channel) на базі криптосистем, що дозволяють особі, що впровадила канал (Розробник), отримати певну чутливу інформацію криптосистеми.

Клептографія, як напрямок інформаційної безпеки, тісно пов'язана як з криптологією так і зі стеганографією, що можна відобразити такою схемою:

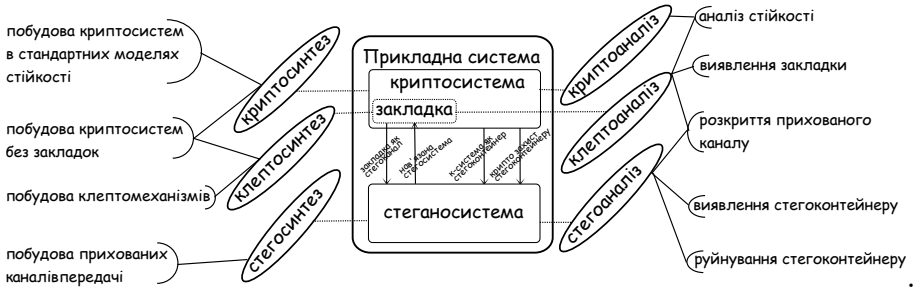

Рисунок 1. Зв'язок між клептографією, криптологією та стеганографією

З рисунку 1 видно, що клептографія пов'язана з криптологією тим, що об'єктом її досліджень є клептографічний механізм (закладка), що не є самостійною сутністю, а є частиною криптосистеми. Отже синтез клептомеханізму супроводжується синтезом або модифікацією криптосистеми (або криптопримітиву). Також, методи криптоаналізу часто використовуються для виявлення закладок, тобто є також інструментами клептоаналізу.

Зв'язок зі стеганографією полягає у тому, що в ряді випадків клептографічний механізм може бути використаним як канал прихованої передачі даних. Тобто, клептомеханізм може фактично грати роль асиметричної криптосистеми з додатковою властивістю прихованості факту передачі інформації. З іншого боку, нав'язана стегосистема може бути використаною для прихованої передачі секретних параметрів криптосистеми, тобто використовуватися як закладка.

На даний момент, основним напрямом клептографічної діяльності є синтез криптосистем та примітивів із закладками, що має практичне значення. Проте методи захисту від клептографічних атак наразі зводяться до традиційного криптоаналізу потенційно вразливих систем, до певних інтуїтивних рекомендацій (яких далеко не завжди дотримуються) щодо процесу розробки та до базових заходів захисту програмно-апаратних комплексів. Одна з найвідоміших формально описаних клептографічних систем є SETUP (Secretly Embedded Trapdoor with Universal Protection), який був запропонований А. Яном та М. Юном у 1996 році та дозволяє організувати непомітну передачу секретного ключа криптосистем на базі RSA та задачі дискретного логарифму. Не зважаючи на широку та давню відомість цих методів, спроби протидії SETUP зводяться переважно до контролю цілісності реалізацій, що може бути не завжди ефективно. Серед вітчизняних та постсовєтських авторів, найближчими до клептографічної тематики є роботи Шелеста М.Є, Задіраки В.К., Прогонова Д.О., Мельникова Ю.Н., Жукова О.Е., Міщенка В.О., проте ці роботи здебільшого сфокусовані на проблемах стеганографії і лише поверхнево – клептографії. Відсутність системного підходу до побудови криптосистем з стійких до клептографічних атак та методів оцінки систем щодо наявності закладок призводить до ризиків інформаційної безпеки, які підвищуються з ростом розповсюдженості

криптосистем та довіри до розробника криптосистеми.

Таким чином, видається доцільним створення методів побудови криптосистем та криптопримітивів з доведеної стійкістю до клептографічних атак, а також створення науково обґрунтованих підходів до оцінки ризиків наявності закладок у примітивах на етапі розробки та конкурсних відборів що і є основою досліджень даної роботи.

Мета та задачі досліджень Метою дисертаційної роботи є підвищення рівня захищеності гібридних криптосистем проти клептографічних атак що передбачає побудову методів виявлення та синтезу клептографічних механізмів в гібридних криптосистемах.

Для досягнення поставленої мети необхідно вирішити такі задачі:

1. Провести аналіз відомих методів побудови клептографічних механізмів.
2. Формалізувати клас протоколів типу «запит-відповідь» у клептографічному контексті.
3. Розробити критерії відсутності клептографічної модифікації реалізації протоколу та побудувати протоколи з доведеною відсутністю клептографічних закладок.
4. Розробити методи побудови клептографічних лазівок геш-функцій з метою визначення загальних принципів впровадження клептографічних лазівок в криптографічні примітиви.
5. Розробити метод виявлення клептографічних лазівок в криптопримітивах.

Об'єктом дослідження дисертаційної роботи є процес клептографічної модифікації реалізацій криптосистем, а предметом дослідження – методи побудови криптосистем з доведеною неможливістю непомітного вбудовування клептографічної закладки у дану криптосистему.

Методи досліджень. Основу дисертаційних досліджень складають теоретичні дослідження. Для аналізу існуючих методів побудови клептомеханізмів застосовувалися методи абстрактної алгебри, теорії складності обчислень та

методів математичної логіки. Для отримання достатніх умов відсутності каналу, а також для зведення стійкості функції стиснення гешу до складності задачі дискретного логарифмування використовувалися методи теорії складності обчислень, теорії алгоритмів та методи лінійної алгебри.

Наукова новизна одержаних результатів. Підсумком розв'язання зазначених задач є такі нові наукові результати, що висуваються на захист:

1. Вперше запропоновано математичну модель для протоколів типу «запит-відповідь» у клептографічному сенсі, яка за рахунок використання моделі практичної нерозрізненості та базових властивостей клептографічного каналу витоку для оцінки стійкості протоколу дозволяє формалізувати клептографічну лазівку у широкому класі криптографічних протоколів.

2. Вперше отримані достатні умови неможливості непомітної клептографічної модифікації криптосистеми, які за рахунок сформульованої теореми та її наслідку дозволяють проектувати протоколи, що є стійкими до побудови каналу непомітного витоку. Можливості використання достатніх отриманих умов продемонстровано на прикладі побудови модифікованих протоколів генерації nonce та Діффі-Хеллмана без SETUP.

3. Вперше розроблено метод побудови функції гешування з клептографічним механізмом, який за рахунок використання перетворення на базі задачі дискретного логарифмування в одній із стандартних конструкцій функції стиснення, що стійкі до колізій, та схемі Меркла-Дамгарда, дозволяє Розробнику ефективно відновлювати частину повідомлення за відомим геш-кодом.

4. Вперше запропонована метрика «клептографічного потенціалу» як кількість інформації Розробника, що передається у структуру примітиву, яка за рахунок оцінювання надлишковості параметрів у структурі криптографічного примітиву дозволяє порівнювати різні криптопримітиви за ризиком наявності клептографічного механізму в них.

5. Отримано подальший розвиток для методу Пренеля побудови шифру на основі схеми Фейстеля з вбудованими диференційними шляхами високої ймовірності. За рахунок застосування його в основі функції стиснення геш функції, дозволяє будувати новий тип закладок, що дозволяє Розробнику отримувати обчислювальну перевагу у спеціальних режимах роботи функції гешування (наприклад, у застосуванні в протоколах консенсусу технології блокчейн).

Практичне значення одержаних результатів.

1. Запропоновані алгоритми базових протоколів «запит-відповідь», які є доведено стійкими до клептографічної модифікації реалізації що дозволяє підвищити рівень захищеності криптосистеми за припущення часткової компрометації одного з учасників.

2. Запропоновані алгоритм та програмна реалізація мовою Python3 функції стиснення, що може бути використана у конструкції Меркла-Дамгарда та містить клептографічний механізм, дозволяє розробнику практично відновлювати повідомлення (за додаткових умов) за відомим геш-кодом.

3. Для лазівки у блокчейн протоколах консенсусу Proof-of-Work отримані оцінки переваги Розробника для різної кількості контрольованих бітів та ймовірності допоміжних диференційних шляхів, що демонструє необхідність подальших досліджень безпеки блокчейн систем у клептографічному контексті.

4. Для ряду примітивів (геш функцій та алгоритмів симетричного шифрування) були отримані оцінки клептографічної надлишковості. Розрахунки показали, що серед розглянутих алгоритмів найбільша клептографічна надлишковість у російського стандарту геш функції ГОСТ Р-34.11-2012 («Стрибог») – 12582.19 біт (тобто за даною метрикою, алгоритм має найбільший ризик наявності клептографічного механізму). Натомість, найменша клептографічна надлишковість спостерігається в стандарті блокового шифрування AES – 32.

1 Сучасний стан клептографічних досліджень

1.1 Клептографічні механізми: визначення

Спершу, введемо неформальне визначення клептографічного механізму.

Визначення 1. *Криптосистему з клептографічним механізмом називатимемо таку криптосистему, що:*

1. *Структура системи згенерована з використанням секрету розробника.*
2. *Секрет розробника практично неможливо отримати шляхом аналізу криптосистеми (або ж сам такий аналіз є неможливим).*
3. *Деякі з криптографічних властивостей системи порушуються у випадку знання секрету розробника.*

Тобто, для клептографічної моделі криптосистеми додається роль «розробника», мета якого модифікувати криптосистему (або побудувати з нуля) таким чином, щоб вона містила «закладку», яка б дозволила в процесі роботи системи непомітно передавати певну секретну інформацію розробнику або ж спростила би розробнику певну задачу, на якій базуються криптографічні властивості системи.

Таке визначення є неформальним і не дозволяє повністю розкрити об'єкт дослідження. Під структурою системи можна розуміти алгоритми примітивів, довгострокові ключі шифрів, кроки протоколу тощо. Також і поняття «неможливості отримання секрету» є доволі розмитим: це може бути використання захищених апаратних компонент із захистом від зламу, або використання асиметричної криптографії для передачі секретів розробнику, але також це можуть бути і програмні компоненти звичайних робочих станцій в окремих зловмисних сценаріях, модель яких не передбачає самих спроб аналізу модифікації за час актуальності процесу передачі секрету жертви.

Більш формалізоване визначення буде сформульоване у наступних розділах.

1.2 Клептографія та стеганографія

Дещо схожим до клептографії напрямом є стеганографія, задачею якої є передача даних з неможливістю розкриття факту передачі. Спільною відправною точкою як клептографії так і стеганографії можна вважати роботи Густава Саймонса, у яких формулюються та досліджуються «проблема ув'язненого» («the prisoner's problem») та приховані канали передачі («subliminal channels») 1—4 Схожість цих напрямів може внести певну неясність в об'єкт дослідження, тож є доречним порівняти їх детальніше.

Відмінності між методами клептографії та стеганографії наведені у таблиці 1.

Таблиця 1

Відмінність у методах клептографії та стеганографії

Критерії порівняння	Клептографія	Стеганографія
задача, що стоїть перед методами	порушення криптографічних властивостей криптосистеми для розробника	передача інформації із приховання факту передачі
рівень абстракції	на рівні криптопротоколу (контекст функціонування не важливий)	на рівні передачі даних (важливий контекст: тип даних, фізичні параметри мережі)
спосіб застосування	таємна модифікація робочої криптосистеми жертви або введення закладки на етапі розробки	попереднє узгодження між учасниками стегосистеми
особливості впровадження	інтегрована в реалізацію криптосистеми	сторонній модуль, програма, плагін
щільність передачі (витоку,всього)	щільна передача (для SETUP – $(n, 2n)$)	відносно низька $(n, >> n)$
вектор протидії	захист криптографічних властивостей системи жертви розробника	унеможливлення обміну прихованою інформацією абонентами
можливість протидії передачі	стійкий клептомеханізм зазвичай не може бути виявленим або відфільтрованим без порушення роботи криптосистеми	часто контейнер може бути умисно псуватися (протидія передачі) без порушення роботи загалом

Разом з тим, функціонування клептографічних та стеганографічних механізмів в певних аспектах схожі чи перетинаються:

1. Таємний клептомеханізму, який виконує власне передачу секрету, може розглядатися як стеганографічний канал.

2. Таємно вбудований стеганографічний може використовуватися розробником для клептографічних атак передачі секрету.

3. Клептографічний механізм та стеганосистема можуть співіснувати незалежно. Наприклад, в механізмі SETUP 5, окрім схеми (1,2) - передачі ключа існують також стегоканали, що базуються на часових затримках відправлення, ймовірнісному контролі певної кількості бітів відкритих випадкових параметрів, імітування збоїв зв'язку тощо.

Варто також окремо акцентувати увагу на те, що в даній роботі при розгляді клептографічних механізмів, особливо при побудові схем з доведеної відсутністю каналу витоку секрету, повністю виключалися з розгляду «залишкові» стеганографічні канали на основі фізичного середовища функціонування (часові затримки, імітація збоїв, тощо). Зазвичай методи виявлення та знешкодження таких каналів є компромісом чутливості систем реагування на інциденти, можливостей їх контролювати фізичне середовище (наприклад, для рандомізації часових затримок з руйнуванням відповідних потенційних стеганоконтейнерів) та ширини пропускання (щільність передачі) прихованого каналу.

1.3 Огляд сучасних клептографічних механізмів

Проблемою усіх практичних клептомеханізмів є те, що навіть при знаходженні закладки чи каналу витоку неможливо практично довести «навмисність» її побудови, оскільки вони також можуть свідчити лише про недостатність наявних методів чи кваліфікації аналітиків. Тож під криптопримітивом із вбудованим клептографічним механізмом ми розумітимемо таку

схему, де лише потенційно може бути навмисно організований канал витоку чи порушення криптографічних властивостей. Розглянемо декілька прикладів алгоритмів, що ймовірно містять клептомеханізм.

1.3.1 Алгоритм шифрування DES

Алгоритм симетричного шифрування DES був запропонований 1974 року фірмою IBM, базується 64-райндовій на мережі Фейстеля, розмір відкритого тексту та повідомлення складає 64 біт, розмір ключа 56 біт. В оригінальну схему Агенством Національної Безпеки США було внесено ряд змін, як то зменшення довжини ключа з 64 до 56 бітів та «допомога» співробітників АНБ у генерації S-блоків, що знизило стійкість алгоритму до атак перебору та диференційного аналізу. Це наводило на підозри, що такі зміни були внесені спеціально для того, щоб спецслужби США, що мали достатні обчислювальні потужності, могли проводити дешифрування повідомлень без знання секретних ключів. Зокрема, є підозри, що вони володіли методами диференційного криптоаналізу до його публікації Біхамом 6—11. Як і у випадку більшості клептографічних механізмів, неможливо довести, що послаблення алгоритму було здійснене навмисно.

1.3.2 Генератор псевдовипадкої послідовності DualEC DRBG

Ще одним прикладом криптопримітиву із вбудованим клептомеханізмом є генератор псевдовипадкової послідовності DualEC DRBG, запропонований Агенством Національної Безпеки США та стандартизований 2006 року (12). В основі стійкості генератора лежить складність вирішення задачі дискретного логарифмування у групі точок еліптичної кривої (рисунок 2).

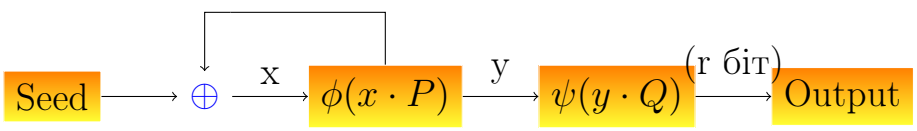

Рисунок 2. Генератор псевдовипадкової послідовності DualEC

де $Seed$ – початкове заповнення генератора, Q, P – випадково та незалежно вибрані на еліптичній кривій точки, ϕ – функція, що повертає координату x точки, ψ – редукція до 30 байтів (старші 2 байти просто відкидаються).

Тобто, стан генератора S на кожній ітерації обчислюється як:

$$S_i = \begin{cases} \phi(S_{i-1} \cdot P), i \neq 0 \\ seed, i = 0 \end{cases} \tag{1.1}$$

А виходом генератора є $\psi(S_i \cdot Q)$, скорочений до 30 байтів.

Значення P та Q згідно зі стандартом (12) обрані незалежно і випадково, та зафіксовані у даному стандарті. Проте у роботах 13—15 було показано, що у випадку, якщо ці параметри згенеровані не випадково і хтось (розробник) володіє секретом $w : Q = w \cdot P$, то він може ефективно відновлювати поточний стан генератора за відомим виходом. Процес відновлення стану відбувається таким чином:

1. Розробник отримує два послідовних виходи генератора R_i та R_{i+1}.

2. Обчислює можливі значення після множення на точку Q (до редукції): $\hat{R}_i^0..\hat{R}_i^{65535}$ (65535 варіантів, оскільки за редукцією відкидаються перші два байти).

3. Для кожного R_i^j, знаючи секретний параметр w, розробник обчислює можливе значення внутрішнього стану генератора: $\hat{R}_i^j = \phi(S_{i+1}^j \cdot Q) = \phi(S_{i+1}^j \cdot w \cdot P) \Rightarrow S_{i+2}^j = w^{-1}\phi^{-1}(\hat{R}_i^j)$

4. Остаточне значення стану є $S_{i+2} = S_{i+2}^j : \psi(\phi(S_{i+2}^j \cdot Q)) = R_{i+1}$.

Знову ж таки, неспростовних доказів того, що АНБ дійсно володіє секретом w немає, проте на підозри наводить, окрім структури генератора, також повідомлення про отримання хабара керівництву компанії RSA Security LLC 16 за впровадження генератора DualEC DRBG як ГПВЧ за замовчуванням у деякі зі свої продуктів, зокрема в RSA BSAFE.

1.3.3 Російський стандарт гешування ГОСТ Р34-11-2012 («Стрибог»)

Російський стандарт гешування ГОСТ Р34-11-12 прийшов на зміну старішого стандарту ГОСТ Р34-11-94. Даний алгоритм має розмір блоку 512 бітів та довжину геш-коду 256/512 бітів, і заявлені складності пошуку прообразу та сильної колізії були $2^{512}/2^{256}$ та $2^{256}/2^{128}$ відповідно (як для ідеальної функції хешування). Однак після стандартизації з'явилося ряд робіт, що демонструють методи побудови колізій та пошуку прообразу усічених версій. З точки зору клептографії, важливий той факт, що методи генерації більшості константних параметрів є невідомими, що наводить на підозру на те, що вони можуть бути секретом розробника, що спрощує для власника секрету певні задачі криптоаналізу. Подібні підозри підсилюють дослідження авторів Riham AlTawy та Amr M. Youssef, де вони демонструють практичну колізію для змінених константних параметрів алгоритму (17), а також роботи, де показана регулярність у структурі нелінійних перетворень 18—23. Схожі результати були також отримані щодо геш функції SHA-1 24. Як і для решти клептомеханізмів, беззаперечних доказів наявності такої закладки немає.

1.3.4 Російський стандарт симетричного шифрування ГОСТ Р34-12-2015 («Кузнєчік»)

Стандарт блокового симетричного шифру ГОСТ Р34-12-2015 розроблений Центром захисту інформації і спеціальним зв'язком ФСБ Росії, являє собою sp-мережу з фейстелівською схемою для ключового розкладу.

Криптологи Алекс Бірюков, Лео Перін та Олексій Удовенко у роботі 25 та інші автори 21—23, 26 показали, що S-блок алгоритму згенерований не дійсно випадковим чином (як це вказано у стандарті), а з використанням генератора, схему якого вдалося відновити. В свою чергу, цей факт хоч напряму і не доводить наявність лазівки, проте наводить на підозри про спрямоване зниження стійкості примітиву з метою спрощення криптоаналізу Розробниками.

1.3.5 Система апаратного шифрування Skipjack та стандарт EES

Стандарт EES (Escrowed Encryption Standard) апаратного шифрування розроблений Агенством Національної Безпеки США у рамках проекту Capstone побудови систем захищенного урядового зв'язку із закладкою. Стандарт включає в себе блоковий алгоритм симетричного шифрування Skipjack та архітектуру LEAF (Law Enforcement Access Field – поле доступу для правоохоронних органів). Для імплементації стандарту використовувався захищений чіп Clipper. Передбачалося, що стійкість шифратора 27—29 базуватиметься на невідомому алгоритмі шифрування (алгоритм Skipjack – секретний), а процес ініціалізації ключів відбуватиметься безпосередньо розробником чіпа. Архітектура LEAF дозволяє використовувати два ключа розшифрування: один для користувача, а інший – для правоохоронних органів. Тож розробники мають змогу розшифровувати перехоплені повідомлення, в той час як звичайні користувачі можуть це робити лише за допомогою власних секретних ключів, що зашиті апаратно.

1.3.6 Канал витоку в системах на основі криптографії на еліптичних кривих

Відомі щонайменше два принципові підходи до побудови лазівки на базі криптографії на еліптичних кривих:

1. Генерація криптографічно слабкої еліптичної кривої, побудова та публікація ізоморфної до неї (ізоморфізм є секретним параметром розробника).

2. Використання стійкої еліптичної кривої такого виду, що відсутність перевірки того, що точка знаходиться на кривій призводить до переводу операцій над класом кривих, у якому розробник може за практичний час вирішувати задачу дискретного логарифмування.

Лазівка першого типу досліджена у роботах 30, 31. Ідея методу полягає у тому, що Розробник спершу обирає еліптичну криву E_s, задачу дискретного

логарифмування якої можна звести до задачі дискретного логарифмування в полі F_{2^N} використовуючи визначену функцію спарювання Вейля 32 так, що остання практично вирішується Розробником. Далі, Розробник будує ізоморфну криву E_{pb}, використовуючи секретне перетворення $\psi : E_s \to E_{pb}$ за методом 33. Далі, крива E_{pb} публікується (наприклад, як частина стандарту) і використовується жертвою. В такому разі задача дискретного логарифмування, наприклад, пошук $x : x \cdot G = P$ за відомих G, P, складна для користувача системи, проте розробник може її звести до задачі на кривій $E_s : P \to \psi^{-1}(P), G \to \psi^{-1}(G)$, що зведенням до задачі дискретного логарифмування над полем лишків за методом 32 дозволяє Розробнику вирішити задачу ECDLP за практичний час.

Клептомеханізм другого типу досліджений у роботах 34—42. Даний метод продемонстрований на прикладі схеми цифрового підпису на базі еліптичних кривих ECKCDSA. Ідея полягає у тому, що в алгоритм генерації цифрового підпису жертви вводиться помилка (секретний параметр розробника), що дозволяє розробнику, перехопивши певну кількість підписів жертви отримати її секретний ключ.

1.3.7 Канали витоку секрету в протоколах

Одним з найвідоміших прикладів таких механізмів є метод SETUP (Secretly Embedded Trapdoor with Universal Protection), що дозволяє організувати витік секретного ключа шляхом зловмисної модифікації реалізації криптосистеми на основі задачі факторизації великих чисел чи задачі дискретного логарифмування в скінчених полях. Наразі цей метод є теоретичним (загальновідомих фактів його застосування нам не відомо), проте є цілком реальним для використання на практиці.

Інший тип атаки – BEAST (CVE-2011-3389) на протокол SSL до версії TLS 1.0, що є використанням комбінації вразливостей XSS(Cross Site Scripting) веб сервісу 43 та Session Fixation у реалізації CBC режиму шифрування протоколу SSL. Зловмисник (розробник) може змусити жертву виконати браузерний код таким чином, що в отриманій зашифрованій послідовності порушуватиметься унікальність стартового вектора (CBC режиму шифру) для

кожного відкритого повідомлення, дозволяючи зловмиснику дешифрувати секретну частину повідомлення. Насправді, малоймовірно, щоб дана атака була спланованим клептографічним механізмом, проте вона має певні ознаки такого: внаслідок втручання у систему жертви утворюється канал непомітного витоку секрету.

Окрему увагу варто звернути на атаки на відкриті реалізації криптопротоколів. Однією з їх особливостей є те, що зміни у відкриті реалізації може робити практично будь-який розробник, при цьому процес аудиту безпеки не завжди йде належним чином. Наведемо декілька прикладів таких вразливостей:

Таблиця 2

Вразливості в реалізації відкритих протоколів, що можуть використовуватися як лазівки

Бібліотека	CVSS2 метрика	Дефект	Опис
OpenSSL	4.3	CVE-2011-3389	Атака фіксації стартового вектора шифру в режимі CBC через додатковий канал контролю відкритого тексту (BEAST)
OpenSSL	5.0	CVE-2014-0160	Помилка в реалізаціїї openssl, що дозволяє зловмиснику, що знає вразливість, ефективно отримувати секретний ключ (heartbleed) 44.
OpenSSL	5.0	CVE-2014-3570	Обхід субекспоненційної задачі з ймовірністю 2^{-128}
OpenSSL	5.0	CVE-2015-3193	Помилка в реалізації алгоритму Монтгомері, що призводить до отримання додаткової інформації про секретний ключ
NSS	6.4	CVE-2016-1938	Помилка в реалізації алгоритму Монтгомері
Nettle	7.5	CVE-2015-8803	Помилка в операціях на стандартній кривій NIST P-256, що зводить задачу DLP до групи меншого порядку
Nettle	7.5	CVE-2015-8805	Помилка в операціях на стандартній кривій NIST P-256, що зводить задачу DLP до групи меншого порядку

1.4 Клептографічні схеми типу SETUP

1.4.1 Побудова SETUP на основі проблеми дискретного логарифмування

У 1994 році, А. Янг і М. Юнг запропонували метод побудови SETUP для протоколів, стійкість яких базується на складності задачі пошуку дискретного логарифму. Ідея схеми у тому, що сторона, яка генерує пари ключів та публікує публічні, має модифіковану реалізацію, що передає захищеним каналом розробнику секретні ключі.

