

# Ассиметричные алгоритмы шифрования

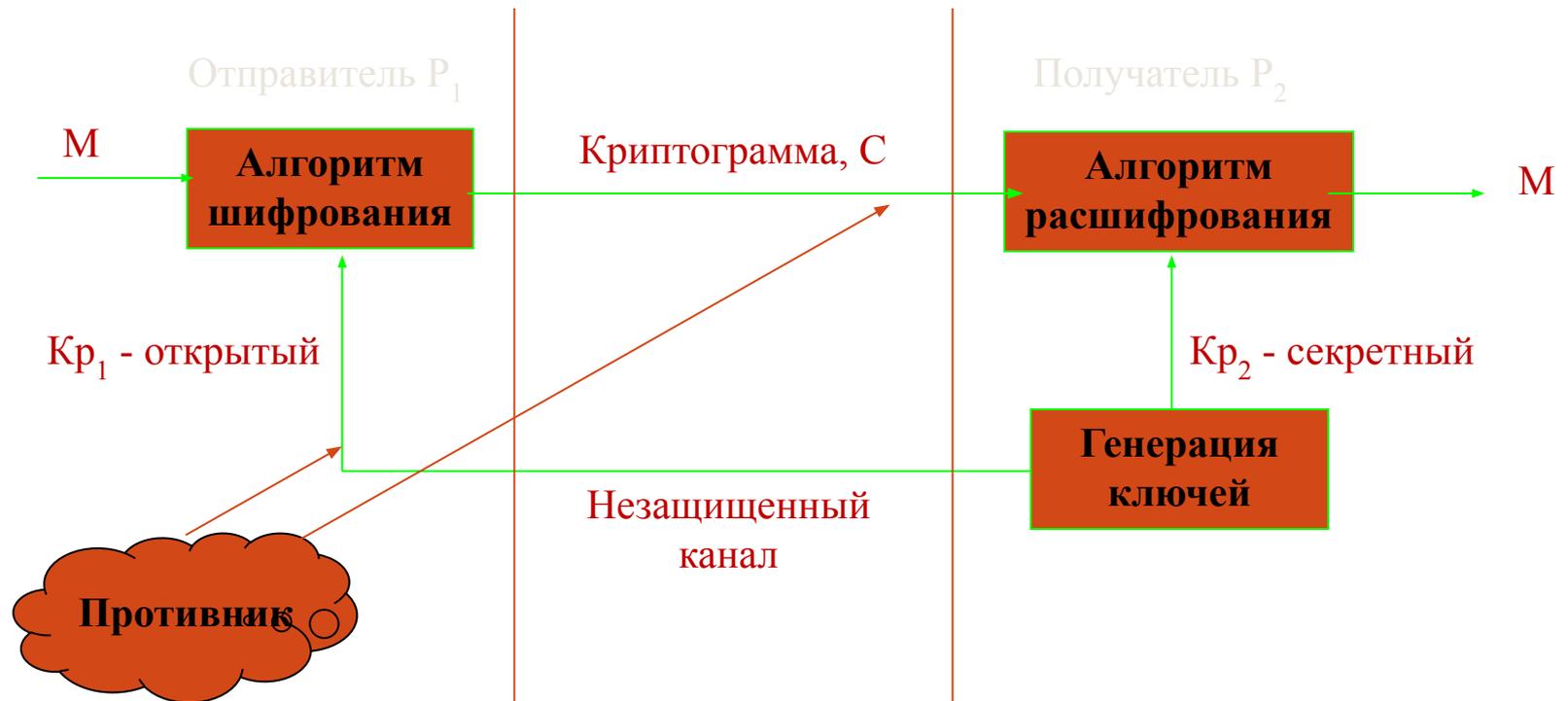
# План

1. Концепция криптосистемы с открытым ключом
2. Элементы теории чисел
3. Односторонние функции
4. Алгоритм Диффи-Хелмана
5. RSA
6. Криптоалгоритмы на основе эллиптических кривых
7. Алгоритм Эль-Гамала (El Gamal)

# 1 Концепция криптосистемы с открытым ключом

Ключевой обмен,  
Электронно-цифровая подпись  
Аутентификация

# Обобщенная схема асимметричной криптосистемы с открытым ключом



## 2 Элементы теории чисел

Под простым числом понимают такое число, которое делится только на 1 и на само себя. Взаимно простыми числами называют такие числа, которые не имеют ни одного общего делителя, кроме 1. Под результатом операции  $i \bmod j$  понимают остаток от целочисленного деления  $i$  на  $j$ .

