Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерної інженерії та управління _____
(повна назва)

Кафедра _____Безпеки інформаційних технологій _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

Атаки на екстрактори квантових генераторів випадкових чисел
(тема)

Виконав:
студент __2_ курсу, групи _БДІРм–20–1_
____Лібін С.Є._____
(прізвище, ініціали)
Спеціальність _____125 Кібербезпека_____
(код і повна назва спеціальності)
Освітня програма ___«Безпека державних інформаційних ресурсів»
(повна назва освітньої програми)
Керівник доцент ГріненкоТ.О.___
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ _____Халімов Г.З.__
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет_____Комп'ютерної інженерії та управління _____

Кафедра_____Безпеки інформаційних технологій___

Рівень вищої освіти_____ другий (магістерський)_____

Спеціальність_____125 Кібербезпека _____
(код і повна назва)

Освітня програма_____«Безпека державних інформаційних ресурсів»___
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові_____Лібіну Сергію_____
(прізвище, ім'я, по батькові)

1. Тема роботи Атаки на екстрактори квантових генераторів випадкових чисел затверджена наказом по університету від _08_ ____11___2021 р. № __1684 ст____

2. Термін подання студентом роботи до екзаменаційної комісії _13_грудня_____2021 р.

3. Вихідні дані до роботи _статистичні дані, методика оцінки ризиків _____

4. Перелік питань, що потрібно опрацювати в роботі _____
Аналіз літературних джерел за темою кваліфікаційної роботи

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _презентаційний матеріал у вигляді слайдів 12. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Видача завдання	01.09.2021	Виконано
2	Аналіз літературних джерел за темою кваліфікаційної роботи	09.11.2021– 13.11.2021	Виконано
3	Обгрунтування вимог до генераторів випадкових чисел	14.11.2021- 16.11.2021	Виконано
4	Вивчення сучасних генераторів випадкових чисел	17.11.2021– 19.11.2021	Виконано
5	Аналіз існуючих екстракторів до генераторів випадкових чисел	20.11.2021– 28.11.2021	Виконано
6	Обгрунтування вимог до екстракторів до генераторів випадкових чисел	29.11.2021– 05.12.2021	Виконано
7	Аналіз атак та загроз на БСШ, ПСШ, Хеш-функцій	06.12.2021– 08.12.2021	Виконано
8	Обгрунтування стійкості до атак на екстрактори	09.12.2021	Виконано
9	Оформлення пояснювальної записки	10.12.2021– 11.12.2021	Виконано

Дата видачі завдання __1___ __вересня_____2021 р.

Студент _____
(підпис)

Керівник роботи _____ _____ доцент Гріненко Т.О._
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 70 с., 2 табл., 3 рис., 58 джерел.

ВИПАДКОВА ПОСЛІДОВНІСТЬ, ГЕНЕРАТОР ВИПАДКОВИХ ЧИСЕЛ, ЕКСТРАКТОР ВИПАДКОВОСТІ, КВАНТОВИЙ ГЕНЕРАТОР ВИПАДКОВИХ ЧИСЕЛ, ЛАНЦЮГ ФЕЙСТЕЛЯ, SPN СТРУКТУРА

Об’єкт дослідження – дослідження атак на екстрактори випадковості, що використовуються в квантових генераторах випадкових чисел.

Предмет дослідження – атаки на екстрактори квантових генераторів випадковиї чисел.

Мета роботи – обгрунтування вимог до генераторів випадкових чисел; аналіз та дослідження методів побудови квантових генераторів випадкових чисел; аналіз методів побудови екстракторів випадковості; обгрунтування вимог, що висуваються до екстракторів випадковості, які доцільно викоритовувати в квантових генераторах випадкових чисел; дослідження атак на екстрактори квантових генераторів випадкових чисел.

Проведений аналіз методів генеруваннння випадкових та псевдовипадкових послідовностей, обгрунтувані вимоги до генераторів випадкових чисел, проведений аналіз та дослідження методів побудови квантових генераторів випадкових чисел, обгрунтувані вимоги, що висуваються до екстракторів випадковості, які доцільно викоритовувати в квантових генераторах випадкових чисел; проведений аналіз та дослідження атак на екстрактори квантових генераторів випадкових чисел.

ABSTRACT

Explanatory note to the qualification work: 70 pages, 2 tables, 3 figures, 58 sources.

RANDOM SEQUENCE, RANDOM NUMBER GENERATOR, RANDOM EXTRACTOR, QUANTUM RANDOM NUMBER GENERATOR, FESTIAL CHAIN, SPN STRUCTURE

Object of research – research of attacks on randomness extractors used in quantum random number generators.

Subject of research – attacks on extractors of quantum random number generators.

Purpose of work – substantiation of requirements to random number generators; analysis and research of methods for constructing quantum random number generators; analysis of methods for constructing random extractors; substantiation of the requirements for random extractors, which should be used in quantum random number generators; study of attacks on extractors of quantum random number generators.

The analysis of methods of generating random and pseudo-random sequences, substantiated requirements for random number generators, analysis and research of methods for constructing quantum random number generators, substantiated requirements for random extractors, which should be used in quantum random number generators; analysis and research of attacks on extractors of quantum random number generators.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ

ГВП	–	Генератор випадкових послідовностей
ГВЧ	–	Генераторів випадкових чисел
ДГВП	–	детермінований генератор випадкових послідовностей
КГВП	–	Квантовий генератор випадкових послідовностей
КГВЧ	–	Квантовий генератор випадкових чисел
ПВП	–	Псевдовипадкова послідовність
AES	–	Advanced Encryption Standard
AIS20	–	Application Notes and Interpretation of the Scheme 20
AIS31	–	Application Notes and Interpretation of the Scheme 31
ANSI	–	American National Standards Institute
FIPS	–	Federal Information Processing Standards
NIST SP	–	National Institute of Standards and Technology Special Publication
NIST STS	–	National Institute of Standards and Technology Statistical Test Suite
QRNG	–	Quantum Random Number Generation

ВСТУП

Рубіж 20-го і 21-го століть можна вважати початком стрімкого розвитку та поширення інформаційних технологій майже в усіх галузях економіки та науки, а також у сфері комунального господарства. Інформаційні технології в багатьох ключових аспектах вимагають врахування в алгоритмах генерування випадкових величин. Отже, проблема генераторів випадкових чисел відіграє фундаментальну роль у сфері інформаційних технологій, зокрема, інформаційної безпеки.

Сучасні застосування генераторів випадкових чисел (ГВЧ) поширюються на область інформаційних технологій з точки зору:

– додатки в області криптографії:

а) для окремих програм користувача;

б) для генерації випадкових послідовностей ініціалізації (так званих початкових) для алгоритмів шифрування, аутентифікації або цифрового підпису;

в) для генерації ключів (для асиметричної та симетричної криптографії), одноразові/ініційні вектори, проблеми аутентифікації, вибір експонентів у протоколі Діффі-Хеллмана;

– інші IT-додатки: теги/токени для протоколів зв'язку, для індексування в базах даних тощо;

– статистичні програми (наприклад, вибір репрезентативної вибірки для статистичного аналізу);

– чисельне моделювання типу Монте-Карло;

– алгоритми AI: нейронні мережі (наприклад, випадкове зважування для мереж) і генетичні алгоритми (наприклад, випадкове введення мутацій, випадкове змішування представників);

– структури та служби підтримки популярних на даний момент криптовалют (наприклад, біткойн-гаманці, біржі біткойн,тощо);

– азартні ігри (наприклад, онлайн-казино);

– випадковість у процесах контролю (важлива проблема відбору зразків для якості процесів контролю);

– випадковість в управлінні (наприклад, встановлення порядку у виборчих списках).

Наведений вище список коротко показує масштаб діапазону застосування випадковості та генераторів випадкових чисел. У цьому контексті якість випадковості та її правдивість стають фундаментальною проблемою.

Наслідки передбачуваності генерованих класичних псевдовипадкових послідовностей очевидні – відсутність справжньої випадковості в будь-якому із зазначених раніше додатків є перешкодою для передбачуваного функціонування. У випадку з криптографічними додатками наслідки можуть бути особливо важкими. Проблема класичних генераторів випадкових чисел, тобто генераторів псевдовипадкових чисел, полягає у можливості знати детермінований процес генерації псевдовипадкових чисел небажаними особами. У випадку криптографії це може призвести до скомпрометації міфу про безпеку. Іншою проблемою може бути неправильна обробка згенерованої послідовності – здебільшого в криптографічних цілях згенерована випадкова послідовність застосовується один раз. Його багаторазове використання може призвести до порушення безпеки (наприклад, у випадку шифру OTP(одноразовий пароль) достатньо довгий ключ має бути дійсно випадковим і використовуватися один раз у цьому протоколі, інакше можна буде зламати код). Масштаб загрози можна проілюструвати вибраними атаками та інформацією про загрози, як показано нижче:

– 2006-2012 – протягом багатьох років було багато повідомлень про атаки на криптографічні ключі, створені слабкими ГПВЧ (що дозволяє, наприклад, здійснити атаку грубої сили на SSH, захищений ключами RSA);

– 2010 р. – була здійснена вражаюча атака на користувачів ігрової консолі Sony PlayStation 3 (PS3) (викрали дані аж 77 млн користувачів). Атака була здійснена з використанням недоліку в реалізації алгоритму ECDSA компанією Sony (розкриті матеріали повідомляли, що одне й те саме випадкове

число було помилково використано кілька разів як так званий одноразовий номер для аутентифікації);

– 2012 – дві групи дослідників виявили численні ключі шифрування RSA, які потім активно використовувалися в Інтернеті як 5secure і ризикували бути зламаними через недостатній генератор випадкових даних, який використовувався для їх створення;

– 2013 – після розкриття Сноуденом цих недоліків Агентству національної безпеки США (АНБ), Reuters та New York Times провели розслідування, які виявили, що АНБ навмисно таємно знижувало безпеку популярних у світі апаратних і програмних рішень з метою крипто- атаки на зашифрований вміст (включаючи атаки на ГВЧ):

a) Для цього було використано подвійний EC DRBG (генератор випадкових розрядів із подвійною еліптичною кривою), ГПВЧ, створений і посилений як стандарт АНБ. Лише в 2013 році з'ясувалося, що АНБ було єдиним, хто мав бекдор для цього генератора, і завдяки цьому АНБ змогло зламати криптографічні ключі, які були згенеровані цими генераторами. Після розкриття інформації RSA Security та Національний інститут стандартів і технологій США (NIST) доручили не використовувати генератор Dual EC DRBG;

b) АНБ здійснило секретний проект під кодовою назвою Bullrun, зосереджений на використанні вразливостей у поширеній ГПВЧ, до якої вона мала випадковий доступ, на різних пристроях (наприклад, Juniper);

c) Генератори випадкових чисел Intel і Via на материнській платі АГВЧ, ймовірно, також мали бекдор. Було зазначено, що інструкції RdRand і Padlock, швидше за все, мають бекдори в Linux ядра до версії 3.13. 9;

– 2013 – Google підтвердив, що клас IBM Java SecureRandom в архітектурі криптографії Java (JCA) генерував повторювані (і, отже, передбачувані) послідовності, що поставило під загрозу безпеку додатків, створених для Android для підтримки електронної валюти Bitcoin;

– 2014 – є підозра, що атака на токійську криптовалютну біржу MtGox, під час якої було вкрадено понад 800 000 біткойнів (в результаті чого MtGox

оголосила про банкрутство), була пов’язана з атакою на ГВЧ;

– 2015 – віддалена атака, яку важко виявити з використанням зовнішньо підключеного апаратного троянського коня на TRNG на основі FPGA;

– 2015 р. – крадіжка 18 866 біткойнів з біржі Bitstamp (12% валюти, що торгується на цій біржі) – підпис атаки RNG;

– 2017 – ANSI x9.31 PRNG сумісний із 2016 FIPS США (Федеральні стандарти обробки інформації) – скомпрометовано, якщо використовується із жорстко закодованим початковим кодом (атака DUHK – не використовуйте жорстко закодовані ключі).

Наведені вище приклади показують, що класичні генератори випадкових чисел можуть бути піддані різним атакам, або можуть мати так звані бекдори. Це виправдовує необхідність розробки альтернативних технологій, які могли б замінити класичні генератори у великих масштабах. Найперспективнішими, оскільки мають принципове обґрунтування випадковості у формалізмі квантової механіки, є квантові генератори випадкових чисел.

Класичні генератори випадкових чисел через детермінований процес генерації (продиктований детермінованими законами класичної фізики або детермінованими математичними інформаційними алгоритмами) генерують послідовності, які, незважаючи на ідеальний баланс між цифрами 0 і 1, неминуче завжди будуть характеризуватися наявністю певних детермінованих довготривалиї закономірностей – кореляції, які можуть становити потенційний ризик для IT-безпеки, несподівані помилки в науковому моделюванні або прогалини в тестуванні фізичних процесів.

Поява ефективного квантового комп’ютера (зараз комерціалізуються псевдоквантові комп’ютери, наприклад, DWave значно перевищує обчислювальну потужність класичних пристроїв, крім того, нещодавно Google представила повністю робочу квантову машину Sycamore, щоб продемонструвати Quantum Supremacy, а пізніше китайські вчені представили фотонний квантовий комп’ютер, званий Jiuzhang) спричинить потенційну загрозу будь-якому класичному генератору випадкових чисел – теоретично,

квантовий комп’ютер знайде детерміновану природу процесу генерації в реальному часі, якщо цей процес заснований на явищі класичної фізики. Відповіддю на цю загрозу, здається, є квантові генератори випадкових чисел, які стають все більш популярними, незважаючи на те, що перспектива створення ефективного великого масштабованого квантового комп’ютера на основі заплутаності все ще відкладається через поточні технологічні обмеження.

1 ОБҐРУНТУВАННЯ ВИМОГ ДО ГЕНЕРАТОРІВ ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

1.1 Математична модель захищеної інформаційно-телекомунікаційної системи

Розглянемо процеси захисту інформації з використанням криптоперетворень за структурною схемою, що наведена на рис. 1.1 [1].

