Житомирська політехніка	Міністерство освіти і науки України Державний університет «Житомирська політехніка»

ЛЕКЦІЯ №6

ШИФРУВАННЯ ІНФОРМАЦІЇ ІЗ ВИКОРИСТАННЯМ КРИПТОГРАФІЧНОГО ШИФРУ ГАМУВАННЯ

ПИТАННЯ:

1. ІСТОРИЧНА ДОВІДКА.
2. ОСНОВИ ШИФРУВАННЯ МЕТОДОМ ГАМУВАННЯ.
3. ГЕНЕРАЦІЯ ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ.
4. ВІДМІННІСТЬ КЛЮЧА ВІД ГАМИ.
5. ДЕШИФРУВАННЯ МЕТОДОМ ГАМУВАННЯ.
6. СТАНДАРТИ ТА СПЕЦИФІКАЦІЇ З ГЕНЕРАЦІЇ ТА ТЕСТУВАННЯ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

ПИТАННЯ №1

ІСТОРИЧНА ДОВІДКА

ВИНИКНЕННЯ ІДЕЇ ГАМУВАННЯ

Гамування виникло дуже давно і має свої коріння ще в античні часи. Одним з ранніх прикладів гамування є "шифр Цезаря", який використовувався римським імператором Юлієм Цезарем для захисту конфіденційних повідомлень. Цей метод полягав у зсуві букв абетки на фіксовану кількість позицій, і цей зсув (ключ) використовувався для шифрування та розшифрування тексту.

ГАМУВАННЯ В СЕРЕДНЬОВІЧЧІ

Середньовіччя було часом інтенсивного розвитку гамування. Гамування використовувалося для шифрування дипломатичних послань, військових комунікацій та інших важливих повідомлень. Наприклад, у середньовіччі використовували гамування з використанням ключів-буквалів, де кожна літера тексту замінювалася на певний ключ.

ВІДНОВЛЕННЯ ІНТЕРЕСУ У 20-МУ СТОЛІТТІ

Після Першої світової війни зросла необхідність у надійних методах шифрування для важливих комунікацій. Гільберт Вернам, американський інженер, розробив метод "гамування одноразовим ключем" (гама великої довжини, яка використовується тільки один раз), який вважається абсолютно надійним, оскільки без ключа неможливо розкрити повідомлення.

ГОЛОДНА ВІЙНА ТА ІНФОРМАЦІЙНА ВІЙНА

Після Другої світової війни гамування стало важливим засобом військової та політичної комунікації під час Холодної війни між США і Радянським Союзом. Обидві сторони використовували складні гамові системи для захисту від перехоплення ворожих повідомлень.

СУЧАСНІСТЬ І СТАНДАРТИ ГАМУВАННЯ

У сучасному світі гамування залишається важливим методом шифрування. Воно застосовується в багатьох сферах, включаючи захист електронної пошти, безпеку фінансових операцій та забезпечення конфіденційності даних у мобільних додатках. Сучасні гамові алгоритми, такі як AES (Advanced Encryption Standard), стали стандартами безпеки і використовуються у багатьох додатках та системах.

КВАНТОВА КРИПТОГРАФІЯ І МАЙБУТНЄ ГАМУВАННЯ

Розвиток обчислювальних технологій і загроза квантових комп'ютерів ставлять під сумнів стійкість багатьох сучасних методів гамування. Тому ведуться активні дослідження в області квантової криптографії, яка може забезпечити надійний захист інформації у майбутньому, використовуючи квантові властивості частинок.

Історія гамування свідчить про його незаперечну важливість у забезпеченні конфіденційності та безпеки комунікацій протягом віків. Гамування залишається ключовим елементом в сфері криптографії та інформаційної безпеки і продовжує еволюціонувати, адаптуючись до сучасних викликів і технологічних зрушень.

ПИТАННЯ №2

ОСНОВИ ШИФРУВАННЯ МЕТОДОМ ГАМУВАННЯ

ОСНОВНІ ПОНЯТТЯ

Гамування є широко застосовуваним криптографічним перетворенням. Під ***гамуванням*** розуміють процес накладення по визначеному законі гами шифру на відкриті дані. ***Гама шифру*** – це псевдо випадкова послідовність, вироблена по заданому алгоритмі для шифровки відкритих даних і дешифрування зашифрованих даних. **Процес шифрування полягає в** генерації гами шифру за допомогою датчика псевдо випадкових чисел і накладенні отриманої гами на вихідний відкритий текст, наприклад з використанням операції додавання по модулю. Нагадаємо, що результатом додавання двох цілих чисел по модулю є залишок від ділення (наприклад, $5+10 \bmod 4 = 15 \bmod 4 = 3$).

Шифри гамування (адитивні шифри) є найефективнішими з погляду стійкості та швидкості перетворень (процедур зашифрування і дешифрування). У літературі шифри цього класу часто називають потоковими, хоча до потокових належать й інші різновиди шифрів. У шифрах гамування може використовуватися додавання по модулю N (загальний випадок) і по модулю 2 (окремий випадок, орієнтований на програмно-апаратну реалізацію).

ДОДАВАННЯ ПО МОДУЛЮ N

У 1888 р. француз маркіз де Віарі в одній зі своїх наукових статей, присвячених криптографії, довів, що в разі заміни літер відкритого/зашифрованого повідомлення і ключа на числа справедливі формули:

$$Ci = (Pi + Ki) \bmod N, \quad (6.1)$$

$$Pi = (Ci + N - Ki) \bmod N, \quad (6.2)$$

де Pi , Ci – і-ий символ відкритого/зашифрованого повідомлення;

N – кількість символів в алфавіті;

Ki – і-ий символ гами (ключа).

Ці формули дають змогу виконати зашифрування/дешифрування за Віженером у разі заміни літер алфавіту числами згідно з такою таблицею кодування символів (стосовно українського алфавіту):

А	Б	В	Г	Ґ	Д	Е	Є	Ж	З	И	І	Ї	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ю	Я
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

ДОДАВАННЯ ПО МОДУЛЮ N

Наприклад, для зашифрування використовується український алфавіт (N = 33), відкритий текст – "АБРАМОВ", гама – "ЖУРІХІН". Під час заміни символів на числа буква А буде представлена як 0, Б – 1, ..., Я – 32.

Приклад адитивного зашифрування по модулю N = 33

Символ								
	відкритого тексту, Рі	А	Б	Р	А	М	О	В
		0	1	20	0	16	18	2
	Гами, Кі	Ж	У	Р	І	Х	І	Н
		8	23	20	11	25	11	17
	Шифрограми, Сі	Ж	Ф	Є	І	Ж	Щ	П
		8	24	7	11	8	29	19

ШИФРОБЛОКНОТИ

Шифри гамування стали використовуватися німцями у своїх дипломатичних представництвах на початку 1920-х рр., англійцями та американцями – під час Другої світової війни. Розвідники-нелегали низки держав використовували шифрблокноти. Шифр Вернама (додавання по модулю 2) застосовували на урядовій "гарячій лінії" між Вашингтоном і Москвою, де ключові матеріали були перфорованими паперовими стрічками, які виготовляли у двох примірниках.