**Теорема 2.3 (основная теорема арифметики).** *Любое целое положительное число может быть представлено в виде произведения простых чисел, причем единственным образом.*

**Пример 2.4.**  $27 = 3 \cdot 3 \cdot 3$ ,  $33 = 3 \cdot 11$ . □

**Определение 2.3.** Два числа называются *взаимно простыми*, если они не имеют ни одного общего делителя кроме единицы.

**Пример 2.5.** Числа 27 и 28 взаимно просты (у них нет общих делителей кроме единицы), числа 27 и 33 — нет (у них есть общий делитель 3). □

**Определение 2.4 (функция Эйлера).** Пусть дано целое число  $N \geq 1$ . Значение функции Эйлера  $\varphi(N)$  равно количеству чисел в ряду  $1, 2, 3, \dots, N - 1$ , взаимно простых с  $N$ .

$$\varphi(10) = ?$$

1, 2, 3, 4, 5, 6, 7, 8, 9,

$$\varphi(10) = 4$$

$$\varphi(12) = ?$$

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

$$\varphi(12) = 4$$

**Утверждение 2.4.** Если  $p$  — простое число, то  $\varphi(p) = p - 1$ .

**Утверждение 2.5.** Пусть  $p$  и  $q$  — два различных простых числа ( $p \neq q$ ). Тогда  $\varphi(pq) = (p - 1)(q - 1)$ .

**Теорема 2.6 (Ферма).** Пусть  $p$  — простое число и  $0 < a < p$ .  
Тогда

$$a^{p-1} \bmod p = 1.$$

**Пример 2.7.**  $p = 13, a = 2$ ;

$$2^{12} \bmod 13 = (2^2)^2 \cdot \left( (2^2)^2 \right)^2 \bmod 13 = 3 \cdot 9 \bmod 13 = 1,$$

$$10^{10} \bmod 11 = 10^2 \cdot \left( (10^2)^2 \right)^2 \bmod 11 = 1 \cdot 1 = 1. \quad \square$$

**Теорема 2.7 (Эйлер).** Пусть  $a$  и  $b$  — взаимно простые числа. Тогда

$$a^{\varphi(b)} \bmod b = 1.$$

**Теорема 2.8.** Если  $p$  и  $q$  — простые числа,  $p \neq q$  и  $k$  — произвольное целое число, то

$$a^{k\varphi(pq)+1} \bmod (pq) = a. \quad (2.12)$$

**Пример 2.9.** Возьмем  $p = 5$ ,  $q = 7$ . Тогда  $pq = 35$ , а функция Эйлера —  $\varphi(35) = 4 \cdot 6 = 24$ . Рассмотрим случай  $k = 2$ , т.е. будем возводить числа в степень  $2 \cdot 24 + 1 = 49$ . Получим

$$9^{49} \bmod 35 = 9, \quad 23^{49} \bmod 35 = 23.$$

**Определение 2.5.** Пусть  $a$  и  $b$  — два целых положительных числа. Наибольший общий делитель чисел  $a$  и  $b$  есть наибольшее число  $c$ , которое делит и  $a$  и  $b$ :

$$c = \gcd(a, b).$$

### Алгоритм 2.1. АЛГОРИТМ ЕВКЛИДА

ВХОД: Положительные целые числа  $a, b$ ,  $a \geq b$ .

ВЫХОД: Наибольший общий делитель  $\gcd(a, b)$ .

