**Ілюстрація роботи RSA на прикладі**

Припустимо, я хочу отримати від вас якісь дані. Ми з вами не хочемо, щоб ці дані дізнався хтось, крім нас. І у нас немає ніякої впевненості в надійності каналу передачі даних.

**Крок перший. Підготовка ключів**

Я повинен виконати попередні дії: згенерувати публічний і приватний ключ.

• Вибираю два простих числа. Нехай це буде p=3 і q=7.

• Обчислюємо модуль — добуток наших p і q: n=p×q=3×7=21.

• Обчислюємо функцію Ейлера: φ=(p-1)×(q-1)=2×6=12.

• Вибираємо число e, що відповідає таким критеріям: воно повинно бути просте, воно повинно бути менше φ — залишаються варіанти: 3, 5, 7, 11, воно повинно бути взаємно просте з φ; залишаються варіанти 5, 7, 11. Оберемо e=5. Це, так звана, відкрита експонента.

Тепер пара чисел {e, n} — це мій відкритий ключ. Я відправляю його вам, щоб ви зашифрували своє повідомлення. Але для мене це ще не все. Я повинен отримати закритий ключ.

Мені потрібно обчислити число d, зворотне е за модулем φ. Тобто залишок від ділення по модулю φ твори d×e повинен дорівнювати 1. Запишемо це у позначеннях, прийнятих у багатьох мовах програмування: (d×е)%φ=1. Або (d×5)%12=1. d може бути одно 5 ((5×5)%12=25%12=1), але щоб воно не плуталася з e надалі оповіданні, давайте візьмемо його рівним 17. Можете перевірити самі, що (17×5)%12 дійсно одно 1 (17×5-12×7=1). Отже d=17. Пара {d, n} — це секретний ключ, його я залишаю у себе. Його не можна повідомляти нікому. Тільки власник секретного ключа може розшифрувати те, що було зашифрувати повідомлення відкритим ключем.

**Крок другий. Шифрування**

Тепер прийшла ваша черга шифрувати повідомлення. Припустимо, ваше повідомлення це число 19. Позначимо його P=19. Крім нього у вас вже є мій відкритий ключ: {e, n} = {5, 21}. Шифрування виконується за наступним алгоритмом:

• Зводите ваше повідомлення ступінь e за модулем n. Тобто, вираховуєте 19 у ступені 5 (2476099) і берете залишок від ділення на 21. Виходить 10 — це ваші закодовані дані.

Строго кажучи, вам зовсім нема чого обчислювати величезне число «19 у ступені 5». При кожному збільшенні достатньо обчислювати не повне твір, а лише залишок від ділення на 21. Але це вже деталі реалізації обчислень, давайте не будемо в них заглиблюватися. Отримані дані E = 10, ви відправляєте мені.

Тут треба зауважити, що повідомлення P = 19 не повинно бути більше n = 21. інакше нічого не вийде.

**Крок третій. розшифровка**

Я отримав ваші дані (E = 10), і у мене є закритий ключ {d, n} = {17, 21}.

Зверніть увагу на те, що відкритий ключ не може розшифрувати повідомлення. А закритий ключ я нікому не говорив. У цьому вся принадність асиметричного шифрування.

Починаємо розкодувати:

• Я роблю операцію, дуже схожу на вашу, але замість e використовую d. Зводжу E в ступінь d: отримую 10 в ступінь 17 (дозвольте, я не буду писати одиничку з сімнадцятьма нулями). Обчислюю залишок від ділення на 21 і отримую 19 - ваше повідомлення.

Зауважте, ніхто, крім мене (навіть ви!) Не може розшифрувати ваше повідомлення (E = 10), так як ні в кого немає закритого ключа.