Вихідні дані:

1. F_p^* – мультиплікативна група з генератором g.
2. $(x, Y = g^x \mod p), x, Y \in F_p^*$ – пара ключів розробника.
3. $W, a, b \in F_p^*$ – фіксовані параметри.

Кроки роботи алгоритму генерації публічних сеансових ключів:

1. Генерується випадковий ключ $c_1 \in F_p^*$. c_1 зберігається для генерації наступного ключа.
2. Обчислюється та публікується перший публічний ключ $\underline{M_1 = g^{c_1} \mod p}$.
3. Генерується випадкове $t \in \{0, 1\}$.
4. Обчислюється $z = g^{c_1 - Wt} Y^{-ac_1 - b} \mod p$.
5. Обчислюється наступний секретний ключ: $c_2 = hash(z), hash : \{0,1\}^* \to F_p^*$.
6. Обчислюється та публікується відкритий ключ $\underline{M_2 = g^{c_2} \mod p}$.

Кроки роботи алгоритму відновлення другого сеансового секретного ключа розробником:

1. $r = M_1^a g^b \mod p$.

2. $z_1 = M_1/r^x \mod p$.

3. $c_2 = hash(z_1)$ або $c_2 = hash(z_1/g^W \mod p)$.

Таким чином, розробник може отримати секретний ключ c_2 і ніхто не зможе його отримати без знання секретного ключа розробника x.

Загалом, побудова SETUP є найбільш дослідженою проблемою клептографії, що висвітлювалася у числених роботах 45—51.

1.4.2 SETUP на базі алгоритму RSA

В роботах 52—55 досліджені можливості побудови клептографічних лазівок на базі алгоритму RSA, зокрема запропоновано схему побудови каналу витоку в реалізації алгоритму RSA. Ідея полягає у тому, що реалізація на стороні жертви генерує параметри RSA таким чином, що зловмисник, знаючи секрет механізму, може ефективно розв'язувати задачу факторизації.

Генерація параметрів немодифікованого RSA влючає генерацію двох простих великих чисел $p, q : \#p = \#q = m, m > 1024$ ($\#(\cdot)$ – бітова довжина), ці значення – секрет криптосистеми. Відкритим параметром є $n = p \cdot q$. Якщо зловмисник модифікує реалізацію криптосистеми так, що зможе ефективно факторизувати n, він зможе також розшифрувати будь-які зашифровані дані.

Схема, запропонована в 52 полягає в тому, що секретний ключ генерується на основі публічного ключа розробника $y = g^x mod P$, де $g, x \in F_P^*$, параметри g, x, y, P – відкриті і $\#P = \#p = m$. Параметри $W, t, a, b \in F_P^*, hash(\cdot)$ мають той же сенс, що і побудові каналу витоку в протоколі Діффі-Хеллмана 56, $G : F_P^* \times K \to \{0,1\}^m$ – симетричний шифр (K – простір ключів). Також задається параметр L – невелике ціле число та фіксований симетричний ключ $k_0 \in K$.

1. Генерація випадкових $c_1, t : c_1 < P - 1, t \in \{0,1\}$.

2. Пошук z у ході розв'язання рівняння $y^{ac_1+b} g^{Wt} z = g^{c_1} mod P$. Якщо не виконується умова $z \in [2..2^m - 1]$, повернутися до кроку 1.

3. $z' = hash(z)$.

4. Молодший розряд z' встановлюється в '1' (z' має бути непарним).

5. Генерація секретного параметру (великого простого числа) p у вигляді $p = z' + num$ (num – найменше натуральне число таке, що p – просте).

6. Для $0 \leq i \leq B$ виконуються кроки:

   (а) Обчислення $U = G(g^{c_1}, k_0 + i)$.

   (б) Генерація випадкового $R \in \{0,1\}^m$.

   (в) Отримання q та цілого $r < p$, що задовольняють рівняння $[U|R] = pq + r$ ($[\cdot|\cdot]$ – конкатенація бітових рядків). Якщо q – не просте, перехід на 6б.

   (г) Обчислення $n = pq = [U|R] - r$.

   (д) Обчислення пари експонент RSA: e, d.

Зловмисник, знаючи секрет лазівки x може відновити параметри p та q. Для цього він виконує такі дії:

1. Отримання U з n (старші m бітів).

2. Для $0 \leq i \leq B$ виконуються кроки:

   (а) Обчислення $m = G^{-1}(U, k_0 + i)$.

   (б) Обчислення $z : m^{xa} g^{b+Wt} z = m(\mod P)$.

   (в) $z' = hash(z)$.

   (г) Молодший розряд z' встановлюється в '1' (z' має бути непарним).

   (д) Пошук такого найменшого s, $p = z' + s$ – просте.

   (е) Якщо $p|n$, зловмисник факторизує n і отримує доступ до каналу витоку.

У випадку, якщо зловмисник не має секрету каналу витоку x, задачі отримання $z = g^{c_1 - Wt} y^{-ac_1 - b}$ (а отже і p) зводитиметься до задачі дискретного логарифму.

1.4.3 Можливості побудови SETUP на базі постквантових примітивів

Загалом, аналізуючи різні алгоритми з вбудованим SETUP можна виділити закономірності, критерії вразливості алгоритму до побудови каналу витоку SETUP:

1. Алгоритм повинен передбачати генерацію ключа з внутрішнього джерела випадковості.

2. Алгоритм має бути асиметричним, оскільки каналом витоку відомих SETUP слугують публічні ключі, що передаються відкрито та перехоплюються розробником.

3. Великий розмір ключів підвищує шанси на побудову SETUP оскільки в цьому випадку часто канал витоку можна побудувати за допомогою іншої криптосистеми (наприклад, для RSA канал витоку базується на DLP).

Насьогодні опубліковано небагато робіт стосовно клептографічних механізмів постквантової криптографії 57, 58. В першу чергу це пов'язано з їхньою малою розповсюдженістю. З асиметричних методів постквантової криптографії, фактично лише алгоритм NTRUENcrypt стандартизований 59, ще декілька, серед яких алгоритми Джентрі 60, McEliece та криптосистема Нідеррайтера 61, є непрактичними через повільність та великі розміри ключів (порядку 2^{20} бітів). Ще чимало систем знаходяться під питанням щодо їх класичної стійкості.

Якщо розглядати конкретно алгоритм NTRUEncrypt, то він потенційно придатний для побудови SETUP: ключі великої довжини (для $N = 251, q = 128$, стандартна стійкість, довжина ключа $\#h = 251 \cdot \log_2 128 = 1757$), схема генерації дещо нагадує таку у RSA (публічний ключ обчислюється як добуток секретних компонентів з усіченого кільця поліномів).

Алгоритм NtruEncrypt базується на кільцях усічених поліномів $Z_p/(X^N + 1)$, $Z_q/(X^N + 1)$, $p << q$ – малі взаємно прості числа (наприклад, $p = 2, q = 127$).

Секретні параметри (випадково згенеровані): $f, g \in Z_p/(X^N+1)$, $f_p = f^{-1}(\mod p)$, $f_q = f^{-1}(\mod q)$.

Відкриті параметри: $p, q, h = p \cdot f_q \cdot g \mod q \in Z_q/(X^N+1)$.

Отже одним з варіантів побудови схеми SETUP може бути генерація таких секретних параметрів f, g, щоб з відкритого параметра h розробник зміг би відновити секретні параметри.

Наразі питання про можливість такої модифікації залишається відкритим, проте цілком ймовірно, що це зробити можливо.

1.5 Схема Пренеля для збалансованих фейстелівських шифрів

При побудові каналу витоку в симетричних криптопримітивах виникають такі додаткові труднощі:

1. Зазвичай симетричні алгоритми працюють швидше за асиметричні. Тож використання асиметричної криптографії для побудови каналу витоку зменшує продуктивність, що може використовуватися для викриття каналу передачі.

2. В симетричних примітивах не завжди наявна рандомізація, яка необхідна для побудови каналу витоку секрету. Це звужує побічний канал. Проте, є можливість використання додаткових стегоконтейнерів: стартові вектори для симетричного шифрування в поточному режимі або «сіль», що використовується при хешуванні.

3. Якщо говорити про побудову каналу в структурі алгоритму, то симетричні примітиви мають невеликі розміри блоків підстановок, що спрощує пошук закладок в них.

Менш із тим, є нечисленні роботи 62, 63 присвячені побудові лазівок у симетричних криптопримітивах.

Б. Пренелем та М. Рейманом у 1997 році запропонований метод побудови каналу витоку секрету (лазівку) у фейстелівських шифрах 64. Суть методу в

тому, що зловмисник спершу генерує випадковий диференційний шлях для майбутнього шифру на основі власного секрету β. Після того, він генерує блок підстановки таким чином, щоб диференційний шлях мав високу ймовірність. Цей диференційний шлях використовується Розробником у майбутньому для ефективного проведення лінійного криптоаналізу. При цьому інші користувачі системи, не знаючи секрету β, не можуть відновити диференційний шлях.

Розглянемо $m \times (n-1)$ S-блок шифру, $S : \{0,1\}^m \to \{0,1\}^{n-1}$. Такий S-блок можна також представити у вигляді, булевих функцій для кожної компоненти виходу: $\{f_i(x)\}_{i=0..n-1, i \neq q}, x \in \{0,1\}^m$, $q \in [0, n-1)$ – фіксоване. Тоді S-блок з каналом витоку T будується доданням до заданого S-блоку функції $f_q = \bigoplus_{i=0..n-1, i \neq q} \beta[i] \cdot f_i(x)$, де $\beta \in \{0,1\}^n$ вибране таким чином, що з високою ймовірністю p_T $\beta \bullet T(x) = 0$ ($\bullet$ – скалярне множення векторів). За достатньо великих значень n та m без знання β зловмиснику «складно» знайти кореляцію у виході блоку. Також зловмиснику чи користувачу «складно» знайти β. Проте розробник, знаючи β може отримати кореляції виходу: $2 \times p_T - 1$ та провести ефективний лінійний криптоаналіз.

Проте уже в 1998 році було доведено 65, що така схема побудови каналу витоку ненадійна в моделі асипмтотичної складності алгоритмів і є лише компромісом ресурсів затрачених на використання каналу розробником та ресурсів затрачених на викриття секрету аналітиком. Слід зазначити, в інших моделях, зокрема в моделі практичної стійкості за відповідного контексту, метод може бути стійким клептографічним механізмом.

1.6 Клептографічні схеми на основі стеганографічного каналу

Іншим розділом, що вивчає методи непомітного витоку секрету, є стеганографія. У 1.2 наводиться порівняння стеганографії та клептографії. Не дивлячись на відмінності у концепціях напрямів, стеганографічні методи можна також застосувати у клептографічних цілях: для непомітної передачі розробнику секретів жертви. В такому разі, Розробник може внести

модифікацію реалізації криптосистеми жертви таким чином, що секрет жертви передаватиметься Розробнику за допомогою стеганографічного контейнеру.

У таблиці 3 наводиться порівняння деяких стегоконтейнерів, що можуть застосовуватися у клептографічних цілях.

Таблиця 3

Порівняння стеганографічних контейнерів

Тип контейнеру	Пропускна здатність за раунд	Інтенсивність раундів
Дані прикладного рівня	Висока	Залежить від застосування
Фізичний мережевий рівень (наприклад, затримки у каналі)	Середня	Висока
Рівень протоколу (наприклад, збої у аутентифікації як метод передачі інформації)	Низька	Середня
Відкриті рандомізовані параметри з частковим контролем (відкриті ключі)	Низька	Низька
Відкриті рандомізовані параметри (стартові вектори, сіль, nonce)	Висока	Низька

З таблиці 3 видно, що контейнер прикладного рівня (наприклад, використання медіа контенту) є доволі містким, проте його можливо використовувати лише за наявності відповідного контенту, він також вимагає додаткового контексту під час модифікації що звужує можливості розробника по впровадженню такої модифікації.

Контейнери фізичного мережевого рівня, на відміну від попереднього, завжди існує: будь яка комунікація передбачає використання каналів передачі. Складність його використання полягає у необхідності виходу за межі контексту криптосистеми та отримання контролю над проміжним програмним забезпеченням, модулями ядра операційної системи, що також сильно ускладнює впровадження клептомеханізму.

Рівень протоколу уже знаходиться у контексті криптосистеми. У якості контейнеру можуть використовуватися навмисні переривання раундів протоколу та часові затримки у роботі. Проблемою використання цього каналу витоку є низька пропускна здатність каналу.

Через частково контрольовані випадкові параметри (певні біти публічних ключів, генераторів групи протоколу Діффі-Хеллмана, тощо) знаходяться цілком у контексті криптосистеми, тож з цієї точки зору найбільш придатні для використання у клептографії. Проте пропускна здатність такого каналу низька (через необхідність контролювати біти параметру ймовірнісним чином).

Контрольовані відкриті випадкові параметри (стартові вектори потокових режимів шифрів, сіль функцій гешування та KDF, nonce протоколів аутентифікації, тощо) мають усі переваги попереднього контейнеру і при цьому мають високу місткість: наприклад, при передачі стартового вектору алгоритму шифрування AES в режимі CBC стартовий вектор – 16 довільних байтів, що можуть використовуватися для передачі інформації Розробнику.

1.7 Клептомеханізми технології блокчейн

1.7.1 Визначення та основні поняття технології блокчейн

Розповсюдження технології блокчейн розпочалося із публікацією Сатоші Накамото 66 та запуском криптовалюти Bitcoin. В основі даної криптовалюти (також в основі усіх інших) лежить технологія блокчейн.

Технологія blockchain є однією із найбільш перспективних сучасних технологій розподіленої обробки інформації. Ефективність її функціонування заснована на стійкості криптографічних систем (які використовуються для роботи blockchain) від криптоаналітичних атак. Мало дослідженими і актуальними є проблеми 67, 68:

1. Розробки та вдосконалення протоколів узгодження в розподілених, ненадійних системах.

2. Дослідження моделей даних в блокчейн-системах. Зокрема актуальною є задача побудови механізму реплікації розподіленої бази даних при різних ступенях надлишковості збереження даних, від традиційної для технології блокчейну повного дублювання інформації в усіх вузлах мережі до мінімальної надлишковості інформації, необхідної лише для зв'язності даних.

3. Модифікація існуючих криптографічних механізмів блокчейну, зокрема наприклад заміна дерев Меркла, використання групових цифрових підписів, заміна задачі обернення геш-функцій на інші складні для обчислення задачі.

4. Аналіз стійкості геш-функцій до обернення із врахуванням особливості їх використання в блокчейні, який відрізняється від традиційної задачі аналізу колізійстійкості та обернення геш-функцій.

5. Аналіз та побудова клептографічних лазівок блокчейну.

6. Побудова криптографічних протоколів на блокчейнах, наприклад протоколів доказу з нульовими знаннями, протоколів анонімізації, доказу інтелектуальної власності, тощо.

7. Розрахунки параметрів практичних блокчейн- систем (оцінки надійності реальних блокчейнів, оцінки довжини буферів очикуваних для включення в блокчейн транзакцій та ін.).

8. Дослідження комплексного впливу атак на реалізацію для поширених та перспективних проектів blockchain, зокрема дослідження атаки на реалізацію механізмів цифрового підпису, що використовуються в існуючих та перспективних проектів blockchain.

9. Дослідження стійкості і ефективності протоколів узгодження для децентралізованих «візантійских» протоколів узгодження та їх сучасних модифікацій (протоколів Practical Byzantine Fault Tolerance, Raft Consensus Algorithm,тощо).

10. Дослідження клептографічних атак технології blockchain для слабого Secretly Embedded Trapdoor with Embedded Protection (weak SETUP) методу атаки на поширені системи, які використовують proof-of-work blockchain, зокрема – криптовалюти Bitcoin.

Основою блокчейн системи є розподілений між усіма учасниками синхронізований реєстр. Суть технології в тому, що блокчейн системою гарантується ідентичність екземплярів реєстру у кожного учасника через механізм консенсусу. Наразі, розповсюдження набули два підходи для встановлення консенсусу:

1. Консенсус через змагання («Proof-of-...»). Ідея підходу полягає у тому, що між учасниками системи створюються умови конкуренції, в результаті якої визначається переможець «змагання». Прикладом такого підходу є Proof-of-Work (змагання потужностей – перемагає той, хто швидше підбирає повідомлення, геш код якого задовольняє заданим вимогам), Proof-of-Stake (змагання гаманців – шансів перемогти більше у того, хто має більший баланс гаманця) тощо. Потім, переможець підтверджує новий блок реєстру, а решта учасників приймають цей блок. Ідентичність блоків гарантується тим, що лише блок одного учасника (з великою ймовірністю) додається до реєстру усіх учасників. У випадку нечесності переможця, інші учасники це виявляють та некоректний блок відкидається.

2. Консенсус на основі Задачі Візантійських Генералів 69. Суть підходу у тому, що виконуючи послідовність кроків протоколу узгодження між учасниками, у результаті усі отримують однаковий новий блок реєстру. При цьому, протокол стійкий до нечесної роботи деякої невеликої множини учасників.

В свою чергу, за першим типом побудовані численні алгоритми консенсусу (таблиця 4).

Таблиця 4

Алгоритми консенсусу на основі змагань

Тип консенсусу	Опис
Proof-of-Work	Учасники протоколу змагаються обчислювальними ресурсами вирішуючи певну складну задачу. Прикладом є криптовалюта Bitcoin (та валюти на його базі), Ethereum (до переходу на PoS) та багато інших.
Proof-of-stake	Пріоритет учасника під час генерації нового блоку в ланцюогу блоків залежить від розміру частки розподіленого цінного обмеженого ресурсу, яким володіє. За генерацію блоку учаснику збільшується його частка цінного ресурсу. Питання щодо того, як розподілити частки між учасниками на етапі ініціалізації протоколу не розглядається.
Proof-of-activity	Пріоритет учасника під час генерації нового блоку в ланцюгу блоків залежить від його обчислювальних ресурсів, частки цінного ресурсу та «активності» в мережі. Що більшими є частка та час перебування в мережі, то вищим є пріоритет. Прикладом практичної реалізації певною мірою можна вважати криптовалюту DASH.
Proof-of-burn	Пріоритет учасника під час генерації нового блоку в ланцюгу блоків залежить від розміру частки розподіленого цінного ресурсу, який був знищений учасником на попередньому етапі.
Proof-of-capacity	Пріоритет учасника під час генерації нового блоку в ланцюгу блоків залежить від розміру частки розподіленого цінного ресурсу, яким є місткість простору для зберігання даних.

Proof-of-delegated-stake	Пріоритет учасника під час генерації нового блоку в ланцюгу блоків формується у два етапи. На першому відбувається вибір підмножин учасників за результатами процедури голосування. Кількість голосів кожного учасника залежить від частки володіння цінним ресурсом. У другому етапі беруть участь тільки учасники, обрані за результатом першого етапу. Пріоритет учасника на другому етапі залежить від наявності учасника у мережі та його обчислювальних ресурсів. Застосовується у системах Bitshares та Steemit.

1.7.2 Задачі клептографії для технології блокчейн

Технологія блокчейн передбачає нові моделі стійкості криптосистем. Наприклад, з точки зору класичної КСЗІ передбачається наявність довірених осіб, що супроводжують обробку визначеної інформації. Натомість, в системі блокчейн такі особи можуть бути відсутніми і процес обробки відбувається одночасно групою учасників з їх постійною взаємною перевіркою.

Також варто згадати ключову властивість блокчейн системи – децентралізацію. З одного боку, децентралізація є фактично інструментом гарантування чесності роботи (кожен кожного контролює), з іншого боку ця властивість є слабко формалізованою та складно оцінюється.

Загальні вектори клептографічних атак, спрямовані проти базових криптопротоколів та примітивів також можуть переноситися і на систему блокчейн. Наприклад, можлива організація витоку секретного ключа власника гаманця або ж побудова системи електронного цифрового підпису з лазівкою, що дозволяє Розробнику створювати транзакції від імені іншого користувача.

Окрім того, існують специфічні вектори, що притаманні саме технології блокчейн 70—72. Наприклад, атаки на консенсус, що дозволяють Розробнику, що знає певний секрет, отримувати певну перевагу в змаганні. Наприклад,

якщо в протоколі консенсусу типу Proof-of-Work одна із сторін (Розробник) має можливість генерувати геш коди заданої складності з певною перевагою відносно інших учасників, у цієї сторони є більша ймовірність виграти змагання потужностей та отримати винагороду за згенерований новий блок. У роботі 73 досліджуються методи побудови диференціальних шляхів для функцій хешування родини MD4. Диференціальні шляхи дозволяють шукати пари повідомлень з ймовірністю появи заданої різниці геш-кодів вищою, за випадковий перебір. В свою чергу, це підвищує ефективність підбору гешу із заданою складністю (за відповідно заданих різницях). Також у роботі 64 автори пропонують метод побудови блокового шифру на базі схеми Фейсталя із заданим диференційним шляхом великої ймовірності. У роботі 74 досліджуються методи переносу побудови диференційних шляхів з блокових шифрів на геш функції, що базуються на їх основі. Детальніше метод побудови лазівки у консенсусі типу Proof-of-Work досліджується у 3.3.

1.8 Проблеми клептографії та постановка задачі

Загалом, можна виділити такі задачі клептографічних досліджень:

1. Побудова каналів витоку інформації шляхом модифікації реалізацій стандартних криптосистем. Модифікація може здійснюватися через шкідливе програмне забезпечення, кібератаки чи інші методи неавторизованого доступу до ресурсів жертви. Наприклад, модель Янга та Юнга 52.

2. Створення криптосистем з вбудованим каналом витоку. Такі системи можуть впроваджуватися шляхом державної стандартизації, лобіювання стандартів, публікацією відкритих (в т.ч. opensource) реалізацій тощо. Найвідомішим прикладом є ГПВЧ DualEC DRBG.

3. Виявлення клептографічних механізмів у криптосистемі.

4. Виявлення клептографічних механізмів у реалізації криптосистеми. Дана задача відрізняється від попередньої в припущені, що базова криптосистема не містить клептографічних механізмів (закладок), закладка впроваджується шляхом модифікації саме конкретної реалізації.

5. Побудова криптосистем з гарантованою відсутністю каналу витоку.

6. Побудова криптосистем (примітивів) із мінімізацією можливостей побудови закладки.

7. Побудова криптосистем стійких до клептографічних атак (наприклад, стійкість до клептографічної модифікації однієї із сторін протоколу).

У ході вирішення даних задач виникає ряд пов'язаниз із ними проблем:

1. Недостатня формалізованість клептографії загалом. Не зважаючи на те, що наразі відомі численні практичні та теоретичні клептографічні системи, досі не сформована більш-менш загальна теорія.

2. Неадекватність наявних моделей криптографічної стійкості у контексті клептографічних задач. Розповсюджені моделі теоретико інформаційної та асимптотичної стійкості є недостатніми для аналізу клептографічних конструкцій оскільки практичні закладки часто не описуються такими моделями.

3. Методи побудови симетричних криптопримітивів з клептомеханізмами є слабо розвинутими. Дійсно, навіть у сучасних конкурсах криптопримітивів основна увага приділяється ефективності реалізації та стійкості до ряду найбільш розповсюджених методів криптоаналізу. При цьому, клептографічні можливості нового шифру чи геш функції зазвичай залишаються в стороні або аналізуються поверхово.

4. Мало розвинуті методи побудови криптосистем стійких до SETUP. При переході проектування криптосистем з підходу КСЗІ до відкритих криптосистем з недовіреними учасниками ризик модифікації реалізації частини учасників є важливою для запобігання компрометації розподіленої системи загалом.

5. Відсутність методів оцінки ризиків, пов'язаних з клептографічними атаками.

6. Відсутність методів побудови криптомеханізмів з доведеною відсутністю клептомеханізмів.

7. Відсутність критеріїв оцінки потенціальних клептографічних можливостей криптосистем.

Об'єктами дослідження є криптографічна система, реалізація якої потенційно може містити клептографічний механізм для витоку секрету. Метрикою для оцінки клептографічних можливостей системи є запропонована автором формальні моделі статичного розподілу інформації та модель клептографічного механізму на базі протоколу типу «запит-відповідь».

Отже, задачу, що ставиться перед автором у ході виконання даної дослідницької роботи можна сформулювати у вигляді задачі оптимізації криптографічного протоколу типу «запит-відповідь» з такими умовами:

1. Допустима множина $\mathbb{X}$ усіх можливих протоколів типу «запит-відповідь».

2. $w \in \mathbb{X}$ – криптографічний протокол типу «запит-відповідь», що необхідно модифікувати.

3. $\mathbb{I}_w : \mathbb{X} \to \{0, 1\}$ – предикат, що для кожного протоколу із $\mathbb{X}$ показує, чи задовольняє він криптографічні властивості та функціонал протоколу w ($\mathbb{I}_w(w) = 1$).

4. Множина можливих модифікацій протоколу $\mathbb{X}^w \in \mathbb{X}, \mathbb{X}^w = \{x : y \in \mathbb{X}, \mathbb{I}_w(x) = 1\}$.

5. Цільова функція $f : \mathbb{X} \to \{0, 1\}$ (предикат), що відображає виконання достатніх умов відсутності клептографічного каналу для заданого протоколу.

6. Критерієм пошуку є максимізація даної функції, тобто результатом вирішення задачі є пошук $\hat{x} \in \mathbb{X}^w$:

$$\hat{x} = \max_{x \in \mathbb{X}^w} f(x) \tag{1.2}$$

Варто також зазначити, що розв'язків задачі оптимізації може бути декілька. У такому випадку достатньо знайти будь-який один з них.

