Механико-математический факультет
Кафедра диф. уравнений и системного анализа

# Криптосистемы с открытым ключом

Чергинец Дмитрий Николаевич

# Криптография

*Криптография* — область знаний, занимающаяся разработкой методов преобразования информации с целью обеспечения ее конфиденциальности, целостности и аутентификации.

# Криптосистема

Пусть $A$ и $B$ — конечные множества, которые будем называть алфавитами. Защищаемую информацию, состоящую из конечного объединения элементов множества $A$, будем называть *открытым текстом*. Преобразованный открытый текст, состоящий из конечного объединения элементов множества $B$, называется *шифрованным текстом* или *шифротекстом*. Через $X$ и $Y$ обозначим множества открытых текстов и шифрованных текстов соответственно. $A$ и $B$ назовём алфавитом открытого текста и шифрованного текста соответственно.

# Симметричная криптосистема

- Функцию $E_k : X \to Y$, где $k$ — параметр функции, называемый *ключом* и принадлежащий множеству ключей $K$, будем называть *функцией шифрования* (encipher [in'saife]).
- Функция $D_k : Y \to X$ называется *функцией расшифрования* (дешифрования).
- *Шифром* или *криптосистемой* называется набор $(A, B, X, Y, K, E_k, D_k)$, удовлетворяющий требованию $D_k(E_k(x)) = x$ для каждого $x \in X$ и $k \in K$.

# DES, AES

- В 1974 году Национальное бюро стандартов США объявило конкурс на стандарт шифрования. Данный конкурс выиграл симметричный шифр DES, разработанный компанией IBM.
- В 1997 году объявлен конкурс на новый шифр.
- В 2002 году на смену DES пришел AES (Advanced Encryption Standard), также известный как Rijndael (Рейндал).