1. WHILE  $b \neq 0$  DO
2.      $r \leftarrow a \bmod b$ ,  $a \leftarrow b$ ,  $b \leftarrow r$ .
3. RETURN  $a$ .

**Теорема 2.9.** Пусть  $a$  и  $b$  — два целых положительных числа. Тогда существуют целые (не обязательно положительные) числа  $x$  и  $y$ , такие, что

$$ax + by = \gcd(a, b). \quad (2.13)$$

**Алгоритм 2.2.** ОБОБЩЕННЫЙ АЛГОРИТМ ЕВКЛИДА

ВХОД: Положительные целые числа  $a, b$ ,  $a \geq b$ .

ВЫХОД:  $\gcd(a, b)$ ,  $x, y$ , удовлетворяющие (2.13).

1.  $U \leftarrow (a, 1, 0), V \leftarrow (b, 0, 1)$ .
2. WHILE  $v_1 \neq 0$  DO
3.      $q \leftarrow u_1 \operatorname{div} v_1$ ;
4.      $T \leftarrow (u_1 \bmod v_1, u_2 - qv_2, u_3 - qv_3)$ ;
5.      $U \leftarrow V, V \leftarrow T$ .
6. RETURN  $U = (\gcd(a, b), x, y)$ .

Пример 2.12. Пусть  $a=28, b=19$ . Найдем числа  $x$  и  $y$ , удовлетворяющие (2.13).

$U$				28	1	0	
$V$	$U$			19	0	1	
$T$	$V$	$U$		9	1	-1	$q = 1$
	$T$	$V$	$U$	1	-2	3	$q = 2$
		$T$	$V$	0	19	-28	$q = 9$

**Алгоритм 2.3.** Возведение в степень (справа-налево)

**ВХОД:** Целые числа  $a$ ,  $x = (x_t x_{t-1} \dots x_0)_2$ ,  $p$ .

**ВЫХОД:** Число  $y = a^x \bmod p$ .

1.  $y \leftarrow 1$ ,  $s \leftarrow a$ .
2. FOR  $i = 0, 1, \dots, t$  DO
3.     IF  $x_i = 1$  THEN  $y \leftarrow y \cdot s \bmod p$ ;
4.      $s \leftarrow s \cdot s \bmod p$ .
5. RETURN  $y$ .

$i$ :	0	1	2	3	4	5	6
$x_i$ :	0	0	1	0	0	1	1
$y$ :	1	1	$a^4$	$a^4$	$a^4$	$a^{36}$	$a^{100}$
$s$ :	$a^2$	$a^4$	$a^8$	$a^{16}$	$a^{32}$	$a^{64}$	$a^{128}$

**Алгоритм 2.4.** Возведение в степень (слева-направо)

**ВХОД:** Целые числа  $a$ ,  $x = (x_t x_{t-1} \dots x_0)_2$ ,  $p$ .

**ВЫХОД:** Число  $y = a^x \bmod p$ .

1.  $y \leftarrow 1$ .
2. FOR  $i = t, t-1, \dots, 0$  DO
3.      $y \leftarrow y \cdot y \bmod p$ ;
4.     IF  $x_i = 1$  THEN  $y \leftarrow y \cdot a \bmod p$ .
5. RETURN  $y$ .

# 3 Односторонние (однонаправленные) функции

Односторонние (однонаправленные) функции обладают следующим свойством:

- Если известно  $X$ , то вычислить  $f(X)$  относительно просто
- Если известно  $y=f(X)$ , то для вычисления  $X$  нет простого (эффективного) пути.

$$L_N(\alpha, \beta) = \exp\left((\beta + o(1)) (\ln N)^\alpha (\ln \ln N)^{1-\alpha}\right)$$

---

– ПДЛ — задача дискретного логарифмирования, о которой мы говорили выше. А именно, по данным  $A, B \in G$  найти такой  $x$ , что  $B = A^x$ .

– ЗДХ — задача Диффи-Хеллмана, которая состоит в следующем: даны элементы  $A \in G, B = A^x$  и  $C = A^y$ ; требуется вычислить  $D = A^{xy}$ .