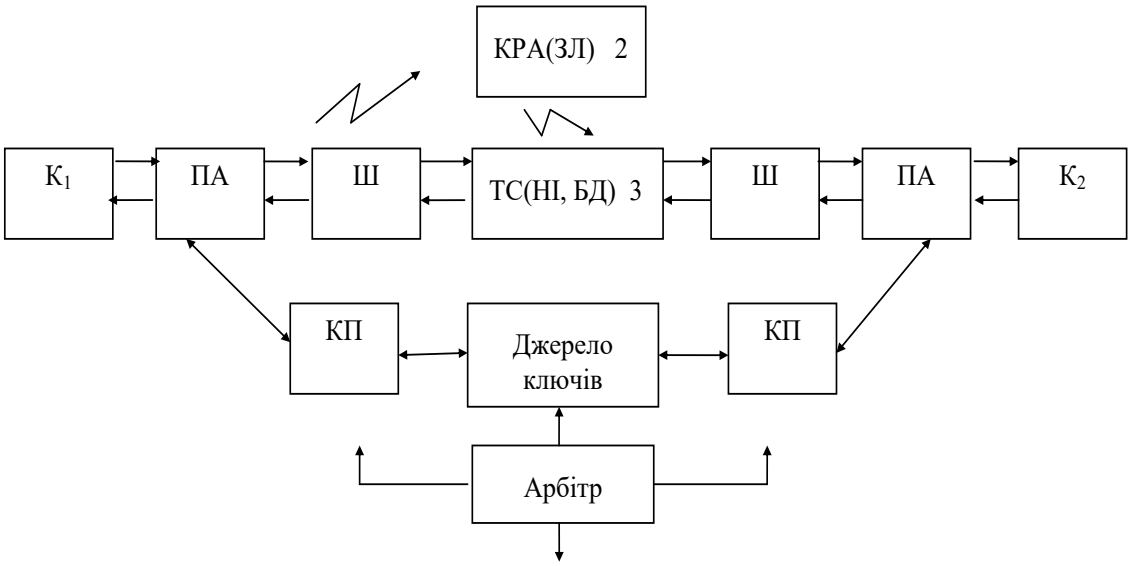

Рисунок 1.1 – Структурна схема захищеної ІТС

Вважатимемо, що К1, К2 є джерелами (передавачами, приймачами) повідомлень M_i для яких відомий апріорний розподіл ймовірностей $P(M_i)$, $i = \overline{1, n_M}$ і відома ентропія джерел повідомлень

$$H = -\sum_{i=1}^{n_M} P(M_i) \log_2 P(M_i). \qquad (1.1)$$

Повідомлення M_i з метою забезпечення його цілісності та справжності автентифікуються в пристрої автентифікації (ПА) засобом криптографічного

перетворення відповідно до ключа K_j^a. В результаті на виході формується M_i^a

.

Шифратор (Ш) здійснює зашифрування за правилом:

$$C_i = F_{пр}(M_i^{'}, K_j^з, P_r), \tag{1.2}$$

де $F_{пр}$ – функція прямого перетворення, $K_j^з$ – ключ зашифрування, P_r – параметри криптоперетворення.

У результаті на виході шифратора формується криптограма C_j. Вважатимемо відомою апріорну статистику появи криптограм на виході шифратора – $P(C_j)$, $j = \overline{1, n_c}$.

Умовна апріорна імовірність

$$P\left(C_j \middle/ M_i\right) = P\left(K_{ij}\right) \tag{1.3}$$

співпадає з імовірністю використання для зашифрування K_{ij} ключа.

При передачі криптограми у телекомунікаційній системі (ТС), а також при зберіганні на носії інформації (НІ) або в базі даних (БД) вона може прийняти значення C_j^* внаслідок дій завад в телекомунікаційній системі або внаслідок її викривлення. Тому на вхід шифратора користувача К2 надходить C_j^*, де * – означає факт можливості викривлення за рахунок перекручування або дій перешкод, а також дій криптоаналітика .

Розшифрування здійснюється за правилом:

$$M_i^* = F_{зв}(C_i^*, K_j^P, P_r), \tag{1.4}$$

де $F_{зв}$ – функція зворотного перетворення, K_j^p – ключ розшифрування. При цьому для розшифровування використовується ключ K_j^p розшифрування.

У результаті на виході шифратора (Ш) з'являється автентифіковане повідомлення M_i^{a*} [2].

В приладі автентифікації (ПА) користувача К2 здійснюється обчислення відкритого підпису за законом ключа $K_j^{a'}$. У результаті на виході ПА формується повідомлення M_i^*. Користувач отримує повідомлення M_i^*.

Для працездатності системи К1, К2 повинні мати та використовувати узгоджено пари ключів, для автентифікації – ($K_J^a, K_j^{a'}$), для шифрування – ($K_j^з, K_j^p$). Ці функції виконує джерело ключів із ключовими пристроями (КП), що є як у користувача К1 так і користувача К2 [2].

Якщо $K_j^a = K_j^{a'}$, то криптосистема називається симетричною, якщо $K_j^a \neq K_j^{a'}$, то асиметричною.

Таким чином, згідно з розглянутою моделлю цілісність та справжність забезпечуються за рахунок здійснення у передавача криптографічного перетворення типу ЕЦП, а у приймача за рахунок перевірки ЕЦП. Конфіденційність та захист від несанкціонованого доступу здійснюються за рахунок виконання криптографічного перетворення типу шифрування (зашифрування/розшифрування).

Криптоаналітик (КРА) в розглянутій моделі може здійснювати протидію, реалізуючи погрози, що були наведені на рис. 1.1 та ін. Для цього він використовує спеціалізовані криптоаналітичні системи та засоби криптоаналізу. В моделі прийнято, що криптоаналітик з великою імовірністю має доступ до каналів телекомунікаційної системи, перехоплює криптограми, які передаються, може їх модифікувати, а також може створювати хибні криптограми. Ці дії здійснюються ним з метою нанесення втрат інформаційній технології або системі [2].

Крім вказаних криптографічних перетворень в ІТС можуть також здійснюватися і стеганографічні перетворення, коли за допомогою криптографічного перетворення здійснюється приховування наявності інформації або її передавання.

1.2 Управління ключами в криптографічних системах

Однією з основних умов забезпечення необхідного рівня гарантій криптографічної стійкості є застосування і відповідне управління ключовими даними та ключовою інформацією. У свою чергу рівень гарантій криптографічної стійкості дозволяє надавати з необхідною якістю такі базові послуги, як конфіденційність. цілісність, автентичність (справжність), неспростовність, доступність тощо. В криптографічних системах створюються спеціальні підсистеми – джерела ключових даних і ключової інформації, а також здійснюється управління ключовими даними (ключами). При цьому під ключовими даними (ключами) розуміється сукупність випадкових або псевдовипадкових значень змінних параметрів криптографічного перетворення інформації, за рахунок яких досягається мета цього перетворення (наприклад, зашифровування, розшифровування, обчислення криптографічного контрольного значення, обчислення електронного цифрового підпису, перевірка електронного цифрового підпису, формування сертифіката відкритого ключа тощо). У ряді випадків також вживається поняття ключова інформація, під якою розуміють ключові дані (ключі), значення таємного ключа, що розділюється, значення для ініціалізації криптографічних засобів та їхніх елементів. синхронізуючі для криптографічних перетворень послідовності тощо. Під управлінням ключовими даними будемо розуміти дії, що пов'язані з генеруванням або придбанням, реєстрацією, розподіленням (розповсюдженням), сертифікацією, доставкою, уведенням у дію (інсталюванням), зміненням, зберіганням, архівуванням, скасуванням, блокуванням, поновленням, зняттям з реєстрації, обліком та знищенням ключової інформації (даних), а також носіїв ключових даних. У ряді застосувань для безпосереднього використання ключових даних використовують ключові документи матеріальні документи із зафіксованими відповідним чином ключовими даними, що призначені для подальшого практичного їх застосування в процесі криптографічних перетворень інформації та управління ключовими даними.

1.3 Вимоги до генераторів випадкових чисел (послідовностей)

Основною складовою, яка визначає якість ключів, е генератори випадкових чисел (послідовностей). Випадкові числа використовуються для побудови гами в поточних криптосистемах, ключів для сеансів (сеансових) та інших ключів у блочних криптосистемах, початкових значень, для генерації параметрів в асиметричних криптосистемах, випадкових значень параметрів для багатьох систем електронного цифрового підпису, «випадкових наборів» даних у протоколах автентифікації тощо.

Визнаним е той факт, що криптографічна стійкість криптографічних перетворень і безпечність реалізації різноманітних криптографічних протоколів суттєво залежать від того, яким чином генеруються та застосовуються різні види ключових даних (ключів). Загальним підходом до генерування ключів с застосування для цього генераторів випадкових послідовностей та/або детермінованих генераторів випадкових послідовностей (бітів). Принциповою відмінністю чисто випадкових послідовностей від псевдовипадкових послідовностей (бітів) є те, що псевдовипадкова послідовність може бути відновлена у просторі й часі без попереднього її запису. Випадкова ж послідовність може бути відновлена тільки якщо її попередньо записати і в подальшому зберігати, розповсюджувати уводити в дію і т д. Ще раз підкреслимо, то, як правило, при генеруванні ключів та ключової інформації вводять і застосовують ключ генератора ключів, який визначає його ентропію як джерела ключів.

До генераторів випадкових послідовностей (ГВП) детермінованих генераторів випадкових послідовностей (ДГВП) (бітів) висуваються складні вимоги щодо генерування символів послідовності випадково, рівномірно, незалежно та однорідно. При цьому факт компрометації ключа є критичним явищем найвищою мірою. Також має бути забезпечене оперативне відновлення ключа у просторі і часі.

Нині загальним підходом до генерування ключів, ключової інформації та параметрів є стандартизація методів, механізмів і практичних (конкретних) алгоритмів їх генерування. Причому, як можна судити із ряду джерел, ці методи, механізми й алгоритми намагаються захистити від розповсюдження, особливо в частості генерування випадкових послідовностей. Також, у зв'язку із суттєвим розвитком інфраструктури відкритих ключів, виникла потреба у створенні апаратних, апаратно-програмних і програмних засобів генерування асиметричних пар ключів. Були розроблені та прийняті спочатку регіональні, а потім і міжнародні стандарти, у яких були визначені вимоги, методи, механізми та алгоритми реалізації генераторів. Причому у зв’язку з необхідністю відновлення ключів та ключової інформації у просторі й часі, у них в повному обсязі розглядаються тільки детерміновані генератори випадкових бітів, що визначає їх особливу актуальність.

Основними вимогами, що висуваються до детермінованих генераторів випадкові бітів (ДГВБ), є непередбачуваність, просторова і часова складність, відновлюваність у просторі й часі, необоротність, а також період повторення.

Запропоновано декілька підходів до визначення рівнів гарантій. Перший із них пов'язаний із тестуванням псевдовипадкових бітів (тобто випадкових бітів, сформованих детермінованим генератором випадкових бітів) на випадковість, для чого, наприклад, застосовується стандарт FIPS 140-1 або AIS 20. Більш детальними є вимоги та механізми реалізації, визначені в AIS 20, що дозволяє реалізувати різні рівні гарантій К1, К2, КЗ, К4. При цьому найвищим рівнем гарантій є рівень К4. В AIS 31 визначено два рівні гарантій Р1 і Р2, у яких, по суті, Р1 дещо еквівалентний К1, К2, а Р2 еквівалентний КЗ, К4. У разі рівня гарантій К4 вимагається, щоб псевдовипадкові біти мали статистичні властивості, подібні до статистичних властивостей псевдовипадкових бітів, що генеровані ідеальним ДВГБ, була задана ентропія джерела ключів (тобто наявність ключа генератора обов’язковою), а також має бути практично виключена можливість обчислення попередніх і наступних бітів генератора при відомому поточному стані.

Аналіз основоположних джерел дозволяє зробити висновок, що необхідною умовою забезпечення криптографічної стійкості є формування ключів, ключової інформації та певних параметрів, що досягається використанням одночасно як засобів формування фізично випадкових і детермінованих випадкових послідовностей. Якщо ця необхідна вимога не виконується, то говорити про певний рівень криптографічної стійкості немає сенсу. Крім того, є визнаним той факт, що криптографічні протоколи з нульовим розголошенням можуть бути реалізовані тільки за умови використання для формування ключів, ключової інформації та параметрів фізично випадкових засобів.

У цілому система управління ключами являє собою комплексну організаційно-технічну систему, що забезпечує надання всіх послуг з питань генерування, реєстрації, накопичування, розподілення, збереження, передавання, приймання, уведення в дію, використання, архівування, знищення ключів та іншого ключового матеріалу, що використовується при здійсненні криптографічних перетворень.

Так, для реалізації потокового симетричного шифру до криптографічно стійкого генератора псевдовипадкової послідовності чисел висуваються три основних вимоги:

– період гами має бути досить великим для шифрування повідомлень різної довжини;

– гама мас бути практично непередбачуваною, що означає неможливість передбачити наступний біт гами, навіть якщо відомі тип генератора і попередній відрізок гами;

– генерування гами не повинне викликати великих технічних складностей.

Із наведеного вище можна зробити висновок, що від якості випадковості формування ключів, ключової інформації та системних параметрів суттєво залежить криптографічна стійкість. При цьому алгоритми генерації і тестування послідовностей випадкових чисел є базовими алгоритмами, що

забезпечують дійсну криптографічну стійкість алгоритмів і механізмів криптографічного захисту інформації.

Базовими міжнародними стандартами, що стандартизують алгоритми генерування послідовностей випадкових чисел, є:

– міжнародний стандарт ISO/IEC 18031 "Information technology Random number generation", який визначає алгоритми генерування псевдовипадкових і випадкових чисел, а також визначає статистичні тести перевірки генераторів;

– міжнародний стандарт ISO/IEC 18032 "Information technology-Prime number generation", який визначає методи генерування простих чисел методи тестування чисел на простоту;

– національний стандарт ДСТУ ISO/IEC19790 «Інформаційна технологія – Методи захисту – Вимоги щодо захисту для криптографічних модулів». Додаткові вимоги до алгоритмів та реалізацій методів і засобів генерування і тестування послідовностей випадкових чисел визначаються національними та промисловими стандартами США- FIPS 140-3, ANSI X9.17, ANSI X9.31, ANSI X9.44 та ін., а також рекомендаціями NIST-NIST SP 800-22 і рекомендаціями органу зі стандартизації Німеччини-AIS-20, AIS-31 та ін.