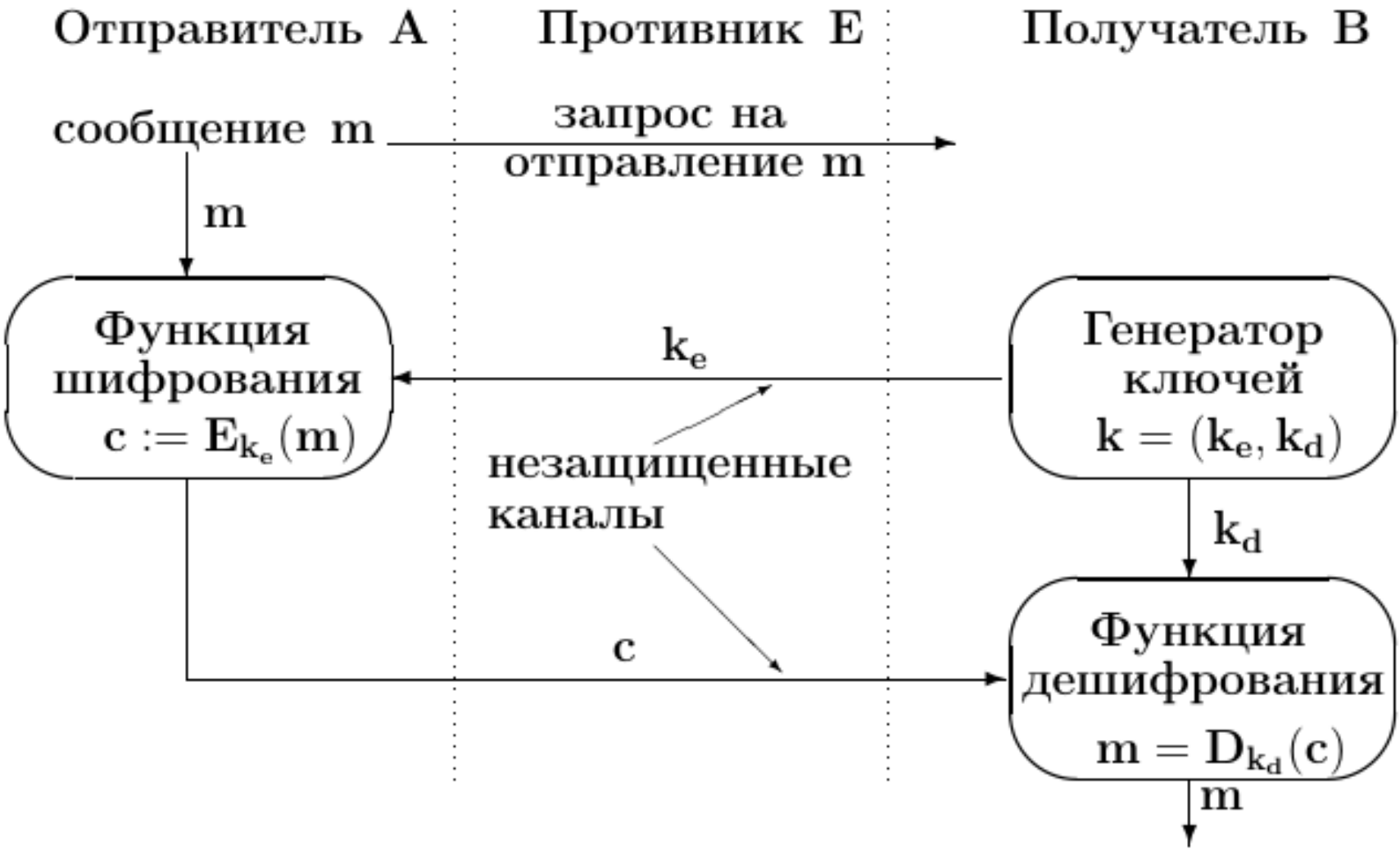
# Парадокс симметричных криптосистем

Криптосистема, в которой для шифрования и расшифрования используется один и тот же (секретный) ключ, называется симметричной. Для того чтобы два человека могли обменяться секретной информацией, ее нужно зашифровать при помощи секретного ключа, но теперь возникает проблема передачи секретного ключа для расшифрования, в этом заключается парадокс симметричных криптосистем.

# Шифры с открытым ключом

Основы криптографии с открытым ключом заложены Уитфильдом Диффи и Мартином Хеллманом в статье «Новые направления в криптографии», опубликованной в 1976 году. Шифры, в которых для шифрования используется один ключ, а для расшифрования — другой, называются *асимметричными* или *криптосистемами с открытым ключом*.

# Шифры с открытым ключом



# Шифры с открытым ключом

Криптосистемой с открытым ключом называется система

$$(X, Y, (k_e, k_d) \in K, E_{k_e}, D_{k_e,k_d}),$$

где алгоритмы шифрования

$$E_{k_e} : X \to Y$$

и дешифрования

$$D_{k_d} : Y \to X$$

являются открытыми, шифрованный текст $c$ и открытый ключ $k_e$ могут передаваться по незащищенному каналу, секретный ключ $k_d$ является, естественно, секретным.

# Шифры с открытым ключом

У. Диффи и М. Хеллман сформулировали основные требования, предъявляемые к криптосистемам с открытым ключом.

1. Для каждой пары ключей $(k_e, k_d) \in K$ и каждого открытого текста $m \in M$ функции $E$ и $D$ легко вычислимы.
2. Получение каждой пары ключей $(k_e, k_d) \in K$ является легко вычислимым.
3. Противник, зная открытый ключ $k_e$ и шифротекст $c$, при попытке вычислить открытый текст $m$ наталкивается на непреодолимую вычислительную проблему.

# Односторонние функции

## Definition

~~Инъективное~~ отображение $f : X \to Y$ называется односторонней (однонаправленной) функцией (one-way function), если
1) существует полиномиальный алгоритм, для каждого $x \in X$ вычисляющий $f(x)$;
2) не существует полиномиального алгоритма, вычисляющего $f^{-1}(y)$ для каждого $y \in f(X)$.

# Односторонние функции с лазейкой

**Definition**

Если для односторонней функции $f$ существует дополнительная информация (лазейка) $k$, зная которую $f_k^{-1}(y)$ вычисляется за полиномиальное время, то данная функция $f$ называется односторонней функцией с лазейкой (one-way trap-door function).

