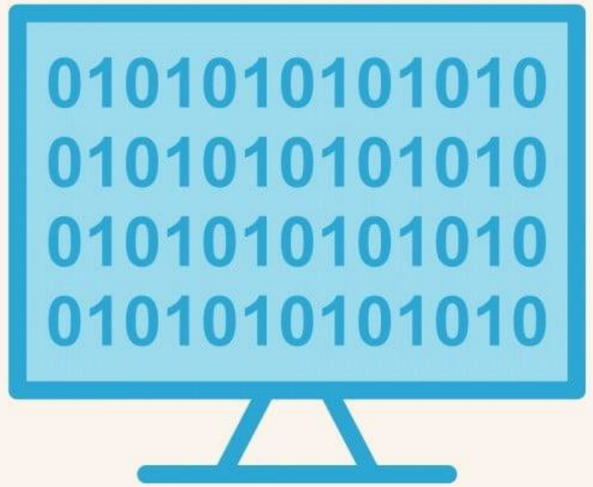
ЛЕКЦІЯ 8

ЖИТОМИРСЬКА ПОЛІТЕХНІКА
ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Потокові симетричні шифри

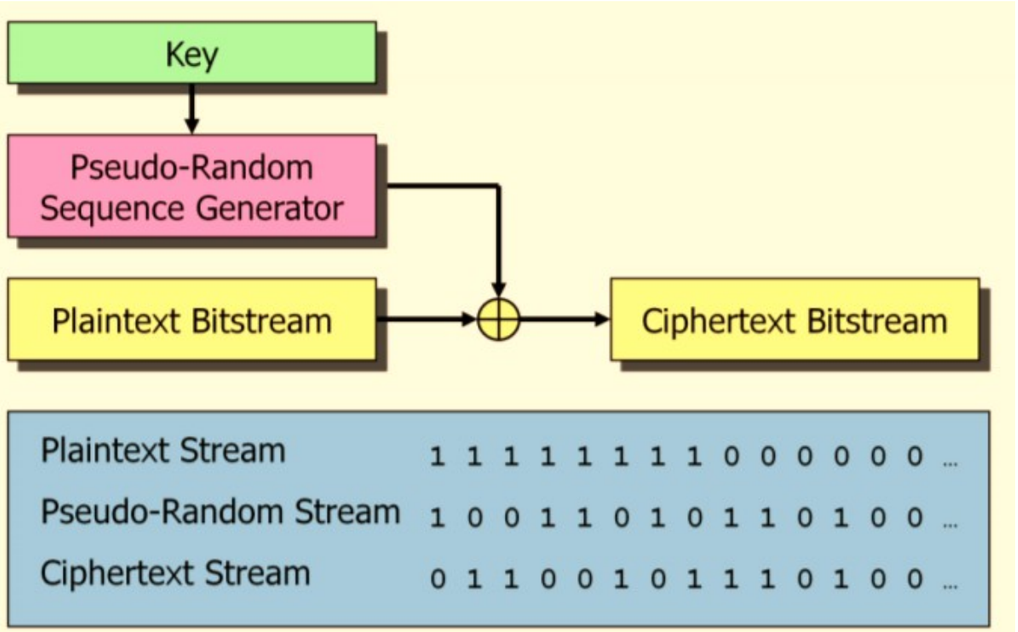


План

1. Загальні відомості про потокові шифри
2. Генератори псевдовипадкових чисел
3. Потоковий шифр RC4
4. Інші потокові шифри

1. Загальні відомості про потокові шифри

Потоковий шифр – шифр, що перетворює кожен символ (літеру, біт або байт) відкритого тексту у символ шифротексту, залежно від ключа та розташування символів у тексті.



Ключовий потік (гама) – це бітова послідовність, що визначається за допомогою ключа шифру.

1. Загальні відомості про потокові шифри

Генератор ключів видає потік бітів K_i, які будуть використовуватися як гама. Джерело повідомлень генерує біти відкритих даних P_i, які додаються за модулем *2* з гамою, внаслідок чого виходять біти зашифрованих даних C_i:

$$C_i = P_i \oplus K_i$$

$$P_i = C_i \oplus K_i$$

1. Загальні відомості про потокові шифри

Критерій	Блокові шифри	Потокові шифри
Принцип шифрування	Шифрують дані блоками фіксованої довжини (наприклад 64 або 128 біт)	Шифрують дані побітно або побайтово
Розмір даних для шифрування	Працюють з блоками фіксованої довжини. Якщо дані менші за блок, необхідне доповнення (padding)	Працюють з потоками будь-якої довжини, padding не потрібен
Швидкість шифрування	Зазвичай повільніші через необхідність обробки блоків	Зазвичай швидші, особливо при шифруванні потокових даних
Реалізація	Часто складніші, оскільки вимагають обробки блоків	Простішою є реалізація, але потрібне налаштування генератора ключів
Атаки	Можуть бути вразливі до атак, пов'язаних з режимами шифрування (наприклад, атаки на CBC)	Вразливі до атак на генератори псевдовипадкових чисел (наприклад, атаки на RC4)
Приклади алгоритмів	AES, DES, 3DES, Blowfish	RC4, STRUMOK, Salsa20, ChaCha20
Використання	Підходять для шифрування великих обсягів даних (файлів, дисків)	Підходять для потокових даних у реальному часі (мережеві протоколи, VPN, GSM)
Ресурси для реалізації	Вимагають більше ресурсів для обробки блоків і доповнення	Можуть бути ефективнішими у системах з обмеженими ресурсами
Криптостійкість	Зазвичай мають високий рівень стійкості, особливо сучасні алгоритми, такі як AES	Залежить від якості генератора ключів; наприклад, RC4 був вразливий до низки атак
Гнучкість використання	Можуть використовувати різні режими шифрування (CBC, ECB, CTR тощо) для різних потреб	Працюють за простішою схемою, зазвичай не потребують різних режимів шифрування

1. Загальні відомості про потокові шифри

Синхронні потокові шифри

Синхронні потокові шифри – це шифри, у яких потік ключів генерується незалежно від відкритого й зашифрованого повідомлення.