ДОДАВАННЯ ПО МОДУЛЮ 2 (ШИФР ВЕРНАМА)

Значний успіх у криптографії пов'язаний з ім'ям американця **Гільберто Вернама**. У 1917 р. він, будучи співробітником телеграфної компанії AT&T, спільно з Мейджором Джозефом Моборном запропонував ідею автоматичного шифрування телеграфних повідомлень. Йшлося про своєрідне накладення гами на знаки алфавіту, представлені відповідно до телетайпного коду Бодо п'ятизначними "імпульсними комбінаціями". Наприклад, літеру А уявляли комбінацією ("— + + +"), а комбінація ("+ + - + + +") означала перехід від букв до цифр. На паперовій стрічці, використовуваній під час роботи телетайпа, знаку "+" відповідала наявність отвору, а знаку "-" – його відсутність. Під час зчитування зі стрічки металеві щупи проходили через отвори, замикали електричний ланцюг і, тим самим, посилали в лінію імпульс струму.

ДОДАВАННЯ ПО МОДУЛЮ 2 (ШИФР ВЕРНАМА)

Шифрувальна машина Siemens M-190

ДОДАВАННЯ ПО МОДУЛЮ 2 (ШИФР ВЕРНАМА)

Вернам запропонував електромеханічно покоординатно складати "імпульси" знаків відкритого тексту з "імпульсами" гами, попередньо нанесеними на стрічку. Додавання проводилося "по модулю 2" (⊕, для булевих величин аналог цієї операції – XOR, "Виключне АБО"). Мається на увазі, що якщо "+" ототожнити з 1, а "-" з 0, то додавання визначається двійковою арифметикою:

⊕	**0**	**1**
0	0	1
1	1	0

ДОДАВАННЯ ПО МОДУЛЮ 2 (ШИФР ВЕРНАМА)

Таким чином, за цього способу шифрування символи тексту і гами подаються в двійковому вигляді, а потім кожна пара двійкових розрядів складається по модулю 2. Процедури зашифрування і дешифрування виконуються за такими формулами:

$$Ci = Pi \oplus Ki \text{ , (6.3)}$$

$$Pi = Ci \oplus Ki \text{ . (6.4)}$$

ДОДАВАННЯ ПО МОДУЛЮ 2 (ШИФР ВЕРНАМА)

Вернам сконструював і пристрій для такого складання. Чудово те, що процес шифрування виявився повністю автоматизованим. Крім того, виявилися злитими воєдино процеси зашифрування/дешифрування і передавання каналом зв'язку.

У 1918 р. два комплекти відповідної апаратури було виготовлено і випробувано. Випробування дали позитивні результати. Єдине незадоволення фахівців-криптографів було пов'язане з гамою. Річ у тім, що спочатку гама була нанесена на стрічку, склеєну в кільце. Незважаючи на те, що знаки гами на стрічці обирали випадково, під час зашифрування довгих повідомлень гама регулярно повторювалася. Цей недолік так само чітко усвідомлювався, як і для шифру Віженера. Уже тоді добре розуміли, що повторне використання гами неприпустиме навіть у межах одного повідомлення. Спроби подовжити гаму призводили до незручностей у роботі з великим кільцем. Тоді було запропоновано варіант із двома стрічками, одна з яких шифрувала іншу, унаслідок чого виходила гама, що має довжину періоду, яка дорівнює добутку довжин вихідних періодів.

ДОДАВАННЯ ПО МОДУЛЮ 2 (ШИФР ВЕРНАМА)

Перед ілюстрацією використання шифру наведемо таблицю кодів символів Windows 1251 та їхнє двійкове представлення.

Буква	Dec-код	Bin-код	Буква	Dec-код	Bin-код	Буква	Dec-код	Bin-код
А	192	1100 0000	І	178	1011 0010	Т	210	1101 0010
Б	193	1100 0001	Ї	175	1010 1111	У	211	1101 0011
В	194	1100 0010	Й	201	1100 1001	Ф	212	1101 0100
Г	195	1100 0011	К	202	1100 1010	Х	213	1101 0101
Ґ	165	1010 0101	Л	203	1100 1011	Ц	214	1101 0110
Д	196	1100 0100	М	204	1100 1100	Ч	215	1101 0111
Е	197	1100 0101	Н	205	1100 1101	Ш	216	1101 1000
Є	170	1010 1010	О	206	1100 1110	Щ	217	1101 1001
Ж	198	1100 0110	П	207	1100 1111	Ь	220	1101 1100
З	199	1100 0111	Р	208	1101 0000	Ю	222	1101 1110
И	200	1100 1000	С	209	1101 0001	Я	223	1101 1111

ДОДАВАННЯ ПО МОДУЛЮ 2 (ШИФР ВЕРНАМА)

Приклад зашифрування відкритого тексту "ВОЛЯ" за допомогою ключа «ЕРА" показано в таблиці. Оскільки довжина ключа менша за довжину відкритого тексту, то для генерації гами він циклічно повторюється.

Відкритий текст, Pi	**Буква**	**В**	**О**	**Л**	**Я**
	Dec-код	194	206	203	223
	Bin-код	1100 0010	1100 1110	1100 1011	1101 1111
Гама, Ki	**Буква**	**Е**	**Р**	**А**	**Е**
	Dec-код	197	208	192	197
	Bin-код	1100 0101	1101 0000	1100 0000	1100 0101
Шифрограма, Ci	**Dec-код**	7	30	11	26
	Bin-код	0000 0111	0001 1110	0000 1011	0001 1010

ДОДАВАННЯ ПО МОДУЛЮ 2 (ШИФР ВЕРНАМА)

Шифрування по модулю 2 має чудову властивість – замість істинної гами супротивникові можна повідомити хибну гаму, яка під час накладення на шифрограму дасть осмислений вираз. Приклад, представлений у таблиці, ілюструє досконалість шифрів гамування. Для перехопленої шифрограми противник може підібрати велику кількість гам, що дають при дешифруванні осмислені тексти і не мають нічого спільного з істинним відкритим текстом. Ба більше, під будь-який хибний відкритий текст знайдеться хибна гама.

Шифрограма, Сі	**Dec-код**	7	30	11	26
	Bin-код	0000 0111	0001 1110	0000 1011	0001 1010
Помилкова гама, К'і	**Буква**	**Н**	**Н**	**А**	**Е**
	Dec-код	205	205	192	197
	Bin-код	1100 1101	1100 1101	1100 0000	1100 0101
Помилковий відкритий текст, Р'і	**Dec-код**	202	211	203	223
	Bin-код	11001010	11010011	1100 1011	1101 1111
	Буква	**К**	**У**	**Л**	**Я**

СТІЙКІСТЬ ШИФРУ

Стійкість адитивних шифрів визначається, головним чином, якістю гами, що залежить від довжини періоду і випадковості розподілу за періодом.

Довжиною періоду гами називається мінімальна кількість символів, після якої послідовність починає повторюватися. ***Випадковість розподілу символів за періодом*** означає відсутність закономірностей між появою різних символів у межах періоду.

Для забезпечення абсолютної стійкості необхідно, щоб послідовність символів у межах періоду гами мала такі властивості:

- була випадковою (має бути відсутня закономірність у появі символів гами);
- розподіл символів алфавіту гами має бути близьким до рівномірного (рівноймовірного);
- збігалася за розміром або була більшою за шифрований відкритий текст;
- застосовувалася тільки один раз.