1.4 Критерії і показники оцінки випадкових і псевдовипадкових послідовностей та вимоги до них

Одним із важливих та необхідних напрямків досліджень при створенні ефективних генераторів випадкових чисел та генераторів псевдовипадкових чисел є розробка методів та засобів оцінки статистичних властивостей випадкових послідовностей. Статистичні показники та побудовані на їх основі критерії оцінки є інструментом перевірки правильності технічних рішень щодо побудови ГВЧ [3]. Дослідження статистичних властивостей здійснюються у рамках методики статистичних випробувань на основі статистичних тестів. Так, наприклад, методика статистичного тестування NIST STS налічує 16 статистичних тестів, які дозволяють з великою мірою довіри відбракувати послідовності, які не відповідають вимогам випадковості.

Стандарт FIPS 140-1 містить чотири основних статистичних тести [14]. По результатам проходження цих тестів приймається рішення про випадковість послідовності.

Методика AIS 20 представляє критерії для оцінки детермінованих генераторів випадкових чисел (ДГВЧ). Основна ідея полягає в тому, що придатність ДГВЧ повинна бути оцінена з урахуванням криптографічних додатків, у яких вони використовуються [3]. Методика налічує 5 статистичних тестів. Методика тестування AIS 20 може застосовуватись як в реальному часі, так і в процесі досліджень, та для технологічного тестування [13].

В AIS 31 [5] представлені критерії оцінки криптографічних властивостей генераторів випадкових чисел, які базуються на математично-технічній основі AIS 20 [3]. Оцінка фізичних генераторів випадкових чисел (ФГВЧ) ґрунтується в основному на статистичних тестах. На основі різних можливих сценаріїв атак розробляються вимоги до властивостей зовнішніх і відповідно внутрішніх випадкових чисел [3]. Методика налічує 8 статистичних тестів. Попередні дослідження та тестування підтвердили, що AIS 31 є надійним тестом і по своїй ефективності забезпечує практично ті ж результати, що і NIST STS. Перевагою AIS 31 є те, що він забезпечує тестування в реальному часі [13].

Властивості ПВП оцінюють з використанням ряду кількісних показників [2]. Основними показниками є:

– період l (довжина) ПВП;

– основа алфавіту m;

– ймовірність перекриття в просторі або в часі двох сегментів Y_r та Y_μ, тобто в різних абонентів або в одного абонента протягом часу, так, що $Y_r = Y_\mu$;

– структурна скритність (еквівалентна складність) S_e послідовності Y;

– кількість (ентропія $N_k(H_k)$ джерела) ключів для випадку, коли генератор ПВП використовується як джерело ключів;

– відстань рівнозначності l_0 конкретної послідовності Y_v;

– безпечний час ГВЧ t_b;

– складність I_y формування послідовності Y;

– довжина параметрів зворотного зв'язку B_2 та B_3;

– властивості випадковості, рівноймовірності, незалежності та однорідності.

За всіма названими показниками до генератора ПВП повинен бути пред'явлений ряд вимог [2]. Так період повторення $l_n \geq l_z$, тобто повинен бути не менше заданого, основа алфавіту *m*, ймовірність перекриття $P_n < P_z$ менше допустимої, структурна скритність $S \geq S_g$, ентропія джерела ключів $H \geq H_g$, відстань рівнозначності $l_0 > l_g$, безпечний час $t_b > t_g$, тобто не менш допустимих. Крім того, реалізація Y_i повинна задовольняти вимогам випадковості, рівноймовірності, незалежності та однозначності.

2 СУЧАСНІ ГЕНЕРАТОРИ ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

Стійкість криптографічних перетворень забезпечується при умові, що параметри генеруються рівномірно і випадково, а також в реальному часі і з високою швидкістю. Для цього використовуються генератори випадкових та псевдовипадкових послідовностей.

Генератором випадкових послідовностей називають пристрій або алгоритм, який формує на своєму виході Y_i послідовність статистично незалежних символів з основою алфавіту m.

Генератором псевдовипадкових послідовностей називають детермінований алгоритм, сукупність алгоритмів чи сукупність алгоритмів та засобів, які для заданої послідовності довжиною k формують при своїй роботі послідовність Y_i символів довжиною $l >> k$, яка володіє більшістю властивостей випадкової послідовності. На рис.2.1 наведено спрощену схему генератора псевдовипадкових послідовностей, де:

1) генератор випадкової послідовності довжиною k;

2) схема формування початкових значень k та П параметрів ГПС;

3) генератор псевдовипадкових символів;

4) логіка зворотного зв'язку.

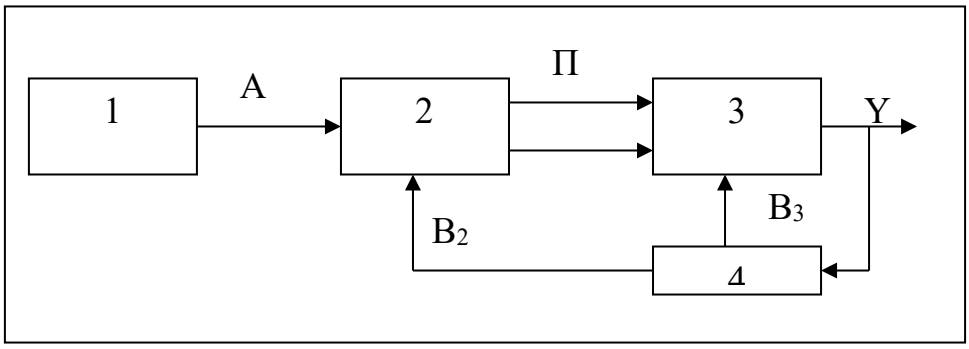

Рисунок 2.1 – Генератор псевдовипадкових послідовностей

В загальному вигляді псевдовипадкову послідовність (ПВП) Y можна задати як залежність:

$$Y = F(\Pi(A_i), K(A_j), A_v)$$

від початкових значень ключа К та параметрів П, а також випадкових послідовностей A_v. При цьому зворотний зв'язок Y призводить до впливу його на параметри, початкові значення та роботу генератора 3.

Основним призначенням ГВЧ в криптографії є формування ключів, початкових значень ключів та параметрів, параметрів та синхромаркерів.

2.1 Класифікація генераторів випадкових послідовностей

Розрізнюють генератори випадкових (фізичні, недетерміновані) чисел, псевдовипадкових (нефізичні, детерміновані) чисел, квантові генератори.

В генераторах випадкових чисел кожен біт вихідних даних базується на непередбаченому фізичному процесі.

Згідно AIS 31 [5] фізичний генератор випадкових чисел (ФГВЧ) виробляє з шумового сигналу внутрішнього фізичного джерела шуму випадкові числа. Значення, одержані безпосередньо дискретизацією аналогового сигналу шуму, називаються надалі дискретизованим шумовим сигналом.

ФГВЧ містить внутрішнє фізичне джерело шуму [15]. Частіше за все він виробляє аналоговий сигнал, який надалі дискретизується. Дискретизований шумовий сигнал при подальшій обробці перетворюється на внутрішню послідовність випадкових чисел, щоб поліпшити розподіл вірогідності дискретизованої послідовності шумових сигналів. Хороше фізичне джерело шуму може без додаткової обробки дискретизований шумовий сигнал передавати безпосередньо у вихідний блок. В цьому випадку послідовність внутрішніх випадкових чисел відповідає дискретизованій послідовності шумового сигналу. Вихідний блок синхронізує безперервну або аперіодичну видачу внутрішньої випадкової послідовності з видачею послідовності випадкових чисел. Ентропія видаваної джерелом шуму послідовності

випадкових чисел підвищується з генерацією кожного випадкового числа. В цьому випадку фізичний ГВЧ представляється ідеальним ГВЧ [5].

Детермінований ГВЧ використовує алгоритм, що виробляє послідовність бітів з початкового значення, обумовленого початковим числом. ДГВЧ вважається реалізованим, коли отримано початкове число і визначене початкове значення. У зв'язку з детермінованим характером процесу вважається, що ДГВЧ виробляє псевдовипадкові, а не випадкові біти. Початкове число, яке використовується для реалізації ДГВЧ, повинно містити достатню ентропію для гарантії випадковості [6]. Для ДГВЧ не можливо збільшити повну (сумарну) ентропію послідовності випадкових чисел за межі ентропії початкового числа, шляхом генерації нових випадкових чисел. Це велика відмінність від фізичних джерел шуму [7]. Тому послідовності випадкових чисел, згенеровані ДГВЧ, не можуть бути дійсно “випадковими”. В кращому разі вони можуть поводитися подібно істинно випадковим послідовностям відносно певних критеріїв.

2.2 Квантові генератори випадкових чисел

В квантових рішеннях ентропія біта вважається апріорно рівною 1, займаючи позицію абсолютної непередбачуваності значення цього біта. У той же час для псевдовипадкового біта ентропія вважається рівною 0, оскільки значення цього біта можна передбачити, хоча часто витрати обчислювальних ресурсів можуть бути дуже високими, що на практиці ускладнює для класичних комп'ютерів вгадати біт (проте, це можливо, і тому ентропія дорівнює 0). Зменшення ентропії квантового біта відбувається в результаті домішки класичного детермінованого хаосу. Використання програмних методів для усунення зміщення та відбілювання сигналу, у свою чергу, збільшує ентропію. Однак ви ніколи не можете бути впевнені в ролі окремих факторів, і вам слід підходити до їх оцінки з обережністю, особливо якщо вони наведені виробником. У цьому контексті, важливою стає можливість вибірки

випадковості послідовності, створеної за допомогою статистичних методів, і прийняття договірних критеріїв якості випадковості.

Передумовою абсолютної випадковості апаратних квантових генераторів випадкових чисел є переконання, що проекція фон Неймана є абсолютно випадковою. Таким чином, вимірювання на стані суперпозиції принаймні двох станів (кубіт) призводить до генерації випадкової послідовності. Тут слід розрізняти два етапи:

1) Підготовка вхідного стану (це може бути той самий відомий квантовий стан або також випадково вибраний стан джерела – в останньому випадку можлива випадковість джерела та його якість також важлива для випадковості другого етапу -вимірювання).

2) Вимірювання ряду стану – цей процес генерує квантову випадковість і в ідеалі гарантує абсолютну випадковість кінцевої послідовності.

Абсолютна випадковість кінцевої послідовності може бути скомпрометована несправним джерелом. Якщо, наприклад, джерело буде забезпечувати власний стан вимірюваної величини з деякою частотою, випадковість результату буде сильно порушена. Тому крок (1) такий же важливий, як і крок (2). Більше того, результат кроку (2) завжди певною мірою змішується з класичним шумом, що виникає в результаті макроскопічної практичної реалізації проекції фон Неймана. Тут слід підкреслити, що проекція фон Неймана завжди виконується за допомогою макроскопічного приладу і лише в ідеалізованій ситуації проведення вимірювального експерименту не вносить випадкових класичних збурень.

Генерована послідовність бітів надзвичайно чутлива до різних форм зміщення. Зменшити зміщення відносно просто, тоді як ідентифікація класичного неявного компонента (кореляції), залученого до створеної послідовності, набагато складніше і не завжди ефективне програмними методами. Скоріше, ми повинні покладатися на фізичне розпізнавання всього явища, фізичне виявлення та мінімізацію класичних компонентів випадковості. Для зменшення зміщення та декореляції доступні різні алгоритми відбілювання сигналу. Вони є найпоширенішою розробкою

алгоритму фон Неймана. Відповідно до цього алгоритму порівнюються два послідовні біти послідовності, якщо вони однакові, обидва відхиляються, якщо вони 0,1, то вважається 0, якщо вони 1,0, це вважається 1. Отримана послідовність є збалансованою, але принаймні вдвічі коротшою та випадковою, оскільки у вихідній послідовності немає кореляції. Досконалішими екстракторами рандомізації є, наприклад, екстрактор Тревізана або екстрактор Тепліца з використанням швидкого перетворення Фур'є. Загалом, відбілювачі випадкової послідовності працюють самі по собі як генератори псевдовипадкових дій. Хорошим прикладом є алгоритм Blum, Blum, Shub (BBS). Він повертає послідовність з вихідного насіння x_0, згідно з формуою,

$$x_i = \left(x_0^{2^i mod \lambda(M)}\right) mod(M),$$

де M = p × q, p, q – високі прості числа. Побітовим результатом процедури є паритет x_{n+1} або, наприклад, останній значущий біт x_{n+1}. Початок x_0 має бути відносно простим до q і p і не може бути 0 або 1. Цікавою особливістю генератора BBS є аналітична форма результату,

$$x_i = \left(x_0^{2^i mod \lambda(M)}\right) mod(M),$$

де λ(M) – функція Кармайкла. Ця функція, визначена додатним цілим числом n, позначеним як λ(n), визначається як найменше натуральне число m, таке, що $a^m = 1 mod(n)$ для кожного цілого відносно простого відносно n. Тому легко вгадати всю випадкову послідовність, знаючи насіння і числа p, q.

Використання різних алгоритмів запобігання зміщенню та антікореляції (балансування та відбілювання розпаду) є програмною обробкою необробленої послідовності та повністю детермінованою (хоча зазвичай

складною з точки зору хеш-функцій). Незважаючи на ефективне усунення зміщення, надмірно складні процедури екстракторів випадковості можуть самі по собі порушувати/заховувати квантову випадковість, що міститься в необробленій послідовності, додаючи власний псевдовипадковий компонент до суміші з класичним внеском шуму. Тому важливим є пошук апаратних рішень квантового генератора випадкових чисел з відносно невеликою класичною домішкою. QRNG, що використовують вимірювання кубіта фон Неймана, наприклад, реєстрацію фотонів, обмежені в швидкості релаксації вимірювального пристрою – одиночні фотонні детектори (наприклад, лавинні діоди або фотопомножувачі) мають інерцію порядку 100 ns, що обмежує швидкість генерації випадкової послідовності до Мбіт/с. Це занадто низька швидкість генерації для криптографічних програм, де необхідна швидкість має становити до Гбіт/с або навіть 100 Гбіт/с. Згідно з оглядом QRNG, таку швидкість гігабайт можна продемонструвати в генераторах, які сильно підтримуються програмним забезпеченням, що є компромісом для продуктивності.