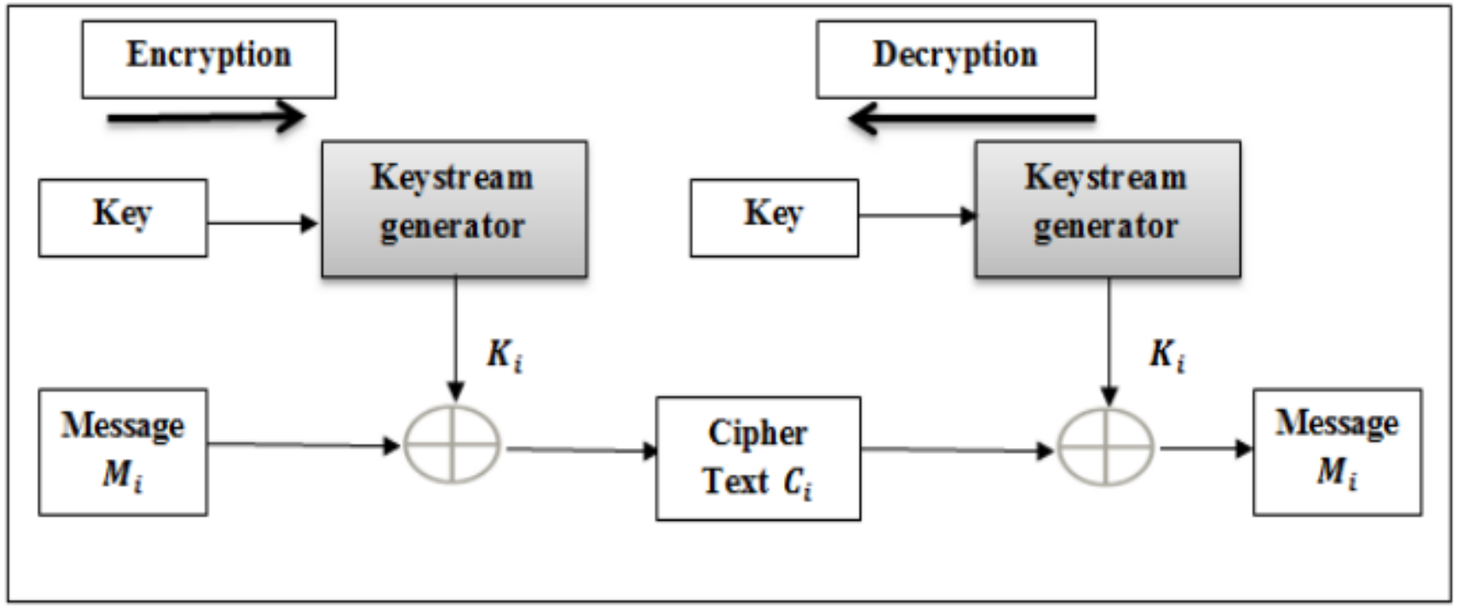
+ Відсутність ефекту поширення помилок (якщо тільки один біт спотворений, то тільки він буде розшифрований неправильно).

+ Захист від вставок і вилучень фрагментів зашифрованого повідомлення, оскільки вони призведуть до втрати синхронізації й будуть виявлені.

– Можлива підміна окремих бітів зашифрованого повідомлення.

1. Загальні відомості про потокові шифри

Синхронні потокові шифри



Синхронізація проводиться вставкою в передане повідомлення спеціальних маркерів. Внаслідок цього пропущений під час передачі знак призводить до неправильного розшифрування лише до того часу, поки не буде прийнятий один з маркерів. Часто для відновлення синхронізації з ціллю отримання правильного відкритого тексту автоматично пробуються різні зсуви.

1. Загальні відомості про потокові шифри

Потокові шифри, що самосинхронізуються

Потокові шифри, що самосинхронізуються (асинхронні потокові шифри) – це шифри, в яких потік ключів створюється функцією ключа й фіксованою кількістю знаків зашифрованого повідомлення (використовується *n* попередніх знаків шифротексту для обчислення потоку ключа).