СТІЙКІСТЬ ШИФРУ

Перші дві властивості (вимоги) перевіряються за допомогою різних тестів. Найпопулярніший і найавторитетніший із них "Набір статистичних тестів для генераторів випадкових і псевдовипадкових чисел для криптографічних застосувань" (англ. NIST SP 800-22 "A statistical test suite for random and pseudorandom number generators for cryptographic applications"), що містить у собі 15 тестів.

Класичний одноразовий шифрувальний блокнот – великий неповторюваний випадковий набір символів ключа, написаний на аркушах паперу, склеєних у блокнот. Шифрувальник при особистій зустрічі забезпечувався блокнотом, кожна сторінка якого містила ключ. Такий самий блокнот був і у приймаючої сторони. Використані сторінки після одноразового використання знищували.

ПИТАННЯ №3

ГЕНЕРАЦІЯ ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

ГЕНЕРАЦІЯ ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

Незважаючи на всі переваги шифрів гамування, однією з ключових проблем їх застосування на практиці є отримання якісних гам. Для процедур зашифрування/дешифрування можна використовувати істинно випадкові або псевдовипадкові гами (послідовності). ***Псевдовипадкова послідовність*** – послідовність чисел, яку було обчислено за певною процедурою, але яка має всі властивості випадкової послідовності чисел у межах розв'язуваної задачі. Відмінність істинно випадкових послідовностей від псевдовипадкових полягає в неможливості передбачення (розрахунку, визначення) символів у ній. Таким чином, будь-який алгоритмічно влаштований програмно-апаратний комплекс не може генерувати істинно випадкові послідовності, тому що він працює за чітко визначеними правилами, а отже результат (гама) передбачуваний. Як сказав Джон фон Нейман, "усякий, хто має слабкість до арифметичних методів отримання випадкових чисел, грішний поза всякими сумнівами".

ГЕНЕРАЦІЯ ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

Істинно випадкові гами можуть бути отримані шляхом оцифрування випадкових фізичних або антропогенних процесів.

Псевдовипадкові гами отримують шляхом застосування рекурентних формул або повноцінних алгоритмів. При цьому відсутність істинної випадковості не заважає отримувати криптографічно стійкі послідовності, зокрема і з нескінченним періодом.

Рекурентна формула – формула вигляду $a_n = f(n, a_{n-1}, a_{n-2}, ..., a_{n-p})$, що визначає кожний член послідовності a_n, як функцію від p попередніх членів і можливо номера члена послідовності n.

ГЕНЕРАТОРИ ІСТИННО ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ. ВИПАДКОВІ ФІЗИЧНІ ПРОЦЕСИ

Генерація істинно випадкових послідовностей можлива шляхом оцифрування випадкових фізичних або антропогенних процесів.

Серед фізичних процесів, які розглядалися як джерело випадкових послідовностей, можна виділити такі:

- ***дробовий шум*** – безладні зміни (флуктуації) сигналу, що приймається або передається, відносно його середнього значення, спричинені дискретністю потоків фотонів і електронів в оптичних і радіоелектронних пристроях;
- ***тепловий шум*** – рівноважний шум, зумовлений тепловим рухом носіїв заряду в провіднику, унаслідок чого на кінцях провідника виникає флуктуйована різниця потенціалів. У мікропроцесорах Intel з архітектурою Ivy Bridge для апаратної генерації випадкових послідовностей використовується тепловий шум у кристалах кремнію. Для посилення ентропії потік бітів проходить додаткову фільтрацію та шифрування AES у режимах CBC і CRT;
- ***рух рідин у лавових лампах***. Компанія CloudFlare, через мережу якої проходить до 10% трафіку Інтернету, як одне з джерел випадкових послідовностей використовує стіну із сотнею лавових ламп;
- ***рух стрічок у потоці повітря вентилятора***;

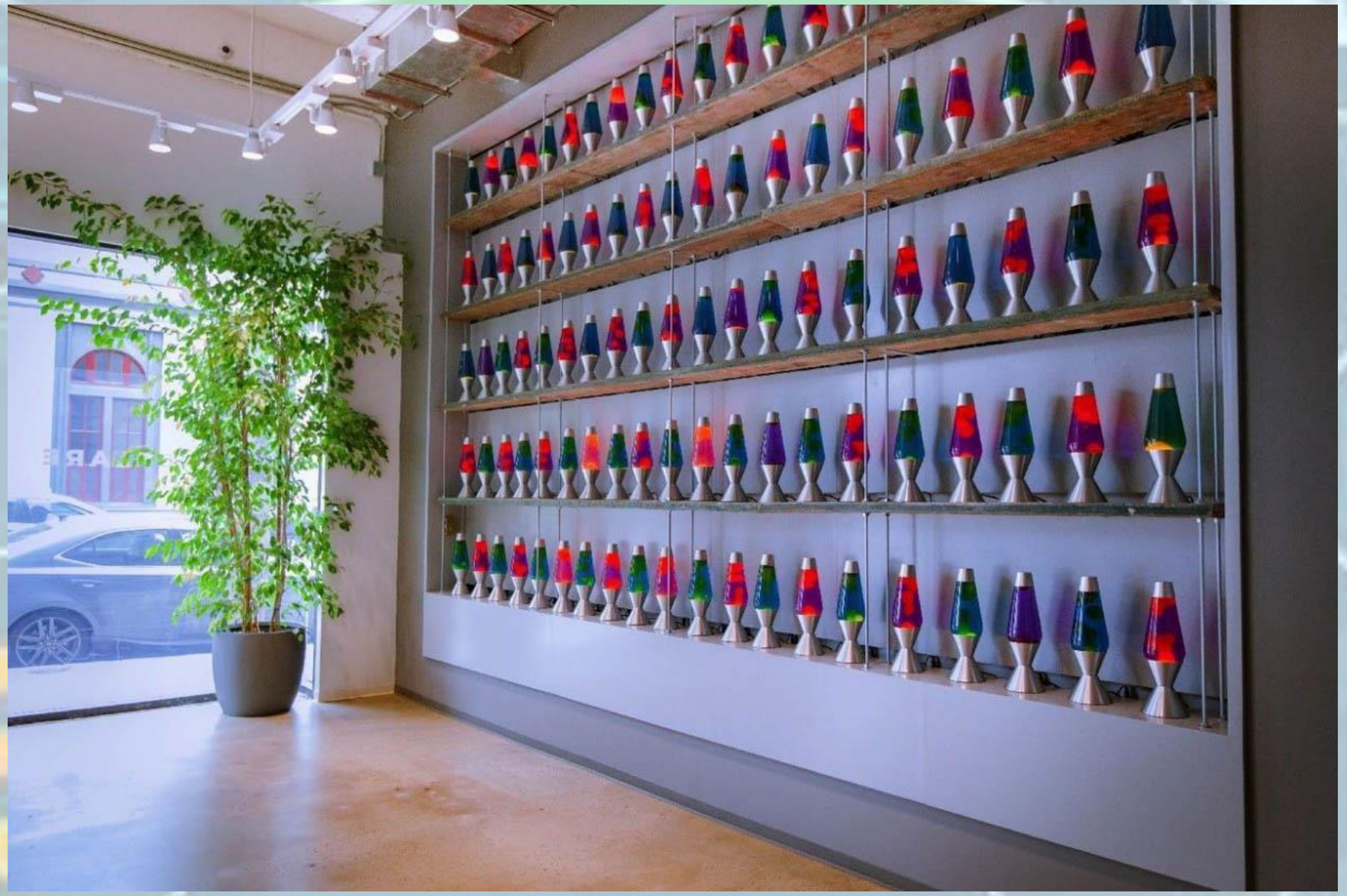
ГЕНЕРАТОРИ ІСТИННО ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ. ВИПАДКОВІ ФІЗИЧНІ ПРОЦЕСИ