Також пропонуються QRNG, які перевіряють квантову випадковість генерованої послідовності. Авторизація квантової випадковості тут використовується для перевірки порушення нерівності Белла та відкидання фрагментів, що не відповідають цьому критерію. Такі генератори отримують високий рівень достовірності навіть з неповністю охарактеризованими та випадковими джерелами. Однак вони сповільнюють рутину.

Квантові апаратні генератори випадкових чиселможна розділити на три категорії:

1) Практичні квантові генератори випадкових чисел– повністю надійні та відкалібровані пристрої. Випадковість залежить від правильного моделювання та реалізації фізичного квантового процесу. Як правило, швидкість генерації помірна, а вартість пристрою відносно невисока. На практиці в цих пристроях квантова випадковість часто змішується з класичним шумом (який, однак, можна видалити, якщо основний квантовий процес змоделювати належним чином). Для цих пристроїв безпека залежить від

довіри до пристрою та його компонентів, що може бути проблемою при роботі зі сторонніми постачальниками.

2) Самотестовані квантові генератори випадкових чисел – згенерована послідовність перевіряється на випадковість через обмежену впевненість у реалізації фізичного процесу. Тестування може базуватися на класичних тестах, а також, наприклад, на перевірці існування квантової заплутаності шляхом перевірки нерівностей Белла. Ці пристрої також відомі як незалежні від пристроїв квантових генераторів випадкових чисел. Через складність процесу тестування такі генератори зазвичай повільні або вимагають додаткових складних перевірочних пристроїв.

3) Напівсамотестовані квантові генератори випадкових чисел – ця категорія включає пристрої, в яких перевірка випадковості була зменшена завдяки впевненості реалізації. Це дозволяє оптимізувати параметри швидкості з вартістю довіри до генерованої випадковості. Деякі компоненти в таких пристроях вважаються безпечними і надійними через їх точні характеристики, інші не можна вважати такими, і тому необхідно проводити більш розширені випробування.

3 ЕКСТРАКТОРИ ВИПАДКОВОСТІ

3.1 Поняття екстрактору випадковості

Екстрактор випадковості, часто званий просто «екстрактором», – це функція, яка застосовується для виведення зі слабко випадкового джерела ентропії разом з коротким, рівномірно випадковим початковим значенням, що генерує дуже випадковий висновок, який здається незалежним від джерела і рівномірно розподіленим. Приклади слабко випадкових джерел включають радіоактивний розпад або тепловий шум; Єдине обмеження на можливі джерела полягає в тому, що їх неможливо повністю контролювати, розраховувати або передбачати і що можна встановити нижню межу їхньої ентропії. Для даного джерела екстрактор випадковості може розглядатися як істинний генератор випадкових чисел (TRNG); але не існує єдиного екстрактора, який, як було доведено, справляв справді випадковий результат із будь-якого типу слабко випадкового джерела.

Іноді термін «систематична помилка» використовується для позначення відхилення слабо випадкового джерела від одноманітності, а в більш ранній літературі деякі екстрактори називаються алгоритмами незміщення, оскільки вони беруть випадковість з деякого, так званого «упередженого» джерела і виводять розподіл, який здається неупередженим. Слабко випадкове джерело завжди буде довшим, ніж потужність екстрактора, але ефективний екстрактор - це той, який максимально знижує це співвідношення довжин, одночасно зберігаючи низьку довжину затравки. Інтуїтивно це означає, що з джерела «витягнуто» якнайбільше випадковості.

Зверніть увагу, що екстрактор має деяку концептуальну подібність із генератором псевдовипадкових чисел (PRNG), але ці дві концепції не ідентичні. Обидві функції приймають на вході невелике рівномірно випадкове початкове число і роблять більш довгий висновок, який виглядає рівномірно випадковим. Деякі псевдовипадкові генератори є також екстракторами. (Коли

PRNG заснований на існуванні основних предикатів, можна думати про слабовипадкове джерело як про набір таблиць істинності таких предикатів і доводити, що результат статистично близький до однорідного.) Однак у загальному визначенні PRG не вказується, що має використовуватися слабо випадкове джерело, і хоча у випадку екстрактора вихід повинен бути статистично близьким до однорідного, в PRNG потрібно лише бути обчислювально невідмінним від одноманітного, дещо слабшого поняття.

NIST Спеціальна публікація 800-90B рекомендує кілька екстракторів, включаючи сімейство геш-функцій SHA, і заявляє, що якщо кількість ентропії, що вводиться, вдвічі перевищує кількість бітів, що виводяться, то вихід можна вважати цілком випадковим.

3.1.1 Формальне визначення

Мін-ентропія розподілення X (визначається як $H_\infty(X)$) є найбільшим матеріальним числом k таким, що $PR[X = x] \leq 2^{-k}$ для будь-якого x з X. По суті, це яка ймовірність, що X прийме його найбільш ймовірне значення, при найгіршому розподілі. Позначимо U_l як рівномірний розподіл на $\{0,1\}^l$, чи $H_\infty(U_l) = l$

Для n-бітного розподілу X з мін-ентропією k кажуть, що X є (n, k) розподіленням.

Визначення (Екстрактор): (k, ε) – екстрактор. Нехай

$$\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$$

– функція, яка приймає на вхід вибірку з (n, k) розподілення X, d-бітне начальне значення з U_d та повертає m-бітну строку. Ext буде являти собою (k, ε) – екстрактором, якщо для всіх (n, k) розподілень X вихідне розподілення не далі від U_m, чим від ε.

У наведеному вище визначенні мається на увазі статистичну відстань.

Таким чином, екстрактор бере слабко випадковий n-бітний вхід, короткий випадковий початковий розмір і видає m-бітний вихід, який виглядає рівномірно випадковим. Мета полягає в тому, щоб зробити маленький d (тобто використовувати якнайменше рівномірної випадковості) і якнайбільше m наскільки це можливо (тобто щоб отримати якомога більше близьких до випадкових біт вихідних даних).

3.1.2 Сильні екстрактори

Екстрактор є сильним, якщо конкатенація початкового значення з виходом екстрактора призводить до розподілу, який є близьким до рівномірного.

Визначення (сильний екстрактор): (k, ε) – екстрактор: A (k, ε) – сильний екстрактор, це функція

$$\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

така, що для кожного (n, k) розподілу X розподіл $U_d \cdot \mathrm{Ext}(X, U_d)$ (дві U_d означають ту саму випадкову величину) далі від рівномірного розподілу не більше ніж на ε по U_{m+d}

3.1.3 Явні екстрактори

Використовуючи ймовірнісний метод, можна показати, що існує (k , ε) – екстрактор. Однак зазвичай недостатньо просто показати, що екстрактор існує. Необхідна явна конструкція.

Визначення (явний екстрактор): для функцій k(n), ε(n), d(n), m(n) сімейство $\mathrm{Ext} = \{\mathrm{Ext}_n\}$ функцій

$$\mathrm{Ext}_n : \{0,1\}^n \times \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$$

є явним (k, ε) – екстрактором, якщо Ext(x, y) може бути обчислено за поліноміальний час (за довжиною його входу) і для кожного n, Ext_n є (k(n), ε(n)) – екстрактором.

Імовірнісним методом можна показати, що існує (k, ε) – екстрактор з початковим значенням

$$d = \log(n-k) + 2\log\left(\frac{1}{\varepsilon}\right) + O(1)$$

та довжиною

$$m = k + d - 2\log\left(\frac{1}{\varepsilon}\right) - O(1).$$[5]

3.2 Екстрактори випадковості в криптографії

Одним з найважливіших аспектів криптографії є генерація випадкових ключів. Часто необхідно генерувати секретні випадкові ключі з напівсекретних джерел, які можуть бути певною мірою скомпрометовані.

ГВЧ зазвичай складається з двох компонентів, джерела ентропії та екстрактора випадковості. У QRNG джерелом ентропії може бути фізичний пристрій, вихід якого принципово непередбачуваний, тоді як екстрактор випадковості може бути алгоритмом, який генерує майже ідеальні випадкові числа з вихідних даних попереднього джерела ентропії, який може бути не зовсім випадковим. Два компоненти QRNG пов’язані кількісним визначенням випадковості з мінімальною ентропією. Мінімальна ентропія джерела ентропії

спочатку оцінюється, а потім подається в екстрактор випадковості як вхідний параметр.

Приймаючи єдиний короткий (і секретний) випадковий ключ як джерело, екстрактор можна використовувати для генерації довшого псевдовипадкового ключа, який можна використовувати для шифрування з відкритим ключем. Зокрема, коли використовується сильний екстрактор, його висновок виглядатиме рівномірно випадковим навіть для того, хто бачить частину (але не все) джерела. Наприклад, якщо джерело відоме, але початкове число невідоме (або навпаки). Ця властивість екстракторів особливо корисна у так званій криптографії, стійкій до дії, в якій необхідний екстрактор використовується як стійка до дії функції (ERF). Стійка до впливу криптографія враховує той факт, що важко зберегти в таємниці початковий обмін даними, який часто відбувається під час ініціалізації шифрування, наприклад, відправник зашифрованої інформації повинен надати отримувачам необхідну інформацію для розшифрування.

Недосконалу випадковість джерела ентропії вже можна побачити в схемах на основі SPD, таких як схема виявлення числа фотонів. Позначаючи N як верхню межу дискримінації детектора, що розрізняє число фотонів, за подію виявлення можна генерувати щонайбільше log2(N) необроблених випадкових бітів. Однак, оскільки числа фотонів джерела когерентного стану мають розподіл Пуассона, необроблені випадкові біти мають нерівномірний розподіл; отже, ми не можемо отримати log2(N) бітів випадкових чисел. Щоб отримати абсолютно випадкові числа, нам потрібна процедура постобробки (тобто екстрактор випадковості).

У QRNG на основі когерентного виявлення квантова випадковість неминуче змішується з класичними шумами, що вносяться детектором, та іншими недоліками системи. Більше того, будь-яка вимірювальна система має скінченну пропускну здатність, що передбачає неминучі кореляції між сусідніми зразками. Після кількісної оцінки ці небажані побічні ефекти можна усунути за допомогою відповідного екстрактора випадковості.

3.2.1 Екстрактор фон Неймана

Один із ранніх прикладів екстракторів випадковості був запропонований Джоном фон Нейманом. Принцип його роботи полягав у наступному: з вхідного потоку він бере пари послідовних (не перекриваючихся) бітів. Якщо два біти збігаються, вихідні дані не генеруються. Якщо розрізняються біти, виводиться значення першого біта. Може бути показано, що екстрактор фон Неймана виробляє рівномірний вихідний сигнал, навіть у тому випадку, коли розподіл вхідних бітів не є рівномірним, якщо кожен біт має однакову ймовірність того, що він дорівнює одиниці, і немає кореляції між послідовними бітами.

Таким чином, він приймає як вхідні дані послідовність Бернуллі з p, необов'язково рівним 1/2, і виводить послідовність Бернуллі з p=1/2. У більш загальному сенсі, це можливо застосувати до будь-якої замінної послідовності – воно засноване тільки на тому факті, що для будь-якої пари однаково ймовірні 01 і 10: для незалежних випробувань вони мають ймовірності p · (1 - p) = (1 - p) · p у той час як для замінної послідовності ймовірність може бути складнішою, але обидві однаково можливі.

3.2.2 БСШ

На нинішній час необхідно виділити три методологічних підходів до побудування перспективних БСШ. По суті вони уже були представлені кандидатами на стандарт БСШ Європейської програми Nessie.

Перший пов’язаний з використанням СПН структур. Загальна структура – SPN (square-type) байт-байт орієнтований шифр. На основі таких структур були розроблені та знайшли визнання БСШ Rijndael та його звужена версія AES (FIPS – 197), що побудовані на основі попередньої розробки авторів – шифрі Square. Цей напрям був достатньо досліджений і за їх результатами запропоновано кандидат національного стандарту БСШ «Калина».

В процесі досліджень була звернута увага на БСШ, що мають IDEA подібну структуру. Відомо, що де факто Європейський стандарт IDEA пройшов великі випробовування часом и до цих пір забезпечує задекларований рівень стійкості. На початку 21 століття Паскалем Юнодом та Сержем Воденеєм був запропонований проект удосконаленого БСШ , що отримав назву FOX. Алгоритм IDEA NXT (раніше відомий як FOX), являє собою блоковий симетричний шифр, розроблений Паскалем Юнодом і Сержем Воденеем з лабораторії EPFL (Лузана, Швейцария). Замислений у період між 2001 і 2003 р., проект спочатку називався FOX і був опублікований в 2003 р. У травні 2005 р. він був анонсований компанією MediaCrypt за назвою IDEA NXT. IDEA NXT є нащадком алгоритму IDEA і використає розширену схему Лэя-Массея, відому своєю стійкістю до крипто аналізу. Він є власністю швейцарської компанії MediaCrypt, якій належать права на поширення IDEA і яка є власником патентів на IDEA NXT. Шифр IDEA NXT являє собою сімейство різних модифікацій шифрів з різними розмірами блоків і розмірами ключів: Standard NXT64 (64-бітовий блок, 128-бітовий ключ, 12 раундів) і Standard NXT128 (128-бітовий блок, bits, 256-бітовий ключ, 12 раундів). Можуть бути також побудовані версії Standard (з розміром ключа від 0 до 256 біт, числом раундів від 2 до 255). А також можуть завантажуватися індивідуальні таблиці (sbox, матриця перестановок - permutation matrix), що заміняють стандартну таблицю.

В основу реалізації третього методологічного підходу покладено уже добре випробувану Фейстель подібну схему. Вона реалізована в випробуваних часом стандартах БСШ DES, DEA, TDEA, ГОСТ 28147 – 89, а також в MISTY1, Cammellia. На нинішній час стандарти БСШ, що мають Фейстель подібну структуру, ще в значній мірі застосовуються на практиці, та не втратили перспективу застосування, можливо при деякому удосконаленні.