1.9 Висновки до розділу 1

У розділі 1вводиться поняття клептографічного механізму та описується поточний стан клептографічних досліджень:

1. Введене неформально поняття клептографічного механізму як розширення поняття криптосистеми, що має додаткову роль Розробника.
2. Наведене порівняння напрямів клептографії та стеганографії.
3. Продемонстровано найвідоміші криптопримітиви із вбудованим клептомеханізмом: алгоритм блокового шифрування DES, генератор псевдовипадкової послідовності DualEC DRBG, система апаратного шифрування Skipjack, а також алгоритм гешування із потенційною лазівкою ГОСТ Р34-11-2012 («Стрибог»).
4. Сформульовано клептографічні задачі технології блокчейн.
5. Наведено клептографічний механізм SETUP авторів Янга і Юнга для протоколів на базі задач дискретного логарифмування та факторизації.
6. Сформульовані задачі та проблеми клептографії.

2 Методи побудови криптосистем з доведеною відсутністю каналів витоку

2.1 Термінологія та визначення

Для уникнення розбіжностей в розумінні, спершу наведемо роз'яснення термінів 75, що вживаються далі у роботі.

Клептографічний механізм – особливість дизайну криптосистеми (криптопримітиву), що дозволяє Розробнику, який впровадив даний механізм, проводити практичний криптоаналіз криптосистеми. Також для позначення клептомеханізмів використовуватимемо синоніми *«закладка»* та *«клептографічна лазівка»*.

Розробник – зловмисник (аналітик) системи захищеного зв'язку, який, окрім аналізу відкритих потоків даних, може також модифікувати або нав'язувати криптосистеми одному або декільком абонентам.

Конструкція – алгоритм без строгої типізації параметрів та операцій (мета-алгоритм). Використовується, наприклад, коли вхідні параметром можуть бути різними об'єктами (група лишків за модулем, група точок еліптичної кривої), або коли структура містить блоковий шифр без уточнення конкретного алгоритму. Алгоритм вважаємо частковим випадком конструкції.

Схема – конструкція без уточнення деяких особливостей проектування. Вважатимемо конструкцію частковим випадком схеми.

Криптографічна система – сукупність криптографічних механізмів, протоколів, системи управління ключами та правил експлуатації криптографічних механізмів.

Криптографічний примітив, криптографічний механізм – базові криптографічні перетворення, які є атомарними блоками криптографічних механізмів.

Клептографічна атака – можливі зловмисні сценарії, що можуть виконуватися Розробником: клептографічна модифікація системи, побудова криптосистеми з закладкою, використання можливості закладок.

Стійкість до клептографічних атак – властивість системи, що полягає у неможливості побудови непомітної (та, що не виявляється за практичний час) закладки або каналів витоку.

Клептографічний канал витоку – канал непомітного витоку секрету, прихований канал витоку, subliminal channel – наявність у сеансах комунікації криптопротоколу чутливої інформації одного або декількох абонентів, яку може практично отримати Розробник закладки і не можуть отримати інші учасники системи.

Основні результати даного розділу були опубліковані у 76.

2.2 Обмеження наявних моделей клептографічної стійкості

Для оцінки криптографічної стійкості наразі розповсюдженими є два основні підходи:

1. Зведення до відомих теоретичних примітивів з відомою асимптотичною складністю.

2. Зведення до відомих конструкцій, практично стійких до відомих методів криптоаналізу.

Проте для опису клептографічних механізмів такі підходи є недостатніми. Дійсно, якщо криптоаналіз лазівки зводиться до алгоритму поліноміальної складності (наприклад, за розміром ключа), це не означає відсутність такої лазівки, оскільки даний алгоритм може практично не обчислюватися наявними ресурсами.

Побудова каналу витоку шляхом модифікації реалізації стандартного протоколу – одна з найбільш вивчених клептографічних проблем. Однією із перших спроб формалізації клептографічного механізму є модель Янга та Юнга 52. У своїй роботі автори вводять поняття SETUP – Secretly Embedded Trapdoor with Universal Protection (таємно вбудований захищений канал витоку секрету). Окрім власне SETUP також розрізняють слабкий та сильний SETUP.

Визначення 2. *SETUP криптосистеми C називають таку її модифікацію C', що:*

1. *Інтерфейс взаємодії (вхідні та вихідні параметри) з C' відповідає заявленому для C стандарту.*
2. *C' ефективно обчислюється.*
3. *Секрет розробника наявний лише у нього і не міститься в C'.*
4. *Секретна інформація, яку C' надсилає до каналу витоку, може бути ефективно розшифрована лише розробником (розробник використовує свій секретний ключ для розшифрування).*
5. *Ніхто, окрім розробника, не може розрізнити за поліноміальний час виходи систем C' та C.*
6. *Після аналізу модифікованої реалізації (отримання всіх необхідних алгоритмів, деструктивний реверс інжиніринг) неможливо відновити попередні або спрогнозувати майбутні ключі.*

Визначення 3. *Слабким SETUP називається SETUP для якого розрізнити виходи C та C' може не лише розробник, а і власник модифікованої реалізації*

Визначення 4. *Сильним SETUP називається SETUP з додатковою умовою – неможливо відновити попередні та спрогнозувати наступні ключі не лише після деструктивного аналізу, але і у випадку аналізу стану системи в режимі реального часу.*

Важливою характеристикою SETUP механізму є ширина пропускання (bandwidth): (n,m)-схемою витоку секрету називається SETUP механізм, якому необхідно передати m повідомлень для здійснення таємного витоку n повідомлень.

Наразі всі відомі SETUP механізми побудовані на базі асиметричних криптопримітивів для багатьох криптосистем: систем цифрового підпису (77), системах, що базуються на проблемах дискретного логарифму (5) та RSA (52).

Модель клептографічного механізму за визначенням 2 має ряд недоліків:

1. Модель зводить лазівку до односторонньої функції, задача обернення якої є експоненційною за входом (модель теоретичної складності алгоритмів). Отже, в ній одразу відкидаються підходи, що базуються на практично складних задачах.

2. Модель є не достатньо формалізованою – поняття «відповідності вхідних на вихідних параметрів» можна трактувати по різному.

3. Модель не покриває закладки у симетричних криптопримітивах.

4. Також не покриваються випадки, коли порушення криптографічних властивостей системи не призводять до витоку секрету: клептографічні механізми підробки підписів, побудови колізій геш функцій, дешифрування без знання ключа зашифрування тощо.

Це створює проблему пошуку більш адекватної моделі для аналізу практичних клептографічних конструкцій.

Далі у розділі пропонуються нові формальні моделі опису клептографічних систем. Зокрема пропонується загальна статична модель розподілу інформації клептографічного механізму, часткова модель для протоколів, що базуються на протоколі типу «запит-відповідь» та розширена класифікація клептографічних механізмів.

2.3 Моделі стійкості

2.3.1 Статична модель розподілу інформації

Для оцінки стійкості клептографічної системи перш за все необхідно визначитися з моделлю складності. В криптографії виділяють декілька основних моделей: теоретико-обчислювальна, теоретико-інформаційна, зведення до практичних криптопримітивів, моделі на основі практичної складності виконання алгоритмів тощо. Ці моделі не завжди повністю

відображатимуть картину стійкості клептографічного механізму. Наприклад, якщо складність побудови лазівки симетричного шифру є екпоненційною відносно розміру внутрішнього стану, але тим не менш побудувати лазівку можливо за практичний час, то такий метод побудови лазівки має право на життя. З іншого боку, якщо асимптотична складність виявлення лазівки є поліномом високої степені від розміру внутрішнього стану, але для заданої параметрів криптосистеми цей час є непрактично великим, то можна говорити про стійкість лазівки до виявлення в практичному сенсі.

У роботі 78 використовується модель, що базується на практичній складності виконання алгоритмів.

У даній роботі пропонується використовувати схожу модель складності, що базується на практичній складності обчислення та їдеях роботи Berbain та Gilbert 78.

Спершу введемо поняття практичного класифікатору ансамблів та практичної нерозрізненості (на противагу поліноміальній нерозрізненістю 79). Необхідність запропонованого поняття випливає з того факту, що відомі клептографічні лазівки є практичними односторонніми функціями, що загалом не підпадають під модель теоретико інформаційної чи обчислювальної складності.

Визначення 5. *(Практичний класифікатор ансамблів) Нехай існує два ансамблі* $E = \{e_1, e_2, ...\}$ *та* $E' = \{e'_1, e'_2, ...\}$, $e_i, e'_i \in S$, *де* S *– скінчена множина. Додатково заданий часовий поріг* t, *максимальний час, що відпускається на виконання алгоритму (наприклад,* $t = 2^{80}$*).*

Класифікатором ансамблів називатимемо ймовірнісний розпізнавальний алгоритм A_t, *обмежений часом роботи* t, *що для вектору довжини* l, $\vec{\nu} \in E^l \cup E'^l$, *повертає значення:*

$$\begin{cases} A_t(E, \vec{\nu}) = 1 & \Leftrightarrow \vec{\nu} \in E^l \\ A_t(E', \vec{\nu}) = 1 & \Leftrightarrow \vec{\nu} \in E'^l \end{cases}$$

Також визначимо перевагу (advantage) практичного класифікатора у розпізнаванні ансамблів як $Adv_{A_t}(E, E', l) = |P\{A_t(E, \vec{\nu}) = 1\} - P\{A_t(E', \vec{\nu}) = 1\}|$, *де* $\vec{\nu} \in E^l \cup E'^l$ *- випадковий вектор довжини* l.

Окремо слід звернути увагу на те, що ми розглядаємо алгоритм A_t не як абстрактний обчислювач, а як оптимізована реалізація на визначених моделлю аналітика ресурсах. Тобто, за один і той же час t можливо отримати різні набори можливих алгоритмів в залежності від передбачених потужностей та відомих методів оптимізацій.

Визначення 6. *(Практична нерозрізненість) Два ансамблі E та E' називаються практично нерозрізнені (practical indistinguishable) для заданого параметру безпеки t, якщо максимальна перевага у розпізнаванні для всіх практичних класифікаторів A_t буде незначною відносно параметра безпеки: $Adv(E, E') = \max_{l,A_t}\{Adv_{A_t}(E, E', l)\} < \varepsilon(t)$, де $\varepsilon(t)$ – порогове значення для «незначної» ймовірності (наприклад, $\varepsilon(t) = 2^{-40}$ при $t = 2^{80}$).*

Надалі, для практично нерозрізнених множин E та E' використовуватимемо позначення $E \simeq_t E'$.

Використовуючи такий підхід до визначення складності спробуємо побудувати формальну модель клептографічної системи.

Визначення 7. *(Статична модель розподілу інформації криптопротоколу). Зрізом розподілу інформації криптопротоколу називатимемо кортеж $\langle S, E(\cdot), R, F(\cdot)\rangle$, де*
S – множина усіх можливих конфігурацій стану (конфігурація включає всі дані, які відомі на момент зрізу: ключі, допоміжні змінні, константи тощо)
$E : S \to V^n$ – ін'єктивна ймовірнісна функція, що кодує конфігурацію стану у двійковий вектор довжини n
R – множина ролей криптосистеми
$F : R \to V^n$ – функція, що для кожної ролі повертає двійкову маску, що визначає доступ до конфігурації стану, а саме: роль r_i володіє інформацією $E(s) \otimes F(r_i)$ ($\oplus$ – порозрядне додавання двійкових векторів за модулем 2, $\otimes$ – порозрядне множення за модулем 2). Для зручності позначимо $f_i = F(r_i)$. При цьому на функцію кодування E та функцію доступу F накладається така додаткова умова: $\forall i = 0..|R| - 1$, $\forall f \in V^n : f \neq 0, f \otimes f_i = 0$, $\max_{A_t} \left| P\{A_t(E(s) \otimes f_i) = E(s) \otimes f\} - \frac{1}{|\mathfrak{U}_f|} \right| < \varepsilon(t)$, $s \overset{rand}{\in} S$, $\mathfrak{U}_f = \{E(s) \otimes$

$f|s \in S\}$, A_t – *реалізація рандомізованого алгоритму, обмежена часом виконання* t.

Проілюструємо статичну модель на прикладі функції хешування з сіллю $H(Salt, Message)$:

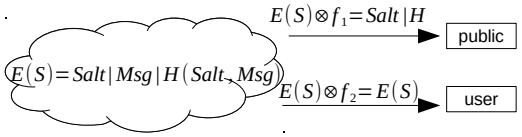

Рисунок 3. Функція хешування з сіллю

де $Salt$ – сіль, Msg – повідомлення.

Або ж, якщо маємо систему узгодження секрету на основі криптографії з відкритим ключем, то схема може виглядати так:

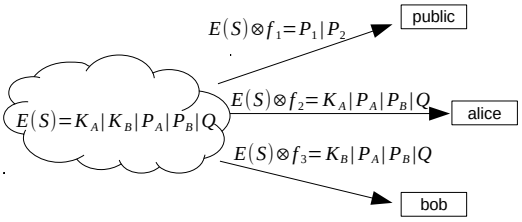

Рисунок 4. Протокол узгодження ключа

де K_A, K_B – секретні ключі, p_A, p_B – відкриті ключі, Q – спільний секрет.

Визначення 8. *(Статична модель розподілу інформації клептографічного механізму). Зрізом розподілу інформації криптосистеми з вбудованим клептомеханізмом називатимемо кортеж* $\langle \omega_0, S, E(\cdot), R, F(\cdot) \rangle$, *де* $S, E(\cdot), R, F(\cdot)$ – *мають той же сенс, що і для криптосистеми (див. визначення 7),*
$\omega_0 \in \Omega$ – *секрет розробника. При цьому мають виконуватися умови:*

1. $\forall i = 0..|R|-1$, $\nexists f \in V^n : f \otimes \overline{f_i} \neq 0$, $\max_{A_t^\omega} P\{A_t^\omega(E(s) \otimes f_i) = E(s) \otimes f\} < \frac{1}{|\mathfrak{U}_f|}$, $s \overset{rand}{\in} S$, $\mathfrak{U}_f = \{E(s) \otimes f | s \in S\}$, $\omega \overset{rand}{\in} \Omega$, A_t^ω – *реалізація рандомізованого алгоритму з додатковим параметром* ω, *обмежена часом виконання* t.

2. $\exists j = 0..|R| - 1$, $\exists f \in V^n : f \otimes \overline{f_i} \neq 0$, $\max_{A_t^{\omega_0}} P\{A_t^{\omega_0}(E(s) \otimes f_i) = E(s) \otimes \overline{f_i}\} > \varepsilon(t)$, $s \overset{rand}{\in} S$,

Наведена модель є доволі загальною, що дозволяє описати широкий клас алгоритмів з вбудованими клептомеханізмами. Проте її загальність призводить до складності її практичного застосування. Тому далі у роботі розглядатимуться моделі для вужчих класів криптосистем, що дозволить застосовувати їх на практиці.

2.4 Класифікація клептографічних механізмів

Клептографічні механізми різняться за сценарієм побудови, за способом захисту каналу розробника, за рівнем абстракції тощо. Наразі, відомі роботи, що торкаються проблем клептографії, фокусуються на окремих алгоритмах з потенційною закладкою або побудовою конкретних протоколів з каналами непомітного витоку секрету, тож різноманітність методів певною мірою розмиває загальну картину напряму.

Однією зі спроб класифікації клептографічних механізмів є робота Б. Шнаєра 80, в якій представлена таксономія різних типів вразливостей реалізацій криптосистем. Для систематизації уявлень про можливі форми існування клептографічних механізмів в даній роботі пропонується удосконалена та адаптована до клептографічних механізмів таксономія Шнаєра.

Таблиця 5

Класифікація клептографічних механізмів

Класифікація	Тип клептомеханізму	Приклади
	відкриті стандартизовані реалізації	програмні бібліотеки, схемотехнічні описи, специфікації алгоритмів

За закритістю реалізації	закриті реалізації	пропрієтарні програмні продукти з обфускацією програмного та потоків даних
	апаратні реалізації	криптографічні мікроконтролери, апаратні криптомодулі
За наслідками аналізу	недеструктивні (з можливістю подальшого використання)	аналіз програмних компонент, маскованих апаратних компонент, логічний аналіз специфікацій тощо
	деструктивні (без можливості подальшого використання)	аналіз криптографічних контролерів, вбудованої EEPROM-пам'яті тощо
За рівнем побудови	модифікація готових криптосистем	додавання каналу витоку в реалізацію системи, наприклад, атака BEAST протоколу TLS1.0 і нижче
	побудова нових криптоалгоритмів з вбудованими закладками	приклади ймовірно таких алгоритмів: DES, DualEC DRGB
За способом впровадження	модифікація працюючих криптосистем на стороні користувача	через троянські програми, використання вразливостей веб інтерфейсу та інші кібератаки
	відкрите розповсюдження клептографічних модифікацій реалізацій алгоритмів	наприклад, у вигляді програмних компонентів
	розповсюдження пропрієтарних закритих криптосистем у вигляді апаратних модулів	

	лобіювання стандартизації клептографічних криптосистем, нав'язування їхнього використання через правові механізми, корпоративні політики чи маркетингові кампанії	

2.5 Модель протоколу типу «запит-відповідь»

Модель криптосистеми визначення 7 є доволі загальною, але її складно застосовувати на практиці. Проте для великої кількості базових протоколів, наприклад, одно- та двохпрохідні протоколи автентифікаці або протоколи доведення з нульовим розголошенням, можливо провести редукцію до протоколу типу «запит-відповідь».

Протокол типу «запит-відповідь» моделюється грою з $N > 1$ учасниками, один з яких є оракулом.

Клептографічний варіант протоколу також включає ще одного учасника – розробника, з яким один з учасників знаходиться у змові.

Визначення 9. *(Клептографічний механізм на базі протоколу «запит-відповідь») Клептографічним варіантом базового протоколу типу «запит-відповідь» називатимемо гру між 3-ма учасниками (Alice, Bob та Dev) з такими правилами:*

1. *Сторони Alice та Dev можуть бути змовниками.*

2. *Сторона Alice очікує на запит сторони Bob, після якого вона має повернути відповідь у форматі, що заздалегідь узгоджений з Bob. Задачами Alice є:*

(а) Сформувати відповідь, що кодує один біт інформації, яку сторона Dev може ефективно відновити.

(б) Кодування має відбуватися таким чином, щоб сторона Bob не могла розпізнати факту передачі.

3. *Сторона Bob посилає довільний запит стороні A і отримує відповідь. Задачею Bob є виявлення факту передачі додаткової інформації стороні Dev.*

4. *Сторона Dev пасивно прослуховує трафік між Alice та Bob. Задачею сторони Dev є відновлення біту інформації від сторони Alice на основі перехоплених повідомлень.*

Також вважатимемо дійсними такі припущення:

1. *Сторона Bob не змовляється ні з Alice ні з Dev.*

2. *Усі сторони використовують стандартні базові криптопримітиви та не містять закладок.*

3. *Alice та Dev не використовують додаткові стеганографічні канали, що базуються на часових затримках протоколу, збоях у роботі тощо.*

З точки зору сторони Bob, яка має виявляти факт витоку секрету, протокол може бути представленим у вигляді кортежу $\langle D_t, V, U\rangle$, де V – множина запитів сторони Bob, U – множина відповідей оракула Alice, $D_t : V \times U \to \{0,1\}$ – ймовірнісний алгоритм, обмежений часом роботи t, що перевіряє відповідь оракула на відповідність протоколу.

З боку розробника Dev, до кортежу також додається $R_t^\omega : V \times U \to \{0,1\}$ – алгоритм, що розшифровує повідомлення, передане стороною Alice.

Визначення 10. *(Протокол «запит-відповідь», формальна модель) Протоколом типу «запит-відповідь» називатимемо кортеж $\langle D_t, V, U, A_t\rangle$, де:*
$D_t : V \times U \to \{0,1\}$ *– ймовірнісний алгоритм, обмежений часом роботи t, що перевіряє відповідь оракула на відповідність протоколу. Кожну коректну пару запит-відповідь алгоритм розпізнає з ймовірністю 1, тобто*

є алгоритмом типу Монте-Карло
V – множина запитів сторони Bob, U – множина відповідей оракула Alice
$A_t : V \to U$ – рандомізований алгоритм оракула Alice без витоку секрету: $\forall v \in V : D_t(A_t(v)) = 1$

Визначення 11. *(Канал витоку на базі протоколу «запит-відповідь», формальна модель) Протоколом типу «запит-відповідь» з каналом витоку називатимемо кортеж з моделі визначення 10 $\langle D_t, V, U, A_t, R_t^\omega, A_t^\omega \rangle$ з додатковими параметрами R_t^ω та A_t^ω (канал витоку), де D_t, V, U, A_t мають той же сенс, що і в моделі за визначенням 10:*
$A_t^\omega : V \times \{0,1\} \to U$ – рандомізований алгоритм оракула Alice з витоком секрету: $\forall v \in V, s \in \{0,1\} : D_t(A_t^\omega(v,s)) = 1$, $v \overset{rand}{\in} V, s \in \{0,1\} : P\{R_t^\omega(v, A_t^\omega(v,s)) = s\} > 1/2 + \varepsilon(t)$
$R_t^\omega : V \times U \to \{0,1\}$ – ймовірнісний алгоритм, що декодує повідомлення, передане стороною Alice, на основі секрету ω.

Додатково накладаються умови секретності та непомітності каналу: множини $H = \{\langle v,u\rangle | v \in V, u \in U : u = A_t(v)\}$, $H_0 = \{\langle v,u\rangle | v \in V, u \in U : u = A_t^\omega(v,1)\}$ та $H_1 = \{\langle v,u\rangle | v \in V, u \in U : u = A_t^\omega(v,0)\}$ є попарно практично нерозрізненими: $H \simeq_t H_0 \simeq_t H_1$

Також додається припущення про відсутність інформації для розробника з виходу алгоритму A_t: $|P\{R_t^\omega(v, A_t(v)) = 0\} - P\{R_t^\omega(v, A_t(v)) = 1\}| < \varepsilon(t)$.

Розроблена модель необхідна в подальшому для вирішення головної проблеми: побудови криптопротоколу з доведеною відсутністю каналів витоку.

Для формулювання теорем у роботі використовується поняття «рівність рандомізованих алгоритмів», що не слід плутати із поняттям еквівалентних алгоритмів.

Визначення 12. *(рівність рандомізованих алгоритмів) Рандомізовані алгоритми $A_t, A'_t : \mathfrak{L}_1 \to \mathfrak{L}_2$, обмежені часом роботи t називатимемо однаковими ($A_t = A'_t$), якщо $P\{A_t(l) \neq A'_t(l)\} < \varepsilon(t)$, $l \overset{rand}{\in} \mathfrak{L}_1$.*

В роботі сформульована та доведена теорема про необхідну умову наявності каналу витоку в моделі протоколу типу «запит-відповідь» (визначення 11).

Теорема 1. *(Необхідна умова наявності каналу витоку) Якщо в протоколі за визначенням 10 існує канал витоку, то* $\exists A_t, A'_t : V \to U$, $A'_t \neq A_t$, *що* $P\{D_t(v, A_t(v)) = 1\} > 1 - \varepsilon(t)$ *і* $P\{D_t(v, A'_t(v)) = 1\} > 1 - \varepsilon(t)$

Доведення. Доведення конструктивне. Розглянемо алгоритми $A^\omega_0(v) \equiv A^\omega(v, 0)$ та $A^\omega_1(v) \equiv A^\omega(v, 1)$, $v \in V$. Тоді виконуватимуться умови теореми:

1. $P\{D_t(A^\omega_0(v)) = 1\} > 1 - \varepsilon(t)$ та $P\{D_t(A^\omega_1(v)) = 1\} > 1 - \varepsilon(t)$. Дійсно, згідно з визначенням 11 накладається вимога на непомітність каналу, тобто $\varepsilon(t) > Adv(A^\omega_1, A_t) \geq P\{D_t(A_t(v))\} - P\{D_t(A^\omega_1(v))\} = 1 - P\{D_t(A^\omega_1(v))\} \Rightarrow P\{D_t(A^\omega_1(v))\} > 1 - \varepsilon(t)$ (для A^ω_0 доведення аналогічне).

2. $A^\omega_0 \neq A^\omega_1$. Доведемо від супротивного: нехай $A^\omega_0 = A^\omega_1$, тоді $P\{A^\omega(v, 0) = A^\omega(v, 1)\} = 1 - \sigma, \sigma \in [0, \varepsilon(t))$ згідно з визначенням 12. Отже, $P\{R^\omega(A^\omega(v, s)) = 0\} = P\{R^\omega(A^\omega(v, s)) = 1\} = \frac{1}{2}$ з ймовірністю $p = 1-\sigma$. З іншого боку, $P\{A^\omega(v, 0) \neq A^\omega(v, 1)\} = \sigma$ і тому $\max_s P\{R^\omega(A^\omega(v, s)) = s\} = \xi, \xi \in (1/2, 1]$. З цього слідує, що повна ймовірність буде $\max_s P\{R^\omega(A^\omega(v, s)) = s\} = \frac{1}{2}(1 - \sigma) + \xi\sigma = \frac{1}{2} + \sigma(\xi - \frac{1}{2}) \in [\frac{1}{2}, \frac{1}{2} + \frac{\varepsilon(t)}{2})$, що суперечить властивостям ймовірнісного алгоритму R^ω_t визначення 11.

Отже, алгоритми A^ω_0 та A^ω_1 є прикладами алгоритмів A'_t та A_t з умови теореми, тож теорема доведена конструктивно. $\lhd$

В роботі сформульовано важливий наслідок теореми – достатні умови відсутності каналу витоку.

Наслідок. Нехай $\exists A_t, \forall v \in V : P\{D_t(v, A_t(v)) = 1\} = 1$ і $\forall A'_t : A'_t \neq A_t, P\{D_t(v, A'_t(v)) = 1\} = \sigma < 1 - \varepsilon(t)$. Тоді в протоколі неможливо побудувати канал непомітного витоку секрету. Більш того, у випадку передачі повідомлення секретним каналом, ймовірність виявлення факту цього складатиме $P \geq 1 - \sigma$.