Серед фізичних процесів, які розглядалися як джерело випадкових послідовностей, можна виділити такі(продовження):

- ***радіоактивний розпад*** – спонтанна зміна складу або внутрішньої будови нестабільних атомних ядер шляхом випускання елементарних частинок, гама-квантів та/або ядерних фрагментів. Компанія CloudFlare як одне з джерел випадкових послідовностей використовує радіоактивний елемент;
- ***електромагнітне випромінювання*** – електромагнітні хвилі, що виникають під час збурення магнітного або електричного поля. Генератори випадкових послідовностей на базі квантових явищ пропонує велика кількість компаній (ID Quantique, QuintessenceLabs, ComScire та ін.);
- ***лавинний пробій у напівпровідниках*** – електричний пробій p-n- переходу, спричинений лавинним розмноженням носіїв заряду під дією сильного електричного поля;
- ***нестабільна частота осцилятора***, що вільно працює. Компанія RAND використала це явище для публікації в 1955 р. книжки "Мільйон випадкових цифр зі стандартним відхиленням 100 000", а компанія AT&T у 1986 р. представила комерційну мікросхему-генератор.

МІКРОПРОЦЕСОР CORE I7-3770 З АРХІТЕКТУРОЮ IVY BRIDGE ТА ІНСТРУКЦІЄЮ ГЕНЕРАЦІЇ ВИПАДКОВИХ ЧИСЕЛ RDRAND

"СТІНА ЕНТРОПІЇ" В КОМПАНІЇ CLOUDFLARE

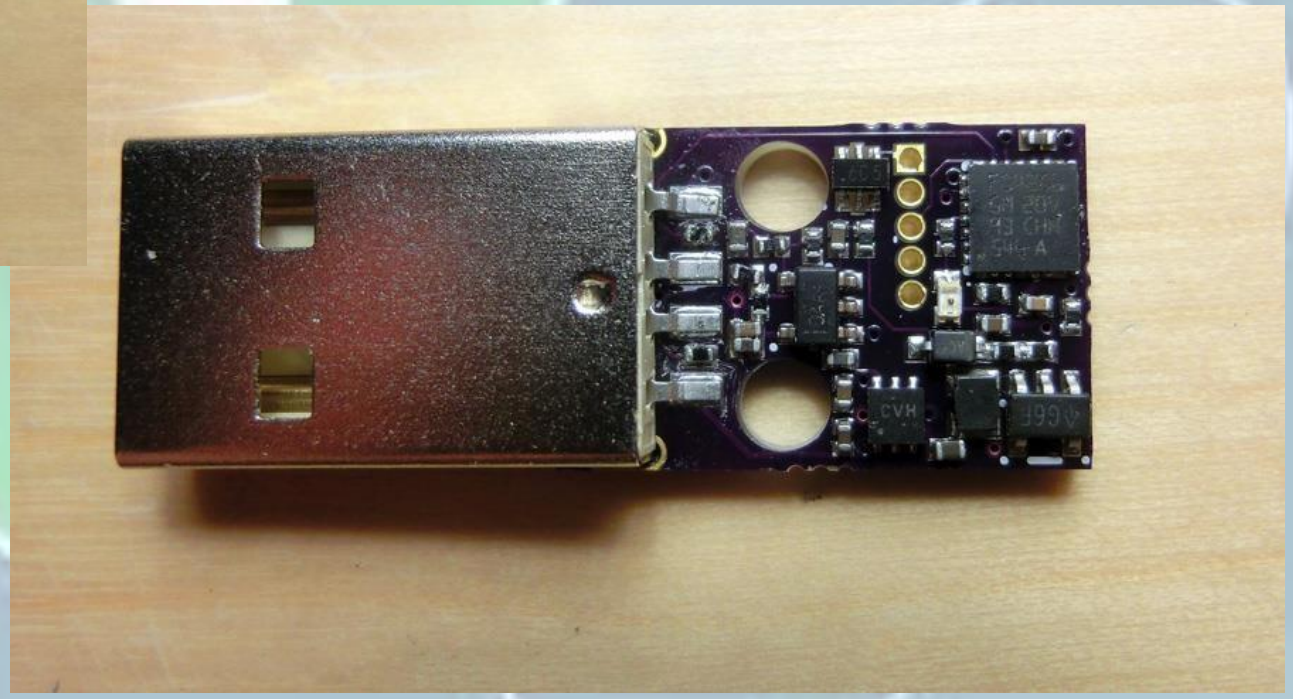


ХАОТИЧНИЙ РУХ СТРІЧОК У ПОТОЦІ ПОВІТРЯ

ЛІЧИЛЬНИК ГЕЙГЕРА

КВАНТОВИЙ ГЕНЕРАТОР ВИПАДКОВИХ ЧИСЕЛ

ПРИСТРІЙ CHAOSKEY



ГЕНЕРАТОРИ ІСТИННО ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ. ВИПАДКОВІ АНТРОПОГЕННІ ПРОЦЕСИ

До антропогенних процесів, які використовують як джерело випадкових послідовностей, можна віднести такі:

- час між натисканнями клавіш;
- руху комп'ютерної миші;
- зміна швидкості обертання жорсткого диска, викликаного турбулентністю повітря;
- час між прийнятими пакетами в мережі (наприклад, пінг сайту Google).

НЕДОЛІКИ ГЕНЕРАТОРІВ ІСТИННО ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

Для більшості розглянутих вище генераторів на базі фізичних та антропогенних процесів характерні певні недоліки:

- необхідність передання гам усім учасникам обміну даними. Оскільки гама генерується в одному місці і вона випадкова, то сторона, що приймає дані, для дешифрування попередньо повинна отримати цю гаму (проблема обміну ключами);
- повільна швидкість генерації числових послідовностей;
- антропогенні процеси можуть мати приховані залежності – кожен користувач має свій "почерк" роботи з комп'ютером.

ГЕНЕРАТОР ІСТИННО ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ ЗА МЕТОДОМ ВЧЕНИХ КОРНЕЛЬСЬКОГО УНІВЕРСИТЕТУ

На закінчення огляду генераторів істинно випадкових послідовностей розглядається оригінальний метод, запропонований у 2013 р. вченими Корнельського університету. Коротко принцип нового методу шифрування полягає в тому, що відправник і одержувач (Аліса й Боб) під час зустрічі формують спільний ключ, опромінюючи шматочки скла зображенням-патерном з IDi (зовні він нагадує QR-код). У результаті відбиття і заломлення, характер якого індивідуальний для кожного шматка скла, в Аліси і Боба виходять власні випадкові зображення, які потім оцифровують (отримують ключі $keyA_i$ і $keyB_i$). Із цих зображень і складається загальний ключ ($keyAB_i = keyAi \oplus keyB_i$).