– ПВДХ — проблема выбора Диффи-Хеллмана. Дано:

$$A \in G, \quad B = A^x, \quad C = A^y \quad \text{и} \quad D = A^z;$$

требуется определить, является ли  $z$  произведением  $z = x \cdot y$ .

# 4 Алгоритм Диффи-Хелмана

Авторы публикации [DifH76] предложили использовать для шифрования одностороннюю функцию-ловушку. *Односторонняя функция* — это функция  $f : A \rightarrow B$  со следующими свойствами:

F1)  $f(a)$  легко вычисляется для любого  $a \in A$ .

F2) вычислительно невозможно найти  $f^{-1}(b)$  почти для всех  $b \in B$ .

*Односторонняя функция-ловушка* — это односторонняя функция с еще одним свойством

F3)  $f^{-1}(b)$ ,  $b \in B$ , легко вычисляется, если известна некоторая дополнительная информация.

Свойство F1 делает такую функцию практичной в использовании, тогда как свойство F2 обеспечивает безопасность при использовании  $f$  в целях шифрования. Свойство F3 делает возможным дешифрование сообщений получателем.

# Алгоритм Диффи – Хеллмана (Diffie - Hellman)

Отправитель  $P_1$

Получатель  $P_2$



2. Случайное число  $X$ ,  
вычисляет  $A = a^x \pmod{n}$



3. Случайное число  $Y$ ,  
вычисляет  $B = a^y \pmod{n}$

4. Вычисление ключа  
 $K_{p_1} = B^x \pmod{n}$



5. Вычисление ключа  
 $K_{p_2} = A^y \pmod{n}$

Пример

$n = 5, a = 7, x = 3, y =$

$$A = 7^3 \pmod{5} = 343 \pmod{5} = 3$$

$$K_{p_1} = 4^3 \pmod{5} = 64 \pmod{5} = 4$$

$$B = 7^2 \pmod{5} = 49 \pmod{5} = 4$$

$$K_{p_2} = 3^2 \pmod{5} = 4$$

# 5 RSA

современной криптографии – *односторонней функцией с «лазейкой»* (trapdoor function).

Эта система базируется на следующих двух фактах из теории чисел:

- 1) задача проверки числа на простоту является сравнительно легкой;
- 2) задача разложения чисел вида  $n = pq$  ( $p$  и  $q$  — простые числа) на множители является очень трудной, если мы знаем только  $n$ , а  $p$  и  $q$  — большие числа (это так называемая задача факторизации).

# АСИММЕТРИЧНЫЕ КРИПТОАЛГОРИТМЫ

## Алгоритм RSA

Чтобы использовать алгоритм RSA, надо сначала сгенерировать открытый и секретный ключи, выполнив следующие шаги.

1. Выбрать два очень больших простых числа  $p$  и  $q$ .
2. Определить  $n$  как результат умножения  $p$  на  $q$  ( $n = pq$ ).
3. Выбрать большое случайное число  $d$ . Оно должно быть взаимно простым с результатом умножения  $(p - 1)(q - 1)$ .
4. Определить такое число  $e$ , для которого является истинным следующее соотношение:  $ed \bmod((p - 1)(q - 1)) = 1$ .
5. Назвать открытым ключом числа  $e$  и  $n$ , а секретным ключом — числа  $d$  и  $n$ .

Далее, чтобы зашифровать данные по известному ключу  $\{e, n\}$ , необходимо разбить шифруемый текст на блоки, каждый из которых может быть представлен в виде числа  $M(i) = 0, 1, \dots, n - 1$ ; зашифровать текст, рассматриваемый как последовательность чисел  $M(i)$  по формуле  $C(i) = M(i)^e \bmod(n)$ .

Чтобы расшифровать эти данные, используя секретный ключ  $\{d, n\}$ , необходимо выполнить следующие вычисления:  $M(i) = C(i)^d \bmod(n)$ . В результате будет получено множество чисел  $M(i)$ , которое представляет собой исходный текст.