Доведення. З $\exists A_t, \forall v \in V : P\{D_t(v, A_t(v)) = 1\} = 1$ і $\forall A'_t : A'_t \neq A_t, P\{D_t(v, A'_t(v)) = 1\} = \sigma < 1 - \varepsilon(t)$ випливає те, що $\forall A'_t, A''_t : A'_t \neq$

$A''_t, P\{D_t(v, A'_t(v)) = 1\} < 1 - \varepsilon(t) \cup P\{D_t(v, A''_t(v)) = 1\} < 1 - \varepsilon(t)$, наслідком чого є достатня умова відсутності каналу непомітного витоку секрету.
Нехай сеанс передачі повідомлення характеризується двома властивостями: $Leak \in \{0, 1\}$ – непомітний витік секрету відбувся та $Detect \in \{0, 1\}$ – витік був виявлений. Тоді повна ймовірність виявлення факту витоку буде $P = P\{Detect = 1|Leak = 0\} + P\{Detect = 1|Leak = 1\}$. Але оскільки $P\{Detect = 1|Leak = 0\} = 0$ (неможливо виявити витік у випадку, якщо він не відбувся), то $P = P\{Detect = 1|Leak = 1\} \geq P\{Detect = 1\} = Adv(A_t, A'_t) \geq |P\{D_t(v, A_t(v)) - P\{D_t(v, A'_t(v))\}| = 1 - \sigma$. $\lhd$

Наслідок теореми 1 є важливим науковим результатом, оскільки він визначає критерії, досягнення яких на етапі розробки криптосистемі забезпечує гарантовану відсутність таємних каналів витоку Розробника. В наступному розділі буде продемонстрована побудова криптопротоколів без SETUP з використанням даного результату.

2.6 Побудова криптосистеми стійких до побудови каналів витоку

2.6.1 Підходи до побудови криптосистеми

До інформаційних потоків, якими оперує реалізація однієї з сторін криптосистеми, належать джерела випадковості, канал прийому даних від інших сторін, канали передачі даних іншим учасникам, а також приховані канали витоків секрету, якщо розглядаємо у контексті клептографічних модифікацій. Для простоти вважатимемо, що наявні лише джерело випадковості (вхідний канал) і канали передачі іншим сторонам та розробнику (вихідні канали). Під таке спрощення підпадає, наприклад, асиметричний криптопротокол на кроці генерації сесійної пари асиметричних ключів.

До наявних зараз методів виявленням 63, 81, 82 лазівок належать моніторинг трафіку та контроль цілісності власне реалізації. Головними

недоліками цих методів є:

1. Моніторинг трафіку не здатен відслідкувати SETUP (одна з вимог до SETUP – неможливість спостерігачу відрізнити модифікацію його від оригіналу).

2. Контроль цілісності ускладнений тим, що в моделі SETUP зловмисних уже має доступ до реалізації для модифікації, отже ймовірно, він також матиме можливість обійти контроль (наприклад, перегенерувати підпис та замінити ключі перевірки або просто вимкнути перевірку).

3. Якщо навіть абонент має можливість контролювати цілісність, він не здатен проконтролювати цілісність реалізації свого опонента.

В роботі пропонується інший метод побудови системи, в основі якої лежить принцип покрокової перевірки роботи протоколу опонента.

Ключовим принципом роботи системи з доведеною відсутністю клептографічних лазівок є те, що жоден з абонентів системи не використовує в протоколах внутрішні джерела випадковості, а всі псевдовипадкові послідовності генеруються на базі публічних унікальних значень (лічильників) з механізмами доведення оригінальності (відсутності модифікацій). Це дозволяє забезпечити виконання достатніх умов, що є умовами наслідку теореми 1 про відсутність каналу непомітного витоку секрету.

Принципи системи з захистом від SETUP така:

1. Кожен учасник системи ідентифікується власною парою асиметричних ключів.

2. Якщо конкретному учаснику необхідно згенерувати випадкову послідовність, він генерує її виключно на основі свого ключа-ідентифікатора та унікальних відкритих значень (лічильників), що генеруються у домовлений та однаковий для всіх спосіб (наприклад, простий інкремент).

3. У випадку генерації такої послідовності, учасник має довести іншим сторонам протоколу чесність генерації.

Тобто, у випадку, коли протокол передбачає використання джерела випадковості, учасник має згенерувати псевдовипадкову послідовність на основі публічного лічильника, з використанням власного секретного ключа, без використання власних джерел випадковості, з неможливістю прогнозування іншими сторонами та можливістю доведення автентичності даних. Для такої задачі можуть бути застосовані схеми цифрового підпису без рандомізації, наприклад алгоритм цифрового підпису BLS (83), що будується на скінченних абелевих групах з визначеними білінійними відображеннями (наприклад, криптографічно стійкі еліптичні криві з визначеними функціями Вейля або Тейта (84)), або алгоритм цифрового підпису RSA без рандомізації.

Підхід до генерації псевдовипадкової послідовності з публічного лічильника, окрім захисту від клептографічних атак, має ще одну корисну властивість – він дозволяє отримати якісний генератор псевдовипадкових послідовностей у випадку відсутності власного джерела випадковості. Це актуально для віртуальних машин (доступ до багатьох апаратних ресурсів обмежений), систем IoT («інтернету речей», вимоги до малих затримок не дозволяють накопичувати достатню кількість випадкового матеріалу), різноманітних систем на базі контролерів без апаратної підтримки джерел випадковості, товстих клієнтах, коли велика частина логіки веб ресурсу перекладається на потужності клієнта при чому як правило засоби обробки такої логіки мають обмежений доступ до системних ресурсів.

2.7 Зменшення клептографічних можливостей криптопримітивів

2.7.1 Клептографічний потенціал

В даній роботі основна увага приділялася захисту стандартних криптопротоколів від зловмисної модифікації розробника, що дозволяє організувати непомітний витік секрету. Проте в основі дослідження лежить

важливе припущення – окремі криптопримітиви, що беруть учать у протоколі, не містять клептографічних механізмів.

В даному параграфі досліджуються підходи до оцінювання потенційних можливостей побудови клептографічної закладки саме у криптопримітиві (наприклад, шляхом лобіювання стандартизації клептомодифікацій). Головною метою дослідження є отримання метрики $\phi : \mathbb{A} \to \mathbb{R}$, де $\mathbb{A}$ – множина усіх можливих модифікацій алгоритму криптопримітива, при чому якщо $\phi(A) < \phi(B), A, B \in \mathbb{A}$, то «ризики» наявності клептомеханізму у примітиві B є вищими, тобто дана функція є метрикою «клептографічного потенціалу» криптосистеми.

Нехай, даний криптопримітив $Prim$ є функцією виду $Prim : Par \to Out$, де Par – простір вхідних параметрів (відкритий текст, ключі, стартові вектори тощо), Out – простір виходів потенційного алгоритму. Визначимо також множину всіх можливих функцій $\mathbb{F} = Par^{Out}, Prim \in \mathbb{F}$ та множину заборонених функцій $\hat{\mathbb{F}} \in \mathbb{F} : Prim \notin \hat{\mathbb{F}}$ (в контексті побудови криптопримітиву, заборонені функції – класи алгоритмів, які виключаються на етапі оцінки кандидата і раніше – алгоритми із слабкими криптографічними властивостями, вразливі до відомих типів теоретичних та практичних атак). Припустимо, Розробник володіє методом (алгоритмом), який реалізує ін'єктивну функцію, що на основі власного секрету Розробника з множини секретів Ω повертає алгоритм криптопримітиву з закладкою:

$$TrapGen_{Prim} : \Omega \to \mathbb{F}_{Prim} \backslash \hat{\mathbb{F}}_{Prim}.$$

Один з корисних інструментів для моделювання криптопримітиву, в тому числі і у клептографічному контексті, який далі використовуватиметься для визначення клептографічного потенціалу є інформаційна ентропія (для зручності розглядаємо шеннонівська 85). Вона дозволяє формалізувати невизначеність структури примітиву та її залежність від інформації, внесеної в структуру примітиву Розробником.

Нехай маємо два ансамблі X Y. Тоді безумовна ентропія Шеннона обчислюється як:

$$H(X) = \sum_{x \in X} p(x) log_2 \frac{1}{p(x)}. \tag{2.1}$$

Суть умовної ентропії $H(X|Y)$ в тому, що вона показує невизначеність, яка залишилася після реалізації змінної ансамблю Y. Умовна обчислюється як:

$$H(X|Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) log_2 \frac{1}{p(x|y)}. \tag{2.2}$$

Тобто, таким чином, якщо обчислювати ентропію структури примітиву за умовою інформації Розробника, вона є метрикою того, скільки у структурі примітиву залишається невизначеності після ініціалізації розробником.

Також в подальшому використовуватиметься властивість ентропії:

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y). \tag{2.3}$$

Варто зазначити, що далі у роботі, для зручності, аргументом безумовної ентропії виступає не ансамбль, а множина. При цьому припускаємо, що ймовірнісний розподіл є рівномірним, що відповідає максимальній невизначеності. Тепер визначимо власне клептографічний потенціал Розробника.

Визначення 13. *(Клептографічний потенціал). Для криптографічного примітиву $Prim$, вимоги до вигляду та властивостей якого задають множину функцій $R_{Prim} = \mathbb{F}_{Prim} \backslash \hat{\mathbb{F}}_{Prim}$, називатимемо клептографічним потенціалом кількість інформації, що закладена розробником у структуру примітиву, тобто:*

$$\phi(Prim) = H(R_{Prim}) - H(R_{Prim}|D) \leq log_2(|R_{Prim}|),$$

де $H(R_{Prim})$ – безумовна ентропія структури примітиву, $H(R_{Prim}|D)$ – ентропія структури примітиву за умови ініціалізації розробником.

Сенс такої метрики в том, що з загальної ентропії структури прибирається частина невизначеності, що не контролюється розробником. В дійсності, у більшості випадків умовна ентропія буде нульова, оскільки Розробник повністю контролює структуру примітиву. Проте можливі випадки його обмежень, про що йтиме мова далі.

Клептографічний потенціал є метрикою можливості (ризику) побудови Розробником закладки у примітиві. Дійсно, з ростом потужності множини

секретів Розробника зростає складність виявлення цього секрету методом прямого перебору, а для великої потужності множини секретів необхідною умовою є велике значення потенціалу і навпаки, низький клептографічний потенціал вказує на неможливість використання Розробником великої множини секретів, що призводить до неможливості побудови прихованої закладки.

Оцінити клептографічний потенціал за визначенням 13 є на практиці складною задачею, оскільки обмеження на множини допустимих та заборонених функцій найчастіше є нетривіальними. Більш того, в ряді випадків необхідно оцінювати клептографічні ризики для уже існуючих криптопримітивів, це ускладнює задачу визначення множини допустимих функцій. Більш практичним є поняття «клептографічної надлишковості», яке вводиться далі.

Розглянемо криптопримітив $Prim$ з множини допустимих алгоритмів $\mathbb{F}\backslash\hat{\mathbb{F}}$. Визначимо також відношення еквівалентності $R^{\simeq}_{Prim} \subset \mathbb{F}\backslash\hat{\mathbb{F}}$, що задає клас примітивів, які ми вважаємо можливими альтернативами кандидата $Prim$. Нехай також ми можемо побудувати відношення строгого порядку «$<$» на множині $R^{\simeq}_{Prim}$.

Визначення 14. *(Клептографічна надлишковість). Клептографічною надлишковістю криптопримітиву $Prim$ із класу альтернативних алгоритмів, заданого відношенням $R^{\simeq}_{Prim}$ називатимемо кількість інформації, що може бути закладена Розробником у структуру примітиву через вибір представника із класу альтернативних алгоритмів:*

$$\rho_{\simeq}(Prim) = H(R^{\simeq}_{Prim}) - H(R^{\simeq}_{Prim}|D) \leq log_2(|R^{\simeq}_{Prim}|),$$

Тобто, клептографічна надлишковість визначається подібним до клептографічного потенціалу чином, відмінність лише у тому, що замість множини допустимих функцій (яку складно отримати на практиці) використовується клас еквівалентних алгоритмів, який можна конкретно визначити.

Важливо зазначити, що відношення еквівалентності, від якого залежить значення клептографічної надлишковості, може обиратися різними способами. Проте чим більшу множину еквівалентних примітивів ми отримуємо,

тим точнішою оцінкою значення клептографічної надлишковості є для клептографічного потенціалу. Загалом, для криптопримітивів можна визначити певний мінімальний набір правил, що формують відношення еквівалентності.

Покажемо, що клептографічна надлишковість є оцінкою знизу клептографічного потенціалу.

Теорема 2. *Клептографічна надлишковість (визначення 14) є оцінкою знизу для клептографічного потенціалу (визначення 13):*

$$\phi(Prim) \geq \rho_{\simeq}(Prim)$$

Доведення. Згідно із властивостями ентропії, $H(X,Y) = H(Y) + H(X|Y)$, тому $H(R_{Prim}) - H(R_{Prim}|D) = H(R_{Prim}) - H(R_{Prim}, D) + H(D) = H(D) - H(D|R_{Prim})$. Оскільки при зменшенні множини R_{Prim} умовна ентропія монотонно не спадає, то $R^{\simeq}_{Prim} \subset R_{Prim} \Rightarrow H(R^{\simeq}_{Prim}) - H(R^{\simeq}_{Prim}|D) < H(R_{Prim}) - H(R_{Prim}|D)$. Отже клептографічна надлишковість визначення 14 дійсно є оцінкою знизу для клептографічного потенціалу (визначення 13). $\lhd$

В роботі наведені базові правила, що можуть визначати відношення еквівалентності криптопримітивів.

Для примітиву $Prim$ називатимемо еквівалентними:

1. Примітив $Prim$ після довільної заміни раундових констант (зі збереженням додаткових умов та криптографічних властивостей примітиву).

2. Примітив $Prim$ після довільної заміни блоків підстановок (зі збереженням додаткових умов та криптографічних властивостей).

3. Примітив $Prim$ після довільної заміни стартових векторів (зі збереженням додаткових умов та криптографічних властивостей).

4. Примітив $Prim$ після довільної заміни інших констант у структурі примітиву (зі збереженням додаткових умов та криптографічних властивостей).

5. Примітив $Prim$ після довільної заміни побітових операцій та нелінійних функцій (зі збереженням криптографічних).

Перелік правил формування відношення еквівалентності можна розширювати в залежності від типу примітиву та його базової структури. Прикладом додаткових умов можуть бути: заборона нульових констант; заборона констант, що не проходять певні тести випадковості; використання лише функції з максимальною нелінійністю в блока підстановки; константи лише прості числа певної довжини; викоритання лише констант, що будуються на основі лічильників (наприклад, 64 32-бітових констант $T_1..T_{64}$ геш функції MD5 не випадкові, а обчислюються як $T_i = \lfloor 2^{32}|sin(i)|\rfloor$) тощо.

Стосовно визначення строгого порядку на множині, головна вимога до впорядкування – Розробник конкретного примітива не повинен впливати на визначення порядку.

2.7.2 Приклади оцінки клептографічного потенціалу

Тривіальні випадки

Розглянемо два граничні випадки – найменшого та найбільшого потенціалу.

Перший випадок виникає тоді, коли вимоги до кандидату-криптопримітиву однозначно задають функцію $Cand_0 : Par \to Out$.

Очевидно, клептографічний потенціал за даних умов є $\phi(Cand_0) = log_2(|\mathbb{F}\backslash\hat{\mathbb{F}}|) = log_2(1) = 0$. Нульовий клептографічний потенціал означає, що побудувати канал витоку неможливо і це інтуїтивно зрозуміло – всі учасники конкурсу мають пред'явити однаковий алгоритм.

Інший випадок виникає тоді, коли вимоги до кандидату мінімальні. Припустимо, єдиною вимогою для криптографічного перетворення $Cand_\infty : Par \to Out$ є розмір входу та виходу, наприклад $|Par| = n$, $|Out| = m$. За таких обмежень, множина $\mathbb{F}\backslash\hat{\mathbb{F}}$ являє собою усі можливі відображення $Par \to Out$. Відповідно, клептопотенціал оцінюється як $\phi(Cand_\infty) = log_2(|\mathbb{F}\backslash\hat{\mathbb{F}}|) = log_2(m^n)$. Наприклад, якщо вхід та вихід – 128 бітові блоки, то $\phi(Cand_\infty) = log_2(2^{128^{2^{128}}}) = 2^{128+7} = 2^{135}$.

Параметризований фреймворк

Зазвичай, під час розробки ітеративного криптопримітиву використовується цілий ряд констант – заповнення S-блоків, матриць перестановки, регістрів ключового розкладу тощо. Деякі значення констант можуть бути слабкими і мають відкидатия. Проте виникає питання, яким чином вибирати ці константи із множини допустимих. В деяких алгоритмах ці константи генеруються у спосіб, що дає менше варіацій, а отже і меншу кількість інформації у структурі алгоритму (раундові константи MD4/MD5 як наближенні значення обчислення тригонометричних функцій, S-блок AES, що є інверсією полінома в полі Галуа), а в деяких алгоритмах вони обираються довільно (S-блоки алгоритму ГОСТ-28147-89, раундові константи геш-функції ГОСТ Р-34-11-12).

Фреймворком назвемо криптопримітив: $Cand_1 : Consts \times Par \rightarrow Out$ з додатковим аргументом із простору $Consts$, даний аргумент задається одноразово під час стандартизації однаковим для всіх користувачів криптосистеми.

Розглянемо дві стратегії вибору констант для криптопримітиву:

1. Обрати $c \in Consts$ використовуючи «просте перетворення» (наприклад, 32 отктетів стартового вектору MD5 згенеровані як послідовний інкремент числа 0x0 до 0xf, а потім декремент від 0xf до 0x0).

2. Обрати $c \in Consts$ випадковим чином.

У першому випадку маємо певний клептографічний потенціал $\phi_0 = \phi(framework)$ фреймворку, а у другому також наявна варіативність у константах: $\phi_1 = \phi(framework) + log_2(|Consts|)$. Отже, порівнюючи клептографічні потенціали примітивів, отримуємо висновок: з використанням другої стратегії вибору констант, клептографічний потенціал збільшується на $\Delta\phi = log_2(|Consts|)$, тобто ризики наявності в такому примітиві клептомеханізму є вищими, ніж за першої стратегії, а максимальна довжина секрету Розробника збільшується на $log_2(|Consts|)$ бітів.

Оцінка клептографічного потенціалу ARX-шифрів

ARX криптографія 86 є доволі цікавою в контексті мінімізації можливого клептографічного потенціалу, оскільки такі примітиви не містять блоків підстановки (що є основним джерелом варіативності) та, часто, інших констант. Менший, в порівняні із фейстелівськими примітивами, клептографічний потенціал зменшує ризики щодо побудови закладки Розробником.

2.7.3 Шляхи зменшення клептографічного потенціалу

В даній роботі пропонуються два підходи до зменшення клептографічного потенціалу у криптопримітивах на етапі проектування. Перший підхід передбачає узгодження між учасниками системи зв'язку (точніше, між організаціями, що відповідальні за впровадження засобів безпеки зв'язку) однозначного методу генерації усіх випадкових параметрів системи. Другий підхід включає у процес розробки алгоритму етап узгодження структури криптопримітиву таким чином, щоб жодна із сторін не мала контролю над нею, таким чином, хоча структура і згенерована з використанням джерел випадковості абонентів, проте вона не може бути використана для побудови лазівки одним із учасників.

Використання генератору констант

Складові підходу до зменшення клептографічного потенціалу нового криптопримітиву:

1. Під час розробки криптопримітиву використовується спеціальний стандартизований генератор констант.

2. Заздалегідь визначається джерело початкового заповнення генератора – це мають бути відкриті дані з низькою ентропією джерела та унікальні для кожного алгоритму. Наприклад, дата першої публікації та прізвища авторів, закодовані у бітовий рядок.

3. Припускається, що ГК базується на криптографічно сильному ГПВЧ та його дизайн не містить лазівок.

4. За необхідності генерації констант (раундових констант, коефіцієнтів ключового розкладу, стартових векторів, блоків підстановок тощо) ГК ініціалізується лічильником, що є унікальним для кожного криптопримітиву. Далі, псевдовипадкова послідовність використовується для формування констант.

Один із можливих варіантів генератора констант зображений на рисунку 5.

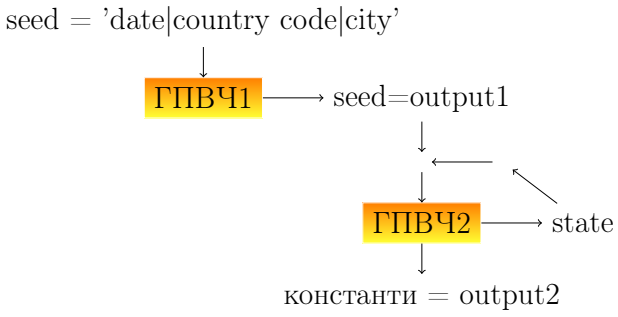

Рисунок 5. Генератор констант із зменшеним клептографічним потенціалом

Кроки зменшення клептографічного потенціалу примітиву:

1. Майбутній криптопримітив проектується у вигляді фреймворку, тобто конструкції, у якій константні параметри подаються у вигляді неініціалізованих змінних. У випадку існуючого примітиву будується клас альтернативних алгоритмів, що теж подається у вигляді фрейморку із додатковими обмеженнями на константи.

2. Задаються алгоритми детермінованої генерації констант для констант, що мають додаткові обмеження: матриці перестановки, бент-функції, поліноми n-го степеня.

3. У процесі власне ініціалізації відбувається ініціалізація генератора констант. При цьому зерном ГПВЧ1 є унікальний для примітива лічильник, наприклад, конвертована у бітовий рядок дата публікації, ідентифікатор країни тощо. ГПВЧ1 працює одну ітерацію.

4. Отриманий вихід генератора використовується як зерно для ГПВЧ2.

5. Вихід генератора ГПВЧ2 використовується для ініціалізації фреймфорка.

6. Для констант, що мають задовольняти додаткові статистичні чи структурні обмеження, генерація констант відбувається детерміновано відповідними заданими методами, з використанням виходу ГПВЧ2.

Тепер оцінимо зменшення клептографічного потенціалу криптопримітиву, ініціалізованого Розробником фреймворку довільним чином та з використанням генератору констант.

Нехай множина можливих констант – $\mathbb{M}_c$. Нехай відношення еквівалентності задається так: усі криптопримітиви, що утворені з фреймворку усіма можливими константами – альтернативні. Для того, щоб задати порядок на класі використовуватимемо Генератор констант. Нехай, константи можуть повністю задаватися за одну ітерацією Генератора констант. Тоді відношення порядку визначимо як $Prim_A < Prim_B$, якщо параметри $Prim_A$ згенеровані на більш ранній ітерації. Тобто $min_A(Prim_A)$ буде найперший примітив, згенерований з фреймворку. Тоді клептографічна надлишковість криптопримітиву із довільною ініціалізацією констант буде $\rho = log_2|\mathbb{M}_c|)$ і саме на це значення клептографічний потенціал буде зменшено при використанні Генератора констант.

Протоколи узгодження структури примітиву

Іншим підходом до побудови криптопримітивів з низькою клептографічною надлишковістю є використання протоколу ініціалізації, під час якого майбутні користувачі узгоджують структуру примітиву таким чином, що жодна із сторін не може безпосередньо її контролювати, а отже інформація структури не може бути використаною для закладки. Це є саме той випадок, коли умовна ентропія у виразі клептографічної надлишковості (визначення 14) є ненульовою.

Розглянемо процес конкурсного відбору криптопримітиву, де $E_1, E_2, ..., E_n$ – учасники конкурсу. Учасники пред'являють криптопримітиви $\{Prim_i\}_{i=1..n}$.

Задачею методу зменшення клептографічного потенціалу є побудова такого процесу конкурсного відбору за яким жодна зі сторін не може контролювати структуру примітиву, таким чином підвищуючи умовну ентропію та знижуючи клептографічний потенціал.

Визначення 15. *(Метод узгодження криптопримітиву з мінімізацією клептографічної надлишковості).*

Вхідні дані:

1. *Кандидати конкурсу* $\{Prim_i\}_{i=1..n}$*, що має такі додаткові властивості:*

   (а) *Усі константні параметри* $< c_0, c_1, ... > \in \mathbb{C}^i$ *: структури спочатку не задані (тобто, примітив подається у вигляді фреймворку).*

   (б) *Кандидат супроводжується алгоритмом, що реалізує бієктивне відображення* $Init^i : Z_k \to \mathbb{C}^i$*, що перетворює k-бітовий рядок в екземпляр структури. Значення* k *є потужністю множини можливих параметрів:* $k = log_2(|\mathbb{C}^i|)$*.*

2. *Незалежний арбітр, що включається в протокол роботи кандидатів та реалізує функцію* $\mathbb{P}^i(s_1, s_2, ...) \in Z_k$*, де* s_i *– таємна випадкова послідовність і-го учасника. Функція* $\mathbb{P}$ *та методи генерації послідовностей* $\{s_i\}$ *має задовольняти умову:* $\forall i = 1..n : H(\mathbb{P}^*(s_1, s_2, ...)| \cup_{j \neq i} s_j) = H(\mathbb{P}^*(s_1, s_2, ...))$*. Іншими словами, жодна кооперація із* $n-1$ *учасників протоколу не здатна контролювати вихід функції, вона не здатна зменшити невизначеність результату роботи протоколу до його завершення.*

Кроки роботи методу:

1. *Обирається фреймворк-переможець конкурсу* $Prim^* \in \{Prim_i\}_{i=1..n}$*. Процедура обрання є поза рамками даного методу.*

2. *Кожен учасник конкурсу* E_i*, таємно від інших, генерує та передає арбітру таємну послідовність* s_i*.*

3. *Арбітр обчислює та публікує результат роботи протоколу узгодження* $S = \mathbb{P}^*(s_1, s_2, ...)$.