Як і в класичній системі Вернама, кожен випадковий патерн (ключі $keyA_i$, $keyB_i$ і $keyAB_i$) можна використовувати лише один раз. У зв'язку з цим Аліса і Боб повинні сформувати достатню кількість ключів, опромінюючи свої шматки скла різними патернами. Після генерації загальні ключі $keyAB_i$ поміщаються в загальнодоступне сховище.

СХЕМА ГЕНЕРАЦІЇ ОДНОГО СПІЛЬНОГО КЛЮЧА

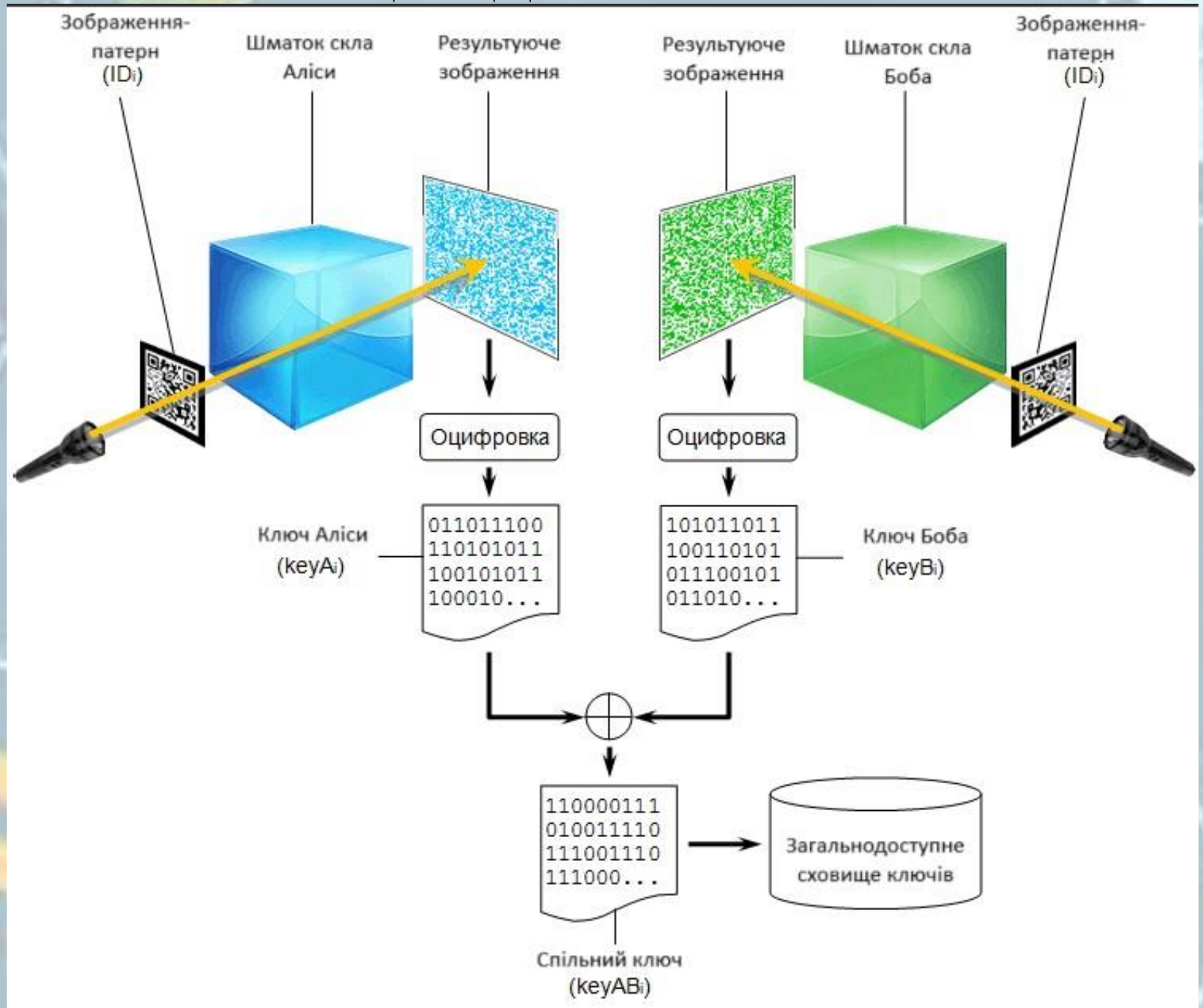
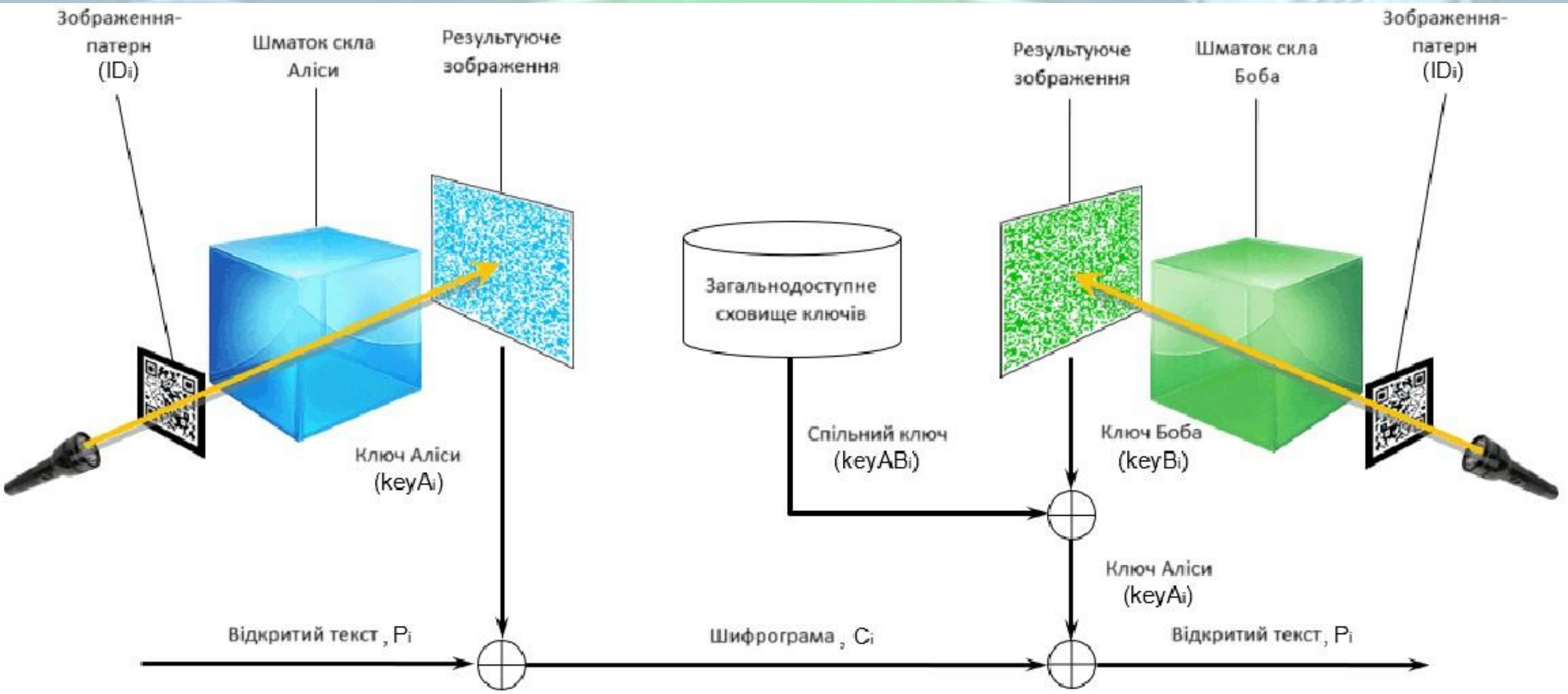


СХЕМА ЗАШИФРУВАННЯ ТА ДЕШИФРУВАННЯ ПОВІДОМЛЕННЯ



ПРОЦЕДУРА ОБМІНУ ШИФРОГРАМАМИ

Процедура обміну шифрограмами має такий вигляд:

1)Аліса вибирає патерн з ID_i, опромінює свій шматочок скла й оцифровує отримане зображення, отримуючи ключ $keyA_i$.