Нині на міжнародному рівні спільними світовими зусиллями розроблено та детально досліджено значну кількість БСШ. Щодо БСШ також прийнятий міжнародний стандарт ISO/IEC 18033-3. Перелік та деякі характеристики

алгоритмів БСШ цього стандарту наведено в табл. 3.1. У стандарті визначено шість БСШ

Таблиця 3.1 – блокові шифри, що визначені ISO/IEC 18033-3

Довжина блока	Назва алгоритму (пункт)	Довжина ключа
64 бітів	TDEA (4.1)	128 або 192 бітів
	MISTY1 (4.2)	128 бітів
	CAST-128 (4.3)	
128 бітів	AES (5.1)	128, 192 або 256 бітів
	Camellia (5.2)	
	SEED (5.3)	128 бітів

3.2.3 ПСШ

Оскільки при проектуванні будь-яких криптосхем їхня стійкість визначається, насамперед, стійкістю до відомих на сучасний момент криптографічних атак, спрямованих на виявлення вразливостей різного роду, представляється доцільним розгляд найпоширеніших методів (алгоритмів) криптоаналіза для вироблення рекомендацій щодо розробки криптографічно стійких схем шифрування.

Аналіз потокових шифрів зводиться до аналізу генератора ключової послідовності (ГКП) для знаходження відхилень від статистичної моделі, відповідно до якої ГКП розглядається як істинно випадкове джерело. Звичайно при дослідженні потокових шифрів розглядаються атаки з відомим відкритим текстом. По суті, це означає, що криптоаналітику відомо великий сегмент ключової послідовності. Знайдені статистичні відхилення використовуються засобами для здійснення таких атак.

Таким чином, успішне проведення криптоаналіза потокових шифрів можливо у випадку, якщо ключова послідовність має відхилення від істинно випадкової. Відомо, що поява будь-якого символу вихідної послідовності

двійкового симетричного джерела є рівноймовірною подією. Тоді успіх криптоаналізу можна визначити в такий спосіб.

Нехай G позначає генератор ключової послідовності, а Sn – двійкова послідовність довжиною n, Sn=s0,s1,...sn-1, si∈{0,1},n>0, яка може бути генерована як G, так і істинно випадковим джерелом, R. І нехай A позначає імовірнісний алгоритм, за допомогою якого необхідно визначити якою моделлю, G або R, була породжена послідовність Sn. На виході алгоритму з'явиться 1, якщо Sn генерована G, інакше, на виході з'явиться 0. Будемо вважати, що в результаті роботи алгоритму було виявлено, що послідовність Sn відмінна від істинно випадкової, якщо існує число const > 0 таке, що виконується умова:

$$|1/2(P(si|sn\leftarrow G)+P(si|sn\leftarrow R))-1/2|\geq const, \qquad (2.6)$$

де P(si | sn ← G) позначає ймовірність появи символів (нулів або одиниць) досліджуваної послідовності у випадку, якщо вона генерована G; P(si | sn ← R) – імовірність появи символів послідовності, отриманої з виходу істинно випадкового джерела.

Точність і час роботи алгоритму A залежить від довжини n досліджуваної послідовності.

Якщо вважати, що криптоаналітик має необмежений час і обчислювальні ресурси, то криптографічна стійкість потокового шифру не залежить від складності алгоритму формування ключового потоку й визначається складністю найбільш ефективної атаки, застосовної до даного шифру.

3.2.4 Геш-функція

Функції гешування є примітивами, що використовуються в різних криптографічних додатках. Найбільш важливі застосування належать до

електронного цифрового підпису та протоколів автентифікації. Можна виділити три основних підходи до побудови функцій гешування:

– функції гешування, що побудовані з використанням блокових шифрів;

– функції гешування, що засновані на арифметиці з перетворенням за певним модулем;

– замовлені функції гешування.

Міжнародна організація зі стандартизації ISO/IEC розробила стандарт для опису різних класів функцій гешування. У частині ISO/IEC 10118-1 подано загальні визначення, вимоги та схеми функцій гешування.

У частині ISO/IEC 10118-2 [87] визначені функції гешування, засновані на блокових шифрах у конструкції Matyas-Meyer-Oseas, коли незалежний блоковий шифр в алгоритмі MDC-2 з двома і більше функціями формує геш-значення подвоєної та потроєної довжини відповідно.

Частина ISO/IEC 10118-3 визначає три замовлених алгоритми: RIPEMD-128, RIPEMD-160 і SHA-1. Ця частина стандарту на цей час переглядається з урахуванням нових криптографічних примітивів, що будуть прийняті як стандарти ISO. Окрім відзначених трьох алгоритмів, широкого застосування набули функції гешування: SHA-2/256, SHA-2/384, SHA-2/512 і Whirlpool.

Частина ISO/IEC 10118-4 описує MASH-1 і MASH-2 функцій гешування, що використовують арифметику з перетворенням за певним модулем. Основні характеристики функції гешування наведено табл. 3.2.

Таблиця 3.2 – Характеристики функцій гешування

Функція гешування	Клас функції	Базові перетворення	Довжина геш-значення, бітів
Whirlpool	Однонаправлена	У кінцевих полях і матрицях	512
SHA-2	Однонаправлена	Логічні й арифметичні	256, 384, 512
ГОСТ 34.311-95	Однонаправлена	Блоковий симетричний шифр	256
HAVAL	Однонаправлена	Логічні й арифметичні	128, 160, 192, 256
SHA-1	Однонаправлена	Логічні й арифметичні	160
RIPEMD-160	Однонаправлена	Логічні й арифметичні	160
MD5	Однонаправлена	Логічні й арифметичні	128
MD4	Однонаправлена	Логічні й арифметичні	128
UMAC	Однонаправлена і вироблення КАП	У кільцях	128, 64
Rijndael CBC-MAC	Вироблення КАП	Блоковий симетричний шифр	128
ГОСТ 28147-89 (режим 4)	Вироблення КАП	Блоковий симетричний шифр	64

4 АТАКИ НА ЕКСТРАКТОРИ КВАНТОВИХ ГЕНЕРАТОРІВ ВИПАДКОВИХ ЧИСЕЛ

На сьогодні кіберпростір є невід'ємною частиною життя кожного громадянина України, що значною мірою впливає на економічний розвиток держави. Впровадження інтернет-технологій у державний сектор створює необхідність захисту інформації в спеціальних інформаційно - телекомунікаційних системах, а кібератаки, спрямовані на порушення роботи цих систем, становлять безпосередню загрозу економічній стабільності та суверенітету України.

Однією із основних задач кібербезспеки є кіберзахист. Відповідно до Закону України «Про основні засади забезпечення кібербезпеки України» під кіберзахистом розуміється сукупність організаційних, правових, інженерно-технічних заходів, а також заходів криптографічного та технічного захисту інформації, спрямованих на запобігання кіберінцидентам, виявлення та захист від кібератак, ліквідацію їх наслідків, відновлення сталості і надійності функціонування комунікаційних, технологічних систем. Очевидним є те, що криптографічні методи захисту інформації та криптосистеми є невід'ємною частиною організації кібербезпеки в Україні.

Основними вимогами до криптосистем є висока швидкодія при реалізаціях на різних обчислювальних платформах та стійкість, тобто спроможність протистояти усім відомим криптоаналітичним атакам.

4.1 Анализ атак на БСШ

Основними задачами криптоаналізу симетричних криптопримітивів є розробка й ефективне застосування методів, систем, комплексів, алгоритмів і засобів аналізу криптографічних систем. Криптоаналіз має здійснюватися за відомих вхідних і вихідних даних, алгоритмах чи засобах криптографічних перетворень, у тому числі можливо на частині ключових даних і ключової

інформації. Основною метою проведення криптоаналізу є визначення криптографічної стійкості криптографічних перетворень, перш за все щодо неможливості визначення спеціальних (ключових) даних тощо. Причому криптоаналіз має проводитися з метою доведення рівня гарантій відносно криптографічної стійкості, перш за все розробником з метою доведення рівня гарантій щодо криптографічної стійкості, що задекларована розробником та очікується замовником.

Щодо умов проведення криптоаналізу вважається, що криптоаналітик може здійснювати криптоатаки в таких умовах:

– атака при відомому шифр-тексті;

– атака на основі відомого відкритого тексту та відповідних йому шифр-текстів;

– атака на основі підібраного відкритого тексту;

– атака на основі адаптивно підібраного відкритого тексту;

– атака на основі підібраного шифр-тексту;

– атака на основі підібраного ключа;

– грабіжницький аналіз.

Аналіз показав, що вибір і особливості застосування методів крипто аналізу відносно БСШ повністю залежать від методу криптографічного перетворення, який реалізований у криптосистемі. Тому при криптоаналізі БСШ будемо розглядати три рівня стійкості криптосистем:

– безумовно стійкі або теоретично не дешифровані криптосистеми, відносно яких криптоаналітик не в змозі виконати криптоаналіз як практично, так і теоретично;

– обчислювально стійкі криптосистеми, відносно яких можна виконати криптоаналіз, але за належним чином вибраних розмірах параметрів і ключів існуючих матеріально технічних ресурсів не достатньо для успішного криптоаналізу;

– обчислювально нестійкі (тимчасової стійкості) криптосистеми, у яких складність криптоаналізу на межі можливостей криптоаналітика, або які мають слабкості.

Надалі аналізу та дослідженню підлягають тільки обчислювально стійкі криптосистеми на основі БСШ.

Відносно БСШ можуть обґрунтовуватись і застосовуватись різноманітні методи криптоаналізу. Але, незважаючи на це, при дослідженні криптостійкості та оцінці їх стійкості методи криптоаналізу можна поділити на два основних великих класи:

– атаки типу «груба сила»;

– аналітичні атаки.

Згідно з вимогами, що висунуті на міжнародному рівні СПБ (БСШ) будемо оцінювати за такими трьома класами стійкості:

– високий рівень безпеки, коли довжина блоку $l_б$=128 біт, а довжина ключа $l_к$=256 бітів;

– нормальний рівень безпеки, коли довжина блоку $l_б$=128 біт, а довжина ключа $l_к$=128 бітів;

– задовільний рівень безпеки, коли довжина блоку $l_б$=64 біт, а довжина ключа $l_к$=128 бітів.

Вимоги, що були висунуті до перспективного БСШ в національному конкурсі, відрізняються від Nessie вимог тим, що у національному конкурсі ще введено надвисокий рівень безпеки (гарантій), коли $l_б \geq 256$ біт, а довжина ключа lк=512 біт. Практично довжина блоку в цьому випадку $l_б$=512 бітів.

У першій фазі конкурсу NESSIE до кандидатів при відборі пред'являлися, цілком обґрунтовані, такі основні вимоги, що є по суті необхідними умовами:

1) Захищеність кандидатів БСШ від крипто аналітичних атак. При цьому обов'язковими методами крипто аналізу, від яких повинен бути забезпечений захист, вважались такі: диференціальний криптоаналіз, розширення для диференціального криптоаналізу, пошук найкращої диференціальної характеристики, лінійний криптоаналіз; інтерполяційне вторгнення;

вторгнення із частковим угадуванням ключа; вторгнення з використанням зв'язаного ключа; вторгнення на основі обробки збоїв; пошук лазівок.

2) Статистична безпечність алгоритму шифрування, під якою розуміється статистична незалежність зашифрованих блоків повідомлень та гам шифрування від блоків відкритої інформації та ключів, що використовуються.

3) Надійність математичної бази в змісті відсутності можливостей здійснювати атаки, універсальне розкриття за рахунок недосконалості або закладеної навмисної специфічної математичної бази. При цьому вважається, що така атака універсального розкриття має складність набагато меншу ніж складність атаки «груба сила».

4) Практична захищеність алгоритму шифрування від силових атак, яка може досягатись на основі використання симетричних блокових криптоперетворень з довжиною блоку не менш ніж 128 бітів та довжиною ключа не менше 128 бітів (в деяких випадках 256 або 512 біт).

5) Відсутність слабких початкових ключів та підозр на існування ключів, при яких складність криптоаналітичної атаки є меншою ніж складність атаки «груба сила».

6) Складність прямого та зворотного перетворень не перевищують допустимої величини, крім того, складність розгортання ключів не перевищує завідомо заданої величини.

7) Особливості конструкції й відкритість структури. Представлені криптоалгоритми повинні володіти зрозумілою, легко аналізованою структурою й ґрунтуватися на надійних математичних і криптографічних принципах.

8) Стійкість при модифікації. Всі кандидати перевіряються на стійкість до різного роду модифікаціям: стійкість до криптоаналітичних атак при зменшенні числа циклів, скороченні компонентів використовуваних алгоритмом і т.п.

9) Складність програмної, апаратної та програмно-апаратної реалізації повинна оцінюватися обсягом пам'яті, як для програмної, так і апаратної

реалізації. У тому числі, при програмній реалізації – кількістю необхідної оперативної пам'яті, розміром вихідного коду, швидкістю роботи програми на різних платформах при реалізації на відомих мовах програмування. При апаратній оцінюється кількістю вентилів і швидкістю в Мб/с.

10) Універсальність криптоалгоритму: можливість роботи з різними довжинами початкових ключів і інформаційних блоків; безпека реалізації на різних платформах і додатках; можливість використання криптографічного алгоритму в основних режимах роботи БСШ.

11) Параметри криптоалгоритму: криптоалгоритм повинен бути симетричним блоковим; розмір блоку даних – 128, 256, 512 біт; розмір разового (сеансу) ключа – 128, 256, 512 біт.

12) Принципи побудови: здатність протистояти відомим методам криптографічного аналізу та мати запас стійкості з урахуванням тенденцій розвитку засобів електронної обчислювальної техніки та криптографічної науки; криптографічні перетворення, що застосовуються, повинні базуватись на надійній та прозорій математичній базі та не мати вбудованих лазівок; швидкодія криптоалгоритму повинна бути не менше, ніж швидкодія існуючого державного стандарту шифрування.

13) Реалізація крипто алгоритму: криптоалгоритм повинен бути орієнтованим для можливості реалізації на 32-х або 64-х розрядних процесорах; зазначені в криптоалгоритмі операції повинні мати ефективну програмну та апаратну реалізацію; необхідний для роботи об’єм пам’яті має враховувати можливість реалізації криптоалгоритму у мікропристроях; передбачити можливість паралельного виконання декількох операцій (за можливості).