4. *Обчислюються параметри фреймворку з опебліковаого значення* S *та з використанням алгоритму обчислення структури примітиву:* $< c_0, c_1, ... >= Init^*(S)$

5. *Ініціалізований фреймворк остаточно оголошується обраним криптопримітивом.*

В результаті роботи методу отримуємо криптопримітив з нульовою клептографічною надлишковістю (множина альтернативних алгоритмів визначається як фреймворк з можливими ініціалізаціями параметрів).

Теорема 3. *Нехай структура криптографічного примітиву* $Prim^*$ *узгоджена за методом визначення 15, поданий у вигляді фреймворку із множиною константних параметрів* $\mathbb{C}^*$. *Нехай також задано відношення еквівалентності* $' \simeq'$ *як множина криптопримітивів, що утворюються із фреймворку шляхом ініціалізації константних параметрів елементами множини* $\mathbb{C}^*$. *Припустимо, що Розробником може бути будь яка кооперація із* $n-1$ *учасників конкурсу. Тоді отриманий примітив матиме нульову клептографічну надлишковість відносно відношення* $' \simeq'$: $\rho_{\simeq}(Prim^*) = 0$.

Доведення. Згідно з визначенням 15 для будь-якого Розробника D (що є кооперацією $n-1$ учасників конкурсу) для функції $\mathbb{P}^*$ виконується умова $H(S|D) = H(S)$, де $S = \mathbb{P}^*(s_1, s_2, ...)$. Оскільки структура примітиву $Init^*(S)$ є бієктивним перетворенням виходу функції арбітра, то така ж рівність справедлива і для неї. Клептографічна надлишковість відносно $' \simeq'$ буде: $\rho_{\simeq}(Prim^*) = H(Init^*(S)|D) - H(Init^*(S)) = 0$. $\lhd$

Варто зазначити, в реальних реалізаціях даного методу узгодження, незалежний арбітр може бути не фізичною стороною комунікації, а певним криптографічним протоколом чи розподіленою системою (наприклад, на основі технології блокчейн). Проте пошук таких рішень та строге доведення нульової клептографічної надлишковості для них виходить за рамки даної роботи та є напрямом подальших досліджень.

2.8 Методи виявлення клептографічних механізмів

Поряд із методами побудови клептомеханізмів з доведеною відсутністю клептографічних механізмів у криптосистемах важливе значення також мають методи виявлення механізмів в існуючих стандартах та реалізаціях.

Існує принципово два різних підходи до клептографічного аналізу:

1. Пошук власне каналу витоку через виявлення модифікацій реалізації, використання класичних методів криптоаналізу примітивів та протоколів, аналіз поведінки абонентів тощо. Тобто вважається, що криптосистема містить клептографічний механізм, якщо цей механізм фактично знайдений.

2. Демонстрація відсутності способів доведення відсутності каналу витоку. Тобто апріорі вважається, що криптосистема містить вбудований клептографічний механізм, а задача аналізу зводиться до обґрунтування неможливості такої побудови. До даного методу, наприклад, належить метод оцінки клептографічного потенціалу 2.7, який не демонструє шляхи побудови закладки, але показує, що гіпотетично така можливість існує.

Переваги першого підходу – підхід є конструктивним, дозволяє застосувати конкретні методи усунення каналу витоку, можлива часткова автоматизація процесу виявлення (наприклад, контроль цілісності при завантажені ПЗ, антивірусні засоби тощо). Недоліками є наявність помилки 2-го роду, тобто реально існуючий клептомеханізм не виявляється в силу замалої обчислювальної потужності аналітика або обмеженість інформації про потенційну лазівку. Більш того, якщо розглядати стійкість клептомеханізму до виявлення, як задачу практичної нерозрізненості (визначення 6), принципово єдина можливість виявлення – це перехід до моделі аналітика, розширеної порівняно з моделлю аналітика, що закладається в дизайн лазівки. В випадку високого порогу передбачених потужностей аналітика це зробити практично неможливо.

Перевагами другого підходу є широке покриття алгоритмів із можливими каналами витоку та простіші (відносно першого підходу) методи до оцінки потенційних каналів. Проте недоліками є наявність помилки 1-го роду, що допускає наявність великих класів криптосистем, що визначаються як такі, що потенційно містять клептографічні механізми, хоча насправді вони відсутні. Фактично, за цим підходом, криптосистемами без лазівки вважаються лише ті, для яких формально доведено задоволення достатніх умов відсутності каналу витоку секрету.

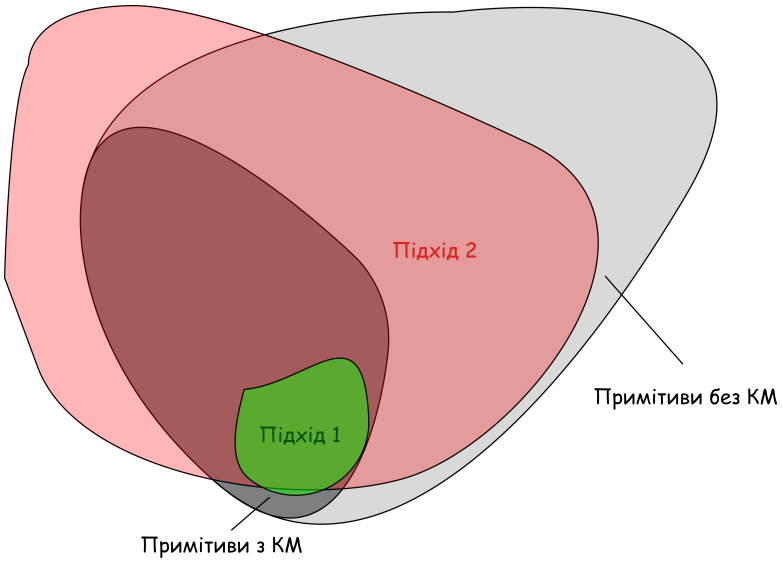

Рисунок 6. Ефективність підходів клептоаналізу для виявлення клептомеханізмів

Рисунок 6 ілюструє якісне співвідношення між підходами до клептографічного аналізу та їх покриттям множини криптопримітивів. Найменша область – область покриття методів прямого виявлення каналів витоку, що виявляють невелику частину простору алгоритмів із вбудованим клептографічним механізмом. Натомість, покриття методів другого підходу покриває значно більшу область, проте вони також реагують на примітиви, що в дійсності лазівок не містять.

2.9 Базова схема побудови гібридних криптосистем без клептомеханізмів

Оскільки головною метою роботи є підвищення рівня захищеності гібридної криптосистеми за рахунок зменшення клептографічних ризиків, покажемо місце окремих напрямів клептографічних досліджень (клептоаналіз, клептосинтез та побудова криптосистем з доведеною відсутністю клептомеханізмів) у процесі підвищення безпеки гібридної криптосистеми.

Побудова гібридних криптосистем сьогодні тісно пов'язана із процесом розробки програмних та програмно-апаратних комплексів, що реалізують дані криптосистеми. Однією з проблем побудови є те, що у зв'язку з високими темпами впроваджень технічних рішень наразі широко розповсюджені «гнучкі» підходи до розробки. Це означає, що на практиці важко впровадити строгий процес контролю безпеки рішень (в т.ч. і в клептографічному контексті) на кожному кроці життєвого циклу продукту. Фактично, програмне проектування, аудит захищеності продукту та провадження заходів щодо зменшення клептографічних ризиків є паралельними та неперервними процесами. В свою чергу, процес підвищення рівня інформаційної захищеності (в клептографічному контексті) також може складатися з паралельних процесів, що зручно вкладається в концепцію мікросервісної архітектури.

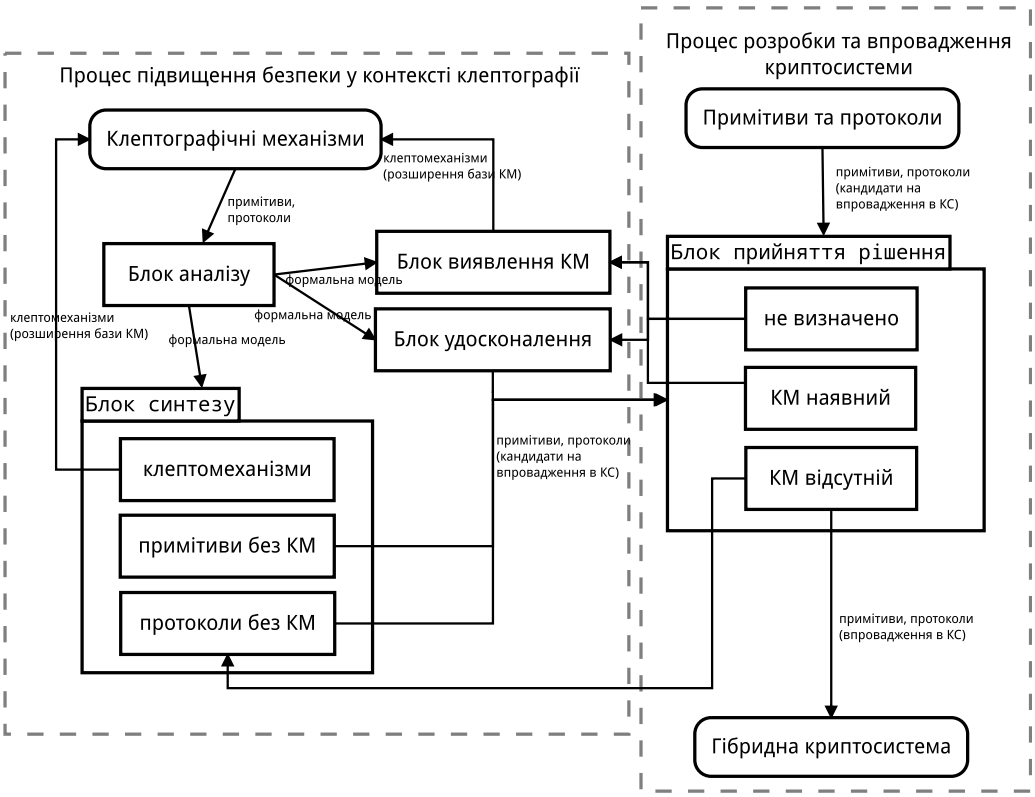

Рисунок 7. Процес зменшення клептографічних ризиків гібридної криптосистеми

В даній схемі вихідними артефактами є база відомих клептографічних механізмів, база відомих криптографічних примітивів та протоколів та власне цільова гібридна криптосистема. Основа схеми – неперервна взаємодія паралельних процесів (блоків), кожен з яких відповідає за свою підзадачу. Пропонуються такі блоки в архітектурі схеми:

1. Блок аналізу – аналіз клептографічних механізмів, класифікація, побудова формальної моделі. На вході блоку – відомі клептографічні механізми, на виході – формальна модель.

2. Блок синтезу – побудова криптосистем з клептомеханізмами та з доведеною відсутністю закладок. Синтезовані клептомеханізми використовуються для розширення простору відомих клептомеханізмів для подальшого аналізу. На вході блоку – формальні моделі закладок та систем без них, на виході – криптосистеми з закладками та даведеною їх відсутністю.

3. Блок виявлення клептомеханізму – виявлення клептомеханізму та клептоаналіз, метою якого є розширення бази відомих клептомеханізмів. На вході блоку – криптосистема, що визначена як та, що містить закладку, на виході – запис в базу клептомеханізмів.

4. Блок удосконалення криптосистеми – модифікація криптосистеми із закладкою або без таким чином, щоб модифікована система була вільною від клептомеханізму. На вході блоку – криптосистема, що визначена як та, що містить закладку, на виході – модифікована криптосистема без закладки.

5. Блок прийняття рішень – визначає чи містить дана криптосистема лазівки.

Блок прийняття рішень – це ключовий елемент взаємодії клептографічних заходів із реальним процесом розробки. Фактично, тут відбувається прийняття ризиків пов'язаних з клептографічними атаками. Наприклад, якщо блок визначає заданий криптопримітив як той, що не містить лазівки, це означає лише те, що в даному випадку використання припускається, що криптопримітив не містить лазівки, а відповідні ризики прийняті.

Наведена вище схема є одним з простих можливих процесів покращення функціонуючої системи з відносно невеликими економічними затратами, порівняно із впровадженням повного циклу розробки безпечного програмного забезпечення, і може застосовуватися до багатьох практичних процесів розробки.

2.10 Висновки до розділу 2

У даному розділі була показана неадекватність існуючих моделей криптографічної стійкості у застосуванні в оцінці стійкості клептографічних механізмів.

Були отримані такі теоретичні результати:

1. Введене поняття практичного класифікатору ансамблів та практичної нерозрізненості алгоритмів (на противагу поліноміальній нерозрізненості).

2. Запропонована статична модель розподілу інформації криптопротоколу.

3. Створена загальна класифікація клептографічних механізмів.

4. Запропонована формальна модель базового протоколу типу «запит-відповідь» із клептографічним каналом витоку.

5. Сформульована концепція побудови протоколів з доведеною відсутністю каналів непомітного витоку секрету.

6. Сформульована та доведена теорема про необхідні умови наявності каналу витоку в протоколі.

7. Введено поняття клептографічного потенціалу як міри потенціальної наявності клептографічного механізму (закладки) у криптопримітиві.

3 Застосування методів клептографії в побудові клептомеханізмів

3.1 Розробка криптосистем з доведеною відсутністю каналів витоку секрету

У розділі 2 був запропонований метод побудови (модифікації) криптопротоколів з доведенням відсутності клептографічного каналу витоку секрету. У даному розділі цей підхід демонструватиметься на двох прикладах – протокол генерації випадкового запису nonce та протокол асиметричного узгодження ключа.

Ключовим принципом роботи методів є те, що жоден з абонентів системи не використовує в протоколах внутрішні джерела випадковості, а всі псевдовипадкові послідовності генеруються на базі публічних унікальних значень (лічильників) з механізмами доведення оригінальності (відсутності модифікацій). Це дозволяє забезпечити виконання достатніх умов, що є умовами наслідку теореми 1 про відсутність каналу непомітного витоку секрету.

В роботі запропоновані два базові протоколи для демонстрації методу: протокол генерації випадкового запиту nonce (що використовується у багатьох протоколах, зокрема протоколах аутентифікації) та протокол узгодження секретного ключа на базі задачі Діффі-Хеллмана.

3.1.1 Схема випадкового запиту nonce

Протокол випадкового запиту nonce є базовим протоколом, що використовується майже у будь-якій системі аутентифікації для запобігання атак повторів.

Розглянемо базову схема протоколу випадкового запиту nonce.

Базова схема протоколу випадкового запиту nonce.

Вихідні дані: абоненти Alice та Bob.

Кроки роботи протоколу:

1. Alice генерує унікальний одноразовий запит nonce та відправляє Bob.

2. Bob посилає відповідь «запит прийняв».

З точки зору клептографії, такий запит може слугувати контейнером з високою ємністю (немає чіткого обмеження на довжину nonce) для таємної передачі секрету Розробнику (у випадку зловмисної модифікації). З огляду на це, є ризик отримати канал витоку секретного ключа через запит.

Нехай, в певному протоколі одним з кроків є передача випадкової послідовності (наприклад, як протидія до атак повторення у протоколах автентифікації). Якщо ця послідовність випадкова та має довжину r бітів, розробник, що модифікує систему, може використовувати r бітів як стегоконтейнер для подальшого витоку, наприклад, секретного ключа. Тож задачею нової схеми є унеможливлення зробити це непомітно.

Модифікація протоколу генерації nonce з викриттям стегоконтейнеру.

Вихідні дані:

1. Абоненти Alice та Bob.
2. Асиметрична криптосистема з простором K секретних ключів та простором Q публічних.
3. $Sign : K \times \{0,1\}^* \to B$ – функція генерації цифрового підпису без рандомізатору, B – простір підписів.
4. $Verify : Q \times \{0,1\}^* \times B \to \{0,1\}$ – функція перевірки цифрового підпису.
5. пара асиметричних ключів абонента $(k_A, p_A), k_A \in K, p_A \in Q$.
6. $\psi : \{0,1\}^* \to \{0,1\}^*$, $\psi_0 : Time \to \{0,1\}^*$ алгоритми генерації нового значення та ініціалізації лічильника.

Кроки роботи протоколу:

1. Абонент Alice обчислює наступне значення лічильника $ctr_i = \psi ctr_{i-1}$. Якщо попереднього значення не існує, можливе його створення, наприклад, на основі мітки часу $ctr_0 = \psi_0(time)$.

2. Абонент Alice обчислює $nonce = Sign(k_A, ctr_i)$ та передає значення $ctr_i|nonce$.

3. Bob перевіряє факт того, що не були задіяні власні джерела ентропії: $Verify(p_A, ctr_i, nonce) = 1$, $ctr_i = \psi(ctr_{i-1})$.

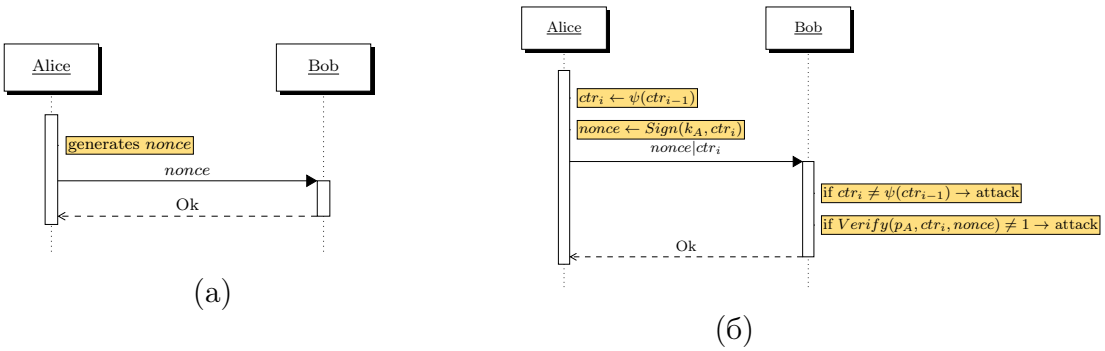

Рисунок 8. Схема генерації nonce: а) базова та б) вдосконалена

Можливі зловмисні сценарії:

1. Зловмисник вгадує (прогнозує) випадкову послідовність до її публічної появи

2. Зловмисник відновлює секретний ключ сторони з перехоплених у відкритому каналі даних

3. Розробник модифікує сторону А таким чином, щоб використовувати випадковий запит як стеганографічний контейнер

У першому випадку, зловмиснику необхідно зі значень ctr_i та p_A отримати значення $Sign(k_A, ctr_i)$, тобто обчислити значення цифрового підпису без знання секретного ключа, що зводиться до задачі підробки цифрового підпису.

У другому випадку, зловмисник перехоплює відкриті дані: ctr_i, $Sign(k_A, ctr_i)$, p_A. На основі цих даних він має отримати секретний ключ k_A. Це можливо зробити або дискретним логарифмуванням публічного ключа (що зводиться до задачі пошуку секретного ключа за відомим публічним) або отримати його з відомого цифрового підпису (що зводиться до задачі пошуку секретного ключа цифрового підпису за значенням підпису та ключа перевірки).

У третьому випадку, алгоритм розробника задає значення випадкового запиту певним повідомленням (стеганограмою) M. Модифікація розробника також контролює секретний ключ k_A. Отже, модифікація розробника, маючи M (значення nonce) та k_A (секретний ключ підпису) має задати таке значення ctr_i, щоб $Sign(k_A, ctr_i) = M$. Перейдемо до більш загальної задачі: модифікація розробника, на основі M повинна згенерувати k'_A, R такі, щоб $Sign(k'_A, R) = M$ і $Verify(p_A, R, M) = 1$. Це також потребує розв'язання задачі підробки підпису.

Формалізуємо даний протокол, користуючись моделлю з визначенням11.

Визначення 16. *(Протокол генерації випадкового запиту з викриттям стегоконтейнеру) Припустимо, абонент є оракулом Alice, сторона Bob займається викриттям каналу витоку.*

Протоколом генерації випадкового запиту з викриттям стегоконтейнера назвемо кортеж $\langle D_t, V, U, A_t \rangle$ *моделі за визначенням 10, де:*
V *– множина можливих значень лічильника*
U *– множина можливих значень виходу абоненту*
$A_t \equiv Sign(k_A, v)$, $v \in V$ *– алгоритм абонента*
$D_t \equiv Verify(p_A, v, u)$, $u \in U$

Для оцінки клептографічної стійкості доведемо таку теорему.

Теорема 4. *Нехай справедливе припущення:* $\forall v \in V$, $\forall A_t : A_t(k_A, v) \neq Sign(k_A, v)$, $P\{Verify(p_A, v, A_t(k_A, v)) = 1\} < \varepsilon(t)$ *(тобто, практично неможливо створити пару різних підписів одного повідомлення).*
Тоді у протоколі передачі випадкової послідовності за визначенням 16 відсутній канал непомітного витоку секрету.

Доведення. В моделі за визначенням 16 класифікатор $D_t \equiv Verify(p_A, v, u)$ задовольняє достатній умові наслідку теореми 1:

1. $\forall v \in V : P\{Verify(p_A, v, Sign(k_A, v)) = 1\} = 1$ за властивістю цифрового підпису
2. $\forall v \in V, A_t(k_A, v) \neq Sign(k_A, v), P\{Verify(p_A, v, A_t(v)) = 1\} = \sigma < \varepsilon(t)$ за припущенням про практичну неможливість створення пари різних підписів

Отже, за наслідком теореми 1, за наведених припущень, для протоколу в моделі за визначенням 10 не існує каналу витоку секрету.
Більш того, у випадку передачі повідомлення прихованим каналом, ймовірність виявлення цього факту буде $P \geq 1 - \sigma \Leftrightarrow P > 1 - \varepsilon(t)$ за умови справедливості припущення теореми 4.
$\lhd$

Отже, запропонований протокол передачі nonce дійсно є протоколом з гарантованою відсутністю таємного каналу витоку.

3.1.2 Схема узгодження спільного ключа на базі Діффі-Хелмана стійка до побудови каналу витоку

Протокол узгодження спільного ключа Діффі-Хелмана є базовим у багатьох системах захищеного зв'язку. Разом з тим, це один приклад криптосистеми, для якої Янг та Юнг розробили методи клептографічної модифікації з утворенням таємного каналу витоку з пропускною здатністю (1,2) 5. У даному розділі пропонується метод модифікації базового протоколу Діффі-Хелмана для узгодження спільного ключа з неможливістю побудови таємного каналу витоку відповідно до моделі за визначенням 2.5.

Для ілюстрації ідей побудови каналу без витоку, використаємо як базовий однопрохідний протокол Діффі-Хеллмана.

Базовий протокол Діффі-Хелмана.

Вихідні дані:

1. Абоненти Alice та Bob.
2. Асиметрична криптосистема для алгоритму Діффі-Хеллмана: G – генератор,K, Q – простір закритих та відкритих ключів $'\cdot' : K \times Q \to Q$ – функція узгодження (піднесення до степеню).
3. Пара асиметричних ключів абонента Alice $(k_A, p_A), k_A \in K, p_A \in Q$.
4. Пара асиметричних ключів абонента Bob $(k_B, p_B), k_B \in K, p_B \in Q$.

Кроки роботи протоколу:

1. Абонент Alice генерує сесійний секретний ключ $x \overset{rand}{\in} K$

2. Alice відправляє відкритий сесійний ключ: $X = x \cdot G$, $W \to$ Bob.

3. Alice обчислює симетричний (спільний) ключ каналу: $s = x \cdot p_B$.

4. Bob обчислює спільний симетричний ключ каналу: $s = k_B \cdot X$.

Розглянемо алгоритм встановлення захищеного каналу зв'язку між абонентами A та B за модифікованим протоколом Діффі-Хелмана, що дозволяє генерувати випадковий сесійний ключ каналу без можливості побудови прихованого каналу витоку SETUP (див. Рисунок 9).

Модифікований протокол Діффі-Хелмана.

Вихідні дані:

1. Асиметрична криптосистема (для цифровиго підпису) з простором $\tilde{K}$ секретних ключів та простором $\tilde{Q}$ публічних.

2. Асиметрична криптосистема для алгоритму Діффі-Хеллмана: G – генератор,K, Q – простір закритих та відкритих ключів $'\cdot' : K \times Q \to Q$ – функція узгодження (піднесення до степеню).

3. Симетричний шифр (E, D) з простором ключів S, та бієктивними функціями $E : S \times B \to B$, $D : \forall s \in S, \forall b \in B, D_s(E_s(b)) = b$.

4. $Sign : \tilde{K} \times \{0,1\}^* \to B$ – функція генерації цифрового підпису без рандомізатору, B – простір підписів.

5. $Verify : \tilde{Q} \times \{0,1\}^* \times B \to \{0,1\}$ – функція перевірки цифрового підпису.

6. Пара асиметричних ключів абонента Alice $(k_A, p_A), k_A \in \tilde{K}, p_A \in \tilde{Q}$.

7. Пара асиметричних ключів абонента Bob $(k_B, p_B), k_B \in K, p_B \in Q$.

8. Криптографічно сильні функції хешування $h1 : B \to K$, $h2 : Q \to S$.

9. $\psi : \{0,1\}^* \to \{0,1\}^*$, $\psi_0 : Time \to \{0,1\}^*$ алгоритми генерації нового значення та ініціалізації лічильника.