2)Відкритий текст P_i Аліса складає по модулю 2 із ключем $keyA_i$ ($C_i = P_i \oplus keyA_i$) і надсилає шифрограму C_i з ідентифікатором патерну ID_i Бобу.

3)Боб за отриманим ідентифікатором ID_i із загальнодоступного сховища зчитує загальний ключ $keyAB_i$ і, опромінюючи патерн з ID_i і свій шматочок скла, генерує власний ключ $keyB_i$.

4)Складаючи по модулю 2 ключі $keyAB_i$ і $keyB_i$, Боб отримує ключ $keyA_i$ ($keyA_i = keyAB_i \oplus keyB_i$).

5) Для прочитання відкритого тексту P_i Боб складає по модулю 2 шифрограму C_i і ключ keyAi ($P_i = C_i \oplus keyA_i$).

КРИПТОСТІЙКІСТЬ ШИФРУ ВЕРНАМА

За твердженням авторів схеми, зламати шифр Вернама можна, тільки вкравши самі шматочки напівпрозорого скла. Інші можливі проблеми при використанні цієї схеми:

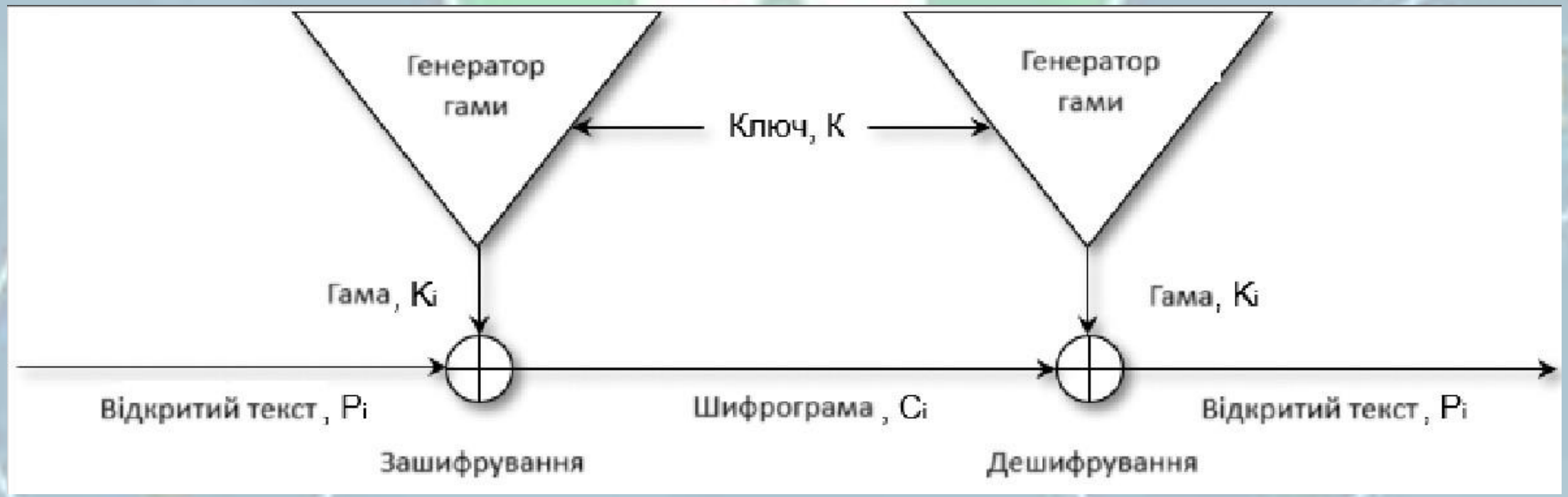
- втрата шматочка скла;
- поява сколів і подряпин на склі;
- необхідність точного взаємного розташування та орієнтації джерела світла, зображення-паттерна, шматочка скла і полотна з результуючим зображенням під час оцифрування гами.

Інформаційна ентропія – міра невизначеності або непередбачуваності інформації, невизначеність появи будь-якого символу первинного алфавіту.

ГЕНЕРАЦІЯ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

Генератори псевдовипадкових послідовностей набули найбільшого поширення. За схемою використання вони поділяються на **синхронні** і **такі, що самосинхронізуються**.

Схема шифрування з використанням синхронних генераторів



БУДОВА СИНХРОННОГО ГЕНЕРАТОРА

При цьому генератор гами, як правило, складається з трьох основних блоків. **Внутрішній стан** описує поточний стан генератора гами. Початковий внутрішній стан, як правило, визначається ключем K. Два генератори, з однаковим ключем і однаковим внутрішнім станом, створюють однакові гами. **Функція переходів** зчитує поточний внутрішній стан і генерує новий внутрішній стан. **Вихідна функція** зчитує внутрішній стан і генерує біт (біти) гами K_i. Будова генератора гами з внутрішнім зворотним зв'язком:

ГЕНЕРАТОР ТИПУ ЛІЧИЛЬНИК. ПЕРЕВАГИ ТА НЕДОЛІКИ СИНХРОННИХ ГЕНЕРАТОРІВ

В іншому різновиді, так званих **генераторах типу лічильник**, відсутній блок із функцією переходів. На відміну від генераторів зі зворотним зв'язком, вони дають змогу обчислити і-ий біт гами, не обчислюючи всіх попередніх бітів. Для цього генератор встановлюється в і-ий внутрішній стан, після чого обчислюється і-ий біт гами, що йому відповідає. Цю властивість корисно використовувати для забезпечення довільного доступу до файлів даних, що дає змогу дешифрувати окремий фрагмент даних, не дешифруючи файл повністю.

