ОГЛАВЛЕНИЕ

Введение 4
Постановка задачи 5
1. Общая информация о криптосистеме Эль-Гамаля 6
1.1 Алгоритм создания открытого и закрытого ключей 7
1.2. Шифрование и расшифрование 7
1.3. Дешифрование 8
1.4. Особенности криптосистемы Эль-Гамаля 8
2. Алгоритмы решения задачи дискретного логарифмирования . . 10
2.1. В произвольной мультипликативной группе 10
2.2. В кольце вычетов по простому модулю 10
2.3. Алгоритмы с экспоненциальной сложностью 11
2.4. Субэкспоненциальные алгоритмы 13
3. Американский стандарт кодирования - ASCII 15
4. Анализ DES, ГОСТ 28147-89, Crypto03, El-Gamal 17
5. Описание электронной обучающей программы "El-Gamal_Tutor" 22
5.1 Общие сведения 22
5.2. Функциональное назначение 22
5.3. Используемые технические средства 22
5.4. Описание логической структуры 23
5.5. Описание алгоритма 24
5.6. Вызов и загрузка 50
5.7. Входные и выходные данные 50
6. Описание сценария лабораторной работы 51
6.1. Постановка задачи 51
6.2. Содержание отчета о выполнении лабораторной работы . 52
Заключение 53

Список используемых источников 54

ВВЕДЕНИЕ

В настоящее время в вузах Российской Федерации базовые стандарты обучения для ряда специальностей включают в себя разделы, связанные с изучением методов и средств защиты информации. Для успешного освоения данных тем необходимо понимание принципов и знание основных элементов криптографического преобразования информации.

В Интернете можно найти десятки описаний лабораторных работ, посвященных криптографической системе Эль Гамаля [1 – 3]. К сожалению, подавляющее большинство из них содержат задания и примеры реализации схемы Эль Гамаля без учета особенностей длинной арифметики, не требуя обоснований алгоритмов и использования обучающих программ, не затрагивая вопросы криптоанализа.

Известно несколько компьютерных обучающих программ, позволяющих быстро и достаточно полно ознакомиться с алгоритмами шифрования и расшифрования данных, используемыми в традиционных симметричных и современных асимметричных криптосистемах. К сожалению, эти программы, представленные в сети Интернет, не сопровождаются исходными текстами, ограничиваются краткой справочной информацией и содержат большое число ошибок и недочетов. В связи с этим и было принято решение: разработать алгоритм и реализовать свою электронную обучающую программу для изучения криптосистемы Эль Гамаля, а также разработать сценарий лабораторной работы с использованием этой программы. Предлагаемый вариант лабораторной работы призван преодолеть указанные недостатки.

ПОСТАНОВКА ЗАДАЧИ

1. Провести анализ криптографического алгоритма Эль Гамаля.
2. Разработать сценарий выполнения лабораторной работы по изучению алгоритма Эль Гамаля.
3. Разработать и реализовать обучающую компьютерную программу "El-Gamal_Tutor".

1. ОБЩАЯ ИНФОРМАЦИЯ О КРИПТОСИСТЕМЕ ЭЛЬ-ГАМАЛЯ

Схема Эль-Гамаля (Elgamal) — криптосистема с открытым ключом, основанная на трудности вычисления дискретных логарифмов в конечном поле. Криптосистема включает в себя алгоритм шифрования и алгоритм цифровой подписи. Схема Эль-Гамаля лежит в основе бывших стандартов электронной цифровой подписи в США (DSA) и России (ГОСТ Р 34.10-94, ГОСТ Р 34.10-2001). Схема была предложена Тахером Эль-Гамалем в 1985 году. Эль-Гамаль разработал один из вариантов алгоритма Диффи-Хеллмана. Он усовершенствовал систему Диффи-Хеллмана и получил два алгоритма, которые использовались для шифрования и для обеспечения аутентификации. В отличие от RSA алгоритм Эль-Гамаля не был запатентован и, поэтому, стал более дешевой альтернативой, так как не требовалась оплата взносов за лицензию. Считается, что алгоритм попадает под действие патента Диффи-Хеллмана.

Криптографические системы с открытым ключом используют так называемые односторонние функции, которые обладают следующим свойством:

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

Под односторонностью понимается не теоретическая однонаправленность, а практическая невозможность вычислить обратное значение, используя современные вычислительные средства, за обозримый интервал времени.

В основу криптографической системы Эль-Гамаля положена сложность задачи дискретного логарифмирования в конечном поле. Для шифрования используется операция возведения в степень по модулю большого числа. Для дешифрования за разумное время необходимо уметь вычислять дискретный логарифм в конечном поле по простому модулю, что является вычислительно трудной задачей.

В криптографической системе с открытым ключом каждый участник располагает как открытым ключом (англ. public key), так и закрытым ключом (англ. private key). В криптографической системе Эль-Гамаля открытый ключ состоит из тройки чисел, а закрытый ключ состоит из одного числа. Каждый участник создаёт свой открытый и закрытый ключ самостоятельно. Закрытый ключ каждый из них держит в секрете, а открытые ключи можно сообщать кому угодно или даже публиковать их.

1.1. Алгоритм создания открытого и закрытого ключей

Ключи в схеме Эль-Гамаля генерируются следующим образом:

1. Генерируется случайное простое число p.
2. Выбирается целое число g — первообразный корень p.
3. Выбирается случайное целое число x, такое, что $1 < x < p$.
4. Вычисляется $y = g^x \ mod \ p$.
5. Открытым ключом является тройка $(p, \ g, \ y)$, закрытым ключом — число x.