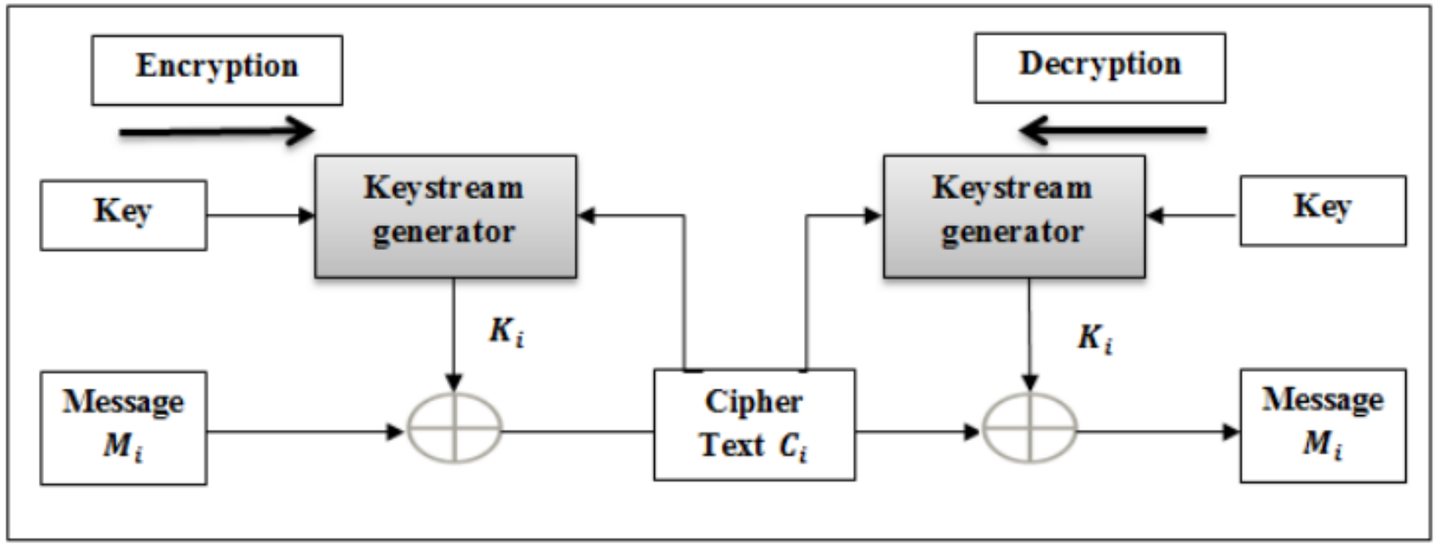
\+ Дешифрувальний генератор, прийнявши *n* бітів, автоматично синхронізується із шифрувальним генератором.

\+ Розсіювання статистики відкритого тексту.

– Чутливий до розкриття повторною передачею.

1. Загальні відомості про потокові шифри

Потокові шифри, що самосинхронізуються



Синхронізація генераторів відбувається за рахунок використання *попередніх шифрованих символів* для відновлення потоку ключів. Якщо відправлене повідомлення втрачає кілька байтів або пошкоджується, отримувач може втратити синхронізацію на кілька символів. Але завдяки тому, що кожен наступний символ використовує шифротекст попереднього, система поступово повертається до правильного стану.

2. Генератори псевдовипадкових чисел

Стійкість потокових шифрів цілком залежить від якості й криптографічної стійкості генератора псевдовипадкових чисел, за допомогою якого отримується потік ключа.

Генератор псевдовипадкових чисел (ГПВЧ) – це пристрій або алгоритм, який за заданими параметрами генерує послідовність *псевдовипадкових чисел* (ПВЧ).

2. Генератори псевдовипадкових чисел

Послідовність випадкових чисел – послідовність, сформована за допомогою фізичного процесу, результат якого є непередбачуваний і не може бути повторно відтворений.

Послідовність псевдовипадкових чисел – відтворювана послідовність, сформована за допомогою детермінованого алгоритму (незалежно від способу його реалізації), що володіє статистичними властивостями послідовностей випадкових чисел.

2. Генератори псевдовипадкових чисел

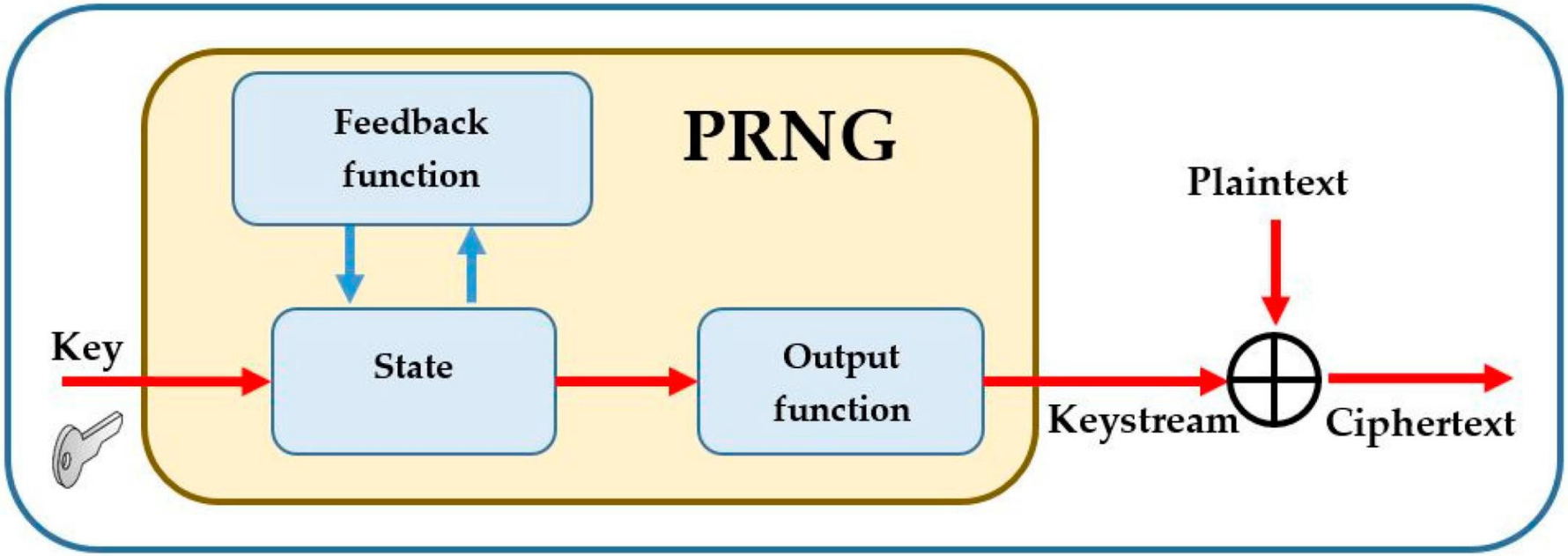
Чи можна вважати дану послідовність псевдовипадковою? Які будуть наступні елементи?