# Пример

Пусть $e, n \in \mathbb{N}$, $1 < e < \varphi(n)$, $\gcd(e, \varphi(n)) = 1$, будем также считать, что $\varphi(n)$ неизвестно и его достаточно трудно вычислить. Рассмотрим функцию $f : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$, действующую по правилу

$$f(x) = x^e \pmod{n}.$$

$f(x)$ вычисляется за полиномиальное время.
С другой стороны, пока не найдено полиномиального алгоритма, вычисляющего $f^{-1}(y)$, т. е. решающего уравнение $x^e \equiv b \pmod{n}$ (но и не доказано, что такого алгоритма не существует).

# Пример

Однако, если мы узнаем число $d$, удовлетворяющее условию $ed \equiv 1 \pmod{\varphi(n)}$, т.е. $ed = 1 + z\varphi(n)$ для некоторого $z \in \mathbb{Z}$, то обратную функцию можно определить формулой

$$f^{-1}(y) := y^d \pmod{n},$$

так как

$$y^d \equiv x^{ed} \equiv xx^{z\varphi(n)} \equiv x \pmod{n}.$$

Мы нашли $f^{-1}$, поэтому $f$ — инъекция, а так как она определена на конечном множестве $\mathbb{Z}_n^*$, то $f$ — биекция.

# Пример

Для того чтобы найти $d$ при помощи расширенного алгоритма Евклида, необходимо знать $\varphi(n)$.
Таким образом, если считать, что не существует полиномиального алгоритма, решающего уравнение $x^e \equiv b \pmod{n}$, то функция $f$ является односторонней функцией с лазейкой.

# Класс *NP*

## Definition

**Классом** $P$ называют множество задач, для которых существует детерминированный полиномиальный алгоритм решения.

**Классом** $NP$ называют множество задач, решение которых можно проверить за полиномиальное время. То есть существует полиномиальное доказательство: для каждого предъявленного решения существуют некие данные, при помощи которых за полиномиальное время можно доказать, что предъявленное решение действительно является решением.

# Факторизация чисел $\in NP$

## Example

Задача разложения числа $n$ на множители принадлежит классу $NP$, Так как предъявленное решение задачи

$$p_1, p_2, \ldots, p_k$$

можно проверить за полиномиальное время, перемножив числа $p_1, p_2, \ldots, p_k$ и сравнив результат с $n$. Дополнительных данных для доказательства в данном случае не требуется.

# Проблема $P = NP$?

- Очевидно, что $P \subset NP$.
- Верно ли обратное включение $NP \subset P$ ?
- Данная задача является одной из семи Задач тысячелетия.
- Проблему $P = NP$ можно переформулировать таким образом: если положительный ответ на какой-то вопрос можно быстро проверить (за полиномиальное время), то правда ли, что ответ на этот вопрос можно быстро найти (за полиномиальное время и используя полиномиальную память)?

# Доказательство существования односторонних функций

На данный момент не доказано существование ни одной односторонней функции.

**Если $P = NP$, то односторонних функций не существует**

Действительно, для $y \in Y$ существует полиномиальный алгоритм проверки того, что $f(x) = y$.
Если $P = NP$, то должен существовать и полиномиальный алгоритм, вычисляющий $f^{-1}(y)$.

# Криптосистема RSA

Название криптосистемы RSA образовано заглавными буквами фамилий ее авторов Рональда Ривеста, Ади Шамира и Леонарда Адлемана. В августе 1977 года в колонке «Математические игры» Мартина Гарднера в журнале Scientific American появилось первое описание криптосистемы RSA. Читателям также было предложено дешифровать английскую фразу, зашифрованную описанной криптосистемой. За расшифровку была обещана награда в 100 долларов США. Фразу расшифровали лишь в 1995 году.