1.2. Шифрование и расшифрование

Предположим, пользователь А хочет послать пользователю Б сообщение . Сообщениями являются целые числа в интервале от 0 до $p-1$. Алгоритм для шифрования:

1. Взять открытый ключ пользователя Б
2. Взять открытый текст М
3. Выбрать сессионный ключ — случайное целое число k такое, что $1 < k < p-1$
4. Зашифровать сообщение с использованием открытого ключа пользователя Б, то есть вычислить числа: $a = g^k \ mod \ p$, и $b = y^k M \ mod \ p$.

Алгоритм для расшифрования:

1. принять зашифрованное сообщение $(a,\ b)$ от пользователя А
2. Взять свой закрытый ключ M
3. Применить закрытый ключ для расшифрования сообщения: $M = b(a^x)^{-1}\ mod\ p$
4. При этом нетрудно проверить, что
$(a^x)^{-1} \equiv g^{-kx}\ (mod\ p)$, и поэтому
$b(a^x)^{-1} \equiv (y^k M)g^{-xk} \equiv (g^{xk}M)g^{-xk} \equiv M\ (mod\ p)$.

1.3. Дешифрование

Дешифрование - получение открытых данных по зашифрованным в условиях, когда алгоритм расшифрования и его секретные параметры не являются полностью известными и расшифрование не может быть выполнено обычным путем. Алгоритм для дешифрования криптосистемы Эль-Гамаля:

1. Перехватить зашифрованное сообщение $(a,\ b)$.
2. Взять открытый ключ $(p,\ g,\ y)$
3. Решить относительно x уравнение $y \equiv g^x\ (mod\ p)$
4. Расшифровать сообщение по формуле $M = b(a^x)^{-1} mod p$

Собственно, самый главный вопрос из этого алгоритма – как по данным $(p,\ g,\ y)$ найти x. Эта задача называется задачей дискретного логарифмирования [2].

1.4. Особенности криптосистемы Эль-Гамаля

- Криптосистема асимметричная (двухключевая).
- Блочная, с длиной блока открытого текста меньше или равной длине открытого (публичного) ключа.

- Длина открытого и закрытого ключей, по современным представлениям, 2048 бит или более.
- Используется лишь один метод шифрования – метод аналитических преобразований.
- Базируется на вычислительно трудной задаче дискретного логарифмирования.
- Предоставляет возможность реализации электронной подписи.

2. АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ ДИСКРЕТНОГО ЛОГАРИФМИРОВАНИЯ

2.1. В произвольной мультипликативной группе

Разрешимости и решению задачи дискретного логарифмирования в произвольной конечной абелевой группе посвящена статья J. Buchmann, M. J. Jacobson и E. Teske [8]. В алгоритме используется таблица, состоящая из $O(\sqrt{|g|})$ пар элементов, и выполняется $O(\sqrt{|g|})$ умножений. Данный алгоритм медленный и не пригоден для практического использования. Для конкретных групп существуют свои, более эффективные, алгоритмы.

2.2. В кольце вычетов по простому модулю

Рассмотрим сравнение

$$a^x \equiv b \ (mod\ p) \tag{1}$$

где p — простое, b не делится на p. Если a является образующим элементом группы $\mathbb{Z}/p\mathbb{Z}$, то сравнение (1) имеет решение при любых b . Такие числа a называются ещё первообразными корнями, и их количество равно $\phi(p) = p - 1$, где ϕ — функция Эйлера. Решение сравнения (1) можно находить по формуле:

$$x \equiv \sum_{i=1}^{p=2} (1 - a^i)^{-1} b^i \ (mod\ p) \tag{2}$$

Однако, сложность вычисления по этой формуле хуже, чем сложность полного перебора.

Следующий алгоритм [3] имеет сложность $O(\sqrt{p} \cdot \log p)$. Алгоритм

1. Присвоить $H := [\sqrt{p}] + 1$
2. Вычислить $c = a^H mod\ p$
3. Составить таблицу значений $c^u\ mod\ p$ для $1 \leq u \leq H$ и отсортировать её.

4. Составить таблицу значений $b \cdot a^v \ mod \ p$ для $0 \leq v \leq H$ и отсортировать её.

5. Найти общие элементы в таблицах. Для них $c^u \equiv b \cdot a^v \ (mod \ p)$ откуда $a^{H \cdot u - v} \equiv b \ (mod \ p)$

6. Выдать $H \cdot u - v$.

Существует также множество других алгоритмов для решения задачи дискретного логарифмирования в поле вычетов [3]. Их принято разделять на экспоненциальные и субэкспоненциальные. Полиномиального алгоритма для решения этой задачи пока не найдено.

2.3. Алгоритмы с экспоненциальной сложностью

Алгоритм Гельфонда-Шенкса (алгоритм больших и малых шагов, baby-step giant-step) был предложен независимо советским математиком Александром Гельфондом в 1962 году и Дэниэлем Шенксом в 1972 году. Относится к методам встречи посередине. Идея алгоритма состоит в выборе оптимального соотношения времени и памяти, а именно в усовершенствованном поиске показателя степени.

Пусть задана циклическая группа G порядка n, генератор группы α и некоторый элемент группы β. Задача сводится к нахождению целого числа x, для которого выполняется $\alpha^x = \beta \mod m$.

Алгоритм Гельфонда — Шенкса основан на представлении x в виде $x = i \cdot m - j$, где $m = \lfloor \sqrt{n} \rfloor + 1$, и переборе $1 \leq i \leq m$ и $0 \leq j \leq m$. Ограничение на i и j следует из того, что порядок группы не превосходит m, а значит указанные диапазоны достаточны для получения всех возможных из полуинтервала $[0; m)$. Такое представление равносильно равенству

$$\alpha^{im} = \beta \alpha^j \tag{3}$$

Алгоритм предварительно вычисляет α^{im} для разных значений i и сохраняет их в структуре данных, позволяющей эффективный поиск, а затем перебирает всевозможные значения j и проверяет, если $\beta \alpha^j$ соответствует какому-то значению i. Таким образом находятся индексы

i и j, которые удовлетворяют соотношению (3) и позволяют вычислить значение $x = i \cdot m - j$.