12, 14, 14, 17, 16, 20,…

2. Генератори псевдовипадкових чисел

Метою використання ГПВЧ у потокових шифрах є отримання нескінченної ключової послідовності, за використання відносно малої довжини самого початкового ключа.

Ключ, який породжує гаму також називають зародком (seed).



2. Генератори псевдовипадкових чисел

Найважливішою характеристикою ГПВЧ є довжина періоду повторення, після якого випадкові числа, на виході ГПВЧ почнуть повторюватися.

Псевдовипадкову послідовність можна використовувати як ключ тільки до повтору, інакше потоковий шифр виродиться в нестійкий алгоритм звичайного XOR.

2. Генератори псевдовипадкових чисел

Вимоги до ГПВЧ

1) Непередбачуваність (неможливо визначити наступне число з імовірністю, більшою за $\frac{1}{m}$, де m– потужність алфавіту генератора);

2) Відтворюваність дозволяє повторити (за відомих початкових значень) з абсолютною точністю послідовність на виході ГПВЧ в довільний момент часу та довільну кількість разів;

3) Великий період повторення (настільки великий, що його неможливо відтворити сучасними технічними засобами).

4) Числа, що генеруються мають бути статистично рівномірно розподілені.

2. Генератори псевдовипадкових чисел

Криптографічно стійкі генератори псевдовипадкових чисел (CSPRNG) — це спеціальний клас PRNG, який має додаткові властивості, що роблять його придатним для використання в криптографічних додатках.

Основні характеристики CSPRNG:

- **Непередбачуваність:** Навіть якщо відомо частину послідовності, вгадати наступні числа має бути практично неможливо.
- **Висока ентропія:** CSPRNG генерує послідовність чисел, які є дуже близькими до дійсно випадкових з точки зору розподілу і непередбачуваності.
- **Стійкість до атак:** Алгоритм повинен бути стійким до атак, наприклад, до атаки на зворотний аналіз або підбір seed. Якщо хакер дізнається про частину послідовності, він не зможе передбачити інші числа.
- **Безпека seed:** Навіть якщо початковий seed скомпрометований, алгоритм повинен залишатися безпечним або важким для зламу.
- **Застосування:** CSPRNG використовується у криптографічних алгоритмах, генерації ключів, цифрових підписах, аутентифікації та інших випадках, де безпека є критично важливою.

2. Генератори псевдовипадкових чисел

Характеристика	PRNG	CSPRNG
Призначення	Загальне використання (симуляції, ігри тощо)	Криптографія, безпечна передача даних
Непередбачуваність	Легко передбачити, якщо відомо seed	Непередбачувані, навіть якщо відома частина послідовності
Відтворюваність	Можна відтворити при відомому seed	Відтворювані, але з додатковими заходами безпеки щодо seed
Стійкість до атак	Вразливі до атак, наприклад, на підбір seed	Стійкі до криптографічних атак
Швидкість	Швидші	Можуть бути повільнішими через додаткові криптографічні операції
Застосування	Моделювання, ігри, тестування	Генерація ключів, токенів, цифрові підписи
Приклади	Лінійний конгруентний генератор (LCG), Mersenne Twister, Xorshift, Middle Square Method, Blum Blum Shub, WELL (Well Equidistributed Long-period Linear)	AES-CTR, HMAC_DRBG, SHA-256 PRNG, Fortuna, ChaCha20-based CSPRNG, Yarrow

2. Генератори псевдовипадкових чисел

Лінійний конгруентний генератор (LCG)

Для обчислення чергового числа x_{i+1} використовується формула:

$$x_{i+1} = (a \cdot x_i + b) mod\, m,$$

де a, b, m – деякі константи, а x_i – попереднє псевдовипадкове число, а x_0 – початкове значення (seed).

Якщо параметри LCG обрані правильно, то генератор буде породжувати випадкові числа з максимальним періодом, що дорівнює m.

2. Генератори псевдовипадкових чисел

Приклад 2.1: $a = 5, b = 3, m = 11, x_0 = 1.$

$x_1 = (5 \cdot 1 + 3) mod\ 11 = 8;$
$x_2 = (5 \cdot 8 + 3) mod\ 11 = 10;$
$x_3 = (5 \cdot 10 + 3) mod\ 11 = 9;$
$x_4 = (5 \cdot 9 + 3) mod\ 11 = 4;$
$x_5 = (5 \cdot 4 + 3) mod\ 11 = 1;$
$x_6 = (5 \cdot 1 + 3) mod\ 11 = 8;$
$x_7 = (5 \cdot 8 + 3) mod\ 11 = 10;$
$x_8 = (5 \cdot 10 + 3) mod\ 11 = 9.$

Лінійні конгруентні генератори не рекомендують використовувати, оскільки криптоаналітики навчилися відновлювати всю послідовність ПВЧ із кількох значень.