# Генерация ключей в RSA.

**Вход:** $L \in \mathbb{N}$.
**Выход:** $k_e$ – открытый ключ,
$k_d$ – секретный ключ.

1. Выбираем два случайных простых числа $p, q$, состоящие из $L$ бит.
2. Вычисляем $n := pq$, $\varphi(n) := (p-1)(q-1)$.
3. Случайным образом выбираем натуральное число $e$, удовлетворяющее условиям $1 < e < \varphi(n)$, $\gcd(e, \varphi(n)) = 1$.
4. При помощи расширенного алгоритма Евклида вычисляем
$$d = e^{-1} \pmod{\varphi(n)}.$$
5. Выдаем ответ $k_e := (e, n)$, $k_d := d$.

# Шифрование и дешифрование в RSA

**Функция шифрования** $E_{k_e} : \mathbb{Z}_n \to \mathbb{Z}_n$ определяется правилом

$$E_{k_e}(m) := m^e \pmod{n}.$$

**Функция дешифрования** задается формулой

$$D_{k_e,k_d}(c) := c^d \pmod{n}.$$

Отметим, что все три приведенных выше алгоритма являются эффективными.

# Корректность RSA

**Theorem**

*Пусть* $n = pq$, $p \neq q$ — *простые*, $e \in \mathbb{N}$, $1 < e < \varphi(n)$, $\gcd(e, \varphi(n)) = 1$.
*Тогда для каждого* $m \in \mathbb{Z}_n$, *справедливо равенство*

$$D_{k_e,k_d}(E_{k_e}(m)) = m.$$

Исходя из определений функций $E$ и $D$ нам необходимо доказать, что

$$m^{ed} \equiv m \pmod{n}.$$

Так как $d := e^{-1} \pmod{\varphi(n)}$, то найдется такое $z \in \mathbb{Z}$, что $ed = 1 + z\varphi(n)$.

# Док-во корректности RSA

1. Пусть сперва $\gcd(m, n) = 1$.
Тогда по теореме Эйлера

$$m^{ed} = m^{1+z\varphi(n)} \equiv m(m^{\varphi(n)})^z \equiv m \pmod{n}.$$

2. Пусть $\gcd(m, n) = n$.
Тогда $m^{ed} \equiv 0 \equiv m \pmod{n}$.
3. Пусть, наконец, $\gcd(m, n) = p$.
Из того, что $\gcd(m, q) = 1$, имеем

$$(m)^{ed} \equiv 0 \equiv m \pmod{p},$$

$$m^{ed} \equiv (m)^{1+z\varphi(p)\varphi(q)} \equiv m(m^{\varphi(q)})^{z\varphi(p)} \equiv m \pmod{q}.$$

Так как $\gcd(p, q) = 1$, то по свойству сравнений
$m^{ed} \equiv m \pmod{n}$.

# Факторизация $n \Leftrightarrow \varphi(n)$

**Theorem**

*Задача разложения на множители числа $n = pq$ и задача вычисления функции Эйлера $\varphi(n)$ полиномиально эквивалентны.*

Если известно разложение числа $n$ на множители $p, q$, то значение функции Эйлера можно вычислить за одну арифметическую операцию $\varphi(n) = (p-1)(q-1)$.

# $\varphi(n) \quad \Rightarrow \quad n = pq$

Обратно, если известно $\varphi(n)$, то множители $p, q$ удовлетворяют системе

$$\begin{cases} pq = n, \\ (p-1)(q-1) = \varphi(n); \end{cases} \Rightarrow \begin{cases} pq = n, \\ p + q = n + 1 - \varphi(n). \end{cases}$$

По теореме Виета $p, q$ – корни уравнения

$$x^2 - (n + 1 - \varphi(n))x + n = 0.$$

Таким образом

$$p = \frac{n + 1 - \varphi(n) + \sqrt{(n + 1 - \varphi(n))^2 - 4n}}{2},$$

$$q = \frac{n + 1 - \varphi(n) - \sqrt{(n + 1 - \varphi(n))^2 - 4n}}{2}.$$

**Theorem**

*Задача вычисления секретного показателя $d$ полиномиально эквивалентна задаче разложения на множители модуля $n$.*

Если известно разложение $n = pq$, то можно вычислить $\varphi(n) = (p-1)(q-1)$, а затем при помощи расширенного алгоритма Евклида найти $d = e^{-1} \pmod{n}$.