Алгоритму Гельфонда — Шенкса требуется $O(n)$ памяти. Возможно выбрать меньшее m на первом шаге алгоритма, но это увеличивает время работы программы до $O(n/m)$.

Рис. 1: Мартин Хеллман

Другим методом дискретного логарифмирования является алгоритм Сильвера-Полига-Хеллмана. Он работает, если известно разложение числа $p - 1 = \prod_{i=1}^{s} q_i^{\alpha_i}$ на простые множители. Сложность оценивается как $O(\sum_{i=1}^{s} \alpha_i(\log p + q_i))$. Если множители, на которые раскладывается $p - 1$, достаточно маленькие, то алгоритм чрезвычайно эффективен. Это необходимо учитывать в выборе параметров при разработке криптографических схем, основанных на вычислительной сложности дискретного логарифмирования, иначе схема будет ненадёжной.

Для применения алгоритма Сильвера-Полига-Хеллмана необходимо знать разложение $p - 1$ на множители. В общем случае задача факторизации — достаточно трудоёмкая, однако если делители числа — небольшие, то это число можно быстро разложить на множители даже методом последовательного деления. Таким образом, в тех случаях, когда эффективен алгоритм Сильвера-Полига-Хеллмана, необходимость факторизации не усложняет задачу.

Ещё одним методом дискретного логарифмирования является ρ-метод Полларда, который был предложен Джоном Поллардом в 1978 году, основные идеи алгоритма похожи на ρ-алгоритм Полларда для факторизации чисел. Условием работы ρ-метода Полларда является простота порядка группы, порождённой основанием a дискретного логарифма по модулю p.

Алгоритм имеет эвристическую оценку сложности $O(p^{\frac{1}{2}})$. По сравнению с другими методами дискретного логарифмирования ρ-метод Полларда является менее затратным как по отношению к вычислительным операциям, так и по отношению к затрачиваемой памяти. Например, при достаточно больших значениях числа p данный алгоритм является вычислительно менее сложным, чем алгоритм COS и алгоритм Адлемана. С другой стороны, условие работы алгоритма накладывает серьёзные ограничения на его использование.

2.4. Субэкспоненциальные алгоритмы

Данные алгоритмы имеют сложность, оцениваемую как $O(\exp(c(\log p \log p \log p)^d))$ арифметических операций, где c и $0 \leq d \leq 1$ — некоторые константы. Эффективность алгоритма во многом зависит от близости c к 1 и d — к 0.

Алгоритм Адлемана [9] появился в 1979 году. Это был первый субэкспоненциальный алгоритм дискретного логарифмирования. На практике он всё же недостаточно эффективен. В этом алгоритме $d = \frac{1}{2}$.

Алгоритм COS [3] был предложен в 1986 году математиками Копперсмитом (Don Coppersmith), Одлыжко (Andrew Odlyzko) и Шреппелем (Richard Schroeppel). В этом алгоритме константа $c = 1$, $d = \frac{1}{2}$. В 1991 году с помощью этого метода было проведено логарифмирование по модулю $p \approx 10^{58}$. В 1997 году Вебер [3] провел дискретное логарифмирование по модулю $p \approx 10^{85}$ с помощью некоторой версии данного алгоритма. Экспериментально показано, что при $p \leq 10^{90}$ алгоритм COS лучше решета числового поля.

Дискретное логарифмирование при помощи решета числового поля [3] было применено к дискретному логарифмированию позднее, чем к факторизации чисел. Первые идеи появились в 1990-х годах. Алгоритм, предложенный Д. Гордоном в 1993 году [3], имел эвристическую сложность $O(\exp 3^{3/2}(\log p \log p \log p)^{\frac{1}{3}})$, но оказался достаточно непрактичным.

Позднее было предложено множество различных улучшений данного алгоритма. Было показано, что при $p \geq 10^{100}$ решето числового поля быстрее, чем COS [3]. Современные рекорды в дискретном логарифмировании получены именно с помощью этого метода.

Наилучшими параметрами в оценке сложности на данный момент является $c = (92 + 26\sqrt{13})^{1/3}/3 \approx 1,902$, $d = \frac{1}{3}$. Для чисел специального вида результат можно улучшить. В некоторых случаях можно построить алгоритм, для которого константы будут $c \approx 1,00475$, $d = \frac{2}{5}$. За счёт того, что константа c достаточно близка к 1, подобные алгоритмы могут обогнать алгоритм с $d = \frac{1}{3}$.

Другая возможность эффективного решения задачи вычисления дискретного логарифма связана с квантовыми вычислениями. Теоретически доказано, что с их помощью дискретный логарифм можно вычислить за полиномиальное время. В любом случае, если полиномиальный алгоритм вычисления дискретного логарифма будет реализован, это будет означать практическую непригодность криптосистем на его основе [3].

3. АМЕРИКАНСКИЙ СТАНДАРТ КОДИРОВАНИЯ - ASCII

ASCII (англ. American Standard Code for Information Interchange) — американская стандартная 7-битная кодировочная таблица для печатных символов и некоторых специальных кодов, использующаяся в компьютерной коммуникации. ASCII представляет собой кодировку для представления десятичных цифр, латинского и национального алфавитов, знаков препинания и управляющих символов.

Таблица была разработана и стандартизована в 1963 году. Множество современных кодировок и стандартов (UTF-8, Win-1251, КОИ-8) являются расширениями стандарта ASCII. В СССJ стандарт был утвержден в 1987 году в виде таблицы международной ссылочной версии кода КОИ-7 Н0 ГОСТ 27463-87 (СТ СЭВ 356-86) «Системы обработки информации. 7-битные кодированные наборы символов» [**?**].

