



Московский институт электроники и  
математики им. А.Н. Тихонова

Кафедра информационной  
безопасности киберфизических  
систем

Москва 2024

# Криптографические методы защиты информации

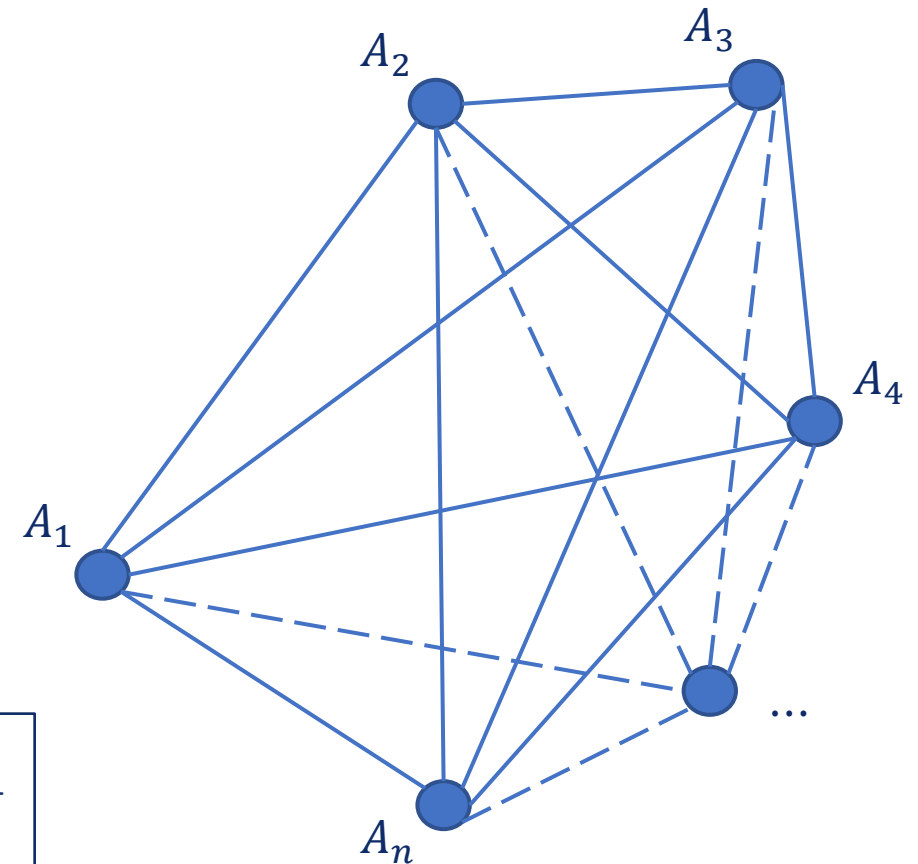
Криптография с открытым ключом



## Общие сведения

## Проблема управления ключами

- Генерация ключевой информации.
- Распределение ключей между пользователями.
- Безопасное хранение долговременных ключей.
- Обновление ключей.
- Уничтожение ключевой информации.



## Криптография с открытым ключом

- **Криптосистемы с открытым ключом (асимметричные криптосистемы)** используют два различных ключа: открытый ключ зашифрования и закрытый ключ расшифрования.
- Асимметричное шифрование основано на использовании **однонаправленных функций с лазейкой** (ловушкой, люком):
  - открытый ключ определяет конкретную реализацию однонаправленной функции;
  - закрытый ключ содержит информацию о лазейке.