# $d \quad \Rightarrow \quad n = pq$

- Пусть известны $e, d$.
- Так как $ed - 1 = z\varphi(n)$, то для $a \in \mathbb{Z}_n^*$ получим

$$a^{ed-1} \equiv (a^{\varphi(n)})^z \equiv 1^z \equiv 1 \pmod{n}.$$

- Так как $\text{НОД}(e, \varphi(n)) = 1$, $\text{НОД}(d, \varphi(n)) = 1$, то $e, d$ – нечетные.
- Представим

$$ed - 1 = 2^s t,$$

где $s, t \in \mathbb{N}$, $t$ – нечетное.

# $d \quad \Rightarrow \quad n = pq$

Найдем такое $\tilde{s} \leq s$, что

$$v := a^{t2^{\tilde{s}}} \equiv 1 \pmod{n}, \qquad u := a^{t2^{\tilde{s}-1}} \not\equiv 1 \pmod{n}.$$

Если оказалось, что

$$a^t \equiv 1 \pmod{n},$$

то возьмем другое $a$.
Мы нашли $u \neq 1$, что

$$u^2 \equiv v \equiv 1 \pmod{n}.$$

Поэтому $u^2 - 1 \equiv (u-1)(u+1) \equiv 0 \pmod{n}$.
Если $u \neq -1$, то

$$p := \gcd(u-1, n),\ q := \gcd(u+1, n).$$

# Разложение $n$, зная $e, d$

**Вход:** $n, e, d \in \mathbb{N}$.
**Выход:** $p, q$.
1. При помощи последовательного деления на 2 представляем число $ed - 1$ в виде $2^s t$, где $t$ — нечетное.
2. Выбираем случайное $a \in \mathbb{N}$, $1 < a < n - 1$.
3. Если $\gcd(a, n) > 1$, то $p := \gcd(a, n)$, $q := \frac{n}{p}$, конец алгоритма.
4. Вычисляем $u := a^t \pmod n$, $v := u^2 \pmod n$.
5. Если $u = 1$, то переходим к шагу 2.
6. Пока $v \neq 1$ вычисляем $u := v$, $v := u^2 \pmod n$.
7. Если $u \equiv -1 \pmod n$, то переходим к шагу 2, иначе вычисляем $p := \gcd(u + 1, n)$, $q := \gcd(u - 1, n)$, конец алгоритма.

# Оценка сложности алгоритма

- На шаге 6 за один цикл выполняется не более $s$ ($s < 2\log_2 n$) возведений в квадрат.
- При $n = pq$ уравнение

$$x^2 \equiv 1 \pmod{n}$$

имеет четыре корня, два из них $\pm 1$.
- На 6 шаге мы находим $u$ – один из корней данного уравнения, отличный от 1.
- С вероятностью $\frac{2}{3}$ корень $u$ отличен от $-1$ и мы получим правильный ответ.

# Алгоритм Лас-Вегаса

**Definition**

Алгоритм Лас-Вегаса это вероятностный алгоритм, который всегда выдает верный ответ (не ошибается), но иногда выдает ответ: "Не знаю". Данный алгоритм можно выполнять снова и снова, пока он не даст правильный ответ.

# Криптосистема Рабина

**Генерация ключей** $G(x)$.

1. Генерируем случайные простые числа $p$, $q$, удовлетворяющие свойствам

$$p, q > x, \qquad p \equiv q \equiv 3 \pmod 4.$$

2. Вычисляем $n := pq$.
3. Выдаем открытый ключ $k_e := n$ и секретный ключ $k_e := (p, q)$.

# Криптосистема Рабина

## Шифорвание

$$E_{k_d}(m) := m^2 \pmod{n}$$

## Дешифрование

$$x^2 \equiv c \pmod{n} \quad \Leftrightarrow \quad \begin{cases} x^2 \equiv c \pmod{p}, \\ x^2 \equiv c \pmod{q}. \end{cases}$$

## Проблема

Из четырех корней нужно выбрать один по какому-то признаку.