# АСИММЕТРИЧНЫЕ КРИПТОАЛГОРИТМЫ

Приведем простой пример использования метода RSA для шифрования сообщения «ЕДА». Для простоты будем использовать очень маленькие числа (на практике используются намного большие числа).

1. Выберем  $p = 3$  и  $q = 11$ .
2. Определим  $n = 3 \cdot 11 = 33$ .
3. Найдем  $(p-1)(q-1) = 20$ . Следовательно, в качестве  $d$  выберем любое число, которое является взаимно простым с 20, например  $d = 3$ .
4. Выберем число  $e$ . В качестве такого числа может быть взято любое число, для которого удовлетворяется соотношение  $e \cdot 3 \bmod(20) = 1$ , например  $e = 7$ .
5. Представим шифруемое сообщение как последовательность целых чисел в диапазоне  $0 \dots 32$ . Пусть буква Е изображается числом 6, буква Д — числом 5, а буква А — числом 1. Тогда сообщение можно представить в виде последовательности чисел 651. Зашифруем сообщение, используя ключ  $\{7, 33\}$ :

$$C_1 = 6^7 \bmod(33) = 279936 \bmod(33) = 30;$$

$$C_2 = 5^7 \bmod(33) = 78125 \bmod(33) = 14;$$

$$C_3 = 1^7 \bmod(33) = 1 \bmod(33) = 1.$$

Попытаемся расшифровать сообщение  $\{30, 14, 1\}$ , полученное в результате зашифрования по известному ключу, на основе секретного ключа  $\{3, 33\}$ :

$$M_1 = 30^3 \bmod(33) = 27000 \bmod(33) = 6;$$

$$M_2 = 14^3 \bmod(33) = 2744 \bmod(33) = 5;$$

$$M_3 = 1^3 \bmod(33) = 1 \bmod(33) = 1.$$

Таким образом, в результате расшифрования сообщения получено исходное сообщение «ЕДА».

# Алгоритм RSA (Rivest-Shamir-Adleman)

## Генерация ключей

- Получатель
1.  $P, Q$  - простые,  $N = P \cdot Q$
  2.  $\varphi(N) = (P-1) \cdot (Q-1)$ ,  $\varphi(N)$  - функция Эйлера

Выбор открытого ключа  $Y$ :

$$1 < Y \leq \varphi(N), \text{НОД}(Y, \varphi(N)) = 1$$

Вычисление секретного ключа  $X$ :

$$X \cdot Y \equiv 1 \pmod{\varphi(N)}$$

$(N, Y) \rightarrow$  отправителю

- Отправитель
- шифрование  $M$  ( $M_i = 0, 1, 2, \dots, N-1$ )
3.  $C_i = M_i^Y \pmod{N}$

- Получатель
- расшифрование  $C$  ( $C_1, C_2, \dots, C_i, \dots$ )
4.  $M_i = C_i^X \pmod{N}$

*Пример*

## Генерация ключей

1.  $P = 3, Q = 11, N = P \cdot Q = 33$
2.  $\varphi(N) = (P-1) \cdot (Q-1) = 2 \cdot 10 = 20$   
 $Y = 7, \text{НОД}(Y, \varphi(N)) = 1$   
 $X \cdot Y = 1 \pmod{20}, 7 \cdot 3 = 1 \pmod{20}, X = 3$

Сообщение:  $M_1 M_2 \rightarrow 32; M_1 = 3 < 33, M_2 = 2 < 33$

## Шифрование

- $C_i = M_i^Y \pmod{N}$
3.  $C_1 = 3^7 \pmod{33} = 2187 \pmod{33} = 9$   
 $C_2 = 2^7 \pmod{33} = 128 \pmod{33} = 29$

Шифротекст 9,29

## Расшифрование

- $M_i = C_i^X \pmod{N}$
4.  $M_1 = 9^3 \pmod{33} = 729 \pmod{33} = 3$   
 $M_2 = 29^3 \pmod{33} = 24389 \pmod{33} = 2$

Восстановленный текст 3,2

# 6 КРИПТОАЛГОРИТМЫ НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

## Общие положения

### Что такое Эллиптическая кривая?

В общем случае эллиптическая кривая описывается математическим уравнением вида:

$$y^2 + axy + by = x^3 + cx^2 + dx + e ,$$

где  $a$ ,  $b$ ,  $c$ ,  $d$  и  $e$  являются действительными числами, удовлетворяющими некоторым простым условиям.