2. Генератори псевдовипадкових чисел

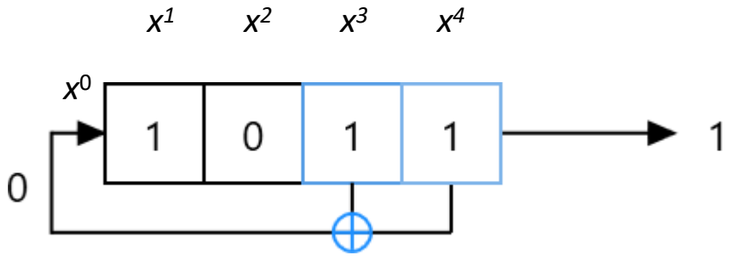
ГПВЧ на основі регістрів зсуву з лінійним зворотним зв'язком (LFSR)

Зворотний зв’язок є функцією XOR певних бітів регістра, які називають ***відводами*** (taps).

Розташування відводів для зворотного зв'язку може бути представлене многочленом з коефіцієнтами 0 або 1.

Наприклад, для 4-бітового LFSR многочлен зворотного зв'язку:

$$x^4 + x^3 + 1$$



2. Генератори псевдовипадкових чисел

Алгоритм LFSR

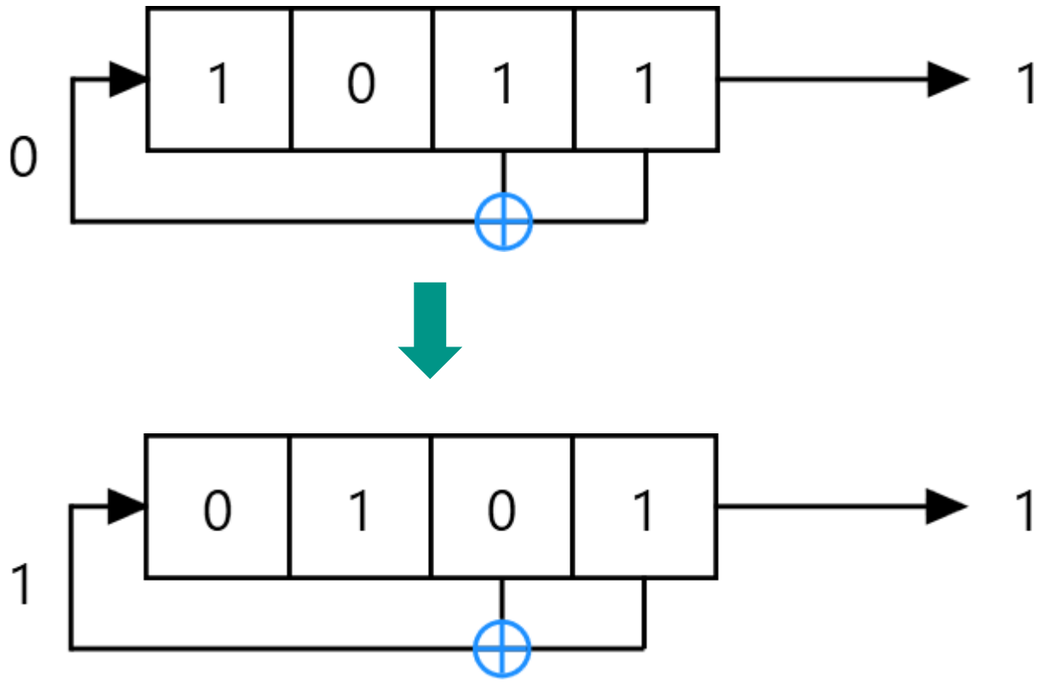
1. У регістр записується початкове значення.
2. Обчислюється XOR відповідних бітів (відводів).
3. Обчислений біт записується до першої позиції регістру ліворуч.
4. Усі решта біт зсуваються на одну позицію праворуч.
5. Останній біт з правого боку усувається з регістру і стає черговим бітом псевдовипадкової послідовності.
6. Алгоритм повторюється з 2 кроку.

2. Генератори псевдовипадкових чисел

Приклад 2.2:

Номер стану	Внутрішній стан Q R S T	Результат обчислення S⊕T	Біт, псевдовипадкової послідовності
1	1 0 1 1	0	1
2	0 1 0 1	1	1
3	1 0 1 0	1	0
4	1 1 0 1	1	1
5	1 1 1 0	1	0
6	1 1 1 1	0	1
7	0 1 1 1	0	1
8	0 0 1 1	0	1
9	0 0 0 1	1	1
10	1 0 0 0	0	0

2. Генератори псевдовипадкових чисел



LFSR розміром *n* бітів може перебувати в одному з $2^n–1$ станів. Тому, теоретично можна генерувати псевдовипадкову послідовність із максимальним періодом $2^n–1$.

2. Генератори псевдовипадкових чисел

Яким буде ключовий потік, що генерує LFSR, якщо його многочлен зворотного зв'язку $x^4 + x + 1$, а початковий стан 1101?