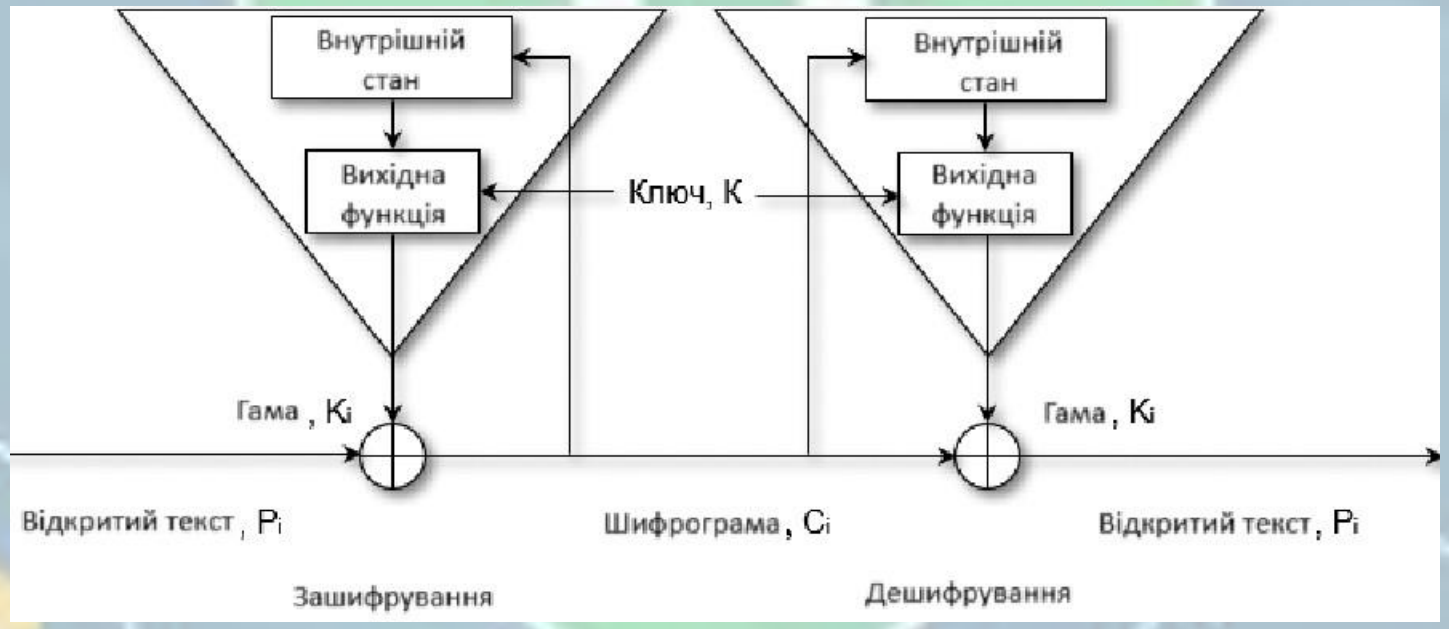
У синхронному генераторі гама генерується незалежно від потоку повідомлення. На стороні шифрування генератор гами послідовно видає біти гами Ki. На стороні, що дешифрує, інший генератор гами один за одним видає ідентичні біти гами. Ця схема працює нормально, якщо обидва генератори синхронізовані.

Недолік синхронного генератора. Якщо один із генераторів пропускає один із циклів або біт шифрограми губиться під час передачі, то всі символи шифрограми, що йдуть за помилкою, дешифруються некоректно. У цьому випадку відправник і одержувач повинні синхронізувати генератори і заново передати некоректно дешифровану частину повідомлення.

Перевага синхронного генератора. Відсутність поширення помилок. Якщо під час передачі біт Ci змінить своє значення, що набагато ймовірніше за його втрату, то некоректно дешифрується тільки один змінений біт.

ГЕНЕРАТОР, ЩО САМОСИНХРОНІЗУЄТЬСЯ

У генераторі, що самосинхронізується, кожен біт гами являє собою функцію фіксованого числа попередніх бітів шифрограми. Використовувані при такому шифруванні генератори гами називають ***генераторами зі зворотним зв'язком за шифрограмою (шифртекстом)***. Схема шифрування з використанням генераторів гами, що самосинхронізуються:



Внутрішній стан залежить від **n** попередніх бітів шифрограми. Кожне повідомлення починається випадковим заголовком (**вектор ініціалізації, синхропосилка**) довжиною **n** біт, після проходження якого обидва генератори гами синхронізуються.

НЕДОЛІКИ ГЕНЕРАТОРА, ЩО САМОСИНХРОНІЗУЄТЬСЯ

1. Поширення помилки. Для кожного біта шифрограми, спотвореного під час передачі, дешифрувальний генератор видає n некоректних бітів гами. Отже, змінений біт впливає на внутрішній стан – кожній помилці шифрограми відповідатиме n помилок відкритого тексту.
2. У разі втрати біта C_i необхідно заново передати частину повідомлення, але на відміну від синхронних генераторів, синхронізація набагато простіша.

Для генерації псевдовипадкових послідовностей використовують рекурентні формули або повноцінні алгоритми. У першому випадку члени числової послідовності не тільки розраховуються рекурентно, а й згодом стають частиною гами. У другому випадку для генерації гами використовують складніші правила, у т.ч. для підвищення ентропії гами можуть застосовувати хеш-функції та шифрування.

РЕКУРЕНТНІ ФОРМУЛИ ГЕНЕРАЦІЇ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

1.Лінійний конгруентний генератор – генератор псевдовипадкових чисел, у якому новий член послідовності розраховується на базі попереднього через лінійну залежність(Наприклад алгоритм RANDU).

2.Інверсний конгруентний генератор (генератор Ейхенауера-Лена) – генератор псевдовипадкових чисел, у якому новий член послідовності розраховується як зворотне число до попереднього по модулю.

3.Адитивний генератор (генератор Фібоначчі із запізненням) – генератор псевдовипадкових чисел, у якому новий член послідовності залежить більш ніж від одного з попередніх. Одну з ранніх реалізацій генератора послідовності Фібоначчі із запізненням було запропоновано 1958 р. Дж. Ж. Мітчелом і Д. Ф. Муром.

4.Регістр зсуву з лінійним зворотним зв'язком (РЗЗЛЗЗ) – упорядкований набір бітів, у якого значення вхідного (зсувного зліва, старшого) біта дорівнює лінійній булевій функції від значень інших бітів регістра до зсуву. Теорію послідовності регістрів зсуву розробив у 1965 р. головний криптограф норвезького уряду Ернст Селмер.

АЛГОРИТМИ ГЕНЕРАЦІЇ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

1.Алгоритм Блюм-Блюм-Шуба (англ. Algorithm Blum - Blum - Shub, BBS) запропонований у 1986 р. Ленор Блюм, Мануелем Блюмом і Майклом Шубом.

2.RC4 (від англ. Rivest cipher або Ron's code) був створений співробітником компанії "RSA Security" Рональдом Рівестом 1987 року. Протягом семи років шифр був комерційною таємницею, і точний опис алгоритму надавали тільки після підписання угоди про нерозголошення, але у вересні 1994 р. його опис анонімно відправили у список розсилки "Cypherpunks«.

3.Trivium був представлений у 2008 р. як частина європейського проекту eSTREAM за профілем 2 (апаратно орієнтовані шифри) і зараз має статус міжнародного стандарту "ISO/IEC 29192-3:2012. Інформаційні технології - Методи безпеки - Легка криптографія - Частина 3: Потокові шифри" (англ. "ISO/IEC 29192-3:2012. Інформаційні технології - Методи безпеки - Легка криптографія - Частина 3: Потокові шифри"). Авторами генератора (шифру) є Крістоф Де Канньєр і Барт Пренел.

4.Генератор на базі ірраціональних чисел. *Ірраціональне число* – дійсне число, яке не є раціональним, тобто не може бути подане у вигляді звичайного дробу.