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	Start of Header	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	Start of Text	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	End of Text	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	End of Transmission	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	Enquiry	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	Acknowledgment	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	Bell	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	Backspace	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	Horizontal Tab	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	Line feed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	Vertical Tab	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	Form feed	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	Carriage return	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	Shift Out	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	Shift In	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	Data Link Escape	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	Device Control 1	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	Device Control 2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	Device Control 3	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	Device Control 4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	Negative Ack.	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	Synchronous idle	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	End of Trans. Block	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	Cancel	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	End of Medium	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	Substitute	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	Escape	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	File Separator	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	Group Separator	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	Record Separator	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	Unit Separator	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	Del

Рис. 2: ASCII коды

В криптографических программах ASCII используется для преобразования символов текста в цифры, чтобы текст было возможно представить в виде чисел и совершать над ним криптографические преобразования. Например: большим буквам английского алфавита соответствуют значения с 97 по 122.

Поскольку на подавляющем большинстве современных компьютеров минимально адресуемой единицей памяти является байт (размером в 8 бит), там используются 8-битные, а не 7-битные символы. Обычно символ ASCII расширяют до 8 бит, подставляя нулевой бит в качестве старшего. Таким образом, каждый преобразованный в число символ занимает ровно один байт. Уменьшение размера одного символа для криптосистем главным образом означает возможность передать больший шифротекст в одном блоке при неизменной длине ключа.

4. АНАЛИЗ DES, ГОСТ 28147-89, CRYPTO03, EL-GAMAL

Перед началом написания программы “El-Gamal_Tutor” были изучены другие приложения для обучения криптосистемам. Одними из них были: DES, ГОСТ 28147-89, Crypto-03 и El-Gamal.

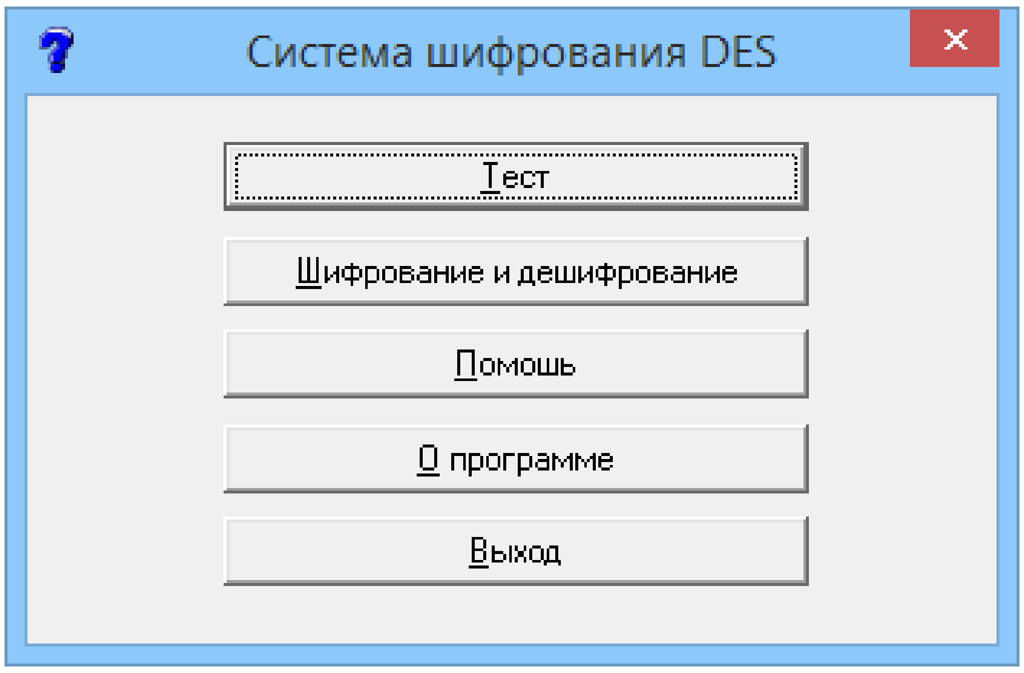

Рис. 3: Главное меню программы "Система шифрования DES"

Программа "Система шифрования DES" предлагает режим обучения симметричной криптосистеме DES, а так же, в качестве дополнительной функции, возможность зашифровать и расшифровать произвольное сообщение используя криптосистему DES.