14) Ключова система: криптоалгоритм може передбачати наявність довгострокового ключа; довжина сінхропосилки – не менше 64 бітів.

15) Режими шифрування. В криптоалгоритмі повинні бути передбачені наступні режими шифрування: проста заміна; зчеплення блоків шифрованого тексту; зворотній зв’язок за входом; зворотній зв’язок за виходом; режим виробки гами («довгого циклу»).

Необхідно підкреслити, що наведені вимоги є необхідними, але не є достатніми. У цей час симетричні БСШ є основним криптографічним засобом забезпечення конфіденційності при обробці інформації в сучасних інформаційно-телекомунікаційних системах. Крім того, блокові шифри використаються для забезпечення цілісності, а також як базовий елемент при побудові інших криптографічних примітивів, таких як генератори псевдовипадкових послідовностей (ГПСЧ), потокові шифри й функції гешування. Рівень стійкості й властивості БСШ, що використовуються в системах, в суттєвій мірі визначають стійкість криптографічного захисту інформації, безпеку криптографічних протоколів і захищеність інформаційно-телекомунікаційної системи в цілому.

4.2 Анализ атак на ПСШ

В ході застосування БСШ (DES, DEA, ГОСТ 28147-89, IDEA тощо) було виявлено ряд недоліків, серед них:

– достатньо велика складність криптографічних перетворень, оскільки потрібно виконувати 10-32 і більше ідентичних циклів;

– середня швидкодія прямих і зворотних перетворень, яка не задовольняє користувачів при їх застосуванні в мережевих додатках, оскільки досягти швидкодії порядку декількох Гбітів/с практично неможливо;

– складність реалізації розпаралелювання криптоперетворень;

– усі блоки в основному режимі зашифровуються з використанням одного і того самого ключа.

У той же час в сучасних ІТС, включаючи хмарні обчислення, необхідно забезпечувати швидкодію в десятки і в сотні Гбітів/с. Розв’язати це протиріччя можна застосувавши потокові симетричні шифри. Підтвердження – ряд проектів NESSIE, eSTREAM тощо. За їх результатами прийнято важливий міжнародний стандарт ISO/IEC 18033-4, який є методологічною основою побудування перспективних ПСШ, а також містить рекомендовані до застосування ПСШ SNOW-2 та MUGI. За результатами виконання

міжнародного проекту eSTREAM розроблено та рекомендовані до застосування два класи ПСШ:

– Програмні шифри – HC 128, Rabbit, Salsa20-12, Sosemanuk.

– Апаратні шифри – F-FCSR-H v2, Grain v1, MISKEY v2, Trivium.

Отже, залежно від цільової спрямованості всі атаки на ПШ можна розділити на два великих класи:

1) Статистичні або атаки на розрізнення (distinguishing attack). Це атаки, що використовують методи розрізнення виходу генератора ключового потоку від випадкової послідовності такої ж довжини, основною метою яких є одержання інформації про наступну ключову послідовність і, потім, визначення відкритого тексту, або виявлення його статистичних властивостей за умови, що відомо алгоритм шифрування. Даний вид атак був успішно застосований до сучасних потокових шифрів Sober- t16 і Sober- t32, а також до шифрів Snow і Scream. Ефективність застосування атак на розрізнення залежить від статистичних властивостей ключової послідовності потокового шифру.

2) Відновлення ключа (key recovery attack). Це великий клас різних методів криптоаналіза, основне завдання яких відновити секретний ключ по відомому сегменту ключової послідовності. Для здійснення даного виду атак існує безліч різних способів, застосовних як для всіх спеціалізованих потокових шифрів, так і для певних схем потокового шифрування.

Різноманіття оригінальних ідей і розробок в області атак, спрямованих на відновлення ключа, сильно ускладнює виявлення загальних зв'язків або ознак між ними. Проте, далі пропонується класифікація найбільш загальних методів.

Силові атаки. До атак даного класу відносяться атаки, які не враховують внутрішню структуру аналізованого алгоритму, і їхня складність залежить тільки від довжини секретного ключа й довжини відомого сегмента ключової послідовності.

Повний перебір ключів (exhaustive key search). Це атака найбільш загального типу, що може застосовуватися до будь-якого потокового шифру. Знаючи

вихідну послідовність ГКП, генеровану за допомогою невідомого ключа, криптоаналітик просто перебирає всі можливі ключі й перевіряє, чи відповідає генеруємий ключовий потік даній послідовності. Складність такої атаки становить, не менш 2^l - 1 шифрувань за допомогою досліджуваної схеми, де l – довжина ключа. Теоретично, даний вид атак повинен бути найбільш ефективним у порівнянні з іншими методами криптоаналіза. Для забезпечення високої складності цієї атаки в шифрах використовують ключі великого розміру. Існує ефективний метод повного перебору ключів потокового шифру – метод компромісу час-пам'ять.

Компроміс час-пам'ять (time-memory tradeoff attacks). Метою даних атак є відновлення початкового стану регістра зсуву, що використовується в ГКП, по фрагменту ключової послідовності за умови, що криптоаналітику відомо схему пристрою. Зокрема, така атака була успішно застосована до криптоалгоритмів A5/1 і LILI- 128. У загальному випадку атака складається із двох етапів:

1) Підготовчий етап, у якому будується великий словник, що включає всі можливі пари <стан, вихід> (однакової розмірності).

2) Основний етап, у якому робиться припущення, що шифр перебуває в певному фіксованому стані, тобто, припущення про певне заповнення всіх комірок пам'яті. На основі цих даних генерується вихід. Далі проглядається перехоплена вихідна послідовність із метою знаходження відповідності зі генерованим потоком. Якщо відповідність відбулася, то фіксований стан з великою ймовірністю вважається початковим заповненням регістра, у противному випадку алгоритм продовжує працювати.

Складність атаки залежить від довжини перехопленого ключового потоку й розміру внутрішнього стану шифру. Під розміром внутрішнього стану шифру, позначимо його як m, мається на увазі довжина початкового заповнення ГКП. Так, наприклад, якщо генератор ключової послідовності шифру заснований на одному ЛРР, то розмір внутрішнього стану буде дорівнювати довжині цього регістра плюс змінні ініціалізації, якщо такі

використовуються. Отже, для протистояння атакам даного типу повинна виконуватися умова:

$$m \geq 2l, \tag{4.7}$$

де m – розмір внутрішнього стану генератора; l – довжина ключа.

Аналітичні. Атаки даного класу використовують специфічні особливості внутрішньої структури конкретного алгоритму (слабості перетворень, що використовуються у шифрі). Криптоалгоритм вважається уразливим до деякої аналітичної атаки, якщо складність її реалізації менше складності силових атак, навіть у тому випадку, коли практичне застосування цієї атаки неможливе на сучасному рівні розвитку обчислювальних засобів.

Математичні. Це атаки, що використовують для відновлення ключа точні математичні методи, складність реалізації яких менше складності силових атак.

Атаки періодичності (periodic attacks). Якщо період ГКП занадто малий, то ключова послідовність буде повторюватися, отже, можливий криптоаналіз використовуючи математичні методи. Період повинен бути достатньо великим, щоб гарантувати, що ключова послідовність не повториться.

Лінійна складність (linear complexity). Лінійна складність послідовності – довжина самого короткого ЛРР, що може відтворити цю послідовність. Лінійна складність легко обчислюється за допомогою алгоритму Берлекампа-Мессі. Відомо, що складність цього алгоритму – $O(n^2)$, де n– довжина відомого фрагмента ключової послідовності. Якщо лінійна складність занадто низька, то криптоаналітик зможе відтворити послідовність за допомогою ЛРР. Існують і інші критерії складності, як, наприклад, dyadic complexity, що оцінює схожі якості.

Для досягнення максимального періоду й високої линійної складності вихідних послідовностей ГКП, заснованих на одному або декількох ЛРР, необхідно, щоб всі регістри використовували примітивні поліноми зворотного зв'язку. Крім того, у випадку, якщо використовується декілька ЛРР, їхні

довжини повинні бути попарно взаємно простими, і у випадку, якщо використовується єдиний ЛРР, то його довжина m і алгебраїчний ступінь d використовуваної нелінійної функції f повинні бути досить великі, так щоб величина $\binom{m}{d}$ була значно більше, ніж передбачувана в додатках довжина ключової послідовності

Алгебраїчні атаки (algebraic attacks). Недавно запропонований алгебраїчний підхід до аналізу потокових шифрів у роботі Куртюа на даний момент є найбільш ефективним методом їх криптоаналіза. Основна ідея його полягає в тому, щоб підібрати систему рівнянь від декількох змінних, що описує залежність внутрішнього стану ГКП від ключової послідовності, і вирішити систему, у випадку, якщо рівняння мають невисокий ступінь.

У загальному випадку алгебраїчні атаки складаються із трьох основних етапів:

– знаходження системи алгебраїчних рівнянь, що зв'язують біти початкового стану й біти ключового потоку;

– підстановка послідовних біт ключової послідовності в отриману систему рівнянь;

– рішення системи.

Перший етап є підготовчим: криптоаналітик підбирає систему рівнянь, перш ніж буде перехоплена ключова послідовність. Другий і третій етапи виконуються при відомому ключовому потоці. При достатній кількості спостережень (і відповідних рівнянь) можна створити систему рівнянь невеликого алгебраїчного ступеня. Кількість одночленів у цій системі відносно мало (тому що ступінь кожного рівняння невисока). Одночлени розглядаються як незалежні змінні, і систему можна вирішити методом лінеаризації, наприклад за допомогою виключень методом Гаусса.

Складність алгебраїчних атак залежить експоненціально від ступеня рівнянь. З метою її зменшення були запропоновані швидкі алгебраїчні атаки. На сьогоднішній день при проектуванні схем потокового шифрування, що використовують у блоці нелінійної фільтрації булеві функції, не існує більше ефективного способу протистояння даному класу атак, чим використати функції, що мають високий алгебраїчний ступінь, однак це значно знижує

швидкість шифрування. Так, якщо вихідна послідовність ГКП залежить від D біт стану ЛРР, то алгебраїчний ступінь, d, нелінійної функції повинен задовольняти умові:

$$d \geq D/2. \tag{4.8}$$

Атаки на основі бінарних схем прийняття рішень (BDD-based attack). Алгоритм використає структури даних для мінімізації й обробки булевых функцій – бінарні діаграми схем прийняття рішень. Проблема знаходження секретного ключа k для заданого алгоритму функціонування ГКП, G, і відомого фрагмента ключової послідовності, z, зводиться до знаходження мінімальної бінарної діаграми, D, за умови, що виконується співвідношення z = G(k). Якщо довжина z приблизно дорівнює відстані одиничності G, то ключ k може бути ефективно обчислений по D. Складність даного методу експоненціально залежить від довжини початкового стану ЛРР, що лежить в основі шифру. Для протистояння таким атакам повинне виконуватися умова, відповідно до якої кожний біт ключової послідовності z є функцією нелінійного перетворення (або функцією стиску) f всіх біт початкового стану регістра, при цьому функція f повинна задовольняти базовим вимогам криптостікості.

Прогнозування. Це великий клас атак, що використовують елементи прогнозування, більш точні, ніж угадування.

Кореляційні атаки (correlation attacks). Найпоширенішими атаками загального типу на потокові шифри, що використовують в основі ЛРР, є кореляційні. Якщо нелінійна функція f пропускає на вихід інформацію про свої внутрішні компоненти, то робота з розкриття такої системи може бути істотно скорочена. Більше того, така інформація існує завжди, навіть якщо функція f задіє пам'ять. У силу даної аксіоми даний клас атак використовує кореляцію вихідної послідовності ГКП із вихідною послідовністю регістрів для відновлення початкового заповнення останніх.

Серед найбільш ефективних методів криптоаналіза, що відносяться до даного класу, можна виділити такі:

– базові кореляційні атаки, які також іменуються в літературі як «атаки декомпозиції» (divide-and-conquer attacks);

– швидкі кореляційні атаки, що базуються на низьковагових перевірках парності;

– атаки, що базуються на використанні конволюційних кодів;

– атаки,щовикористовують техніку турбо-кодів;

– атаки, що базуються на відновленні лінійних поліномів;

– швидка кореляційна атака Чепижова-Йоханссона-Смітса.

Ефективність застосування кореляційних атак залежить від вибору функції f блоку нелінійного перетворення й від параметрів ЛРР, використовуваних в основі генератора ключової послідовності.

Так, для оцінки ступеня уразливості генератора ключової послідовності до розглянутого методу криптоаналізу автором базових кореляційних атак Томасом Зигенталером була запропонована концепція, що одержала назву «кореляційно-імунні функції». Булеві функції називають кореляційно-імунними, якщо на виході не просочується ніяка інформація про їхні вхідні дані.

4.3 Анализ атак на Геш функції

Будь яка атака. Маючи згортку $H(m_1)$ повідомлення m_1, криптоаналітик повинен методом підбору знайти повідомлення m_2 ($m_2 \neq m_1$), для якого $H(m_1) = H(m_2)$ (тоді він може заявляти, що пред'явлена згортка відповідає повідомленню m_2, а не m_1). Якщо хеш-функція на виході дає n-бітовий рядок, складність цього методу $O(2^n)$.

Атака з урахуванням «парадоксу днів народжень». Для хеш-функцій універсальним методом пошуку колізій є метод, що ґрунтується на відомому статистичному завданні – «парадоксі дня народження».

Парадокс дня народження – це парадоксальне твердження, що ймовірність збігу днів народження (дати) хоча б у двох членів групи з 23 і більше осіб, перевищує 0,5. Для 60 і більше людей ймовірність такого збігу перевищує 0,99, хоча 1,0 вона досягає, тільки коли в групі не менше ніж 367 осіб. Таке твердження може здатися неочевидним, тому що ймовірність збігу днів народження двох осіб у будь-який день року (1/365 = 0,0027), помножена на кількість осіб у групі з 23, дає лише 23/365 = 0,063. Це міркування неправильно, оскільки кількість можливих пар (253) значно перевищує кількість осіб у групі. Логічного протиріччя цьому немає, а феномен полягає лише у відмінностях між інтуїтивним сприйняттям і математичним розрахунком.