## Известные криптосистемы с открытым ключом

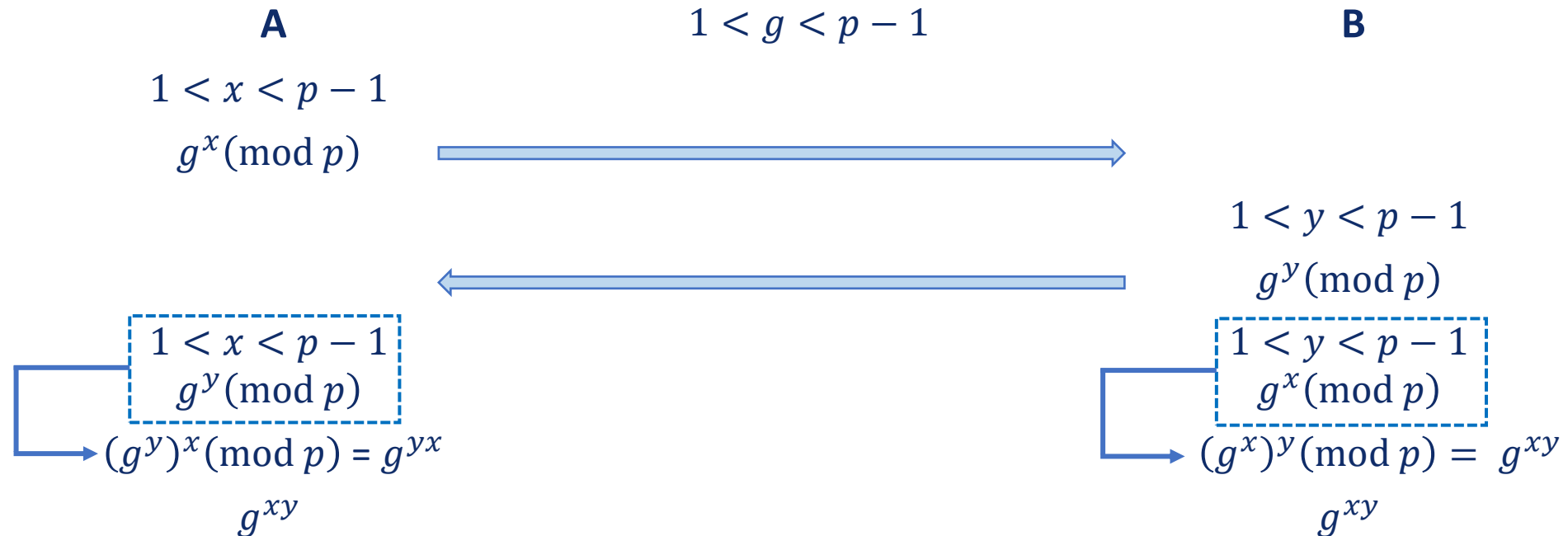
- Основные математические задачи:
  - факторизация целых чисел;
  - дискретное логарифмирование.
- Асимметричные криптосистемы:
  - криптосистема RSA;
  - криптосистема Рабина;
  - криптосистема Эль-Гамала.



## Протокол Диффи-Хеллмана

## Протокол Диффи-Хеллмана

- Предназначен для формирования общего секретного ключа двумя (или более) пользователями по открытому каналу связи.
- Основан на проблеме **дискретного логарифмирования**:
  - $g^x \pmod{p} = c, x = \log_g c = ?$





## Атака «человек посередине»

**A**

$$1 < x < p - 1$$

$$g^x \pmod{p}$$



$$(g^m)^x \pmod{p} = g^{mx}$$

$$g^{mx}$$

**E**

$$1 < m < p - 1$$

$$g^m \pmod{p}$$

$$(g^x)^m \pmod{p} = g^{xm}$$

$$g^{mx}$$

$$1 < n < p - 1$$

$$g^n \pmod{p}$$

$$(g^y)^n \pmod{p} = g^{yn}$$

$$g^{yn}$$

**B**

$$1 < y < p - 1$$

$$g^y \pmod{p}$$

$$(g^n)^y \pmod{p} = g^{ny}$$

$$g^{yn}$$







# Криптосистема RSA



## Общее описание

- Первая криптосистема с открытым ключом (1978).
- Криптосистема RSA основана на задаче **факторизации** целых чисел:
  - $n = p_1 p_2 \dots p_k$ ,
  - $n$  – известно,  $p_1 = ?$ ,  $p_2 = ?$ , ...,  $p_k = ?$

$1125899839733759 \cdot 489133282872437279 = 55071508479452453847089478640176$	Легко
$70951999110841802331327559 = n_1 \cdot n_2 = ?$	<b>Сложно</b>