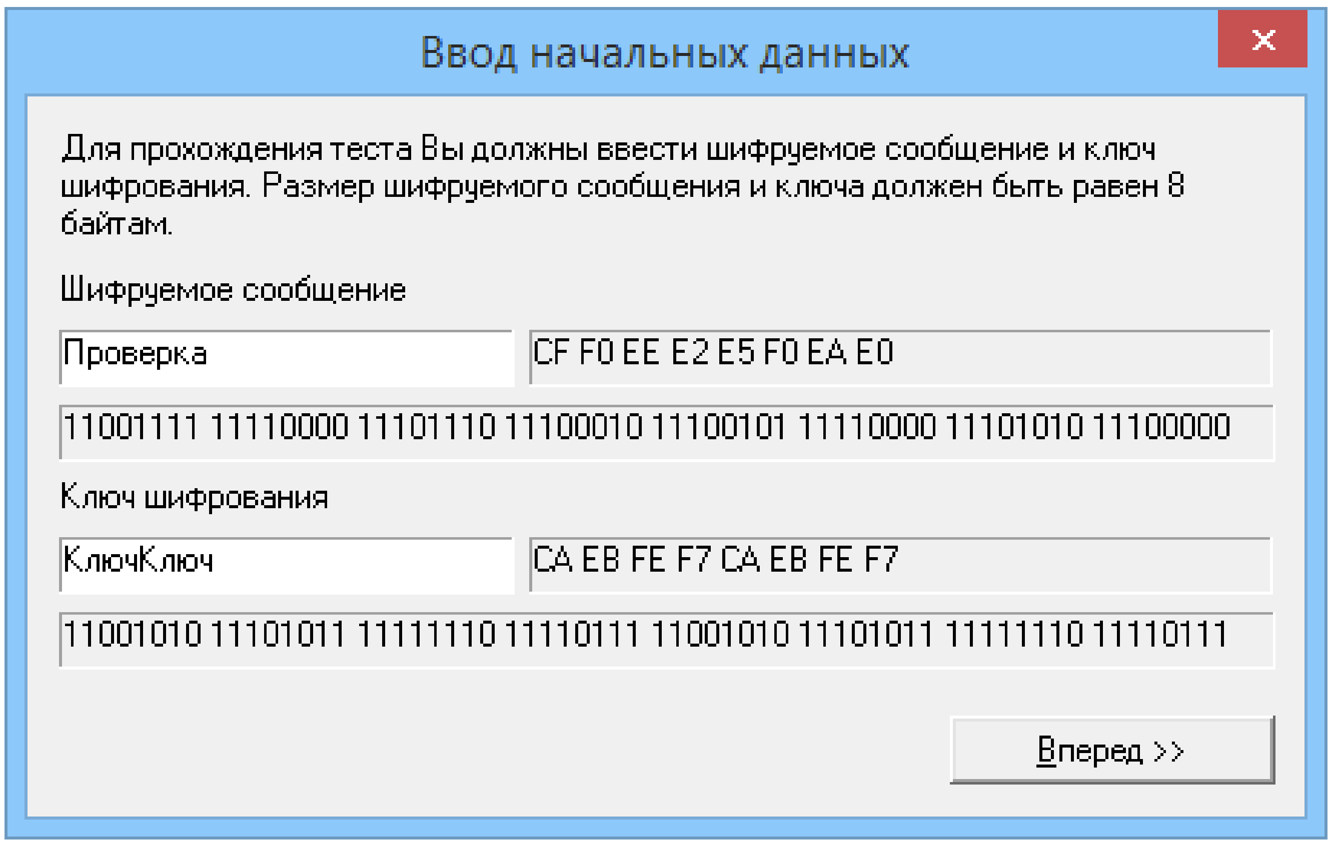

Рис. 4: Ввод начальных данных

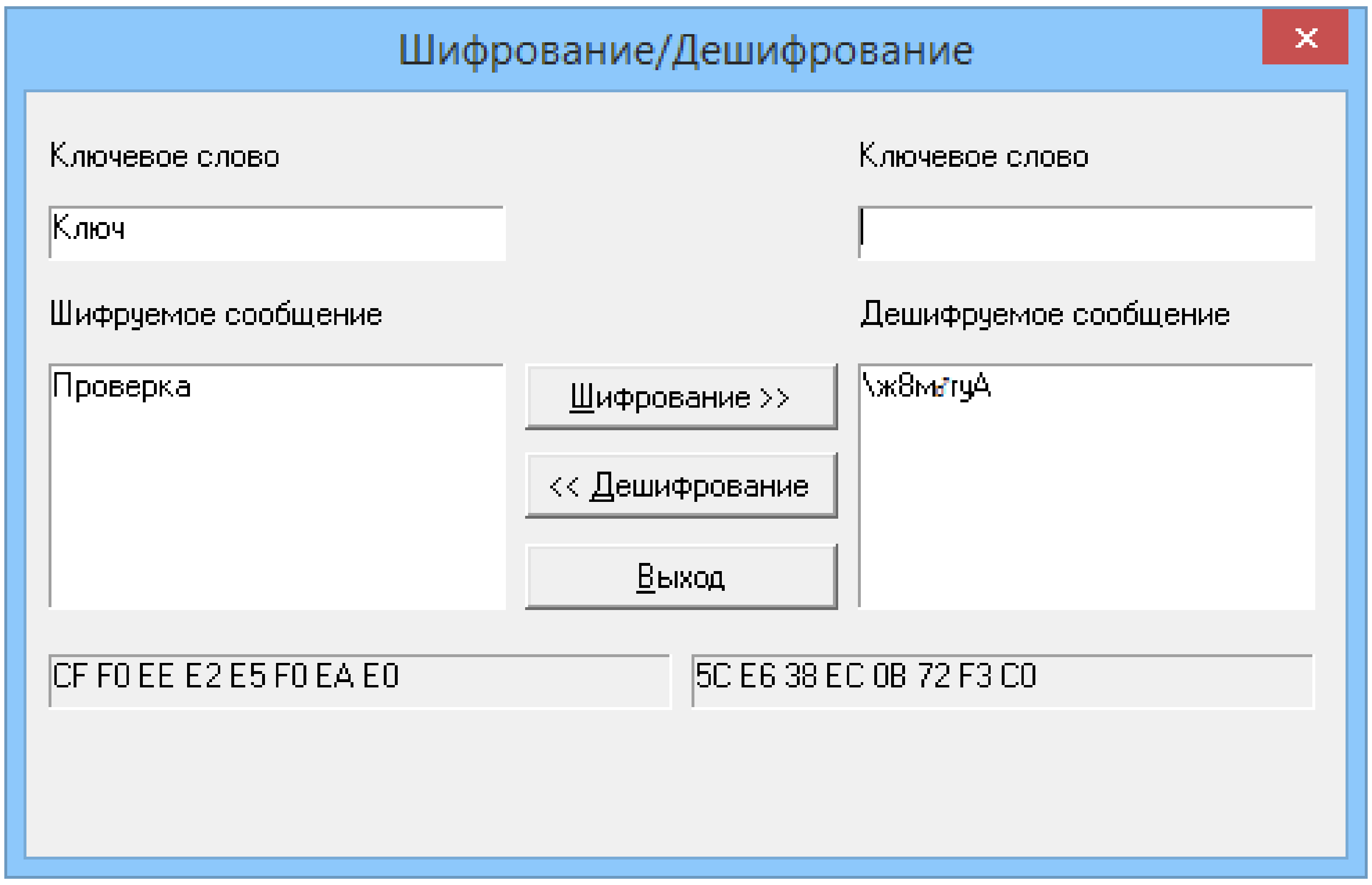

Рис. 5: Шифрование/Расшифрование [4]

В данном случае в программе режим называется неправильно, так как на самом деле вместо дешифрования происходит расшифрование.