ВИКОРИСТАННЯ АЛГОРИТМУ RC4

RC4 набув широкого поширення в криптосистемах і протоколах, зокрема:

- WEP (англ. Wired Equivalent Privacy) – алгоритм для забезпечення безпеки мереж Wi-Fi;
- WPA (англ. Wi-Fi Protected Access) – оновлений алгоритм сертифікації пристроїв мереж Wi-Fi;
- BitTorrent protocol encryption – протоколи піринговых файлообмінних мереж;
- SSL (англ. Secure Sockets Layer) – криптографічний протокол передавання даних у мережі;
- Kerberos – сервер аутентифікації Kerberos;
- PDF (англ. Portable Document Format) – міжплатформний формат електронних документів, розроблений фірмою Adobe Systems;
- Skype - програмне забезпечення IP-телефонії;
- та ін.

ПРОБЛЕМИ ВИКОРИСТАННЯ ІРРАЦІОНАЛЬНИХ ЧИСЕЛ

Ірраціональне число може бути представлено у вигляді нескінченного неперіодичного десяткового дробу. Іншими словами, у дробовій частині такого числа нескінченна кількість цифр і в їхньому записі відсутній період. Це говорить про те, що будь-яка кінцева випадкова числова послідовність рано чи пізно обов'язково буде частиною ірраціонального числа. Ці обставини можуть зробити ірраціональні числа цінним джерелом псевдовипадкових послідовностей. Для генерації гами можна буде вказати номер позиції в дробовій частині, з якої слід почати вибирати послідовність цифр для гами. Очевидно, що цей номер має бути дуже великим, щоб потенційний противник не зміг за прийнятний час його визначити методом перебору.

До проблем використання ірраціональних чисел як псевдовипадкових числових послідовностей належать:

- відсутність доказу нормальності чисел;
- відносно складний алгоритм розрахунку, що передбачає, як правило, використання рекурентних процедур;
- наявність у своєму складі "поганих" послідовностей (наприклад, послідовностей необхідної для гами довжини, але таких, що складаються з одних нулів, або таких, що мають період, який легко визначається).

ПИТАННЯ №4

ВІДМІННІСТЬ КЛЮЧА ВІД ГАМИ

ВІДМІННІСТЬ КЛЮЧА ВІД ГАМИ

Дуже часто поняття "ключ" і "гама" ототожнюють, але між ними є принципові відмінності. Нагадаємо, ключ – мінімально необхідна інформація (за винятком повідомлення, алфавітів і алгоритму), що використовується для зашифрування і дешифрування повідомлень. Гамою є вся числова послідовність, яка використовується для зашифрування або дешифрування повідомлення, з довжиною, не меншою за довжину цього повідомлення.

ВІДМІННІСТЬ КЛЮЧА ВІД ГАМИ

Метод генерації	Ключ	Гама
На базі випадкових фізичних або антропогенних процесів	Ключ відповідає гамі	
На базі слова або фрази	Слово або фраза	Слово або фраза, що циклічно повторюється
Лінійний або інверсний конгруентний генератор	Початкове число і коефіцієнти формул	Числова послідовність, що генерується
Регістр зсуву з лінійним зворотним зв'язком	Вид поліному і вихідний стан регістра	Числова послідовність, що генерується
RC4	Ключ К	Числова послідовність із частин S-блоку
На базі числа π	Номер початкової позиції в дробовій частині, з якої вибираються цифри числа π	Послідовність цифр числа π, починаючи із заданої позиції

ПИТАННЯ №5

ДЕШИФРУВАННЯ МЕТОДОМ ГАМУВАННЯ

ДЕШИФРУВАННЯ ЗА МЕТОДОМ ГАМУВАННЯ

У даного методу шифрування є два вразливих місця. По-перше, якщо датчик випадкових чисел, використовуваний для створення гами побудований неграмотно і послідовно генеровані ним числа мають пряму чи непряму залежність один від одного, то це практично гарантує криптоаналітику успіх у зламі. Злам робиться таким чином:

1. Припускається, що в початковому тексті є будь-яка характерна фраза або окреме слово (чим довша фраза, тим краще). Необхідно сказати, що переважна більшість документів в даний час зберігаються у вигляді файлів стандартних форматів, а формат файлу практично завжди визначається його текстовим заголовком або чим-небудь в тому ж роді.
2. У тексті шифровки шляхом послідовного підбору шукається рядок символів такої ж довжини, закон формування якої відповідає відомій особливості датчика ПВП. Нагадаємо, що текст шифровки виходить складанням по модулю вихідного тексту і псевдовипадкової гами. Перший символ рядка вважається породжуючим, а наступні символи ітераційно обчислюються з нього.

ДЕШИФРУВАННЯ ЗА МЕТОДОМ ГАМУВАННЯ

3. Якщо шуканий рядок виявлено, то переходимо до наступного пункту, а якщо ні, то пробуємо знайти в тексті іншу характерну фразу.
4. Коли рядок символів виявлено, то далі потрібно тільки добудувати псевдовипадкову послідовність вперед і назад, тим самим, відкривши вихідний текст.

Другий спосіб зламу шифрування гамуванням припускає наявність можливості з боку криптоаналітика або його спільника особисто посилати тестові повідомлення по криптографічному каналу, а потім, перехопивши їх, аналізувати вироблені шифроалгоритмом зміни. У такій ситуації, шляхом вирішення детермінованою математичною системи рівнянь, стає відомим алгоритм генерації навіть найкращого генератора ПВП. Далі, обгрунтовано вважаючи, що алгоритм генерації не буде змінюватися і в найближчому майбутньому, коли за криптографічним каналом буде передаватися не тестове повідомлення, а корисна інформація, дії по-злому будуть повністю аналогічні діям з попереднього методу.

ПИТАННЯ №6

СТАНДАРТИ ТА СПЕЦИФІКАЦІЇ З ГЕНЕРАЦІЇ ТА ТЕСТУВАННЯ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

СТАНДАРТИ І СПЕЦИФІКАЦІЇ США

Розвинена система стандартів і специфікацій щодо генерації та тестування псевдовипадкових послідовностей існує у США. **ANSI** (Американський національний інститут стандартів - англ. American National Standards Institute) і **NIST** (Національний інститут стандартів і технологій, США - англ. National Institute of Standards and Technology) розробили такі документи:

стандарти ANSI серії X9.82:

-ANSI X9.82 "Random Number Generation. Part 1: Overview and Basic Principles" (укр. "Генерація випадкових чисел. Частина 1: Огляд та основні принципи");

-ANSI X9.82 "Financial Services - Random Number Generation. Part 2: Entropy Sources" (укр. "Фінансові послуги - Генерація випадкових чисел. Частина 2: Джерела ентропії");

-ANSI X9.82 "Random Number Generation. Part 3: Deterministic Random Bit Generators" (укр. "Генерація випадкових чисел. Частина 3: Детерміновані генератори випадкових бітів");

- ANSI X9.82 "Random Number Generation. Part 4: Random Bit Generator Constructions" (укр. "Генерація випадкових чисел. Частина 4: Конструкції генератора випадкових бітів");

спеціальні публікації NIST серії 800:

-NIST SP 800-22 Rev.1a "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications" (укр. "Набір статистичних тестів для генераторів випадкових і псевдовипадкових чисел для криптографічних додатків");

-NIST SP 800-90A "Recommendation for Random Number Generation Using Deterministic Random Bit Generators" (укр. "Рекомендації для генерації випадкових чисел з використанням детермінованих генераторів випадкових бітів").

ДЯКУЮ ЗА УВАГУ!