Кроки роботи протоколу:

1. Абонент Alice обчислює наступне значення лічильника $ctr_i = \psi ctr_{i-1}$. Якщо попереднього значення не існує, можливе його створення, наприклад, на основі мітки часу $ctr_0 = \psi_0(time)$.

2. Alice:

   - Генерує сесійний секретний ключ $q = Sign(k_A, ctr_i|Alice|Bob)$
   - Симетричний ключ каналу: $s = h2(h1(q) \cdot p_B)$
   - Відправляє відкритий сесійний ключ, лічильник та ідентифікатори абонентів: $W = h1(q) \cdot G, (W, ctr_i, Alice, Bob) \rightarrow$ Bob

3. Bob обчислює спільний симетричний ключ каналу: $s = h2(k_B \cdot W)$

4. Alice відправляє свій секретний сесійний ключ закритим каналом: $E_s(q) \rightarrow Bob$

5. Bob розшифровує ключ q. Перевіряє, чи дійсно цей ключ згенерований на основі публічної послідовності:

   - Перевірка $ctr_i = \psi(ctr_{i-1})$. Якщо ні – лічильник не обчислений узгодженим алгоритмом, підозра на канал витоку, роз'єднання.
   - Перевірка $h1(q) \cdot G = W$. Якщо ні – ключ недійсний, з'єднання розривається.
   - Перевірка $Verify(p_A, ctr_i|Alice|Bob, q) = 1$. Якщо ні, то сесійний секретний ключ згенеровано не на основі відкритого лічильника, підозра на канал витоку, роз'єднання.

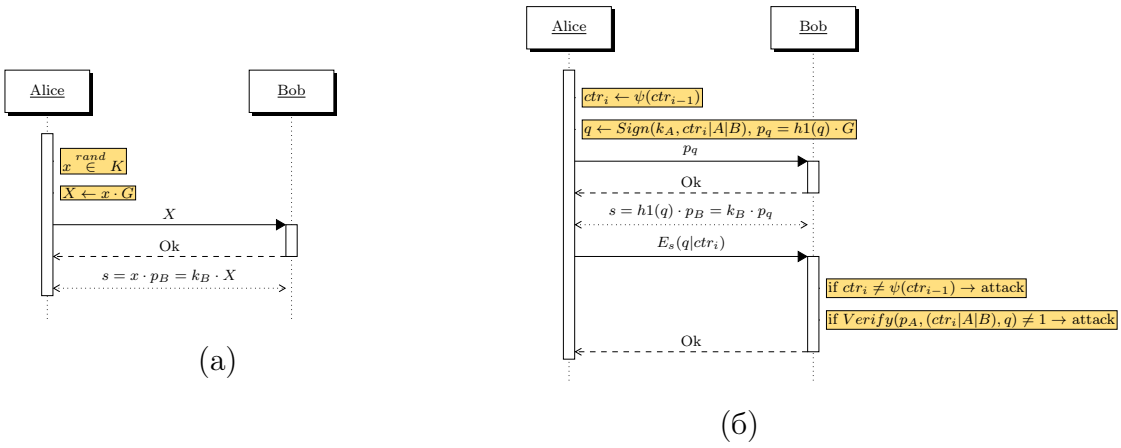

Рисунок 9. Схема узгодження ключа Діффі-Хеллмана: а) базова та б) удосконалена

Можливі зловмисні сценарії:

1. Зловмисник вгадує (прогнозує) сесійний секретний ключ сторони *Alice*.
2. Зловмисник відновлює секретний ключ сторони *Alice* з перехоплених у відкритому каналі даних.
3. Зловмисник – сторона *Bob* – намагається використати отриманий сесійний секретний ключ як свій секретний ключ для отримання спільного секрету з іншою стороною
4. Розробник модифікує *Alice* таким чином, щоб використати дані, що будуть відправлені до відкритого каналу, для організації непомітного витоку секрету (секретного ключа або сесійного секретного ключа криптосистеми).

Перші два сценарії розглядалися в 3.1.1.

У третьому випадку, сторона *Bob* отримує секретний сесійний ключ сторони *Alice* під час роботи протоколу та потім намагається його використати для узгодження спільного секрету з іншою стороною (сторона *Dev*). Проте тоді *Bob* має відправити разом з публічним ключем ідентифікатори *Alice* та *Bob*, що сторона *Dev* розпізнає як недійсну сесію. У випадку ж, коли *Bob* відправляє ідентифікатори *Alice* та *Dev*, сторона *Dev* викриває обман на етапі перевірки цифрового підпису: $Verify(p_A, (R_i|A|B), q)$

Нехай модифікація розробника має передавати певним чином закодований секрет до відкритого каналу. У випадку, якщо для цього будуть використовуватися додаткові відкриті повідомлення (окрім публічного ключа), це може бути виявлено зовнішнім спостерігачем, що порушує першу властивість SETUP (див. Визначення 2), а отже лазівка буде викрита.

Отже, у такому випадку, модифікація розробника може лише контролювати пару сесійних ключів (позначимо їх $(r, R), R = r \cdot G$), маючи при цьому доступ до секретного ключа k_A сторони *Alice*. Додаткові умови, які накладає протокол: $Verify(k_A, (S|A|B), r) = 1$, де S – значення публічного лічильника. А це означає, що максимальна кількість можливих секретних ключів, що зможе згенерувати модифікація розробника дорівнює кількості практично згенерованих лічильників, а отже практично неможливо зафіксувати довільний секретний ключ.

Для формального доведення відсутності каналу непомітного витоку секрету, проведемо редукцію схеми до моделі за визначенням 11.

Визначення 17. *(Протокол узгодження спільного ключа без каналу непомітного витоку секрету) Розглянемо сеанс комунікації абонентів Alice (оракул) та Bob (він же займається викриттям каналу витоку).*

Протоколом узгодження спільного сесійного ключа без каналу непомітного витоку секрету назвемо кортеж $\langle D_t, V, U, A_t \rangle$ *моделі за визначенням 10, де:*

V *– множина можливих значень лічильника*

U *– множина можливих значень виходу Alice*

$A_t \equiv h1(q) \times G)|e$ *– алгоритм абонента Alice,* $q = Sign(k_A, v), e = E_s(q)$, $v \in V$

$D_t(v, u) \equiv Verify(p_A, v, getq(u)) * \mathbb{I}(getp(u) = h1(D_{gets(u)}(gete(u))) \cdot G$, $u \in U$, *де функції* $getq, getp, gets, gete$ *обчислюються таким чином:*

$getp(u) = h1(q) \cdot G$,

$gete(u) = e$,

$gets(u) = h2(k_B \cdot getp(u))$,

$getq(u) = D_{gets(u)}(gete(u))$.

Для оцінки клептографічної стійкості сформульована та доведена така теорема.

Теорема 5. *Нехай справедливі припущення:*

1. $\forall v \in V$, $\forall A_t : A_t(k_A, v) \neq Sign(k_A, v)$, $P\{Verify(p_A, v, A_t(k_A, v)) = 1\} < \varepsilon(t)$ *(тобто, практично неможливо створити пару різних підписів одного повідомлення).*

2. *Функції* $getq, getp, gets, gete$ *обчислюються за час, яким можна знехтувати*

Тоді у протоколі за визначенням 17 відсутній канал непомітного витоку секрету.

Доведення. Користуючись наслідком теореми 1 покажемо, що справджується достатня умова відсутності каналу витоку. В моделі за визначенням 17 є класифікатор D_t який задовольняє достатній умові наслідку теореми 1:

1. $\forall v \in V : P\{D_t(p_A, v, A_t(v)) = 1\} = 1$. Дійсно, алгоритм D_t виглядає таким чином:

   (а) Отримати $q, h1(q) \cdot G), s, e$ з u за допомогою функції $getq, getp, gets, gete$, ймовірність отримання правильних значень $p = 1$.

   (б) Обчислити значення індикатору $\mathbb{I}(getp(u) = h1(D_{gets(u)}(gete(u))) \cdot G$. У випадку, якщо протокол проходить чесно, $h1(D_{gets(u)}(gete(u))) \cdot G) = h1(D_{gets(u)}(E_s(q))) \cdot G = h1(D_s(E_s(q))) \cdot G = h1(q) \cdot G$, тобто значення індикатору буде 1 з ймовірністю 1

   (в) перевірити підпис $Verify(p_A, v, getq(u)) = Verify(p_A, v, q) = 1$ з ймовірністю 1 за властивістю цифрового підпису

2. Нехай $A'_t \neq A_t$, тобто $\exists v \in V : A'_t(v) \neq A_t(v)$, $A_t(v) = g^{h1(q)}|e$, $A'_t(v) = g^w|e'$. Оцінимо ймовірність розпізнавання A'_t класифікатором D_t. В даному випадку можливі три ситуації:

   (а) $h1(q) \cdot G \neq w \cdot G \wedge e = e'$. Тоді $s \neq s', s' = h2(w \cdot k_B \cdot G) \Rightarrow D_{s'}(e) = q' \neq q$, $P\{Verify(p_A, v, q') = 1\} < \varepsilon(t)$ (згідно з припущенням про практичну неможливість створення пари різних підписів одного повідомлення)

(б) $h1(q)\cdot G = w\cdot G \wedge e \neq e'$. Тоді $q' = D_s(e') \neq D_s(e)$ (в силу бієктивності функцій (E, D)), $P\{Verify(p_A, v, q') = 1\} < \varepsilon(t)$

(в) $h1(q) \cdot G \neq w \cdot G \wedge e \neq e'$. Тоді $q' = D_{s'}(e')$. У випадку, якщо $q' \neq q$, $P\{Verify(p_A, v, q') = 1\} < \varepsilon(t)$. У випадку, якщо $q' = q$ ймовірність $P\{Verify(p_A, v, q') = 1\} = 1$. Згідно з протоколом, далі алгоритм D_t виконує перевірку $w \cdot G = h1(q') \cdot G$, що суперечить ситуації $h1(q) \cdot G \neq w \cdot G \wedge e \neq e'$

Тож максимальна ймовірність $P\{D_t(v, A'_t(v)) = 1\} = \sigma < \varepsilon(t)$

Отже, за наслідком теореми 1, за наведених припущень, для протоколу в моделі за визначенням 10 не існує каналу витоку секрету.
Більш того, у випадку передачі повідомлення прихованим каналом, ймовірність виявлення цього факту буде $P \geq 1 - \sigma \Leftrightarrow P > 1 - \varepsilon(t)$ за умови справедливості припущення теореми 5. $\lhd$

Отже, запропонована модифікація протоколу узгодження ключа Діффі-Хеллмана дійсно є протоколом з гарантованою відсутністю таємного каналу витоку. Для нього неможливо побудувати SETUP.

3.2 Розробка примітивів з вбудованим клептографічним механізмом

Синтез клептографічних механізмів займає важливе місце в процесі зменшення клептографіних властивостей системи: отримані приклади нових клептомеханізмів розширяють можливі сценарії Розробника, що в свою чергу дозволяє будувати нові методи захисту від таких сценаріїв.

3.2.1 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 74 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 73 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.

Далі, розглянемо один з можливих методів клептографічного синтезу в застосуванні до функцій гешування.

3.2.2 Схеми побудови функцій гешування

Найпоширенішою схемою побудови функцій гешування є конструкція Меркла-Дамгарда 87, що лягла в основу зокрема таких алгоритмів як MD5, SHA1, SHA256. Вона складається з ланцюжка функцій стиснення, що мають вигляд:

$$F : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^n \tag{3.1}$$

$$\Phi = F(F(F(F(IV, M_0), M_1), ...), M_k), \tag{3.2}$$

де $\{M_i\}_{i=0..k}$ – блоки повідомлень бітової довжини t

Авторами конструкції 87 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 88, 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 89 розглядаються 64 конструкцій, серед яких 20 є ймовірно стійкими до пошуку колізій (див. табл. 6).

Таблиця 6

Конструкції функцій стиснення, що ймовірно стійкими 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:

$$F : V^n \to V^n, \tag{3.3}$$

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 $m_1, m_2, ...m_k$ по r бітів кожен. Результатом обчислення буде:

$$H(M) = G(F(F(F(m_1 || \underbrace{0...0}_{c}) \oplus m_2) \oplus m_3...)) \tag{3.4}$$

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.

3.2.3 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 $2^{n/2}$. Для клептографічного застосування є сенс знизити вимоги до до стійкості криптопримітиву, що витікає з інших вимог безпеки до клептографічних закладок, наприклад, якщо клептографічна закладка значно знижує теоретичну стійкість примітиву, проте на практиці доступні ресурси аналітика не достатні для проведення криптоаналізу, такий тип закладок не можна відкидати. Введемо визначення практичної стійкості функцій гешування.

Визначення 18. *(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 19. *(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 20. *(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.

В ході досліджень було запропоновано базове перетворення для використанні у функції стиснення.

Визначення 21. *(базове перетворення на основі проблеми дискретного логарифмування) Нехай задана циклічна група* $\langle G, + \rangle$ *з генератором порядку* $g : ord(g) = n$. *Для групи можна визначити операцію «·»,* $w \cdot a = \underbrace{a + a + ... + a}_{w}$.

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

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

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

Далі, доведена необхідна в подальшому лема про ін'єктивність базового перетворення:

Лема 1. *(умови ін'єктивності функції* $T_k(\vec{v}, \cdot)$*)*
Якщо для заданого $\vec{v} = \langle c_0 \cdot g, c_1 \cdot g, ..., c_{k-1} \cdot g \rangle \in G^k, T_k(\vec{v}, \vec{x}) = \vec{h}$ *і* $rank(||a_{i,j}||) = k$ *над* Z_n, $a_{i,j} = c_{j-i \bmod k}$, *то* $\forall \vec{x}' : \vec{x}' \neq \vec{x} : T_k(\vec{v}, \vec{x}') \neq \vec{h}$.

Доведення.

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

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

Звідси випливає:

$$\begin{bmatrix} c_0 & c_1 & ... & c_{k-1} \\ c_{k-1} & c_0 & ... & c_{k-2} \\ ... & ... & ... & ... \\ c_1 & c_2 & ... & c_0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ ... \\ x_{k-1} \end{bmatrix} = \begin{bmatrix} \tilde{h}_0 \\ \tilde{h}_1 \\ ... \\ \tilde{h}_{k-1} \end{bmatrix} \tag{3.5}$$

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

$\lhd$

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

Теорема 6. *(стійкість функції T_k)*
Задача пошуку довільної невідомої компоненти x_r за відомими $\vec{v}, \vec{h} = T_k(\vec{v}, \vec{x})$ та $\{x_i\}_{i \neq r}$, що задовольняють умові з леми 1, поліноміально зводиться до задачі дискретного логарифмування в групі $\langle G, + \rangle$ відносно 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 $(a_{i,j} = c_{j-i \ mod \ k})$ 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 21, 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 3.5 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 3.5 над Z_n. Якщо вважати, що час виконання операцій $(*, +)$ над Z_n^k нехтовно малий відносно τ, а параметр k – константний, то $\tau_k = O(\tau)$.

$\lhd$

На базі даного перетворення T_k (визначення 21) та конструкції f_{17} побудована функція стиснення:

Визначення 22. *(Функція стиснення на базі перетворення T_k)*
Функцією стиснення на базі перетворення T_k називатимемо функцію:

$$F(\vec{v}, \vec{m}) = T_k(\vec{v}, \vec{m}) \boxplus \vec{m} \tag{3.6}$$

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

Теорема 7. *Для $\forall k \geq 2, G, Z_n$ конструкція за визначенням 22 з базовим перетворенням $T_k : G \times Z_n \to Z_n$: дозволяє непомітно передати розробнику, який обрав та зафіксував $\vec{v}$, фрагмент m_r блоку повідомлення $\vec{m}$ за умови знання $\{m_i\}, i = 0..k-1, i \neq r$ та розкладу $\vec{v} = \langle s_0 \cdot g, ..., s_{k-1} \cdot g \rangle$. При цьому складність задачі відновлення m_r, без знання секрету розробника, 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 $\vec{M} = \langle m_0, ..., m_k \rangle, m_i \in Z_n$, 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 21 та 22:

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

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

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

При цьому, для всіх інших учасників протоколу, за відсутності інформації про секретні параметри $s_0, ..., s_{k-1}$, складність відновлення повідомлення m_r зводиться до задачі дискретного логарифмування над $\langle G, + \rangle$ відносно операції «$\cdot$» згідно з теоремою 6. $\lhd$

В результаті, отримуємо метод побудови 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).

Метод побудови геш функції з закладкою.

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: $\vec{v} = \langle s_0 \cdot g, .., s_{k-1} \cdot g \rangle$ такого, що задовольняє умови леми 1.

2. Побудова перетворення T_k (визначення 21) на основі параметру k.

3. Побудова функції стиснення F (визначення 22) на основі перетворення T_k та стартового вектора $\vec{v}$.

Згідно з теоремою 7 дана конструкція дозволяє непомітно передати розробнику частину повідомлення m_r, якщо він володіє секретом $\vec{s} \in Z_n$ та частин блоку повідомлення $\{m_i\}_{i \neq r}$. При цьому, запропонована функція (визначення 22) має вигляд стандартної конструкції f_{17} (таблиця 6).

3.3 Побудова лазівки у протоколі консенсусу типу Proof-of-Work

У випадку, якщо геш функція використовується у специфічних режимах, модель її стійкості дозволяє розширити сценарії клептографічних атак. Одним з таких специфічних режимів є використання у системах технології блокчейн, де функція гешування є основою протоколів консенсусу.

Метод побудови лазівки спирається на можливість побудови геш функції з секретом Розробника, що дозволяє Розробнику ефективно будувати пари повідомлень, що ведуть до часткової колізії. Для цього, наприклад, можна використати метод Пренеля для побудови симетричного шифру із прихованим диференційним шляхом, а потім побудувати геш функцію на основі цього шифру у одному з можливих режимів (Девіса-Мейера, Міягучі-Пренеля тощо. Таким чином, матимемо геш функцію із фіксованими різницями повідомлень, що з високою ймовірністю ведуть до часткової колізії.

У контексті звичного використання геш функцій метод не є строго кажучи клептографічним оскільки, через те, що диференціальні шляхи є константними, у випадку декількох використань зовнішній спостерігач може легко виявити фіксовану різницю геш кодів, а з вихідних повідомлень отримати необхідну різницю на повідомлення. Іншими словами, лазівка викривається і надалі будь хто здатен ефективно будувати схожі колізії.

Проте у контексті використання геш функцій у Proof-of-Work консенсусі з'являється можливість використання даного методу Розробником для отримання переваги у змаганні потужностей для підвищення ймовірності підписати блок та отримати винагороду.

Припустимо, що Розробник має множину диференціальних шляхів, що дають йому можливість контролювати m з n бітів складності з ймовірністю p_{dev} (мається на увазі, що обравши довільне повідомлення M Розробник накладає на нього один з диференційних шляхів, що з ймовірністю p_{dev} інвертує контрольовану сукупність біт геш коду). Тепер задача Розробника – слідувати такій схемі перебору повідомлень, щоб збільшити ймовірність появи перших n нулів відносно ідеального випадку $p = \frac{1}{2^n}$.

Розробник виконує такі кроки для оптимізації перебору:

1. Обирає множину диференціальних шляхів $\{\delta_i\}_{i=1..2^m}$ з ймовірностями відповідно $\{p_i\}_{i=1..2^m}$, що дозволяють інвертувати будь яку комбінацію з m бітів (примітка: кількість диференційних шляхів може і не бути степенем 2, в такому випадку, для решти шляхів ймовірність буде нульова).

2. Підбирає початкове повідомлення (випадковим чином) таке, що решта $n - m$ неконтрольовані біти були нульовими.

3. Застосовує необхідний диференціальний шлях з $\{\delta_i\}_{i=1..2^m}$ так, щоб контрольовані m бітів стали нульовими. Ймовірність успішного застосування диференційного шляху буде:
$$p_{dev} = \frac{\Sigma_{i=1}^{2^m} p_i}{2^m} \tag{3.7}$$

4. У випадку, якщо геш код модифікованого повідомлення містить усі n старших біт нульовими – завершує роботу (складність гешу досягнуто). В іншому випадку – переходить до кроку 2. Остаточна ймовірність того, що усі n старших біт є добутком ймовірності, що 2^{n-m} неконтрольовані біти стануть нульовими та ймовірності, що диференційний шлях спрацьовує або ж він не спрацьовує, але умова досягається:
$$P_{success} = \frac{1}{2^{n-m}} \left(\frac{1}{2^m} + \left(1 - \frac{1}{2^m}\right) \left(p_{dev} + (1 - p_{dev}) \frac{1}{2^n} \right) \right). \tag{3.8}$$
Проте необхідно врахувати, що час перебору збільшиться (оскільки при нульових n-m бітах потрібно додатково проводити одну операцію гешування). Отже, середній час досягнення складності блоку (n перших нульових бітів) складатиме:
$$L_{dev} = \frac{1}{P_{success}(1 - \frac{1}{2^{n-m}} + \frac{1}{2^n})}. \tag{3.9}$$

Таким чином, отримуємо перевагу Розробника в обчислені геш-функції:
$$\eta = \frac{L_{dev} - 2^n}{2^n} \cdot 100\%. \tag{3.10}$$

Отже даний метод дозволяє Розробнику, у випадку застосування в системі блокчейн, отримати перевагу в продуктивності обчислень порівняно із звичайним учасником мережі.

3.4 Висновки до розділу 3

Даний розділ містить приклади застосування клептографічних методів для побудови криптосистем з каналами витоку та з доведеною їх відсутністю:

1. Запропоновано модифікацію базового протоколу запиту унікальної послідовності nonce та формально доведено відсутність можливості створення непомітного каналу витоку секрету на його основі.

2. Запропоновано модифікацію протоколу узгодження ключа Діффі-Хеллмана, що є стійкою до атаки Янга та Юнга, формально доведено відсутність можливості створення непомітного каналу витоку секрету під час роботи протоколу.

3. Продемонстровано можливість побудови геш функції на базі конструкції Меркла-Дамгарда із вбудованим клептографічним механізмом, що дозволяє Розробнику, який знає секрет у структурі примітиву, ефективно знаходити прообраз (за певних додаткових умов). Доведена неможливість ефективного пошуку прообразу для аналітика, що не володіє секретом.

4. Продемонстрована потенціальна можливість використовувати геш функції з набором таємних диференційних шляхів високої ймовірності для отримання переваги Розробником у системі технології розподіленого реєстру блокчейн.

4 Оцінка ефективності клептографічних методів

4.1 Оцінка ефективності лазівки в консенсусі технології блокчейн

У розділі 3 запропоновано метод побудови блокчейн системи із Proof-of-Work алгоритмом консенсусу із закладкою, що надає Розробнику перевагу у пошуку геш коду необхідної складності. В даному розділі проводиться аналіз ефективності методу для різних початкових умов.

Ефективність застосування методу можна оцінити за формулою 3.10. Покажемо перевагу Розробника для різних значень кількості контрольованих бітів та середньої ймовірності диференційного шляху p_{dev} (складність блоку взята фіксовано $n = 40$, приблизно така поточна складність блоку Bitcoin).

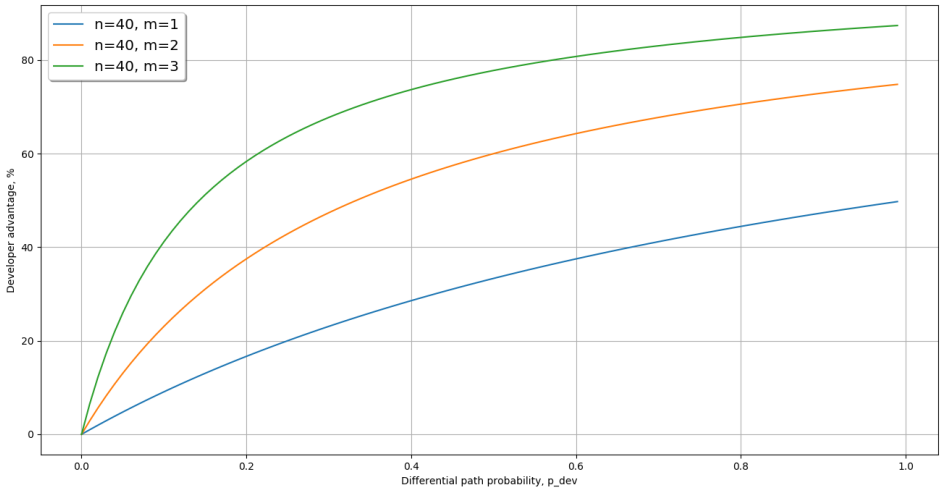

Рисунок 10. Залежність переваги розробника від середньої ймовірності диференційного шляху p_{dev}

З рисунку 10 видно, що з ростом середньої ймовірності диференційних шляхів, що використовуються для контролю m бітів геш коду росте і перевага Розробника, досягаючи максимуму при $p_{dev} = 1$.

Тепер розглянемо максимальну перевагу в залежності від кількості контрольованих бітів.