К сожалению, программа не предлагает дополнительных возможностей, таких, как отдельный режим проведения криптографических вычислений и преобразований.

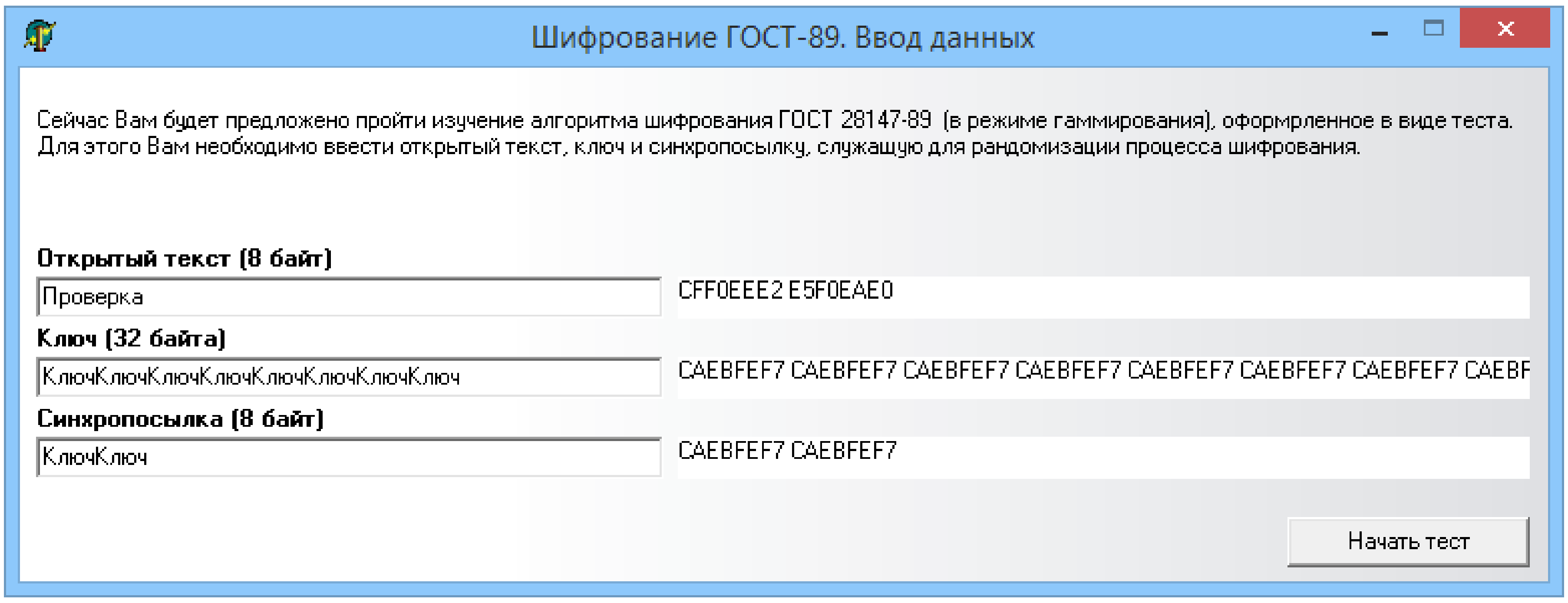

Рис. 6: Ввод данных в обучающей программе ГОСТ 28147-89

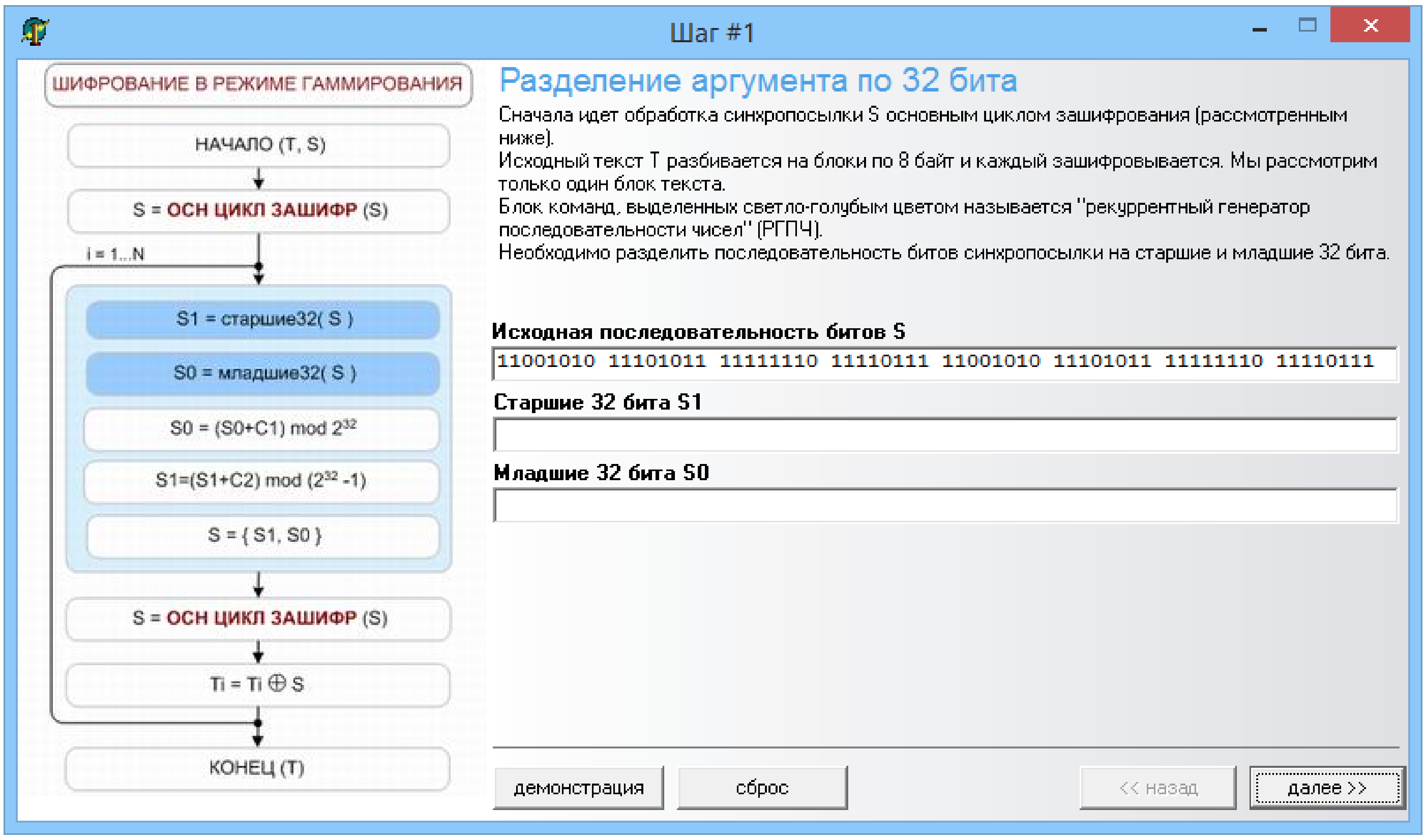

Рис. 7: Первый шаг обучения ГОСТ 28147-89 [6]

Как и программа "Система шифрования DES", программа ГОСТ 28147-89 не предлагает дополнительных криптографических или математических возможностей. Программа не предлагает дополнительных