а) 1101010110;

б) 1101011001;

в) 1101011110;

г) 0010101011.

2. Генератори псевдовипадкових чисел

ГПВЧ на основі алгоритму BBS

Широке поширення в криптографії набув алгоритм Блюма – Блюма – Шуба (від прізвищ авторів – *L. Blum, M. Blum, M. Shub*), який називають ще генератором квадратичних лишків.

Послідовність, сформована за допомогою цього методу, має статистичні властивості, близькі до генераторів випадкових чисел, а метод є досить крипостійким.

2. Генератори псевдовипадкових чисел

Алгоритм BBS

1. Обирають два великих простих числа p та q, які конгруентні. При діленні цих чисел на *4* повинен виходити однаковий залишок *3*.

2. Обчислюється $M = p \cdot q$ – *ціле число Блюма*.

3. Вибирається інше випадкове ціле число x, взаємно просте з M.

4. Обчислюється $x_0 = x^2 mod\ M$ – *стартове число генератора*.

5. На кожному *n*-му кроці обчислюється $x_{n+1} = x^2 mod\ M$.

6. Результатом *n*-го кроку є один (зазвичай молодший) біт числа x_{n+1}.

2. Генератори псевдовипадкових чисел

Приклад 2.3: Нехай $p = 11$, $q = 19$, (переконуємося, що $11\ mod\ 4 = 3$, $19\ mod\ 4 = 3$).

Тоді $M = p \cdot q = 11 \cdot 19 = 209$.

Виберемо x, взаємно просте з M: нехай $x = 3$.

Обчислимо стартове число генератора x_0:

$$x_0 = x^2 mod\ M = 3^2 mod\ 209 = 9\ mod\ 209 = 9$$

Далі обчислимо перші десять чисел x_i за алгоритмом *BBS*. Як випадковий біт будемо брати молодший біт у двійковому записі числа x_i.

2. Генератори псевдовипадкових чисел

$x_1 = 9^2 \bmod 209 = 81 \bmod 209 = 81$ – молодший біт: *1*;

$x_2 = 81^2 \bmod 209 = 6561 \bmod 209 = 82$ – молодший біт: *0*;

$x_3 = 82^2 \bmod 209 = 6724 \bmod 209 = 36$ – молодший біт: *0*;

$x_4 = 36^2 \bmod 209 = 1296 \bmod 209 = 42$ – молодший біт: *0*;

$x_5 = 42^2 \bmod 209 = 1764 \bmod 209 = 92$ – молодший біт: *0;*

$x_6 = 92^2 \bmod 209 = 8464 \bmod 209 = 104$ – молодший біт: *0;*

$x_7 = 104^2 \bmod 209 = 10816 \bmod 209 = 157$ – молодший біт: *1*;

$x_8 = 157^2 \bmod 209 = 24649 \bmod 209 = 196$ – молодший біт: *0*;

$x_9 = 196^2 \bmod 209 = 38416 \bmod 209 = 169$ – молодший біт: *1*;

$x_{10} = 169^2 \bmod 209 = 28561 \bmod 209 = 137$ – молодший біт: *1*.

3. Потоковий шифр RC4

RC4 (Rivest Cipher 4, Ron's Code) – потоковий шифр, який розробив Рональд Рівест у 1987 р.

Шифр RC4 застосовується в деяких широко поширених стандартах і протоколах таких, як WEP, WPA і TLS (до 2015 року)

Рональд Лінн Рівест

3. Потоковий шифр RC4

Ініціалізація матриці стану і масиву ключів

Матриця стану (256 байт) початково заповнюється значеннями:

$S[0] = 0, S[1] = 1, S[2] = 2, \ldots, S[254] = 254, S[255] = 255.$

Ключем є послідовність байтів довжини L, що записується у масив K:

$K[0], K[1], K[2], \ldots, K[254], K[255].$

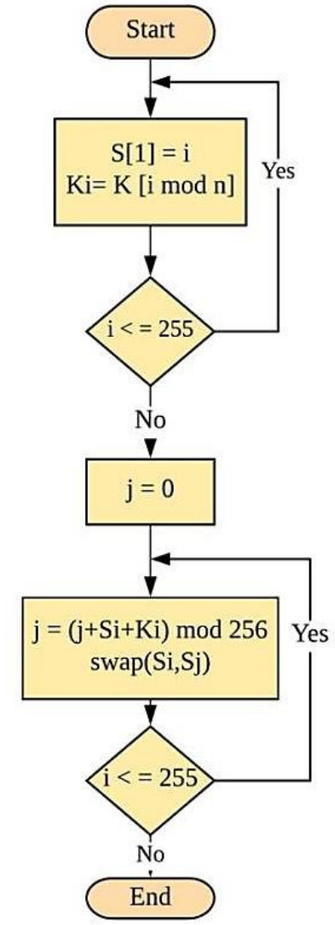
Якщо ключ шифрування має точно *256* байтів, то байти копіюються в масив K, інакше байти повторюються, поки не заповниться масив K

3. Потоковий шифр RC4

Перестановка матриці стану на основі значень ключа

Кожен черговий елемент S_i обмінюється місцями з елементом S_j, номер j якого визначається сумою номера елементу j, самого елементу S_i та елементу ключа K_i.

Значення лічильників i та j спочатку дорівнюють 0.