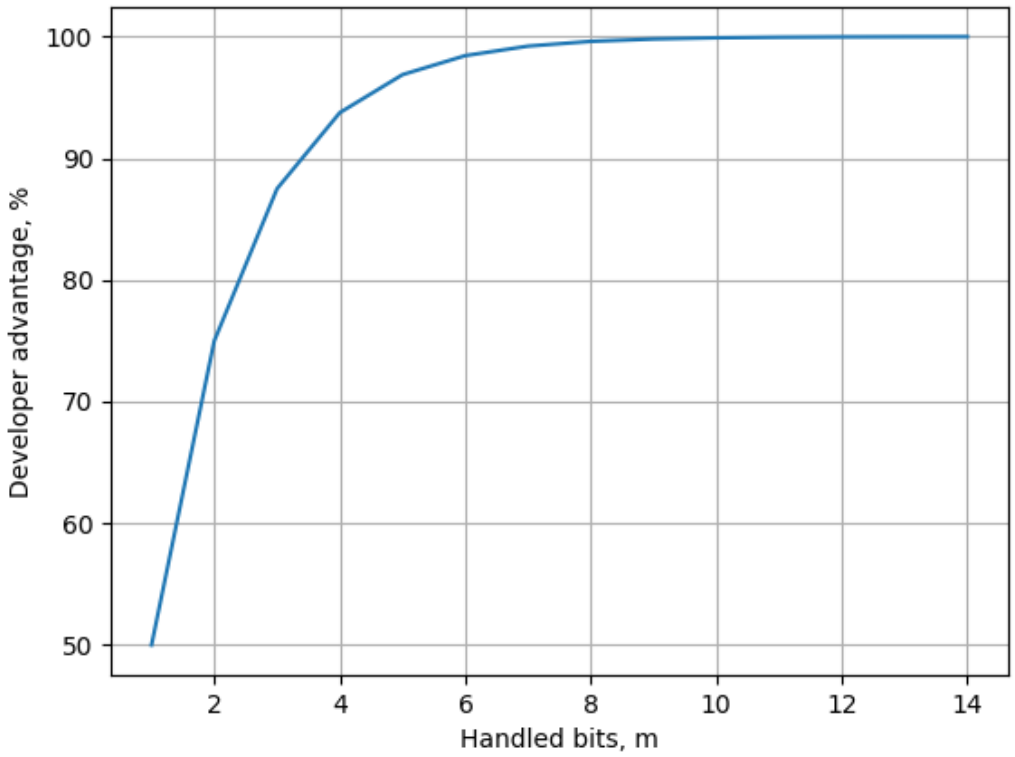

Рисунок 11. Залежність переваги розробника від кількості контрольованих біт m

З рисунку 11 видно, що уже при контролі одного біту перевага Розробника може складати до 50% і характер її росту є експоненційним за m.

З отриманих розрахунків видно, що підхід до клептографічної модифікації геш функції алгоритму консенсусу технології блокчейн дійсно може надавати суттєву перевагу Розробнику у змаганнях потужностей Proof-of-Work. При цьому, така модифікація не виявляється іншими учасниками системи, що не володіють секретом Розробника.

4.2 Оцінка ефективності зменшення клептографічного потенціалу симетричних криптопримітивів через генератор констант

В розділі 2 запропоновано метод зменшення можливостей побудови клептографічних механізмів під час розробки нових криптопримітивів. Спершу була введена метрика клептографічного потенціалу 13, яка відображає потенціальну можливість побудови клептомеханізму у криптосистеми, що спирається на той факт, що розмір секрету Розобника не може перевищувати кількість інформації у структурі криптосистеми. Дана метрика є відношенням нестрогого часткового порядку тож дозволяє порівнювати примітиви за потенціалом.

У даному параграфі проводиться оцінка ефективності підходу на прикладі оцінки клептографічної надлишковості 14 різних стандартних криптопримітивів.

Для визначення надлишковості спершу необхідно визначитися із потужністю класу обраного примітиву, тобто визначити відношення еквівалентності. Наприклад, у такому вигляді:

1. При заміні адитивних констант на довільні інші (якщо не задані інші критерії їх вибору), вважаємо, що модифікований алгоритм еквівалентний оригінальному.

   У випадку довільних адитивних констант загальної бітової довжини n вважаємо, що варіативність криптопримітиву за рахунок них буде 2^n:

   $$L_{const}(n) = 2^n \tag{4.1}$$

2. При заміні S-блока на інше довільне нелінійне перетворення (якщо не задані інші критерії його вибору) з тим самим розміром входів та виходів, вважаємо, що модифікований алгоритм еквівалентний оригінальному.

   У випадку S-блока, що є перетворенням $\pi_n : V^n \to V^n$, оцінюється кількість альтернативних s-блоків. В даному випадку, припускаємо, що s-блок може бути довільним нелінійним перетворенням (вимоги до s-блоків можуть бути іншими, і обмеженнями на степінь нелінійності,

довжиною циклу тощо. Обмеження наведені тут є ілюстративними, що є достатнім для оцінки зверху клептографічної надлишковості). Кожен S-блок може бути представленим у вигляді вектору поліномів АНФ степеня n $\vec{f} = \langle f_1, ..., f_n \rangle$. Кожен поліном включає 2^n коефіцієнтів, отже загальна кількість перетворень $2^{n \cdot 2^n}$. Кількість лінійних перетворень (таких, що жоден з поліномів f_i не містить ненульових коефіцієнтів при членах степені вищої за 1) – $2^{(n+1) \cdot n}$. Отже загальна кількість нелінійних перетворень вигляду π_n буде:

$$L_{sbox}(n) = 2^{n \cdot 2^n} - 2^{(n+1) \cdot n} \tag{4.2}$$

3. При заміні перестановки на іншу (за виключенням тривіальної перестановки $(1, 2, ..., n) \to (1, 2, ..., n)$ і якщо не задані інші критерії її вибору), вважаємо, що модифікований алгоритм еквівалентний оригінальному.
   У
   випадку перестановки елементів $\tau_n : (1, 2, ..., n) \to (tau_n(1), ..., tau_n(n))$, кількість альтернативних перестановок оцінюватиметься як:

$$L_{perm}(n) = n! - 1 \tag{4.3}$$

4. При заміні бієктивного лінійного перетворення $l : V^n \to V^n$ на будь яке інше бієктивне нелінійне перетворення (якщо не задані інші критерії його вибору), вважаємо, що модифікований алгоритм еквівалентний оригінальному.
   Таке перетворення можна задати невиродженою матрицею над GF(2) розміру $n \times n$. Кількість альтернативних перетворень (кількість таких матриць) буде:

$$L_{matr}(n) = \prod_{i=0}^{n-1} (2^n - 2^i) \tag{4.4}$$

AES

AES – алгоритм блокового шифрування, стандартизований 2012 року, розмір блоку – 128 біт, довжина ключа – 128, 192 або 256 біт.

Потенційні константи, що можуть генеруватися на основі ключа розробника:

1. Блок підстановки (процедура SubBytes). S-блок – нелінійне перетворення байт $V^8 \to V^8$. Воно складається з інвертування у мультиплікативній групі поля Галуа та подальшого афінного перетворення. Афінне перетворення виглядає як:

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{4.5}$$

   Матриця афінного перетворення будується на основі одного вектора $\langle 1,0,0,0,0,1,1,1 \rangle$ так що рядок i є циклічним зсувом вектора на i позицій вправо. Отже кількість альтернативних реалізацій – кількість можливих базових векторів: $L_1 = 2^8 - 2$ (без нульового вектора, та вектора з усіма одиницями, інакше матриця буде виродженою). Також можна варіювати адитивною константою – це ще $L_1 = 2^8$ варіантів.
   В результаті, отримуємо кількість альтернативних примітивів: $L_{subbytes} = (2^8 - 2) \cdot 2^8$.

2. Процедура MixColumns. Тут відбувається перетворення груп по 4 байти на поліном 3-го степеня над $GF(2^8)$ та множиться за модулем $x^4 + 1$ на поліном $3x^3 + x^2 + x + 2$.
   Кількість можливих комбінацій поліному: $L_{mixcolumn} = (2^8)^4 - 2 = 2^{32} - 2$.

Отже загальна потужність множини альтернативних алгоритмів:

$$L = L_{subbytes} \cdot L_{mixcolumn} = 2^{32} + 2^{16} - 2^9 - 2 \tag{4.6}$$

А клептографічна надлишковість примітиву (згідно з визначенням 14):

$$R = log_2(L) = log_2(2^{32} + 2^{16} - 2^9 - 2) \approx 32 \tag{4.7}$$

ГОСТ 28147-89

Радянський стандарт блокового шифрування ГОСТ 28147-89 базується на збалансованій схемі Фейстеля. Розмір блоку – 64 біт, довжина ключа – 256 біт. Основним джерелом варіативності Розробника є блоки підстановки, що стандартом затверджується. Фактично, даний стандарт уже є певною мірою фреймворком оскільки заповнення цих таблиць може варіюватися.

Блок підстановки – це 8 нелінійних перетворень виду $V^4 \to V^4$. Тож кількість альтернативних примітивів тут:

$$L = L_{sbox}(4)^8 = (2^{4 \cdot 2^4} - 2^{5 \cdot 4})^8 = (2^{64} - 2^{20})^8 \tag{4.8}$$

А клептографічна надлишковість примітиву (згідно з визначенням 14):

$$R = log_2(2^{64} - 2^{20})^8 \approx 512 \tag{4.9}$$

ДСТУ 7624:2014 «Калина»

Державний стандарт симетричного шифрування «Калина» побудований на основі SP-мережі, оперує блоками станів розмірами 128, 256 та 512 біт. Такими ж можуть бути розміри секретного ключа.

Головне джерело варіативності – чотири S-блоки кожен з яких є нелінійним перетворенням виду $\pi_i : V^8 \to V^8$ (байтова підстановка). Блоки підстановки задані таблицями проте також передбачається можливість, в окремих випадках, використовувати альтернативні S-блоки, подібно до того, як це відбувалося в старому стандарті ДСТУ ГОСТ 28147:2009 (ГОСТ 28147-89).

Отже загальна потужність множини альтернативних алгоритмів:

$$L = L^4_{subbytes}(8) = (2^{8 \cdot 2^8} - 2^{9 \cdot 8})^4 = 2^{8192} - 2^{288} \tag{4.10}$$

А клептографічна надлишковість примітиву (згідно з визначенням 14):

$$R = log_2(L) = log_2(2^{8192} - 2^{288}) \approx 8192 \tag{4.11}$$

ГОСТ Р 34.12-2015 «Кузнєчік»

ГОСТ Р 34.12-2015 – Російський стандарт блокового шифрування, являє собою sp-мережу із фейстелівською мережею для генерації ключа. Розмір блоку – 128 біт, довжина ключа – 256 біт.

Алгоритм містить такі константи:

1. Блок підстановки виду $V^8 \to V^8 : S =$(252, 238, 221, 17, 207, 110, 49, 22, 251, 196, 250, 218, 35, 197, 4, 77, 233, 119, 240, 219, 147, 46, 153, 186, 23, 54, 241,187, 20, 205, 95, 193, 249, 24, 101,90, 226, 92, 239, 33, 129, 28, 60, 66, 139, 1, 142, 79, 5, 132, 2, 174, 227, 106, 143, 160, 6, 11,237, 152, 127, 212, 211, 31,235, 52, 44, 81,234, 200, 72, 171, 242, 42, 104, 162, 253, 58, 206, 204, 181, 112, 14, 86, 8, 12, 118, 18, 191, 114, 19, 71,156, 183, 93, 135, 21, 161, 150, 41, 16, 123, 154, 199, 243, 145, 120, 111, 157, 158, 178, 177, 50, 117, 25, 61, 255, 53, 138, 126, 109, 84, 198, 128, 195, 189, 13, 87, 223, 245, 36, 169, 62, 168, 67, 201,215, 121,214, 246, 124, 34, 185, 3, 224, 15, 236, 222, 122, 148, 176, 188, 220, 232, 40, 80, 78, 51, 10, 74, 167, 151, 96, 115, 30, 0, 98, 68, 26, 184, 56, 130, 100, 159, 38, 65, 173, 69, 70, 146, 39, 94, 85, 47, 140, 163, 165, 125, 105, 213, 149, 59, 7, 88, 179, 64, 134, 172, 29, 247, 48, 55, 107, 228, 136, 217, 231, 137, 225,27, 131,73, 76, 63, 248, 254, 141,83,170, 144, 202, 216, 133, 97, 32, 113, 103, 164, 45, 43, 9, 91,203, 155, 37, 208, 190, 229, 108, 82, 89, 166, 116, 210, 230, 244, 180, 192, 209, 102, 175, 194, 57, 75, 99, 182).
   Кількість альтернатив примітиву: $L_{sbox}(8) = 2^{8 \cdot 2^8} - 2^{(8+1) \cdot 8} = 2^{2048} - 2^{72}$.

2. Лінійне перетворення блоку з 16 байтів $(a_0, a_2, ..., a_15)$ відбувається таким чином:

$$\begin{cases} a_i = a_{i+1}, i = 0..14 \\ a_{15} = \delta^{-1}(148\delta(a_0) + 32\delta(a_1) + 133\delta(a_2) + 16\delta(a_3) + 194\delta(a_4) + 192\delta(a_5) + \\ 251\delta(a_7) + \delta(a_8) + 192\delta(a_9) + 194\delta(a_10) + 16\delta(a_11) + 133\delta(a_12) + 32\delta(a_13) \end{cases} \tag{4.12}$$

   де $delta : V^8 \to GF(2^8)$ – бієктивне перетворення з бітового вектора в поліном поля Галуа.

   Кількість можливих комбінацій коефіцієнтів: $L_{coef} = (2^1 28)$.

Отже загальна потужність множини альтернативних алгоритмів:

$$L = L_{sbox}(8) \cdot L_{coef} \tag{4.13}$$

А клептографічна надлишковість примітиву (згідно з визначенням 14):

$$R = log_2(L) = log_2((2^{2048} - 2^{72}) \cdot (2^1 28)) \approx 2176 \tag{4.14}$$

SHA-256

SHA-256 наразі найбільш розповсюджений алгоритм гешування, базується на конструкції Меркла-Дамгарда, в основі конструкції стиснення лежить незбалансована схема Фейстеля. Стан функції стиснення – 256 біт, розмір блоку повідомлення – 512 біт.

Алгоритм містить такі константи:

1. Стартове заповнення (стан першого блоку повідомлення):
   . h0 = 0x6A09E667,
   h1 = 0xBB67AE85,
   h2 = 0x3C6EF372,
   h3 = 0xA54FF53A,
   h4 = 0x510E527F,
   h5 = 0x9B05688C,
   h6 = 0x1F83D9AB,
   h7 = 0x5BE0CD19,
   Проте заповнюється не довільним чином, а є hex-записом перших 32 бітів квадратного кореня перших 8 простих чисел. Тобто можемо вважати, що кількість можливих альтернативних алгоритмів тут незначна.

2. Раундові константи:
   k[0..63] =
   0x428A2F98, 0x71374491, 0xB5C0FBCF, 0xE9B5DBA5, 0x3956C25B, 0x59F111F1, 0x923F82A4, 0xAB1C5ED5,
   0xD807AA98, 0x12835B01, 0x243185BE, 0x550C7DC3, 0x72BE5D74, 0x80DEB1FE, 0x9BDC06A7, 0xC19BF174,

0xE49B69C1, 0xEFBE4786, 0x0FC19DC6, 0x240CA1CC, 0x2DE92C6F, 0x4A7484AA, 0x5CB0A9DC, 0x76F988DA,
0x983E5152, 0xA831C66D, 0xB00327C8, 0xBF597FC7, 0xC6E00BF3, 0xD5A79147, 0x06CA6351, 0x14292967,
0x27B70A85, 0x2E1B2138, 0x4D2C6DFC, 0x53380D13, 0x650A7354, 0x766A0ABB, 0x81C2C92E, 0x92722C85,
0xA2BFE8A1, 0xA81A664B, 0xC24B8B70, 0xC76C51A3, 0xD192E819, 0xD6990624, 0xF40E3585, 0x106AA070,
0x19A4C116, 0x1E376C08, 0x2748774C, 0x34B0BCB5, 0x391C0CB3, 0x4ED8AA4A, 0x5B9CCA4F, 0x682E6FF3,
0x748F82EE, 0x78A5636F, 0x84C87814, 0x8CC70208, 0x90BEFFFA, 0xA4506CEB, 0xBEF9A3F7, 0xC67178F2

Проте заповнюється не довільним чином, а є hex-записом перших 32 бітів кубічного кореня перших 64 простих чисел. Тобто можемо вважати, що кількість можливих альтернативних алгоритмів тут незначна.

3. Два блок підстановки, що задаються функціями:
$Ch(A,B,C) = (A \wedge B) \oplus (\neg A \wedge C)$
$Ma(A,B,C) = (A \wedge B) \oplus (B \wedge C) \oplus (A \wedge C)$
$\Sigma_0(A) = (A >>> 2) \oplus (A >>> 13) \oplus (A >>> 22)$
$\Sigma_1(A) = (A >>> 6) \oplus (A >>> 11) \oplus (A >>> 25)$
де операції $\wedge, \oplus, \neg$ – побітові над 32 бітовими векторами.
Отже, вважатимемо, що криптопримітив буде еквівалентним при заміні функцій Ch, Ma на довільне нелінійне побудове перетворення над (A,B,C). Також вважатимемо, що утворюється еквівалентний криптопримітив у випадку заміни зсувів у Σ_0, Σ_1 на довільні інші. Таким чином, кількість альтернатив за рахунок блоку підстановки буде: $L = L^2_{sbox}(3) \cdot (32^3)^2 = 2^{78} - 2^{54}$.

Отже загальна потужність множини альтернативних алгоритмів:

$$L = 2^{78} - 2^{54} \tag{4.15}$$

А клептографічна надлишковість примітиву (згідно з визначенням 14):

$$R = log_2(L) = log_2(2^{78} - 2^{54}) \approx 78 \tag{4.16}$$

Геш функція ГОСТ Р 34.11-2012 «Стрибог»

Російський стандарт гешування ГОСТ Р 34.11-2012 90 затверджений на зміну ГОСТ Р 34.11-94. В основі нього лежить конструкція Меркла-Дамгарда. Функція стиснення містить ряд констант, що обрані авторами у невідомий спосіб:

1. Блок підстановки π = (252, 238, 221, 17, 207, 110, 49, 22, 251, 196, 250, 218, 35, 197, 4, 77, 233, 119, 240, 219, 147, 46, 153, 186, 23, 54, 241,187, 20, 205, 95, 193, 249, 24, 101,90, 226, 92, 239, 33, 129, 28, 60, 66, 139, 1, 142, 79, 5, 132, 2, 174, 227, 106, 143, 160, 6, 11,237, 152, 127, 212, 211, 31,235, 52, 44, 81,234, 200, 72, 171, 242, 42, 104, 162, 253, 58, 206, 204, 181, 112, 14, 86, 8, 12, 118, 18, 191, 114, 19, 71,156, 183, 93, 135, 21, 161, 150, 41, 16, 123, 154, 199, 243, 145, 120, 111, 157, 158, 178, 177, 50, 117, 25, 61, 255, 53, 138, 126, 109, 84, 198, 128, 195, 189, 13, 87, 223, 245, 36, 169, 62, 168, 67, 201,215, 121,214, 246, 124, 34, 185, 3, 224, 15, 236, 222, 122, 148, 176, 188, 220, 232, 40, 80, 78, 51, 10, 74, 167, 151, 96, 115, 30, 0, 98, 68, 26, 184, 56, 130, 100, 159, 38, 65, 173, 69, 70, 146, 39, 94, 85, 47, 140, 163, 165, 125, 105, 213, 149, 59, 7, 88, 179, 64, 134, 172, 29, 247, 48, 55, 107, 228, 136, 217, 231, 137, 225,27, 131,73, 76, 63, 248, 254, 141,83,170, 144, 202, 216, 133, 97, 32, 113, 103, 164, 45, 43, 9, 91,203, 155, 37, 208, 190, 229, 108, 82, 89, 166, 116, 210, 230, 244, 180, 192, 209, 102, 175, 194, 57, 75, 99, 182).
   Кількість альтернатив примітиву: $L_{sbox}(8) = 2^{8 \cdot 2^8} - 2^{(8+1) \cdot 8} = 2^{2048} - 2^{72}$.

2. Перестановка 64 байтів: τ =(0, 8, 16, 24, 32, 40, 48, 56, 1,9, 17, 25, 33, 41,49, 57, 2, 10, 18, 26, 34, 42, 50, 58, 3, 11, 19, 27, 35, 43, 51,59, 4, 12, 20, 28, 36, 44, 52, 60, 5, 13, 21,29, 37, 45, 53, 61,6, 14, 22, 30, 38, 46, 54, 62, 7, 15, 23, 31,39, 47, 55, 63).
   Кількість альтернатив примітиву: $L_{perm}(64) = 64!$

3. Лінійне перетворення $V^{64} \to V^{64}$.
   Кількість альтернатив примітиву: $L_{matr}(64) = \prod_{i=0}^{63}(2^6 4 - 2^i)$

4. 12 ітераційних констант довжиною по 512 бітів (загальна довжина 6144 бітів): C1 = 085bda1ecadae9eb...

C2 = 6fa3b58aa99d2f1a...

C3 = f574dcac2bce2fc7...

C4 = ef1fdfb3e81566d2...

C5 = 4bea6bacad474799...

C6 = ae4faeae1d3ad3d9...

C7 = f4c70e16eeaac5ec...

C8 = 9b1f5b424d93c9a7...

C9 = 378f5a541631229b...

C10 = abbedea680056f5d...

C11 = 7bcd9ed0efc889fb...

C12 = 378ee767f11631bad...

Отже, Кількість альтернатив примітиву: $L_{const}(6144) = 2^{6144}$.

Отже загальна потужність множини альтернативних алгоритмів:

$$L = L_{sbox}(8) \cdot L_{perm}(64) \cdot L_{matr}(64) \tag{4.17}$$

А клептографічна надлишковість примітиву (згідно з визначенням 14):

$$R = log_2(L) = log_2(2^{2048} - 2^{72}) + log_2 64! + log_2 \prod_{i=0}^{63}(2^6 4 - 2^i) + log_2(2^{6144}) \approx$$
$$\approx 2048 + 295.99 + 4094.2 + 6144 = 12582.19 \tag{4.18}$$

Результуюча оцінка

Проаналізуємо результати оцінки клептографічної надлишковості різних симетричних криптопримітивів (таблиця 7).

Таблиця 7

Клептографічна надлишковість різних симетричних клептографічних примітивів

Примітив	Конструкція	Джерела надлишковості	Клептографічна надлишковість
AES	SP-мережа	процедури SubBytes та MixColumns	32
SHA-256	схема Фейстеля незбалансована	нелінійні функції блоку підстановки	78
ГОСТ 28147-89	схема Фейстеля	S-блок	512
ГОСТ Р 34.12-2015 «Кузнєчік»	SP-мережа	S-блок та лінійне перетворення	2176
ДСТУ 7624:2014 «Калина»	SP-мережа	чотири S-блоки	8192
ГОСТ Р 34.11-2012 «Стрибог»	SP-мережа	S-блок, байтова перестановка, лінійне перетворення $V^8 \to V^8$, раундові константи	12582.19

Варто нагадати, що клептографічна надлишковість – це нижня оцінка клептографічного потенціалу, тобто значення КП може бути більшим (наприклад, альтернативних алгоритмів AES може бути дещо більше за 32). Одержані результати демонструють різні можливості щодо побудови клептографічного механізму для різних криптопримітивах. Найбільше ризиковий з обраних алгоритмів є російський стандарт геш функції «Стрибог» – 12582.19. Така кількість інформації необгрунтовано обраних констант наводить на підозру про можливість існування лазівки розробника. Натомість, для стандарту блокового шифрування AES знайдено лише 32 біти надлишкової інформації у структурі, що потенційно задається секретом Розробника, тож можна припустити, що лазівка відсутня або ж алгоритм є вразливим.

4.3 Висновки до розділу 4

У даному розділі наводяться оцінки ефективності деяких клептографічних атак та методів клептографічного аналізу.

Були отримані такі практичні результати:

1. Для лазівки у блокчейн протоколах консенсусу Proof-of-Work отримані оцінки переваги Розробника для різної кількості контрольованих бітів та ймовірності допоміжних диференційних шляхів. Наприклад, у випадку контролю лише одного біта, перевага Розробника може сягати 50%.

2. Для ряду примітивів (геш функцій та алгоритмів симетричного шифрування) були отримані оцінки клептографічної надлишковості. Розрахунки показали, що серед розглянутих алгоритмів найбільша клептографічна надлишковість у російського стандарту геш функції ГОСТ Р-34.11-2012 («Стрибог») – 12582.19 біт (тобто за даною метрикою, алгоритм має найбільший ризик містити клептографічний механізм). Натомість, найменша клептографічна надлишковість спостерігається в стандарті блокового шифрування AES – 32.

Висновки

В роботі розв'язані наукові задачі побудови та дослідження моделей клептографічних механізмів та визначення методів побудови криптосистем з доведеною відсутністю клептографічних каналів витоку.

У процесі виконання дисертаційної роботи були отримані такі вагомі результати:

1. Вперше запропоновано математичну модель для протоколів типу «запит-відповідь» у клептографічному сенсі, в результаті чого отримана можливість строгої оцінки клептографічної стійкості протоколів, що зводяться до протоколів типу «запит-відповідь».

2. Вперше отримані достатні умови неможливості непомітної клептографічної модифікації криптосистеми, в результаті чого отримана можливість строгого доведення відсутності клептографічної модифікації у криптографічних протоколах.

3. Вперше розроблено метод побудови функції гешування з клептографічним механізмом, в результаті чого можливе створення геш-функції з лазівкою, що дозволяє Розробнику частково відновлювати повідомлення за відомим геш-кодом.

4. Вперше запропонована метрика «клептографічного потенціалу», в результаті чого отримана можливість порівнювати клептографічні примітиви за ризиком наявності у них закладок.

5. Отримані значення клептографічної надлишковості для відомих криптопримітивів, наприклад, в стандарті блокового шифрування AES клептографічний потенціал можна зменшити на 32 біти, а у російської геш-функції ГОСТ Р 34.11-2012 («Стрибог») – на 12582.19 бітів.

6. Удосконалено загальну класифікацію клептографічних систем Шнаєра, в результаті чого отримані вектори клептографічних атак на криптографічні системи та примітиви.