криптографических или математических функций и не предлагает возможности опробовать криптографический алгоритм ГОСТ-89 на произвольном сообщении без необходимости проходить при этом шаги обучения криптосистеме.

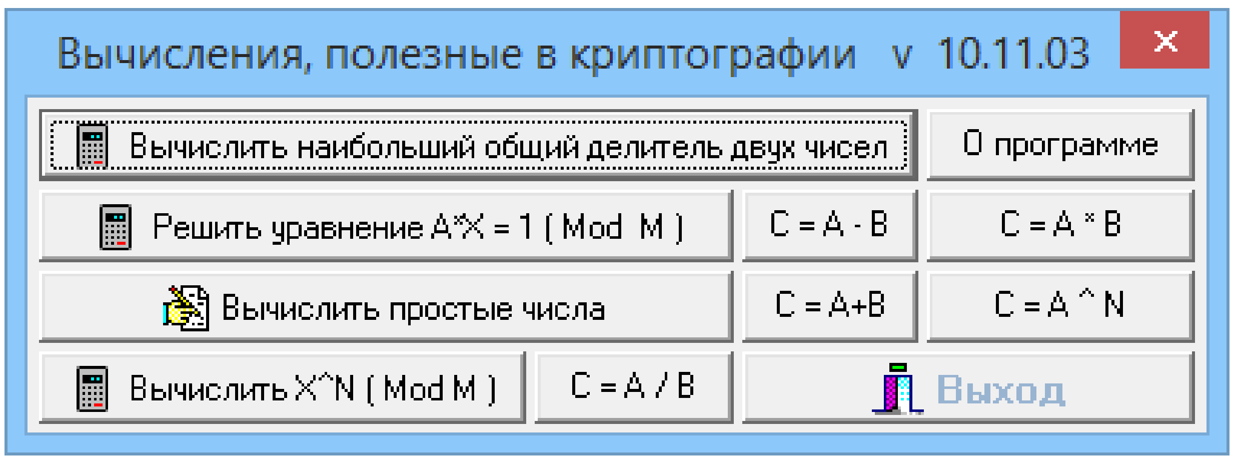

Рис. 8: Основная форма программы вычислений, полезных в криптографии v.10.11.2003 – Crypto03 [4]

Программа Crypto03 представляет собой своего рода криптографический калькулятор, содержащий в себе ряд вычислительных функций, полезных в криптографии. Она не предоставляет режима обучения.