# Теорема из прошлой лекции

## Theorem

*Пусть $n = pq$ – число Блюма, $a$ — квадратичный вычет по модулю $n$.*
*Тогда уравнение $x^2 \equiv a \pmod{n}$ имеет четыре корня $x_1, x_2, x_3, x_4 \in \mathbb{Z}_n$, удовлетворяющие условиям:*

$$0 < x_1, x_2 \leq \frac{n-1}{2} < x_3, x_4 < n,$$

$$\left(\frac{x_1}{n}\right) = \left(\frac{x_3}{n}\right) = 1, \qquad \left(\frac{x_2}{n}\right) = \left(\frac{x_4}{n}\right) = -1.$$

$x_1 \in Q_n$, $x_2, x_3, x_4 \in \overline{Q}_n$.

# Функция шифрования

**Вход:** $k_e = n$, $m \in M$,

$$M := \{m \in \mathbb{Z} \mid 0 \leq m \leq \frac{n-1}{2} - 2\left[\sqrt{n}\right]\}.$$

**Выход:** $c = (a, JS)$ – шифр, $JS \in \{-1, 1\}$.

1. Вычисляем $m_0 := m + 2\left[\sqrt{n}\right]$.
2. Вычисляем $a := m_0^2 \pmod{n}$.
3. Вычисляем символ Якоби $JS := \left(\frac{m_0}{n}\right)$.
4. Выдаем шифртекст $c := (a, JS)$.

# Алгоритм дешифрования

**Вход:** $k_e = n$, $k_d = (p, q)$, $c = (a, JS)$.
**Выход:** $m$ – открытый текст.
1. Находим корни $x_p, x_q$ уравнений

$$x^2 \equiv a \pmod{p}, \qquad x^2 \equiv a \pmod{q}$$

$$x_p := a^{(p+1)/4} \pmod{p}, \qquad x_q := a^{(p+1)/4} \pmod{q}.$$

2. Четыре корня $x_1, x_2, x_3, x_4$ уравнения $x^2 \equiv a \pmod{n}$ находим при помощи Китайской теоремы об остатках, решая четыре системы

$$\begin{array}{llll} 1) & x \equiv x_p & \pmod{p}, & x \equiv x_q & \pmod{q}, \\ 2) & x \equiv x_p & \pmod{p}, & x \equiv -x_q & \pmod{q}, \\ 3) & x \equiv -x_p & \pmod{p}, & x \equiv x_q & \pmod{q}, \\ 4) & x \equiv -x_p & \pmod{p}, & x \equiv -x_q & \pmod{q}. \end{array} \qquad (1)$$

# Алгоритм дешифрования

3. Выбираем корень $x$ из корней $x_1, x_2, x_3, x_4$, который удовлетворяет двум условиям:
   1) неравенству $0 < x \leq \frac{n-1}{2}$;
   2) символ Якоби $\left(\frac{x}{n}\right) = JS$.
4. Выдаем ответ $m = x - 2[\sqrt{n}]$.

# Криптостойкость Рабина

Допустим, что у нас имеется алгоритм, который находит четыре корня $x_1, x_2, x_3, x_4$ уравнения

$$x^2 \equiv a \pmod{p}.$$

Тогда вычисляя следующие наибольшие общие делители

$$\gcd(x_1 - x_2, n) = p, \qquad \gcd(x_1 - x_3, n) = q,$$

получаем делители числа $n$.
Таким образом, криптостойкость криптосистемы Рабина эквивалентна проблеме факторизации.

# RSA vs Rabin

- Шифр Рабина быстрее шифрует.
- Атака на основе подобранного шифротекста. В криптосистеме Рабина по открытому тексту $m$ и шифру $c$ можно получить секретный ключ $k_d$ :

$$\gcd(x_1 - x_2, n) = p, \qquad \gcd(x_1 - x_3, n) = q.$$