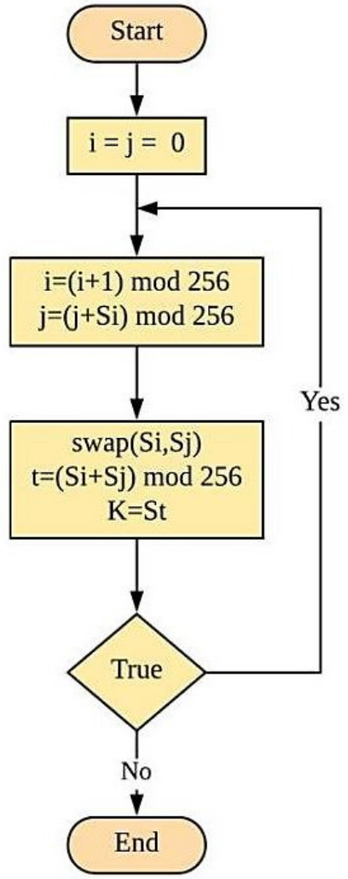


3. Потоковий шифр RC4

Перестановка матриці стану і генерація ключового потоку

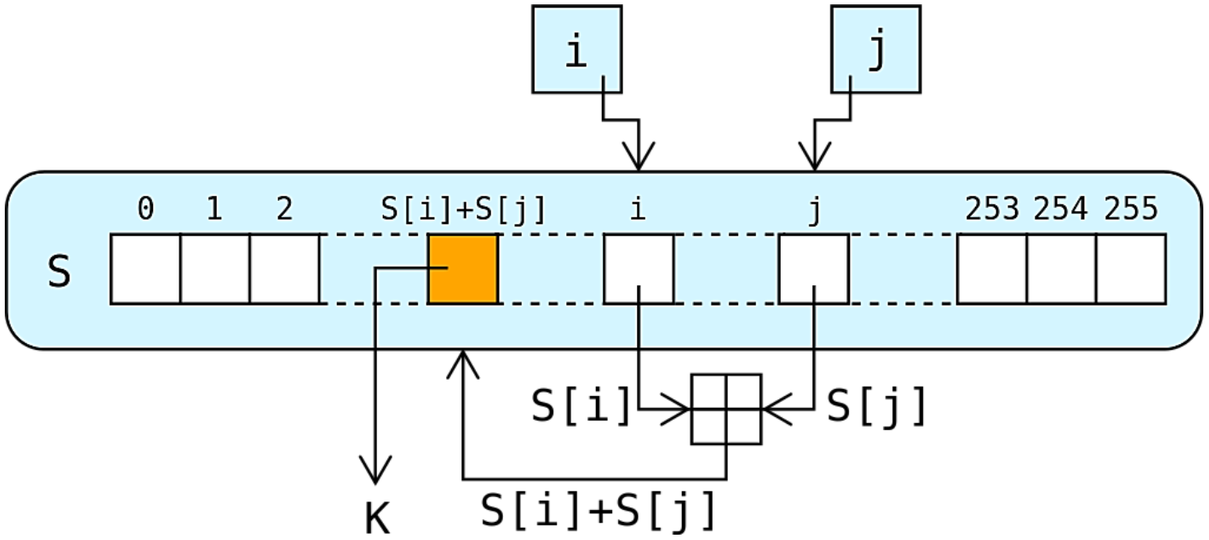
Два елементи матриці стану міняються місцями на основі двох індивідуальних змінних *i* та *j*, що обчислюються за відповідними формулами.

Потім сума значень елементів S_i та S_j визначає індекс *t* елементу S_t, який використовуватиметься як ключ *K*.



3. Потоковий шифр RC4

Генерація ключового потоку (ПВЧ) у RC4



Шифрування RC4

При шифруванні байт *K* додають за модулем 2 з байтом відкритого тексту.

3. Потоковий шифр RC4

Переваги RC4

- **Простота реалізації:** Алгоритм RC4 дуже простий і може бути легко реалізований навіть на низькорівневому обладнанні.
- **Висока швидкість:** RC4 є одним із найшвидших шифрів, що робить його привабливим для застосувань, де потрібно швидке шифрування великих обсягів даних.
- **Мінімальні вимоги до пам'яті:** RC4 використовує мінімум ресурсів (масив із 256 байтів для зберігання стану), що робить його придатним для використання на пристроях із обмеженою пам'яттю.

Недоліки RC4

1.Низька криптографічна стійкість: RC4 не забезпечує надійного захисту через відомі атаки на перші байти потоку та передбачувані шаблони в ключовій послідовності.

2.Атаки на протокол: RC4 у протоколах, таких як WEP та TLS, виявився вразливим до численних атак, що призвело до його поступової заміни іншими шифрами.

3.Втрата безпеки при повторному використанні ключа: Повторне використання того ж ключа в RC4 може призвести до катастрофічних наслідків для безпеки, адже це дозволяє зловмисникам виконувати атаки на ключовий потік.

Використання

1.WEP (Wireless Equivalent Privacy): Стандарт шифрування для безпровідних мереж Wi-Fi. Вразливість RC4 у WEP зробила цей протокол застарілим.

2.TLS/SSL: RC4 використовувався як один з алгоритмів шифрування в ранніх версіях TLS/SSL для захисту трафіку в інтернеті. Після виявлення вразливостей його використання було заборонено.

3.Microsoft Remote Desktop Protocol (RDP): RC4 використовувався для шифрування в ранніх версіях цього протоколу.