# КРИПТОАЛГОРИТМЫ НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

## Общие положения

В случае криптографии с использованием эллиптических кривых приходится иметь дело с редуцированной формой эллиптической кривой, которая определяется над конечным полем.

Особый интерес для криптографии представляет объект, называемый эллиптической группой по модулю  $p$ , где  $p$  является простым числом.

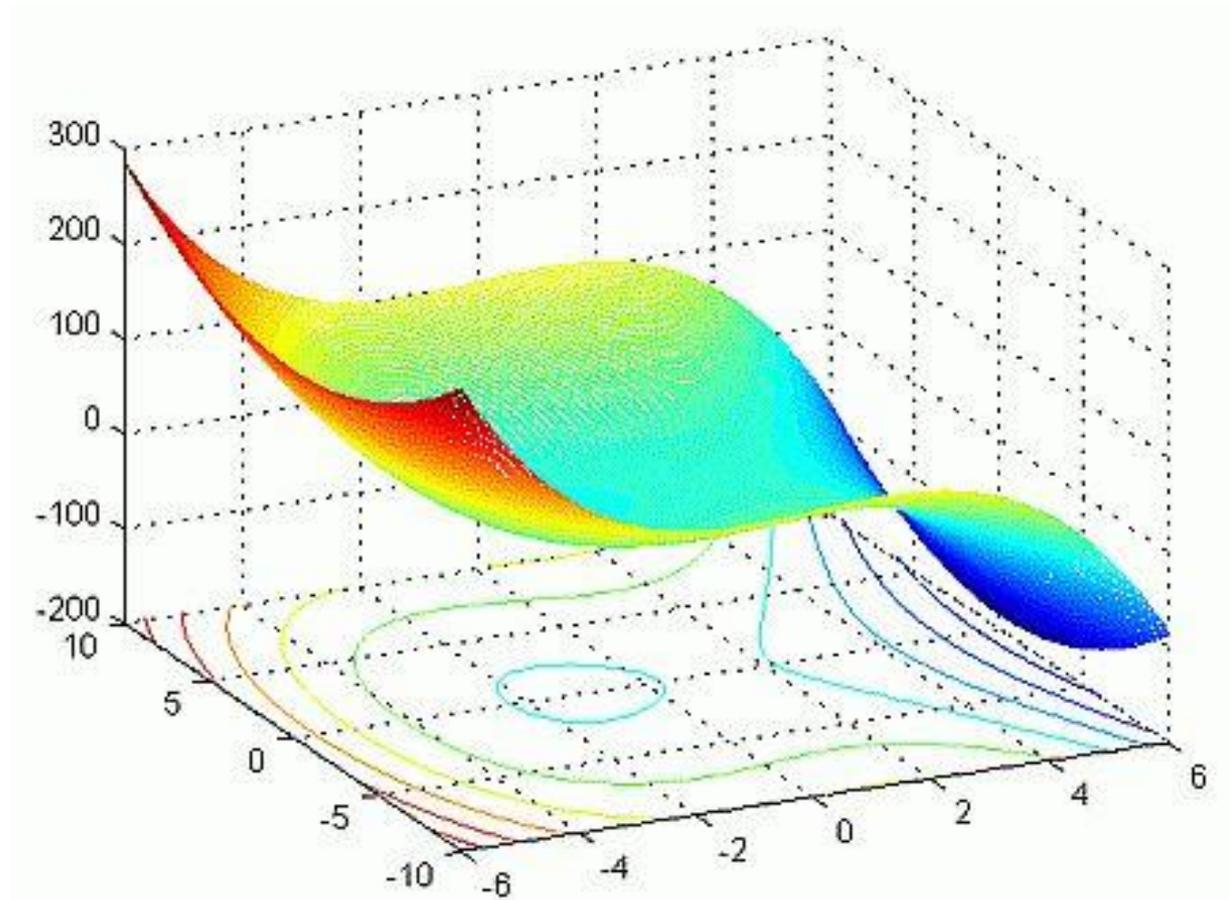
Эллиптическая кривая над конечным полем задаётся уравнением

