9. Слайд
Одним із прикладів асиметричного шифрування є шифр RSA. Це криптографічний алгоритм з відкритим ключем, який базується на обчислювальній складності задачі факторизації великих цілих чисел. Він став першим такого типу, що був придатним як для шифрування, так і для цифрового підпису

10. слайд
Цей алгоритм був розроблений в 1977 році Рональдом Рівестом, Аді Шаміром та Леонардом Адлеманом, коли вони були науковцями в Массачусетському технологічному інституті (MIT).

У цей період з'явилася ідея використання публічного ключа для шифрування повідомлень. І Рівест, Шамір та Адлеман запропонували такий метод шифрування, який би використовував відкритий ключ для шифрування повідомлення і приватний ключ для розшифрування. Їхній метод став відомий як RSA.

Перші публічні описи алгоритму з'явилися у 1978 році на конференції по криптографії. Після цього RSA став широко використовуватися для безпеки інформації і його розробники були визнані світовими лідерами у галузі криптографії. Рівест, Шамір та Адлеман отримали багато нагород та визнання за свою роботу, включаючи Національну медаль науки від президента США в 1999 році.

11. слайд
Опис алгоритму
Тепер перейдемо до опису алгоритму RSA який складається з 4 етапів: генерації ключів, розповсюдження ключів, шифрування та розшифрування.
Так як згадувалося раніше RSA є асиметричним шифром відповідно він використовує два ключі - публічний та приватний. Публічний ключ відомий усім, хто може зашифрувати повідомлення. Приватний ключ відомий лише одній стороні, яка може розшифрувати повідомлення.

12. слайд
Тож розглянемо перший етап - це генерація цих ключів. Для цього потрібно вибрати два великих простих числа p та q. Потім потрібно обчислити їх добуток n = p * q. Це число буде використовуватись як модуль шифрування та розшифрування. Тоді вже обчислюється функція Ейлера

1. Вибираються два великі прості числа p і q
2. Обчислюється їх добуток $n = pq$
3. Обчислюється функція Ейлера $\varphi(n) = (p-1)(q-1)$
4. Вибирається ціле число e таке, що $1 < e < \varphi(n)$ та e взаємно просте з $\varphi(n)$
5. За допомогою розширеного алгоритму Евкліда знаходиться число d таке, що $ed \equiv 1 \pmod{\varphi(n)}$

Тоді наші числа e і d — відкритою й заркритою експонентами (англ. encryption and decryption exponents), відповідно. Пари чисел (n,e) є публічним ключем, а (n,d) — приватним. Числа p і q після генерації пари ключів можуть бути знищені, але в жодному разі не повинні бути розкриті.

13. Слайд

Наступний етап це розповсюдження ключ. Нехай у нас є Боб та Аліса. Боб хоче відправити свої секретні повідомлення Алісі. Для того щоб він міг це зробити Аліса повинна передати свій відкритий ключ (n, e) Бобу через надійний, але не обов'язково секретний маршрут. Секретний ключ d ніколи не розповсюджується.

14. слайд

Припустимо, що Боб хотів би відправити повідомлення *M* Алісі. Спочатку він перетворює *M* в ціле число *m* так, щоб $0 \le m < n$ за допомогою узгодженого оборотного протоколу, відомого як схеми доповнення. Потім він обчислює зашифрований текст *c*, використовуючи відкритий ключ Аліси *e*, за допомогою рівняння:

$$c = m^e \bmod n.$$

Потім Боб передає *c* Алісі.

15.слайд

Для розшифрування повідомлення Боба Алісі потрібно використати приватний ключ щоб обчислити таку рівність:

Для розшифрування повідомлення Боба *m* Алісі потрібно обчислити таку рівність:

$$m = c^d \bmod n.$$

16 слайд

Наприклад, якщо ми маємо публічний ключ, який має значення n = 35 та e = 5, і ми хочемо зашифрувати повідомлення "Hello", ми можемо перетворити кожен символ в його числовий еквівалент і застосувати на ньому публічний ключ:

H = 72 -> зашифроване значення = 72^5 mod 35 = 22
e = 101 -> зашифроване значення = 101^5 mod 35 = 11
l = 108 -> зашифроване значення = 108^5 mod 35 = 18
l = 108 -> зашифроване значення = 108^5 mod 35 = 18
o = 111 -> зашифроване значення = 111^5 mod 35 = 1

17 слайд
Тепер, якщо ми хочемо розшифрувати повідомлення, нам потрібно використати приватний ключ, який містить d = 29. Застосувавши його до кожного зашифрованого символу, ми можемо отримати початкові символи:

22^29 mod 35 = 72 -> H
11^29 mod 35 = 101 -> e
18^29 mod 35 = 108 -> l
18^29 mod 35 = 108 -> l
1^29 mod 35 = 111 -> o

18 слайд
RSA гарантує коректність тому, що базується на складності обернення функції шифрування за публічним ключем без знання відповідного приватного ключа. Ця складність пов'язана з тим, що обернення функції шифрування за публічним ключем є трудомістким завданням для будь-якої сторони, яка не має відповідного приватного ключа.