## Алгоритм генерации ключей

1. Алиса генерирует два больших простых числа  $p$  и  $q$ , отличных друг от друга, причем  $|p - q|$  – большое число.
2. Держа  $p$  и  $q$  в секрете, Алиса вычисляет их произведение  $n = p \cdot q$ , которое называют **модулем алгоритма**.
3. Алиса вычисляет значение функции Эйлера для  $n$  по формуле  $\varphi(n) = (p - 1)(q - 1)$ .
4. Алиса выбирает целое число  $e$ , взаимно простое со значением функции  $\varphi(n)$ . Это число называется **экспонентой зашифрования**.
5. Алиса вычисляет значение  $d$ , удовлетворяющее соотношению  $e \cdot d \equiv 1 \pmod{\varphi(n)}$ . Это значение называется **экспонентой расшифрования**.
6. Пара  $(e, n)$  публикуется в качестве открытого ключа Алисы,  $d$  является закрытым ключом и держится в секрете.



## Алгоритмы зашифрования и расшифрования

- **Алгоритм зашифрования:**

1. Боб получает аутентичную копию открытого ключа Алисы  $(e, n)$ .
2. Боб представляет сообщение в виде числа  $m$ , меньшего модуля алгоритма  $n$ , либо в виде последовательности таких чисел  $m_1, m_2, \dots, m_k$ .
3. Боб вычисляет  $c_i = m_i^e \pmod{n}$ .
4. Боб отправляет шифртекст Алисе.

- **Алгоритм расшифрования:**

1. Алиса получает от Боба шифртекст в виде числа  $c$ , меньшего модуля алгоритма  $n$ , либо в виде последовательности таких чисел  $c_1, c_2, \dots, c_k$ .
2. Алиса вычисляет  $m_i = c_i^d \pmod{n}$ .

## Доказательство корректности шифрования в RSA

- Покажем, что  $m^{ed} \equiv m \pmod{n}$  для любого  $n = p \cdot q$ , где  $e \cdot d \equiv 1 \pmod{\varphi(n)}$ .
- **Случай 1:**  $\text{НОД}(m, n) = 1$ . Тогда верно следующее
  - $m \in \mathbb{Z}_n^*$  и по теореме Эйлера  $m^{\varphi(n)} \equiv 1 \pmod{n}$ ;
  - $m^{ed} = m^{1+k \cdot \varphi(n)} = m \cdot (m^{\varphi(n)})^k \equiv m \pmod{n}$ .
- **Случай 2:**  $\text{НОД}(m, n) = p$ . Тогда верно следующее:
  - $m^{ed} \equiv 0 \pmod{p}$ ;
  - $m \in \mathbb{Z}_q^*$  и по теореме Эйлера  $m^{\varphi(q)} \equiv 1 \pmod{q}$ ;
  - $m^{ed} = m^{1+k \cdot (p-1) \cdot (q-1)} = m \cdot (m^{(q-1)})^{k \cdot (p-1)} \equiv m \pmod{q}$ ;
  - $m^{ed} \equiv m \pmod{n}$  согласно китайской теореме об остатках.

## Доказательство корректности шифрования в RSA

- **Применение китайской теоремы об остатках:**

- $$\begin{cases} m^{ed} \equiv 0 \pmod{p}; \\ m^{ed} \equiv m \pmod{q}. \end{cases}$$
- $a_1 = 0, a_2 = m, n_1 = p, n_2 = q; N = pq, N_1 = N/n_1 = q, N_2 = N/n_2 = p;$
- $v_1 n_1 + u_1 N_1 = 1$ , следовательно  $v_1 p + u_1 q = 1$ ;
- $v_2 n_2 + u_2 N_2 = 1$ , следовательно  $v_2 q + u_2 p = 1$ ;
- $a \equiv (\sum_{i=1}^k a_i u_i N_i) \pmod{N} \equiv (0 \cdot u_1 q + m u_2 p) \pmod{pq} \equiv (m u_2 p) \pmod{pq};$
- $u_2 p = 1 - v_2 q$  и  $m = sp$ , так как  $\text{НОД}(m, n) = p$ ;
- $a \equiv (m(1 - v_2 q)) \pmod{pq} \equiv (m - v_2 spq) \pmod{pq} \equiv m \pmod{pq}.$



## Пример шифрования

- Генерация ключей:

1. Пара простых чисел:
2. Модуль алгоритма:
3. Значение функции Эйлера:
4. Экспонента шифрования:
5. Экспонента расшифрования:

$$p = 113, q = 191$$

$$n = p \cdot q = 113 \cdot 191 = 21583$$

$$\varphi(n) = (113 - 1)(191 - 1) = 21280$$

$$e = 13$$

$$d = 1637$$

$q$	$r$	$y$	$\varphi(n)$	$e$	$y_2$	$y_1$
–	–	–	21280	13	0	1
1636	12	–1636	13	12	1	–1636
1	1	1637	12	1	–1636	1637
12	0		1		1637	



## Пример шифрования

- **Зашифрование:**

1. Открытый ключ:  $e = 13, n = 21583$

2. Открытый текст:

- Символы: *CRYPTO*
- ASCII-коды: (0x43, 0x52, 0x59, 0x50, 0x54, 0x4F)
- Двоичная строка: 010000110101001001011001010100000101010001001111
- Длина блока:  $\lfloor \log_2 21583 \rfloor = 14$
- Блоки:  
 $m_4 = 00000000010000_2 = 16_{10}$   
 $m_3 = 11010100100101_2 = 13605_{10}$   
 $m_2 = 10010101000001_2 = 9537_{10}$   
 $m_1 = 01010001001111_2 = 5199_{10}$





## Пример шифрования

- **Зашифрование:**

- 3. Шифртекст:

- Длина блока:  $\lfloor \log_2 21583 \rfloor + 1 = 15$
    - Блоки:
$$c_4 = 16^{13} \bmod 21583 = 12649_{10} = 011000101101001_2$$
$$c_3 = 13605^{13} \bmod 21583 = 5288_{10} = 001010010101000_2$$
$$c_2 = 9537^{13} \bmod 21583 = 12068_{10} = 010111100100100_2$$
$$c_1 = 5199^{13} \bmod 21583 = 2148_{10} = 000100001100100_2$$
    - Двоичная строка: 011000101101001001010010101000010111100100100000100001100100
    - ASCII-коды: (0x06, 0x2D, 0x25, 0x2A, 0x17, 0x92, 0x08, 0x64)
    - Символы: □-%\*↑T□d



## Криптосистема Рабина



## Алгоритм генерации ключей

1. Алиса генерирует два больших простых числа  $p$  и  $q$  таких, что  $p \equiv q \equiv 3 \pmod{4}$ . Такой специальный вид простых чисел ускоряет процедуру извлечения квадратных корней по модулю  $p$  и  $q$ .
2. Алиса вычисляет  $n = pq$ .
3. Открытый ключ Алисы есть  $n$ .
4. Закрытый ключ Алисы есть пара  $(p, q)$ .



## Алгоритмы зашифрования и расшифрования

- **Алгоритм зашифрования:**

1. Боб получает аутентичную копию ключа Алисы  $n$ .
2. Боб представляет сообщение в виде числа  $m$ , меньшего  $n$ , либо в виде последовательности таких чисел  $m_1, m_2, \dots, m_k$ .
3. Боб вычисляет  $c_i = m_i^2 \pmod{n}$ .
4. Боб отправляет шифртекст Алисе.

- **Алгоритм расшифрования:**

1. Алиса получает от Боба шифртекст в виде числа  $c$ , меньшего  $n$ , либо в виде последовательности таких чисел  $c_1, c_2, \dots, c_k$ .
2. Алиса извлекает из каждого из значений  $c_1, c_2, \dots, c_k$  4 квадратных корня по модулю  $n$ .
3. Алиса определяет нужные значения  $m_1, m_2, \dots, m_k$  для каждой четверки корней



## Пример шифрования

- **Генерация ключей:**

1. Пара простых чисел:
2. Модуль алгоритма:

$$p = 127, q = 199$$

$$n = p \cdot q = 127 \cdot 199 = 25273$$



## Пример шифрования

- **Зашифрование:**


1. Открытый ключ:  $n = 25273$
2. Открытый текст:
  - Символы: *CRYPTO*
  - ASCII-коды:  $(0x43, 0x52, 0x59, 0x50, 0x54, 0x4F)$
  - Двоичная строка:  $010000110101001001011001010100000101010001001111$
  - Длина блока:  $\lfloor \log_2 25273 \rfloor = 14$
  - Блоки:
$$m_4 = 00000000010000_2 = 16_{10}$$
$$m_3 = 11010100100101_2 = 13605_{10}$$
$$m_2 = 10010101000001_2 = 9537_{10}$$
$$m_1 = 01010001001111_2 = 5199_{10}$$