$$y^2 = x^3 + ax + b \pmod{p}.$$

# КРИПТОАЛГОРИТМЫ НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

## Общие положения

Пространственный график эллиптической кривой  $y^2 = x^3 - 5x + 1$



# КРИПТОАЛГОРИТМЫ НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

## Общие положения

Криптоалгоритм основан на “**Проблеме Дискретного Логарифма Эллиптической Кривой**”  
(**Elliptic Curve Discrete Logarithm Problem – ECDLP**):

“Даны “базовая точка”  $P$  и расположенная на кривой точка  $kP$ ; найти значение  $k$ ”.

Для эллиптических кривых и базовых точек решение таких уравнений представляет весьма и весьма большую трудность!

С точки же зрения криптографии имеется возможность определить новую криптографическую систему на основе эллиптических кривых.

Любая стандартная система, основанная на проблеме дискретного логарифма, аналогична системе основанной на **ECDLP**. Например, Эллиптическая Кривая DSA (ECDSA) уже стандартизирована (ANSI X9.62 – Ref. 4) и на ее основе может быть реализован протокол открытого обмена ключами Diffie-Hellman.

# КРИПТОАЛГОРИТМЫ НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

## Общие положения

размеры параметров эллиптических систем и RSA, обеспечивающих одинаковую стойкость шифра

Система на основе эллиптической кривой (базовая точка P)	RSA (длина модуля n)
106 бит	512бит
132бит	768бит
160бит	1024бит
224бит	2048бит

# КРИПТОАЛГОРИТМЫ НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

## Общие положения

Использование эллиптических кривых позволяет строить высоко защищенные системы с ключами явно меньших размеров по сравнению с аналогичными “традиционными” системами типа RSA или DSA.

В частности такие системы менее требовательны к вычислительной мощности и объему памяти оборудования и потому хорошо подходят, например, для смарт-карт или портативных телефонов.

Разумеется существуют и проблемы, которые ограничивают повсеместное распространение криптографических систем на основе эллиптических кривых.

# КРИПТОАЛГОРИТМЫ НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

## Некоторые проблемы и трудности в использовании систем на основе Эллиптической Кривой

**1. Истинная сложность ECDLP ещё не осознана полностью.** Исследования показывают, что некоторые использовавшиеся для отработки алгоритмов шифрования эллиптические кривые, фактически не подходят для таких операций. Такие кривые называются «**аномальными**».

**2. Чрезвычайно трудно создать подходящую кривую и точку  $P$ .** Координаты базовой точки  $P$  должны иметь достаточно большое значение, чтобы гарантировать трудность взлома ECDLP.

**3. Относительно медленная проверка цифровой подписи.**

**4. Проблема лицензирования и патентования криптосистем на основе эллиптической кривой еще не решена.** В этой области существует множество патентов, но главным образом для применения в частных случаях.

# КРИПТОАЛГОРИТМЫ НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

## Скорость обработки

Сравнительные характеристики алгоритмов RSA и ECDSA при создании и проверки подписей.

Алгоритмы выполнялись на параллельных процессорах Motorola 56303 DSP (66 МГц).