Сценарій проведення атаки хеш-функції на основі «парадоксу дня народження». Нехай шахрай А хоче піддати атаці банкіра і підписати в нього контракт, невигідний для В.

– А готує дві версії контракту – «хорошу» M_1 та «погану» M_2 робить кілька малозначних змін до кожного документу, маніпулюючи з точками, комами, пробілами, прийменниками та ін.

– А вибирає для всіх версій договору хеши, порівнює набори хешей, підшукуючи однакові пари. Якщо довжина хеша 64 біта, то зазвичай вистачає 2^{32} пари. А вибирає пару з однаковим хешем.

– А вибирає пару з однаковим хешем і дає на підпис банкіру В погану версію M_2 . Тоді А в суді може доводити, що В підписав невигідний контракт свідомо.

При роботі з ключовими хеш-функціями небезпечно дописувати ключ на початок або в кінці вихідного повідомлення.

Нехай ключ k доданий на початку повідомлення, а функція, що стискає, побудована за схемою Меркеля - Дамгарда. Тоді за відомим повідомленням M та його згортці H = H (k || M) можна знайти значення згортки для всіх повідомлень виду M || M ′, де після M дописано будь-яке повідомлення M′. Справді, в силу ітераційного характеру хеш-функції для знаходження H′ =

H(k||M||M′) не потрібно знати ключ. Достатньо використовувати у обчислене "проміжне" значення H.

Нехай ключ k додано до кінця повідомлення H = H(M || k). Знання колізії для хеш-функції, тобто. пари M_1, M_2 , $M_1 \neq M_2$ але $H(M_1) = H(M_2)$, дозволяє обчислити хеші $H(M_1,k) = H(M_2,k)$ для будь-якого ключа k. ⇒ трудомісткість зміни повідомлення M1 вже оцінюється не величиною $O(2^n)$. Атакуючи хеш-функцію на основі «парадоксу днів народження» можна знизити трудомісткість до $O(2^{n/2})$.

Щоб уникнути цих небезпек, рекомендують ключ приписувати кілька разів: H=H(k||y||M||k) або $H=H(k||y_1||H(k||y_2||M))$, де y, y1, y2 – доповнення ключа до розміру, кратного довжині блоку. Так побудовані безключові функції стійкі до атак колізії, але їх недолік - велика довжина згортки.

Властивості, які повинні бути притаманні криптографічній хеш-функції:

1) Стійкість пошуку першого прообразу – відсутність ефективного поліноміального алгоритму обчислення зворотної функції, тобто. не можна відновити текст m за його згорткою H(m) за реальний час (необоротність). Ця властивість еквівалентно з того що хеш-функція є односторонньою функцією.

2) Стійкість до пошуку другого прообразу (колізій першого роду) – обчислювально неможливо, знаючи повідомлення m та його згортку H (m) , знайти інше повідомлення $m' \neq m$, щоб H(m) = H(m′).

3) Стійкість до колізій (колізій другого роду) – Колізією для хеш-функції називається така пара значень m і m′, $m \neq m'$, для якої H(m) = H(m′) . Оскільки кількість можливих відкритих текстів більше за кількість можливих значень згортки, то для деякої згортки знайдеться багато прообразів, це означає, що колізії для хеш-функцій обов'язково існують. Наприклад, нехай довжина хеш-прообразу 6 бітів, довжина згортки 4 біти. Тоді число різних згорток – 2^4 =16, а число хеш-прообразів – 2^6 =64, тобто. у 4 рази більше, а саме хоча б один згорток із усіх відповідає 4 прообразам. Стійкість хеш-функції до колізій означає, що немає ефективного поліноміального алгоритму, що дозволяє знаходити колізії.

Властивості не є незалежними:

– зворотна функція нестійка до відновлення другого прообразу та колізій;

– функція, нестійка до відновлення другого прообразу, нестійка до колізій; зворотне неправильне.

– функція стійка до колізій, стійка до знаходження другого прообразу;

– стійка до колізій хеш-функція не обов'язково одностороння.

Для криптографії важливо, щоб значення хеш-функції сильно змінювалися при найменшому зміні аргументу, тобто, їй має бути притаманний лавинний ефект.

Значення хеша не повинно давати витоку інформації навіть про окремі біти аргументу.

4.4 Людина посередині

Атака «людина посередині» (MITM) – це загальний термін, коли зловмисник позиціонує себе в розмові між користувачем і програмою – або підслуховувати, або видавати себе за одну зі сторін, створюючи вигляд, ніби звичайний обмін інформацією триває. Метою атаки є викрадення особистої інформації, такої як облікові дані для входу, дані облікового запису та номери кредитних карток або ж згенерованного випадкового числа з генератора. Цілями, якого в нашому випадку є компрометувати шифроване випадкове число яке буде використовуватися в подальших алгоритмах чи потребах других пристроїв чи алгоритмах. Інформація, отримана під час атаки, може бути використана для багатьох цілей, включаючи крадіжку особистих даних, несанкціоновані перекази коштів або незаконну зміну пароля. Загалом, атака MITM є еквівалентом того, що людина відкриває вашу банківську виписку, записує дані вашого рахунку, а потім знову запечатує конверт і доставляє його до ваших дверей.

Рисунок 3.1 Схема роботи MITM

Атаки MITM можуть бути різними якщо ми говоримо про QRNG тому що атака може буди проведена між генератором та екстрактором та компрометувати генероване випадкове число, якщо екстрактор в нашому генераторі е самостійним пристроєм. Також ця атака може проводитися по каналах зв’язку між QRNG та пристроєм до якого він підключається, наприклад комп’ютер де зловмисник може перехоплювати сигнал по проводах між материнською платою та пристроєм чи програмно перехоплювати та компрометувати вихідні данні з генератора.

4.5 Стійкість до атак

Як зазначалося вище, джерело випадковості є абсолютно непередбачуваним, тільки якщо воно має квантову природу. Класичний шум, своєю чергою, не лише принципово детермінованим, але, що найважливіше, може піддаватися зовнішньому впливу із боку противника. Таким чином, при строгому підході для отримання істинно-випадкової послідовності біт, слід використовувати лише квантову складову сигналу. Однак розділити класичні та квантові шуми неможливо, оскільки генерація електричних імпульсів, а також детектування оптичного сигналу здійснюється з використанням класичних пристроїв. Тому слід якимось чином оцінити співвідношення R_{QC} квантових та класичних шумів. Якщо це відношення досить високе, і при

цьому оцифрування сигналу не вносить нерівномірності до випадкової послідовності біт, то можна знехтувати вкладом класичних флуктуацій і використовувати необроблену випадкову послідовність. При цьому, якщо класичними флуктуаціями не можна знехтувати, слід враховувати можливий вплив противника, який потенційно має доступ до класичного шуму і, таким чином, може поставити під загрозу квантовий ГВЧ, ввівши кореляції в його вихідні дані. Співвідношення R_{QC} можна розглядати як міра таких кореляцій. Таким чином, можна припустити, що, позбавляючись цих кореляцій за допомогою різних екстракторів випадковості (наприклад, використовуючи екстрактор фон Неймана або криптографічну хеш-функцію), можна позбутися вкладу класичних флуктуацій.

Ставлення квантового шуму до класичного шуму було оцінено рядом авторів для різних квантових ГВЧ. У їхніх роботах передбачалося, що шумовий сигнал від фотоприймача містить класичний та квантовий вклади у мультиплікативній формі. При такому припущенні R_{QC} значення може бути визначено через відношення дисперсій класичного і квантового шумів, або обчислення різниці між ентропіями Шеннона квантового і класичного сигналів. На жаль, для заявленої конструкції квантового ГВЧ мультиплікативна модель шуму не підходить, і шуми повинні розглядатися адитивно. Як наслідок, оцінка R_{QC} у заявленому рішенні навряд чи є можливою; тому пропонується інший підхід, заснований на оцінці так званого ефективного коефіцієнта стиснення, пов'язаного з коефіцієнтом стиснення, який зазвичай використовується в процедурах вилучення випадковості.

Процедуру отримання випадковості можна розглядати як стиснення необробленої L-бітної послідовності (яка зазвичай є неоднорідною) до однорідної M-бітної послідовності:

$$\{0,1\}^{l} \xrightarrow{\text{RE}} \{0,1\}^{m}$$

де можна ввести (звичайний) коефіцієнт стиснення у як:

$$y = \frac{l}{m}.$$

Коефіцієнт у зазвичай оцінюється за мінімальною ентропією необробленої послідовності. Таким чином, з послідовності $\{X_1,X_2,\ldots X_N\}$ з $N>>1$, в якій кожне X_i є словом в n-біт, можна отримати NH_{min} рівномірно розподілених біт, тобто необроблена бітова послідовність довжини N * n має бути стиснута в $y=n/H_{min}$ раз. Min-ентропія, своєю чергою, визначається так:

$$H_{min} = -\log_2 p_{max},$$

де p_{max} - максимальна ймовірність вгадування елемента із послідовності {X1,X2,…XN}.

Коефіцієнт стиснення залежить від того, як оцифровується сигнал фотоприймача. При використанні АЦП слід враховувати, що сигнал буде з більшою ймовірністю потрапляти в осередки, що відповідають більшій ймовірності, тобто оцифровка сигналу з нерівномірним розподілом щільності автоматично призводить до випадкової неоднорідної послідовності. Навпаки, при оцифровці сигналу за допомогою компаратора слід просто підібрати таку граничну напругу, щоб число нулів та одиниць у вихідній послідовності було однаковим. Дійсно, беручи до уваги справжню випадковість фази $\Delta\varphi$, ми можемо вважати результуючу послідовність біт на виході з компаратора справді випадковою.

Слід зазначити, що з такої реалізації лише один біт витягується однією вибіркою, тобто частота генерації випадкових біт обмежена частотою

повторення лазерних імпульсів. Незважаючи на це обмеження, використання компаратора є дуже вигідним, тому що:

– він дозволяє уникнути дорогих АЦП;

– дозволяє витягувати випадкові біти без необхідності їх обробки і, отже, без зменшення втрат (за умови, що класичний шум малий).

Коефіцієнт стиснення у для схеми квантового ГВЧ із компаратором (n=1) можна визначити просто як $y=1/H_{min}$. Однак це визначення не дозволяє врахувати наявність класичного шуму сигналу фотоприймача. Щоб врахувати класичні флуктуації, було запроваджено ефективний коефіцієнт стиснення Г, використовуючи наступний метод. По-перше, передбачається, що гранична напруга компаратора V_{th} відповідає центру розподілу сигналу, тобто області під кривою $p_s(x)$ зліва та справа V_{th} рівні. Випливає, що: $V_{th} = S_{min} + w_{\Delta\varphi}/2$. Потім визначається квантова min-ентропія так $??' = ??''$:

$$H_{min}^{Q} = -\log_2\left(\int_{S_{min}}^{S_{min}+w_{\Delta\varphi}/2} \rho_S^{ideal}(x)dx\right) = 1.$$

Далі необхідно відзначити, що розподіл щільності сигналу S' стає ширшим зі збільшенням вкладу ψ класичного шуму. Внаслідок цього ймовірність того, що напруга з фотоприймача потрапляє в «бін» від S_{min} до $S_{min} + w_{\Delta\varphi}/2$, менше реального сигналу, ніж ідеального квантового сигналу. Вважатимемо, що й вклади від класичного і квантового шумів однакові, то ГВЧ перестає бути квантовим, а ефективний чинник стиску має дорівнювати нескінченності: $\Gamma \to \infty$. Навпаки, якщо класичні флуктуації незначно малі, можна використовувати необроблену послідовність, т. е. $\Gamma = 1$. Використовуючи це, можна визначити ефективний коефіцієнт стиснення наступним чином:

$$\Gamma = \frac{1}{2 - H_{min}};$$

де

$$H_{min} = -\log_2 \left(\int_{S_{min}}^{V_{th}} \rho_S(x) dx \right);$$

і де $\rho_S(x)$ - експериментальний розподіл щільності сигналу від фотоприймача. Очевидно, якщо немає класичного шуму, то $H_{min} = H_{min}^Q = 1$ і $\Gamma = 1$. Якщо, класична мінентропія дорівнює квантової мінентропії, $H_{min} = H_{min}^Q + H_{min}^C = 2H_{min}^Q = 2$ то тоді $\Gamma \to \infty$.

Цей підхід може бути легко поширений у разі використання АЦП. Для цього відзначимо спочатку, що розподіл щільності $\rho_S(x)$ має яскраво виражений абсолютний максимум при $x = S_{min}$ так, що p_{max} завжди буде відповідати ймовірності відповідного «біна». Тому ми можемо написати для ефективного коефіцієнта стиснення:

$$\Gamma = \frac{n}{2H_{min}^Q - H_{min}},$$

де n - дозвіл АЦП в бітах, і:

$$H_{min}^{Q} = -\log_2\left(\int_{S_{min}}^{S_{min}+\Delta u} \rho_S^{ideal}(x)dx\right),$$

$$H_{min} = -\log_2\left(\int_{S_{min}}^{S_{min}+\Delta u} \rho_S(x)dx\right),$$

з розміром осередку $\Delta u = \Delta U / 2^n$, де ΔU динамічний діапазон АЦП.

Оскільки значення ефективного коефіцієнта стиснення відбиває внесок від класичних флуктуацій, знання робить квантовий ГСЧ стійким до атаки, яка підробляє класичний шум. Іншими словами, опір атаці на сигнал, що генерується, зводиться до обчислення «на льоту» , що, у свою чергу, вимагає обчислення min-ентропії. Оскільки інтеграл містить експериментальний розподіл щільності основне призначення блоку КС полягає у визначенні функції.

ВИСНОВКИ

На сьогодні кіберпростір є невід'ємною частиною життя, що значною мірою впливає на економічний розвиток держави та підприемств. Впровадження інтернет-технологій у державний сектор та приватний сектор створює необхідність захисту інформації в спеціальних інформаційно - телекомунікаційних системах, а кібератаки, спрямовані на порушення роботи цих систем, становлять безпосередню загрозу економічній стабільності, конфіденційності та захищенності.