## Пример шифрования

- **Зашифрование:**

- 3. Шифртекст:

- Длина блока:  $\lfloor \log_2 25273 \rfloor + 1 = 15$
    - Блоки:
$$c_4 = 16^2 \bmod 25273 = 256_{10} = 000000100000000_2$$
$$c_3 = 13605^2 \bmod 25273 = 21846_{10} = 101010101010110_2$$
$$c_2 = 9537^2 \bmod 25273 = 22115_{10} = 101011001100011_2$$
$$c_1 = 5199^2 \bmod 25273 = 12764_{10} = 011000111011100_2$$
    - Двоичная строка: 100000000101010101010110101011001100011011000111011100
    - ASCII-коды: (0x20, 0x15, 0x55, 0xAB, 0x31, 0xB1, 0xDC)
    - Символы: §Ул1



## Криптосистема Эль-Гамала



## Общее описание

- Основана на задаче дискретного логарифмирования в мультипликативной группе конечного поля  $F_p^*$  или группе точек эллиптической кривой  $E_{a,b}(F_p)$ .
- Использует параметры домена, общие для некоторой группы пользователей и не держащиеся в секрете.
- Параметры домена для криптосистемы Эль-Гамала над  $F_p^*$ :
  - большое простое число  $p$ ;
  - число  $g \in F_p^*$ .
- Параметры домена для криптосистемы Эль-Гамала над  $E_{a,b}(F_p)$ :
  - параметры эллиптической кривой  $a, b, p$ ;
  - точка  $G \in E_{a,b}(F_p)$ .



## Алгоритм генерации ключей

1. Алиса выбирает случайное число  $x$  в интервале  $1 < x < p - 1$ .
2. Алиса вычисляет  $h = g^x \pmod{p}$ .
3. Открытый ключ Алисы есть  $h$ , закрытый ключ Алисы есть  $x$ .



## Алгоритм зашифрования

1. Боб получает аутентичную копию открытого ключа Алисы — число  $h$ .
2. Боб представляет сообщение в виде числа  $m$  в интервале  $1 < m < p - 1$ .
3. Боб выбирает сеансовый ключ  $k$  в интервале  $1 < k < p - 1$ .
4. Боб вычисляет два значения:
  - $C_1 = g^k \pmod{p}$ ;
  - $C_2 = m \cdot h^k \pmod{p}$ .
5. Боб отправляет пару  $(C_1, C_2)$  Алисе.



## Алгоритм расшифрования

1. Алиса получает шифртекст — пару  $(C_1, C_2)$ .
2. Алиса, используя свой секретный ключ, осуществляет расшифрование по следующей формуле:

$$\frac{C_2}{(C_1)^x} = \frac{m \cdot h^k}{(g^k)^x} = \frac{m \cdot (g^x)^k}{(g^k)^x} = m$$



## Пример шифрования

- **Выбор параметров домена:**
  - $p = 9973$ ,
  - $g = 5, O(5) = 9972$ .
- **Генерация ключей:**
  - Закрытый ключ  $x = 3157$ ,
  - Открытый ключ  $h = 5^{3157} = 1808$ .



## Пример шифрования

- **Зашифрование:**

1. Открытый ключ  $h = 1808$ .

2. Открытый текст:

- Символы: *CRYPTO*
- ASCII-коды: (0x43, 0x52, 0x59, 0x50, 0x54, 0x4F)
- Двоичная строка: 010000110101001001011001010100000101010001001111
- Длина блока:  $\lfloor \log_2 p \rfloor = \lfloor \log_2 9973 \rfloor = 13$
- Блоки:  
 $m_4 = 0000010000110_2 = 134_{10}$   
 $m_3 = 1010010010110_2 = 5270_{10}$   
 $m_2 = 0101010000010_2 = 2690_{10}$   
 $m_1 = 1010001001111_2 = 5199_{10}$