4.PDF: RC4 застосовувався для шифрування в ранніх версіях шифрування PDF-документів.

4. Інші потокові шифри

Потоковий шифр SNOW 2.0

Потоковий симетричний шифр SNOW 2.0 є генератором ключових потоків, який використовує як вхідні дані 128 або 256-бітовий секретний ключ K і 128-бітовий вектор ініціалізації IV. На сьогодні цей шифр є стандартизованим та являє собою один з найбільш швидких програмно орієнтованих потокових шифрів.

4. Інші потокові шифри

Потоковий шифр «СТРУМОК»

Потоковий симетричний шифр описаний у національному стандарті України *ДСТУ 8845:2019 «Інформаційні технології. Криптографічний захист інформації.* Алгоритм симетричного потокового перетворення». Стандарт набрав чинності з 1 жовтня 2019 року наказом ДП «УкрНДНЦ» від 2 квітня 2019 року.

Основні властивості:

- **Криптографічна стійкість:** STRUMOK розроблений із врахуванням сучасних криптографічних вимог і має високу стійкість до основних атак, включаючи атаки на основі аналізу ключа, статистичні атаки та відомі види криптоаналізу потокових шифрів.
- **Ефективність:** Алгоритм оптимізований для використання на пристроях з обмеженими ресурсами, таких як мобільні пристрої та вбудовані системи.
- **Безпека ключа:** STRUMOK підтримує різні довжини ключів і забезпечує криптографічний захист навіть при обмеженій кількості обчислювальних ресурсів.

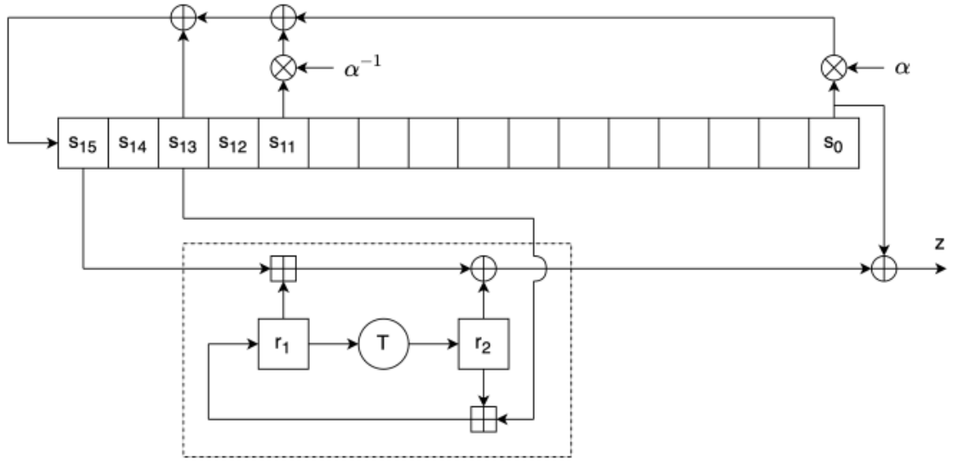
4. Інші потокові шифри

Алгоритм працює зі словами довжиною 64 біти. Використовує 256 бітовий вектор ініціалізації IV та ключ довжиною 256 або 512 бітів

Базові компоненти шифру :

- лінійний регістр зсуву зі зворотним зв'язком (LFSR) (складається з 16 комірок, кожна з яких – 64-бітове слово);
- скінченний автомат (FSM – finite state machine) (основа - 2 регістри по 64 біти.)

Шифр «Струмок» у режимі вироблення ключового потоку



Основні процедури шифру :

- **INIT.** Функція ініціалізації, яка приймає на вхід вектор ініціалізації IV довжиною 256 бітів, ключ шифрування K довжиною 256 або 512 бітів та виконує ініціалізацію початкового стану шифру
- **NEXT.** Функція зміни стану, яка приймає на вхід стан $S(i) = (s(i), r(i))$, де $s(i)$, $r(i)$ – стани лінійного регістру зсуву та регістрів скінченного автомата відповідно в проміжок часу i.
- **STRM.** Функція вироблення гами, яка приймає на вхід зміну стану $S(i) = (s(i), r(i))$ та повертає на виході 64 біти ключової гами.

4. Інші потокові шифри

Потоковий шифр Salsa 20

Програмно-орієнтований алгоритм Salsa 20 став переможцем міжнародного конкурсу eSTREAM. Для ініціалізації внутрішнього стану використовується ключ довжиною 256 біт, 64-бітний nonce та 64-бітна позиція блоку ключового потоку. Максимальна довжина псевдовипадкової ключової послідовності дорівнює 2^{70} біт.

4. Інші потокові шифри

Потоковий шифр Trivium

Trivium – це симетричний апаратно-орієнтований паралельний потоковий шифр. Trivium найбільш простий шифр проекту eSTREAM, який демонструє відмінні результати криптостійкості. Trivium призначений для генерації 264 біт ключового потоку з 80 біт секретного ключа і 80 біт вектору ініціалізації. Шифр є біт-орієнтованим.

4. Інші потокові шифри

Потоковий шифр Grain

Grain – синхронний поточний апаратно-орієнтований шифр. Залежно від апаратної реалізації може бути біт-орієнтованим або слово-орієнтованим. На вхід подається ключ довжиною 80 біт та IV довжиною 64 біти. Довжина ключового потоку, який може бути вироблений на одній парі ключ/вектор – 2^{44} біт.