	<b>Создание подписи</b>	<b>Проверка подписи</b>
<b>RSA (1024 бита)</b>	<b>25 ms</b>	<b>&lt; 2 ms</b>
<b>ECDSA (160 бит)</b>	<b>32 ms</b>	<b>33 ms</b>
<b>RSA (2048 битов)</b>	<b>120 ms</b>	<b>5ms</b>
<b>ECDSA (216 битов)</b>	<b>68 ms</b>	<b>70 ms</b>

# КРИПТОАЛГОРИТМЫ НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

## Заключительные замечания

Криптосистемы на основе эллиптической кривой получают все большее распространение скорее **как альтернатива, а не замена** системам на основе RSA, поскольку системы на основе ECDLP имеют некоторые преимущества, особенно при использовании в устройствах с маломощными процессорами и/или маленькой памятью.

# 7 Алгоритм Эль-Гамала (El

## Gamal)

### Генерация ключей

1.  $P, G$  - простые ( $P > G$ )
2.  $X$  - секретный ключ, (случайное целое  $X < P$ )
3.  $Y$  - открытый ключ  $Y = G^X \text{ mod } P$

### Шифрование $M$

4.  $K$  - случайное целое,  $1 < K < (P-1)$ ,  $\text{НОД}(K, P-1) = 1$   
 $a = G^K \text{ mod } P$      $b = Y^K M \text{ mod } P$      $(a, b)$  - шифротекст

### Расшифрование $(a, b)$

5.  $M = (b / a^X) \text{ mod } P$

*Пример  $p=19$   $G=2$   $x=3$   $k=5$   $M=10$*

### Шифрование $M=5$

1.  $P = 11, G = 2$  ( $P > G$ )
2.  $X < P, X = 8$  - секретный ключ
3.  $Y = G^X \text{ mod } P = 2^8 \text{ mod } 11 = 256 \text{ mod } 11 = 3$   
 $Y = 3$  - открытый ключ
4.  $K = 9, \text{НОД}(K, P-1) = 1, \text{НОД}(9, 10) = 1$   
 $a = G^K \text{ mod } P = 2^9 \text{ mod } 11 = 512 \text{ mod } 11 = 6$   
 $b = Y^K M \text{ mod } P = 3^9 \cdot 5 \text{ mod } 11 = 19683 \cdot 5 \text{ mod } 11 = 9$   
 $(a, b) = (6, 9)$  - шифротекст

### Расшифрование

5.  $M = (b / a^X) \text{ mod } P = 9 / 6^8 \text{ mod } 11$   
 $6^8 M = 9 \text{ mod } 11$   
 $1679619 \cdot M = 9 \text{ mod } 11$   
 $M = 5$

# ЭЦП RSA

## Генерация ключей

1.  $P, Q$  - большие простые числа.
2. Модуль  $N = P \cdot Q$ ;  $\varphi(N) = (P-1) \cdot (Q-1)$ ,  $\varphi(N)$  - функция Эйлера
3. Открытый ключ  $E \leq \varphi(N)$ ;  $\text{НОД}(E, \varphi(N)) = 1$
4. Секретный ключ  $D < N$ ;  $E \cdot D = 1 \pmod{\varphi(N)}$

## Постановка подписи

5. Вычисление хэш-функции  $H = h(M)$ ,  $M$  - сообщение
6. Подпись  $(M, S) \rightarrow S = H^D \pmod{N}$

## Проверка подписи

7. Вычисление хэш-функции  $H' = h(M)$
8. Вычисление  $H'' = S^E \pmod{N}$
9.  $H' = H''$  ?

### *Пример*

## Генерация ключей

1.  $P = 3, Q = 11$
2.  $N = 33$ ;  $\varphi(N) = 20$
3.  $E = 7$ ,  $\text{НОД}(7, 20) = 1$
4.  $D = 3$ ,  $7 \cdot 3 = 1 \pmod{20}$

## Постановка подписи

5.  $H = 4$
6.  $S = 4^3 \pmod{33} = 31$

## Проверка подписи

7.  $H' = 4$
8.  $H'' = 31^7 \pmod{33} = 27512614111 \pmod{33} = 4$
9.  $H' = H'' = 4$  — подпись верна

# Обобщенная схема формирования ЭЦП

*Отправитель  
(постановка ЭЦП)*

канал

*Получатель  
(проверка ЭЦП)*