## Пример шифрования

- **Зашифрование:**

### 3. Шифртекст:

- Длина блока:  $\lceil \log_2 9973 \rceil + 1 = 14$
- Сеансовый ключ:  $k = 47$
- Блоки:
  - $C^1 = 5^{47} \bmod 9973 = 5065_{10} = 000001001111001001_2$
  - $C_4^2 = 134 \cdot 1808^{47} \bmod 9973 = 8702_{10} = 000010000111111110_2$
  - $C_3^2 = 5270 \cdot 1808^{47} \bmod 9973 = 8512_{10} = 000010000101000000_2$
  - $C_2^2 = 2690 \cdot 1808^{47} \bmod 9973 = 4553_{10} = 000001000111001001_2$
  - $C_1^2 = 5199 \cdot 1808^{47} \bmod 9973 = 6134_{10} = 000001011111110110_2$
- ASCII-коды: (0x21, 0xFE, 0x08, 0x50, 0x01, 0x1C, 0x90, 0x5F, 0xD8, 0x13, 0xC9)
- Символы: !■□P□LP\_≠!!⌈



## Тесты на простоту





## Генерация простых чисел

- Генерация больших простых чисел является распространенной операцией в криптографии с открытым ключом.
- Известны алгоритмы, позволяющие проверить число  $n \in \mathbb{N}$  на простоту со сколь угодно малой вероятностью ошибки, и называемые **тестами на простоту**:
  - тест Ферма;
  - тест Миллера-Рабина;
  - и др.
- Тест Ферма основан на многократном возведении случайных чисел из  $\mathbb{Z}_n$  в степень  $n - 1$ . Если результат отличен от 1, то  $n$  — составное число.
- Существуют составные числа, называемые числами Кармайкла, для которых тест Ферма демонстрирует недостаточную эффективность. Числа Кармайкла обладают свойством: если  $\text{НОД}(a, n) = 1$ , то  $a^{n-1} \equiv 1 \pmod{n}$ .
- Тест Миллера-Рабина включает дополнительную проверку того, что у сравнения  $x^2 \equiv 1 \pmod{n}$  нет нетривиальных решений. Наличие таких решений является признаком составного модуля.



## Тест Ферма

**Вход:** нечетное число  $n$ , число итераций  $k$ .

**Выход:** ответ на вопрос «является ли  $n$  простым».

Шаг 1. Для  $i = \overline{1, k}$  выполнить следующее:

Шаг 1.1. Выбрать случайное число  $a$  из интервала  $[2, \dots, n - 1]$ .

Шаг 1.2. Вычислить  $r = a^{n-1} \pmod{n}$ .

Шаг 1.3. Если  $r \neq 1$ , то возврат « $n$  — составное».

Шаг 2. Возврат « $n$  — предположительно простое».



## Тест Миллера-Рабина

**Вход:** нечетное число  $n$ , число итераций  $k$ .

**Выход:** ответ на вопрос «является ли  $n$  простым».

Шаг 1. Представить  $n - 1 = 2^s \cdot t$ , где  $t$  — нечетное число.

Шаг 2. Для  $i = \overline{1, k}$  выполнить следующее:

Шаг 2.1. Выбрать случайное число  $a$  из интервала  $[2, \dots, n - 1]$ .

Шаг 2.2. Вычислить  $r = a^t \pmod{n}$ .

Шаг 2.3. Если  $r \neq 1$ , то выполнить следующее для  $j = \overline{1, s}$ :

Шаг 2.3.1. Если  $r = n - 1$ , то прервать цикл и перейти к шагу 2.1.

Шаг 2.3.2. Если  $j = s - 1$ , то возврат « $n$  — составное».

Шаг 2.3.2. Вычислить  $r = r^2 \pmod{n}$ .

Шаг 3. Возврат « $n$  — предположительно простое».



Московский институт электроники и  
математики им. А.Н. Тихонова

Кафедра информационной  
безопасности киберфизических  
систем

Криптографические методы  
защиты информации

# Спасибо за внимание!

**Евсютин Олег Олегович**

Заведующий кафедрой информационной безопасности киберфизических систем  
Канд. техн. наук, доцент

+7 923 403 09 21

oevsyutin@hse.ru