Стійкість криптографічних перетворень забезпечується при умові, що параметри генеруються рівномірно і випадково, а також в реальному часі і з високою швидкістю. Для цього використовуються генератори випадкових та псевдовипадкових послідовностей.

У цій роботі було досліджено атаки на екстрактори випадковості, що використовуються в квантових генераторах випадкових чисел.

Були обгрунтувані вимоги до генераторів випадкових чисел. Проведений аналіз та дослідження методів побудови квантових генераторів випадкових чисел. Також був зроблений аналіз методів побудови екстракторів випадковості та обгрунтувані вимоги, що висуваються до екстракторів випадковості, які доцільно викоритовувати в квантових генераторах випадкових чисел Були розглянуті атаи на екстрактори квантових генераторів випадкових чисел.

Проведений аналіз методів генерування випадкових та псевдовипадкових послідовностей, обгрунтувані вимоги до генераторів випадкових чисел, проведений аналіз та дослідження методів побудови квантових генераторів випадкових чисел, обгрунтувані вимоги, що висуваються до екстракторів випадковості, які доцільно викоритовувати в квантових генераторах випадкових чисел; проведений аналіз та дослідження атак на екстрактори квантових генераторів випадкових чисел.

На цей час розроблено ряд алгоритмів та засобів формування екстракторів. Їх особливістю є те, що вони будуються, як правило, для двійкової основи $m = 2$. Тому, важливою є задача розробки методів і засобів формування екстракторів із необхідними властивостями випадковості та довільною (певною) основою алфавіту. Найбільш перспективним, вважається, серед класів таких перетворень є клас багатомодульних перетворень.

У цілому метод побудови алгоритму екстрактора на основі багатомодульного перетворення може знайти застосування у криптографічних додатках, в яких висуваються умови високої рівноймовірності та довільної основи появи символів ПВП.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Гріненко Т.О. & Нарєжній О.П. (2015) Квантові генератори випадкових чисел в криптографії. Системи обробки інформації: збірник наукових праць. –Х.: ХУПС. – Випуск 10(135). – 288 с. С. 86-89.

2. Горбенко І.Д. & Гріненко Т.О. & Нарєжній О.П. (2016) Методика вимірювання спектральної щільності потужності шума квантової радіооптичної системи генератора випадкових чисел. Радиотехника: всеукр. межвед. науч.-техн. сб. Харьков: ХТУРЕ. – Вып. 186. – С. 172-183.

3. ДСТУ ISO/IEC 18033-3:2015 (ISO/IEC 18033-3:2010, IDT) “Інформаційні технології. Методи захисту. Алгоритми шифрування. Частина 3. Блокові шифри”.

4. A.A. Ruhkin. (2010). Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publication 800-22rev1a. Режим доступу : https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf

5. The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness [Электронный ресурс]. – URL http://stat.fsu.edu/pub/diehard/

6_1. ANSI NIST SP 800-56. Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography. 2005.

6_2. ANSI NIST SP 800-57. Recommendation for Key Management. 2006

6_3. ANSI NIST SP 800-90. Recommendation for Random Number Generation Using Deterministic Random Bit Generators. 2006

6. ДСТУ ISO/IEC 11770-1:2014. Інформаційні технології. Методи захисту. Управління ключами захисту. – Частина 1: Структура [на заміну ДСТУ ISO/IEC 11770-1:2009].

7. Горбенко І. Д. Прикладна криптологія: Монографія / Горбенко І. Д., Горбенко Ю.І. видання 2-ге. – Харків : Форт, 2012. – 868 с.

8. Горбенко І. Д. Прикладна криптологія: Підручник / Горбенко І. Д., Горбенко Ю. І.; видання 2-ге. – Харків : Форт, 2013. – 878 с.

9. Горбенко Ю.І. Інфраструктури відкритих ключів. Системи ЕЦП. Теорія та практика. / Горбенко Ю. І., Горбенко І. Д. – Харків: Форт, 2010. – 593 с.

10. ISO/IEC 18031:2011 Information technology - Security techniques - Random bit generation.

11, ДСТУ ISO/IEC 11770-2:2014. Інформаційні технології. Методи захисту. Управління ключами захисту. - Частина 2: Механізми, що використовують симетричні методи (на заміну ДСТУ ISO/IEC 11770-2:2002].

12. ДСТУ ISO/IEC 11770-3:2014. Інформаційні технології. Методи захисту. Управління ключами захисту. Частина 3: Механізми, що використовують асиметричні ме розроблення [на заміну ДСТУ ISO/IEC 11770-3:2002].

13. ДСТУ ISO/IEC 11770-4:2014. Інформаційні технології. Методи захисту. Управління ключами захисту. – Частина 4: Механізми, засновані на нестійких секретах.

15. ДСТУ ISO/IEC 11770-5:2014. Інформаційні технології. Методи забезпе чення безпеки. Керування ключами. – Частина 5: Група керування ключами.

15. Горбенко І. Д. Захист інформації в інформаційно-телекомунікаційних системах: Навч. посібник / Горбенко І. Д., Гриненко Т. О. – Ч. 1: Криптографічний захист інформації. – Харків:ХНУРЕ, 2004. – 368 с.

16. Гриненко Т. А. Метод формирования и свойства ПВП на эллиптических кривых /Гриненко Т.А., Горбенко Ю. И., Орлова С. Ю. //Радиотехника: Всеукр.межвед. науч.-техн. сб. – 2001. – Вып. 119. - С. 119-123.

17. Гріненко Т. О. Властивості та перспективи застосування генераторів псевдовипадкових послідовностей на еліптичних кривих /Гріненко Т. О., Горбенко Ю. І., Мордвінов Р. І. // Системи обробки інформації. – 2011. – Вип. 2(92). – С. 76-81.

18.Горбеко І.Д. Методи та засоби тестування послідовностей випадкових чисел та особливості їх застосування. Генерування псевдовипадкових послідовностей на основі багатомодульних перетворень в скінченних полях / Горбенко І. Д.,Гріненко Т. О. // Прикладная радиоэлектроника. – Харьков : ХНУРЭ, 2006. –Том 5. – №1. – С. 115-127.

19. Горбенко Ю. І. Методи та засоби генерування псевдовипадкових послідовностей / Горбенко Ю. І., Шапочка Н. В., Гріненко Т. О., Нейванов А. В.,Мордвінов Р. // Прикладна радіоелектроніка. - Харків : ХНУРЕ, 2010. –Т. 10. - №2.

20. Гріненко Т. О. Властивості детермінованих випадкових послідовностей, що генеруються на основі багатомодульних перетворень в полях Галуа / Гріненко Т. О., Горбенко Ю. І. // Збірник наукових праць Харківського університету повітряних сил. – Харків, 2011. – Вип. 1. – С. 136-139.

21. Горбенко Ю.І. Методи оцінки необоротності генератора псевдовипадкових послідовностей на основі багатомодульних перетворень в скінченних полях / Горбенко Ю.І. // Радиотехника : Всеукр. межвед. науч.-техн. сб. – Харків: ХНУРЕ, 2011. – Вып. 165. - С. 249-253,

22. ДСТУ ISO/IEC 9796-3:2014. Інформаційні технології. Методи забезпечення безпеки. Цифрові схеми підпису, що забезпечують відновлення повідом Частина 3: Основні механізми дискретного логарифма.

23 ДСТУ ISO/IEC 9798-5: 2014. Інформаційні технології.Методи захисту. Автентифікація об'єктів. – Частина 5: Механізми, що використовують методи нульової обізнаності.

24. Application Notes and Interpretation of the Scheme (AIS) 31. Functionality classes and evaluation methodology for physical random number generators. Certification body of the BSI in context of certification scheme. BSI, 2001, 38 pages.

25. Application Notes and Interpretation of the Scheme (AIS) 20. Functionality classes and evaluation methodology for Deterministic random number generators. 1999.

26. Горбенко Ю. И. Сущность и анализ криптографических требований стандарта NIST SP 800-90B / Ю. И. Горбенко, Р.И. Мордвинов // Радиотехника : Всеукр. межвед. науч.-техн. сб. - Харьков: ХНУРЭ, 2012. – Вып. 171. - С. 99–108.

27. Попович Е. В. Критерии оценки требований к генераторам случайных последовательностей в криптографических системах / Попович Е. В., Горбенко Ю. И. // Радіотехніка. – Харків: ХНУРЕ, 2005. – №141.

28. ISO/IEC 18033-3:2006 Information technology - Security techniques Encryption algorithms - Part 3: Block ciphers.

29. ДСТУ ISO/IEC 10116:2014. Інформаційні технології. Методи забезпечення безпеки. Режими роботи для N-розрядного блочного шифру. Вперше.

30. Daniel R. L. Brown. A Security Analysis of the NIST SP 800-90 Elliptic Curve Random Number Generator [Електронний ресурс] / Daniel R. L. Brown, Kristian Gjosteen f – 2007. – Режим доступу: http://eprint. iacr. org/.

31. Ju-Sung Kang. Security frameworks for pseudorandom number generators / Ju-Sung Kang // Information Center for Mathematical Sciences. - Vol. 8, Number 1. – 2005. – P. 1-11.

32. Дональд Кнут. Искусство программирования / Дональд Кнут // Получисленные алгоритмы. The Art of Computer Programming. – Vol. 2. Seminumerical Algorithms. – 3-е изд. – М. : ИД «Вильямс», 2007. – 832 с.

33. A.Rukhin, J.Soto. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publication 800-22, 09.2000.

34. Методичні вказівки до лабораторних робіт за дисципліною „Стандартизація та сертифікація в галузі інформаційної безпеки” / Упоряд. Т.О. Гріненко – Харків: ХНУРЕ, 2020.

35. Federal Information Processing Standard (FIPS) 140-1. Security Requirements for Cryptographic Modules 1994 pdf https://csrc.nist.gov/CSRC/media/Publications/fips/140/1/archive/1994-01-11/documents/fips1401.pdf

36. Federal Information Processing Standard (FIPS) 140-2. Security Requirements for Cryptographic Modules 2002 pdf https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf

37.Federal Information Processing Standard (FIPS) 140-3. Security Requirements for Cryptographic Modules 2019 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf

38. Darren Hurley-Smith; Constantinos Patsakis; Julio Hernandez-Castro. On the unbearable lightness of FIPS 140-2 randomness tests. IEEE Transactions on Information Forensics and Security 2020. P.13

39. The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness [Электронный ресурс]. – URL http://stat.fsu.edu/pub/diehard/

40. Mohammed M. Alani. Testing Randomness in Ciphertext of Block-Ciphers Using DieHard Tests International Journal of Computer Science and Network Security, VOL.10 No.4, April 2010

41. Application Notes and Interpretation of the Scheme (AIS) 31 – Functionality Classes and Evaluation Methodology for Physical Random Number Generators, Version 1 (25.09.2001), English translation.

42. Functionality classes and evaluation methodology for physical random number generators, reference: AIS31 version 1, 25/09/2001, BSI,

43. Functionality classes and evaluation methodology for deterministic random number generators, reference: AIS 20, version 1, 02/12/1999, BSI.

44. A proposal for : Functionality classes and evaluation methodology for true (physical) random number generators, version 3.1, 25.09.2001, W. Killmann (T-system), W. Schindler (BSI),

45. Горбенко, Ю.І. Побудування та аналіз систем, протоколів і засобів криптографічного захисту інформації : монографія. – Ч. 1: Методи побудування та аналізу, стандартизація та застосування криптографічних систем ; за заг. ред. І.Д. Горбенко. – Харків : Форт, 2016. – 960 с.

46. A Fast and Compact Quantum Random Number Generator/ Thomas Jennewien, Ulrich Achleitner, Gregor Weihs, Harald Weinfurter and Anton

Zeilinger – 4/III D-80799 Munchen, Germany February 1, 2008. – pp. 1–21. – [Електронний ресурс] – Режим доступу до матеріалів: https://arxiv.org/pdf/quant-ph/9912118.pdf.

47. Achleitner U. Diploma Thesis, Innsbruck University (1997).

48. Martino, A. J. , Morris, G. M. Applied Optics 30, 981 (1991).

49. Morris, G. M. Opt. Engin. 24, 86 (1985); J. Marron, A. J. Martino, G. M. Morris, Applied Optics 25, 26 (1986).

50. W. M. Itano, J. C. Bergquist, R. G. Hulet, and D. J. Wineland // Phys. Rev. Lett. 59, 2732 (1987).

51.Th. Sauter, W. Neuhauser, R. Blatt, and P. E. Toschek // Phys. Rev. Lett. 57, 1696 (1986).

52. Тетерич, Н.М. Генераторы шума и измерение шумових характеристик. – М. : Энергия, 1968. – 216 с.

53. Стандарты частоты: принципы и приложения / Ф. Риле ; пер. с англ. Н. Н. Колачевского. – М. : Физматлит, 2009. – 511 с.

54. Квантовая радиофизика. Квантовые стандарты частоты с оптической накачкой : учеб. пособие / В.В. Семенов, Г.М. Смирнова, В.М. Хуторщиков ; С.-Петерб. гос. техн. ун-т СПб. : Изд-во СПбГТУ, 1999. – 536 с.

55. Нарежний, А. П. Идентификация скрытых периодичностей в нестационарных фазовых флуктуациях прецизионных мер частоты / А. П. Нарежний // Прикладная радиоэлектроника. – 2005. – Т.4. – № 2. – С. 148–152.

56. NIST SP 800-90B. Recommendation for the Entropy Sources Used for Random Bit Generation. Meltem Sönmez Turan, Elaine Barker, John Kelsey, Kerry A. McKay, Mary L. Baish, Mike Boyle. https://doi.org/10.6028/NIST.SP.800-90B

57. Hoeffding W. Probability inequalities for sums of bounded random variables. J. Amer. Statist. Assoc. 1963. Vol. 58. No 301. P. 13 – 30.

58. Zhang B., Xu C., Meier W. Fast correlation attacks over extension fields, large-unit linear approximation and cryptanalysis of SNOW 2.0. Cryptology ePrint Archive, Report 2016/311. URL: http://eprint.iacr.org/2016/311 (дата звернення: 27.01.2020).