Конкретно, для того, щоб отримати оригінальне повідомлення з його зашифрованого вигляду за публічним ключем, необхідно виконати обернення функції шифрування, яке вимагає знання приватного ключа. Без цього знання обернення функції шифрування є надзвичайно складним завданням, навіть для суперкомп'ютерів, через велику довжину ключа.

19 слайд
Таким чином, RSA гарантує коректність, тому що зашифровані дані не можуть бути розшифровані без знання відповідного приватного ключа, що забезпечує конфіденційність та безпеку обміну інформацією в Інтернеті та в інших мережах.

20 слайд
Варто також зазначити, що коректність RSA може бути порушена в певних практичних ситуаціях, таких як використання ненадійного генератора випадкових чисел або неправильне зберігання приватного ключа. Тому важливо дотримуватись рекомендацій щодо безпеки при використанні RSA.

21 слайд
RSA є однією з найбільш надійних криптографічних систем на сьогоднішній день, але все ж можливі атаки, які можуть порушити її коректність. Давайте розглянемо які атаки можуть порушити цю коректність)

22 слайд
Brute-force атаки: це метод спробувати всі можливі комбінації приватного ключа, доки не буде знайдено той, який може розшифрувати зашифрований текст. Цей метод є дуже неефективним, оскільки простір ключів RSA є дуже великим і складно обчислюваним.

23 слайд
Факторизація чисел: це метод спробувати розкласти велике число, що використовується у системі RSA, на прості множники. Якщо атакуюча сторона зможе розкласти число, то вона зможе використовувати цю інформацію для обчислення приватного ключа. Цей метод є більш ефективним, ніж brute-force атака, але все ще складним і обчислювально важким.

24 слайд
Атаки на повідомлення з відомим відкритим текстом: це метод, коли атакуюча сторона має доступ до зашифрованого повідомлення та його відкритого тексту, і намагається відновити приватний ключ. Цей метод може бути успішним, якщо повідомлення містить інформацію про ключ або якщо атакуюча сторона може знайти відповідність між відкритим і зашифрованим текстом.

Слайд 25:
Привіт! Сьогодні я хотів би розповісти вам про надійність RSA у криптозахисті інформації. RSA - це криптографічний алгоритм, який використовується для шифрування та розшифрування інформації та для підпису даних. Основна ідея RSA полягає в тому, що для шифрування та розшифрування використовується пара ключів: публічний ключ та приватний ключ.

Після слів в основному розглянемо такі тези:
Теза полягає у складності факторизації.
Теза стосується довжини ключа
теза стосується безпеки публічного ключа

Слайд 26:
Перша теза полягає у складності факторизації. Основа надійності RSA полягає в тому, що шифрування за його допомогою ґрунтується на складності факторизації великих чисел. Наразі, немає жодного ефективного способу факторизації великих простих чисел за прийнятний час.

Факториза́ція цілого числа — розкладання заданого цілого числа на прості множники. На відміну від задачі розпізнавання простоти числа, факторизація ймовірно є складною задачею. Передбачувана складність задачі факторизації лежить в основі криптостійкості деяких алгоритмів шифрування з відкритим ключем, таких як RSA.

Слайд 27:

Друга теза стосується довжини ключа. Для того, щоб забезпечити безпеку RSA, довжина ключа повинна бути достатньо великою. Наразі, рекомендована довжина ключа для захисту від квантових обчислювачів складає 2048 біт.

Слайд 28:
Третя теза стосується безпеки публічного ключа.
Зважаючи на те, що RSA є одним з найбільш поширених криптографічних алгоритмів, важливо знати, як його використовувати належним чином, щоб уникнути можливих атак на систему.
Публічний ключ RSA може бути розголошений без будь-яких наслідків для безпеки системи, оскільки для його використання потрібен приватний ключ. Це дозволяє застосовувати RSA для безпечної передачі публічного ключа без захисту.
RSA може бути піддано атакам не тільки з боку зловмисників, але й з боку квантових обчислювачів. Квантові обчислювачі можуть бути використані для розкладу великих простих чисел за значно менший час, ніж класичні обчислювачі, що може ускладнити надійність RSA у майбутньому.
Отже, використання RSA вимагає відповідального та обережного підходу, що включає в себе використання стійких ключів, захист від можливих атак та регулярне оновлення безпеки системи. При правильному застосуванні та захисті, RSA може забезпечити надійний криптозахист інформації.