7. Удосконалено базові протоколи запиту nonce та узгодження ключа Діффі-Хеллмана, в результаті чого отримана база для побудови криптографічних протоколів зі строго доведенною відсутністю клептографічного каналу витоку.

8. Отримано подальший розвиток методу Пренеля побудови шифру для побудови клептографічної функції гешування, в результаті чого, у випадку використання такої функції у протоколах PoW консенсусу блокчейн перевага Розробника підвищується до 50% порівняно зі звичайним учасником.

9. Для лазівки у блокчейн протоколах консенсусу Proof-of-Work отримані оцінки переваги Розробника для різної кількості контрольованих бітів та ймовірності допоміжних диференційних шляхів. У випадку контролю лише одного біта, перевага Розробника може сягати 50%.

А Відомості щодо впровадження результатів дослідження

"ЗАТВЕРДЖУЮ"
Директор департаменту безпеки
Національного банку України

О.А. Скомаровський

"___" _____ 2020 року

АКТ
впровадження результатів досліджень дисертаційної роботи
Коваленка Богдана Анатолійовича в системі захисту інформації Національного банку України

Комісія у складі голови комісії Кльока Олександра Вадимовича і членів комісії: кандидата фізико-математичних наук Кабиша Юрія Михайловича, Пригодського Костянтина Івановича та кандидата технічних наук Шевцова Артура Сергійовича встановила, що у системі захисту інформації Національного банку України впроваджені отримані особисто Коваленко Богданом Анатолійовичем такі наукові результати:

метрика «клептографічного потенціалу» як кількість інформації Розробника, що передається у структуру примітиву, яка за рахунок оцінювання надлишковості параметрів у структурі криптографічного примітиву дозволяє порівнювати різні криптопримітиви за ризиком наявності клептографічного механізму в них.

Зазначені наукові результати дозволяють:

підвищити рівень захисту банківської інформації, яка обробляється в інформаційних системах Національного банку України за рахунок отримання оцінок клептографічної надлишковості для криптографічних алгоритмів гешування та симетричного шифрування, що дозволяє виявляти клептографічні лазівки зазначених алгоритмів на етапі проектування та впровадження систем захисту інформації.

Голова комісії:
Начальник відділу управління безпеки інформації НБУ

О. Кльок

Члени комісії:

Кабиш Ю.М.
Пригодський К.І.
Шевцов А.С.

Прим. №1

"ЗАТВЕРДЖУЮ"
Заступник командира
військової частини Р 9000

С. Борсук

"08" 08 2019 р.

АКТ
впровадження результатів досліджень дисертаційної роботи
Коваленка Богдана Анатолійовича в науково-дослідній роботі
"Дослідження математичних методів аналізу властивостей
односпрямованих перетворень та їх використання у дослідженні
криптографічних систем" (шифр "Родоліт") у військовій частини Р 9000

Комісія у складі голови комісії *Ігнатенка С.М.*, та членів комісії *Куслія Д.І., Андрійчука А.В.* встановила, що у військовій частині Р 9000 при виконанні деяких функціональних задач використовуються такі результати, отримані особисто Коваленком Богданом Анатолійовичем в процесі виконання науково-дослідної роботи "Дослідження математичних методів аналізу властивостей односпрямованих перетворень та їх використання у дослідженні криптографічних систем" (шифр "Родоліт"), номер держреєстрації 0111U007473:

Формальна модель клептографічних механізмів, побудова протоколів з доведеною відсутністю клептографічних лазівок, побудови клептографічних лазівок в геш функціях.

Голова комісії: С. Ігнатенко

Члени комісії: А. Андрійчук

Д. Куслій

Б Реалізація функції гешування на базі T_k-перетворення

```
GROUP = 157
GENERATOR = 17
ORD = GROUP-1

def eta(x):
    return x #(x**(ORD-1))%GROUP
def eta1(x):
    return x #(x**(ORD-1))%GROUP

def pow_mod(x, k, p):
    return (x**k)%p

def T_transform(k, v, x): #k-int, x-Znk, v-Gk
    if k != len(v) or k != len(x):
        raise NameError("Different vector size")

    res = []
    for j in range(k):
        r = 1
        for i in range(k):
            r *= pow_mod(v[i], x[(i+j)%k], GROUP)
        res += [eta(r%GROUP)]

    return res


def F(v, m): #v-Gk, m-Znk
    if len(v)!=len(m):
        raise NameError("Different vector size")
    r = T_transform(len(v), v, m)
    return [(r[i]+m[i])%GROUP for i in range(len(r))]

def Hash(M, k=4):
    IV = range(k)
    block_module = GROUP**k
    M_hex = int(''.join(["%2.2X"%ord(i) for i in M]) \
        + "1" + "%10.10x"%(len(M)*8), 16)
    block = M_hex%block_module
```

```
    state = IV
    while M_hex != 0:
        block_array = [(block/(GROUP**i))%GROUP for i in rang
        state = F(state, block_array)
        M_hex = M_hex/block_module
        block = M_hex%block_module
    return state

def test():
    x = [1,2,3,4]
    v = [3,7,9,1]
    print(T_transform(4, v, x))
    print(F(v,x))
    print(Hash("quick brown fox jumps over the lazzy dog",7))
    print(Hash("quick brown fox jumps over the lazzy dog1",7))


def test_eta():
    for i in range(1,10):
        print(i, eta(i), eta1(eta(i)))

def test_pow_mod():
    print(pow_mod(5, 46, 47))
test()
```

Література

1. Simmons, GJ. The Prisoners' Problem and the Subliminal Channel. в: *Advances in Cryptology: Proceedings of Crypto 83*. за ред. Chaum, D. Boston, MA: Springer US, 1984:51—67. DOI: 10.1007/978-1-4684-4730-9_5. URL: https://doi.org/10.1007/978-1-4684-4730-9_5.
2. Simmons, GJ. The Subliminal Channel and Digital Signatures. в: *Advances in Cryptology*. за ред. Beth, T., Cot, N., Ingemarsson, I. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985:364—378.
3. Simmons, GJ. Subliminal Communication is Easy Using the DSA. в: *Advances in Cryptology — EUROCRYPT '93*. за ред. Helleseth, T. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994:218—232.
4. Simmons, GJ. The History of Subliminal Channels. в: *Proceedings of the First International Workshop on Information Hiding*. London, UK, UK: Springer-Verlag, 1996:237—256. URL: http://dl.acm.org/citation.cfm?id=647594.731524.
5. Young, A., Yung, M. Kleptography: Using Cryptography Against Cryptography. в: *Advances in Cryptology — EUROCRYPT '97: International Conference on the Theory and Application of Cryptographic Techniques Konstanz, Germany, May 11–15, 1997 Proceedings*. за ред. Fumy, W. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997:62—74.
6. Biham, E., Shamir, A. Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology 1991;4:3—72.
7. Knudsen, LR. Truncated and Higher Order Differentials. в: *Fast Software Encryption - Second International Workshop, Leuven, Belgium, LNCS 1008*. Springer-Verlag, 1995:196—211.
8. Coppersmith, D. The Data Encryption Standard (DES) and Its Strength Against Attacks. IBM J. Res. Dev. 1994;38:243—250.
9. Matsui, M. Linear Cryptanalysis Method for DES Cipher. в: *Advances in Cryptology — EUROCRYPT '93*. за ред. Helleseth, T. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994:386—397.

10. Angelova, V., Borissov, Y. Plaintext Recovery in DES-like Cryptosystems Based on S-boxes with Embedded Parity Check. Serdica Journal of Computing 2013;7:257—270.

11. Biryukov, A., Perrin, L. On Reverse-Engineering S-Boxes with Hidden Design Criteria or Structure. в: *Advances in Cryptology – CRYPTO 2015*. за ред. Gennaro, R., Robshaw, M. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015:116—140.

12. Barker, EB., Kelsey, JM. SP 800-90A. Recommendation for Random Number Generation Using Deterministic Random Bit Generators. тех. звіт. Gaithersburg, MD, United States, 2012.

13. Brown, D., GjA steen, K. A Security Analysis of the NIST SP 800-90 Elliptic Curve Random Number Generator. в: *Advances in Cryptology - CRYPTO 2007*. за ред. Menezes, A. т. 4622. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007:466—481. DOI: 10.1007/978-3-540-74143-5_26.

14. Landau, S. NSA and Dual EC_DRBG: Déjà Vu All Over Again? The Mathematical Intelligencer 2015;37:72—83.

15. Bernstein, DJ., Lange, T., Niederhagen, R. Dual EC: A Standardized Back Door. в: *LNCS Essays on The New Codebreakers - Volume 9100*. Berlin, Heidelberg: Springer-Verlag, 2016:256—281. DOI: 10.1007/978-3-662-49301-4_17. URL: https://doi.org/10.1007/978-3-662-49301-4_17.

16. Menn, J. Exclusive: Secret contract tied NSA and security industry pioneer. за ред. Weber, J., McCool, G. [Online; posted 20-December-2013]. 2013. URL: https://www.reuters.com/article/us-usa-security-rsa/exclusive-secret-contract-tied-nsa-and-security-industry-pioneer-idUSBRE9BJ1C220131220.

17. AlTawy, R., Youssef, AM. Watch your Constants: Malicious Streebog. Cryptology ePrint Archive, Report 2014/879. https://eprint.iacr.org/2014/879. 2014.

18. AlTawy, R., Youssef, AM. Preimage Attacks on Reduced-Round Stribog. в: *Progress in Cryptology – AFRICACRYPT 2014*. за ред. Pointcheval, D., Vergnaud, D. Cham: Springer International Publishing, 2014:109—125.

19. Abdelkhalek, A., AlTawy, R., Youssef, AM. Impossible Differential Properties of Reduced Round Streebog. в: *Codes, Cryptology, and Information Security*. за ред. El Hajji, S., Nitaj, A., Carlet, C., Souidi, EM. Cham: Springer International Publishing, 2015:274—286.

20. AlTawy, R., Kircanski, A., Youssef, AM. Rebound attacks on Stribog. в: *IACR Cryptology ePrint Archive*. 2013.

21. Menyachikhin, A. Spectral-linear and spectral-differential methods for generating S-boxes having almost optimal cryptographic parameters. Cryptology ePrint Archive, Report 2018/370. `https://eprint.iacr.org/2018/370`. 2018.

22. Biryukov, A., Perrin, L., Udovenko, A. The Secret Structure of the S-Box of Streebog, Kuznechik and Stribob. Cryptology ePrint Archive, Report 2015/812. `https://eprint.iacr.org/2015/812`. 2015.

23. Perrin, L. Partitions in the S-Box of Streebog and Kuznyechik. Cryptology ePrint Archive, Report 2019/092. `https://eprint.iacr.org/2019/092`. 2019.

24. Albertini, A., Aumasson, JP., Eichlseder, M., Mendel, F., Schläffer, M. Malicious Hashing: Eve's Variant of SHA-1. в: *Selected Areas in Cryptography – SAC 2014*. за ред. Joux, A., Youssef, A. Cham: Springer International Publishing, 2014:1—19.

25. Biryukov, A., Perrin, L., Udovenko, A. Reverse-Engineering the S-Box of Streebog, Kuznyechik and STRIBOBr1 (Full Version). Cryptology ePrint Archive, Report 2016/071. `http://eprint.iacr.org/2016/071`. 2016.

26. AlTawy, R., Duman, O., Youssef, AM. Fault Analysis of Kuznyechik. Cryptology ePrint Archive, Report 2015/347. `https://eprint.iacr.org/2015/347`. 2015.

27. Granboulan, L. Flaws in Differential Cryptanalysis of Skipjack. в: *Fast Software Encryption*. за ред. Matsui, M. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002:328—335.

28. Wang, M., Cui, T., Chen, H., Sun, L., Wen, L., Bogdanov, A. Integrals Go Statistical: Cryptanalysis of Full Skipjack Variants. в: *Fast Software Encryption*. за ред. Peyrin, T. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016:399—415.

29. Kim, J., Phan, RCW. Advanced Differential-Style Cryptanalysis of the NSA's Skipjack Block Cipher. Cryptologia 2009;33:246—270.

30. Dubois, R. Trapping ECC with Invalid Curve Bug Attacks. Cryptology ePrint Archive, Report 2017/554. `https://eprint.iacr.org/2017/554`. 2017.

31. Neves, S., Tibouchi, M. Degenerate Curve Attacks: Extending Invalid Curve Attacks to Edwards Curves and Other Models. IET Information Security 2017;12.

32. Menezes, A., Teske, E. Cryptographic Implications of Hess' Generalized GHS Attack. Cryptology ePrint Archive, Report 2004/235. `https://eprint.iacr.org/2004/235`. 2004.

33. Galbraith, SD., Hess, F., Smart, NP. Extending the GHS Weil Descent Attack. в: *Advances in Cryptology — EUROCRYPT 2002*. за ред. Knudsen, LR. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002:29—44.

34. Teske, E. An Elliptic Curve Trapdoor System. Cryptology ePrint Archive, Report 2003/058. `https://eprint.iacr.org/2003/058`. 2003.

35. Baignères, T., Delerablée, C., Finiasz, M., Goubin, L., Lepoint, T., Rivain, M. Trap Me If You Can - Million Dollar Curve. IACR Cryptology ePrint Archive 2015;2015:1249.

36. Bernstein, DJ., Chou, T., Chuengsatiansup, C та ін. How to Manipulate Curve Standards: A White Paper for the Black Hat Http://Bada55.Cr.Yp.To. в: *Proceedings of the Second International Conference on Security Standardisation Research - Volume 9497*. SSR 2015. Tokyo, Japan: Springer-Verlag, 2015:109—139. DOI: `10.1007/978-3-319-27152-1_6`. URL: `https://doi.org/10.1007/978-3-319-27152-1_6`.

37. Biehl, I., Meyer, B., Müller, V. Differential Fault Attacks on Elliptic Curve Cryptosystems. в: *Advances in Cryptology — CRYPTO 2000*. за ред. Bellare, M. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000:131—146.

38. Biham, E., Carmeli, Y., Shamir, A. Bug Attacks. в: *Advances in Cryptology – CRYPTO 2008*. за ред. Wagner, D. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008:221—240.

39. Boneh, D., DeMillo, RA., Lipton, RJ. On the Importance of Eliminating Errors in Cryptographic Computations. Journal of Cryptology 2001;14:101—119.

40. Brumley, BB., Barbosa, M., Page, D., Vercauteren, F. Practical Realisation and Elimination of an ECC-Related Software Bug Attack. в: *Topics in Cryptology – CT-RSA 2012*. за ред. Dunkelman, O. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012:171—186.

41. Coron, JS. Resistance Against Differential Power Analysis For Elliptic Curve Cryptosystems. в: *Cryptographic Hardware and Embedded Systems*. за ред. Koç, ÇK., Paar, C. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999:292—302.

42. Fouque, PA., Lercier, R., Réal, D., Valette, F. Fault Attack on Elliptic Curve Montgomery Ladder Implementation. в: 2008:92—98. DOI: 10.1109/FDTC.2008.15.

43. Project, OWAS., 73, PRP. OWASP Top 10: The Top 10 Most Critical Web Application Security Threats Enhanced with Text Analytics and Content by PageKicker Robot Phil 73. USA: CreateSpace Independent Publishing Platform, 2014.

44. Durumeric, Z., Li, F., Kasten, J та ін. The Matter of Heartbleed. в: *Proceedings of the 2014 Conference on Internet Measurement Conference*. IMC '14. Vancouver, BC, Canada: ACM, 2014:475—488. DOI: 10.1145/2663716.2663755. URL: http://doi.acm.org/10.1145/2663716.2663755.

45. Schneier, B., Fredrikson, M., Kohno, T., Ristenpart, T. Surreptitiously Weakening Cryptographic Systems. Cryptology ePrint Archive, Report 2015/097. https://eprint.iacr.org/2015/097. 2015.

46. Young, A., Yung, M. The prevalence of kleptographic attacks on discrete-log based cryptosystems. в: *Advances in Cryptology — CRYPTO '97*. за ред. Kaliski, BS. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997:264—276.

47. Teşeleanu, G. Managing Your Kleptographic Subscription Plan. в: *Codes, Cryptology and Information Security*. за ред. Carlet, C., Guilley, S., Nitaj, A., Souidi, EM. Cham: Springer International Publishing, 2019:452—461.

48. Teşeleanu, G. Threshold Kleptographic Attacks on Discrete Logarithm Based Signatures. в: *Progress in Cryptology – LATINCRYPT 2017*. за ред. Lange, T., Dunkelman, O. Cham: Springer International Publishing, 2019:401—414.

49. Russell, A., Tang, Q., Yung, M., Zhou, HS. Cliptography: Clipping the Power of Kleptographic Attacks. в: *Advances in Cryptology – ASIACRYPT 2016*. за ред. Cheon, JH., Takagi, T. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016:34—64.

50. Giacomelli, I., Olimid, RF., Ranellucci, S. Security of Linear Secret-Sharing Schemes Against Mass Surveillance. в: *Cryptology and Network Security*. за ред. Reiter, M., Naccache, D. Cham: Springer International Publishing, 2015:43—58.

51. Bonnetain, X., Perrin, L., Tian, S. Anomalies and Vector Space Search: Tools for S-Box Reverse-Engineering. IACR Cryptology ePrint Archive 2019;2019:528.

52. Young, A., Yung, M. The Dark Side of "Black-Box" Cryptography or: Should We Trust Capstone? в: *Advances in Cryptology — CRYPTO '96: 16th Annual International Cryptology Conference Santa Barbara, California, USA August 18–22, 1996 Proceedings*. за ред. Koblitz, N. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996:89—103.

53. Crépeau, C., Slakmon, A. Simple Backdoors for RSA Key Generation. в: *Topics in Cryptology — CT-RSA 2003*. за ред. Joye, M. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003:403—416.

54. Robshaw, M. Trapdoor One-Way Function. в: *Encyclopedia of Cryptography and Security*. за ред. Tilborg, HCA van. Boston, MA: Springer US, 2005:625—626. DOI: 10.1007/0-387-23483-7_436. URL: https://doi.org/10.1007/0-387-23483-7_436.

55. Yilek, S., Rescorla, E., Shacham, H., Enright, B., Savage, S. When Private Keys Are Public: Results from the 2008 Debian OpenSSL Vulnerability. в: *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*. IMC '09. Chicago, Illinois, USA: ACM, 2009:15—27. DOI: 10.1145/1644893.1644896. URL: http://doi.acm.org/10.1145/1644893.1644896.

56. Diffie, W., Hellman, M. New Directions in Cryptography. IEEE Trans. Inf. Theor. 2006;22:644—654.

57. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P. Post-quantum key exchange - a new hope. Cryptology ePrint Archive, Report 2015/1092. https://eprint.iacr.org/2015/1092. 2015.

58. Kwant, R., Lange, T., Thissen, K. Lattice Klepto. в: *Selected Areas in Cryptography – SAC 2017*. за ред. Adams, C., Camenisch, J. Cham: Springer International Publishing, 2018:336—354.

59. Hoffstein, J., Pipher, J., Silverman, JH. NTRU: A ring-based public key cryptosystem. в: *Algorithmic Number Theory: Third International Symposiun, ANTS-III Portland, Oregon, USA, June 21–25, 1998 Proceedings*. за ред. Buhler, JP. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998:267—288. DOI: 10.1007/BFb0054868.

60. Gentry, C., Halevi, S. Implementing Gentry's Fully-Homomorphic Encryption Scheme. в: *Advances in Cryptology – EUROCRYPT 2011*. за ред. Paterson, KG. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011:129—148.

61. Baldi, M. The McEliece and Niederreiter Cryptosystems. в: *QC-LDPC Code-Based Cryptography*. Cham: Springer International Publishing, 2014:65—89. DOI: 10.1007/978-3-319-02556-8_5. URL: https://doi.org/10.1007/978-3-319-02556-8_5.

62. Young, A., Yung, M. Monkey: Black-Box Symmetric Ciphers Designed for MONopolizing KEYs. в: *Proceedings of the 5th International Workshop on Fast Software Encryption*. FSE '98. London, UK, UK: Springer-Verlag, 1998:122—133. URL: http://dl.acm.org/citation.cfm?id=647933.740753.

63. Bellare, M., Paterson, KG., Rogaway, P. Security of Symmetric Encryption against Mass Surveillance. в: *Advances in Cryptology – CRYPTO 2014*. за ред. Garay, JA., Gennaro, R. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014:1—19.

64. Rijmen, V., Preneel, B. A Family of Trapdoor Ciphers. Fast Software Encryption, LNCS 1267, E. Biham, Ed., Springer-Verlag 1997;1267:139—148.

65. Ohta, K., Pei, D, ред. Advances in Cryptology - ASIACRYPT '98, International Conference on the Theory and Applications of Cryptology and Information Security, Beijing, China, October 18-22, 1998, Proceedings. т. 1514. Lecture Notes in Computer Science. Springer, 1998.

66. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf.

67. Кудін, А., Коваленко, Б., Швідченко, І. Технологія блокчейн: питання аналізу та синтезу. Кібернетика та системний аналіз 2019;55:165—173.

68. Кудін, А., Ковальчук, Л., Коваленко, Б. Теоретичні засади та застосування блокчейн-технологій: імплементація нових протоколів консенсусу та краудсорсінг обчислень. Математичне та комп'ютерне моделювання 2019:62—68.

69. Lamport, L., Shostak, R., Pease, M. The Byzantine Generals Problem. ACM Trans. Program. Lang. Syst. 1982;4:382—401.

70. Atzei, N., Bartoletti, M., Cimoli, T. A Survey of Attacks on Ethereum Smart Contracts SoK. в: *Proceedings of the 6th International Conference on Principles of Security and Trust - Volume 10204*. New York, NY, USA: Springer-Verlag New York, Inc., 2017:164—186. DOI: 10.1007/978-3-662-54455-6_8. URL: https://doi.org/10.1007/978-3-662-54455-6_8.

71. Romano, D., Schmid, G. Beyond Bitcoin: A Critical Look at Blockchain-Based Systems. Cryptography 2017;1:15.

72. Badertscher, C., Garay, J., Maurer, U., Tschudi, D., Zikas, V. But Why Does It Work? A Rational Protocol Design Treatment of Bitcoin. в: *Advances in Cryptology – EUROCRYPT 2018*. за ред. Nielsen, JB., Rijmen, V. Cham: Springer International Publishing, 2018:34—65.

73. Коваленко, Б., Кудін, А. Алгоритмічні аспекти пошуку прообразів геш-функцій на прикладі MD5. Захист інформації 2015;17:205—210.

74. Коваленко, Б., Кудін, А. Диференційний аналіз функцій хешування та блокових шифрів: узагальнений підхід. Безпека інформації 2015;21:159—164.

75. Богуш, В., Кривуца, В., Кудін, А. Інформаційна безпека: Термінологічний навчальний довідник. Київ, Україна, 2004.

76. Коваленко, Б. Побудова криптографічних протоколів вільних від клептографічних модифікацій. Безпека інформації 2019;25:122—126.

77. Atanasiu, A., Olimid, R., Simion, E. On the Security of Black-Box Implementation of Visual Secret Sharing Schemes. Journal of Mobile, Embedded and Distributed Systems 2012;4:1—11.

78. Berbain, C., Gilbert, H. On the Security of IV Dependent Stream Ciphers. в: *Fast Software Encryption*. за ред. Biryukov, A. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007:254—273.

79. Mao, W. Modern Cryptography: Theory and Practice. Prentice Hall Professional Technical Reference, 2003.

80. Schneier, B., Fredrikson, M., Kohno, T., Ristenpart, T. Surreptitiously Weakening Cryptographic Systems. Cryptology ePrint Archive, Report 2015/097. `https://eprint.iacr.org/2015/097`. 2015.

81. Almeida, JB., Barbosa, MB., Barthe, G., Dupressoir, F. Certified computer-aided cryptography: efficient provably secure machine code from high-level implementations. в: *ACM Conference on Computer and Communications Security*. Berlin, Germany, 2013:1217—1230. URL: `http://doi.acm.org/10.1145/2508859.2516652`.

82. Blanchet, B. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. в: *Proceedings of the 14th IEEE Workshop on Computer Security Foundations*. CSFW '01. Washington, DC, USA: IEEE Computer Society, 2001:82—. URL: http://dl.acm.org/citation.cfm?id=872752.873511.

83. Boneh, D., Lynn, B., Shacham, H. Short Signatures from the Weil Pairing. Journal of Cryptology 2004;17:297—319.

84. Menezes, A. An introduction to pairing-based cryptography. 1991.

85. Стратонович, Р. Теория информации. Москва, СССР: Сов. радио, 1975.

86. Dinu, D., Perrin, L., Udovenko, A., Velichkov, V., Grosschadl, J., Biryukov, A. Design Strategies for ARX with Provable Bounds: SPARX and LAX (Full Version). Cryptology ePrint Archive, Report 2016/984. https://eprint.iacr.org/2016/984. 2016.

87. Merkle, RC. Secrecy, authentication, and public key systems. AAI8001972. дис. ... док. Stanford, CA, USA, 1979.

88. Black, J., Rogaway, P., Shrimpton, T. Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. в: *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*. т. 2442. Lecture Notes in Computer Science. Springer, 2002:320—335. DOI: 10.1007/3-540-45708-9_21. URL: http://www.iacr.org/cryptodb/archive/2002/CRYPTO/889/889.pdf.

89. Black, J., Rogaway, P., Shrimpton, T. Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. в: *Advances in Cryptology — CRYPTO 2002*. за ред. Yung, M. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002:320—335.

90. Kazymyrov, O., Kazymyrova, V. Algebraic Aspects of the Russian Hash Standard GOST R 34.11-2012. Cryptology ePrint Archive, Report 2013/556. https://eprint.iacr.org/2013/556. 2013.