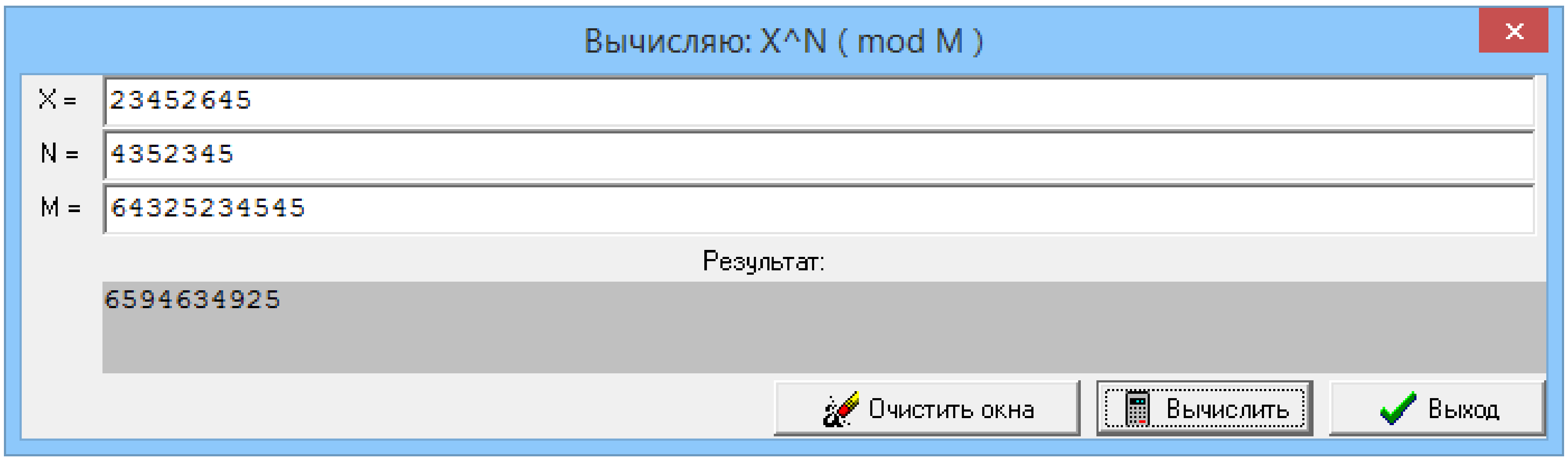

Рис. 9: Вычисление $X^N \mod M$ в программе Crypto03

Программа El-Gamal - обучающая программа, посвящённая криптосистеме Эль-Гамаля. Основными недостатками программы являются скупая подача обучающего материала и весьма неудобный интерфейс. Программа не предлагает дополнительных криптографических или математических функций, а также проблематична в освоении без использования документации.

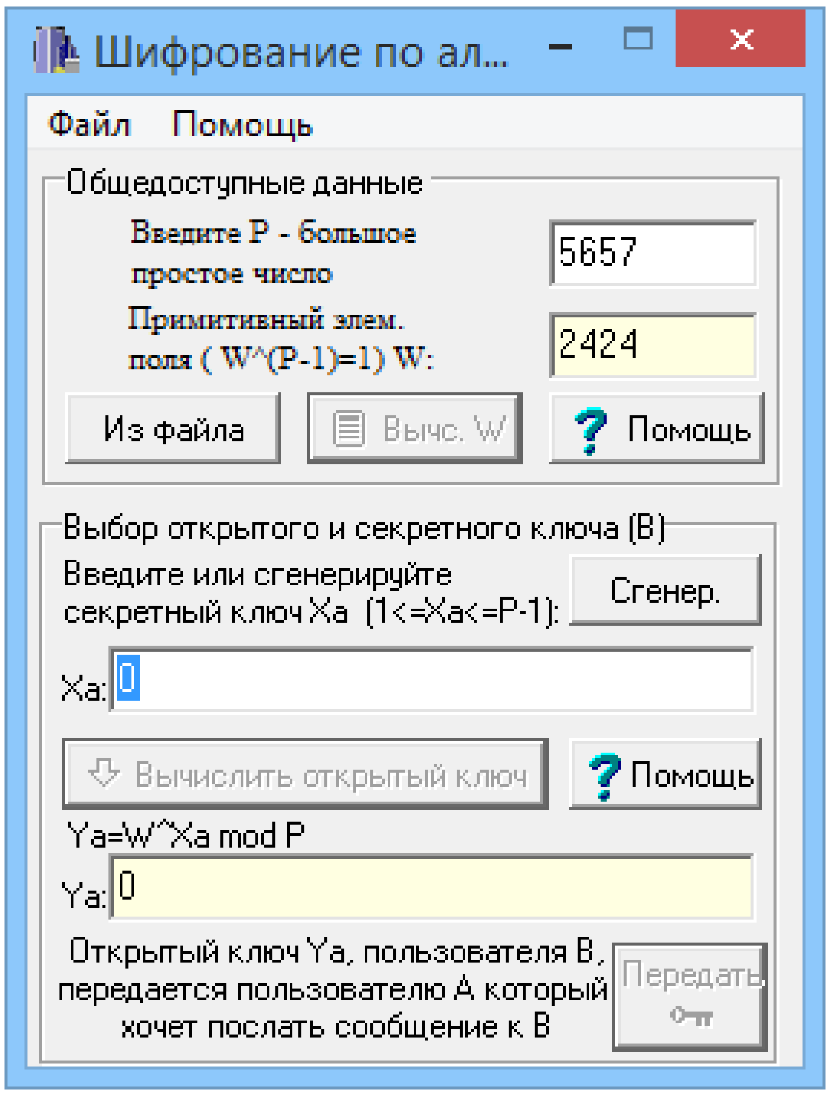

Рис. 10: Основная форма обучающей программы El-Gamal [5]

После рассмотрения всех этих программ, была сформирована картина того, как должна выглядеть будущая электронная обучающая программа El-Gamal_Tutor.

5. ОПИСАНИЕ ЭЛЕКТРОННОЙ ОБУЧАЮЩЕЙ ПРОГРАММЫ "EL-GAMAL_TUTOR"

Посредством среды программирования Microsoft Visual Studio Community 2017 создано приложение, предназначенное для обучения основам криптографической системы Эль-Гамаля.

5.1. Общие сведения

Программа написана на языке программирования C# в визуальной среде Microsoft Visual Studio 2017 Community Edition с использованием программной платформы Microsoft .NET Framework 4.5. Проект общим объемом 1.84 Мб. Программа функционирует в операционной системе Windows 7 или новее.

При разработке использовались модули: System.Collections.Generic, System.ComponentModel, System.Data, System.Drawing, System.Linq, System.Text, System.Threading.Tasks, System.Windows.Forms, System.Numerics.

Размер генерируемых программой ключей теоретически ничем не ограничен, практически же он ограничен в соответствии с характеристиками компьютера, на котором запускается программа.

5.2. Функциональное назначение

Приложение предназначено для обучения методам и алгоритмам, используемым при реализации асимметричной криптографической системы Эль-Гамаля, а также частичной проверки знаний учащегося.

Дополнительные функции приложения позволяют использовать его в качестве программы для небольших полезных в криптографии вычислений.

5.3. Используемые технические средства

Компьютер с шестиядерным процессором 3.2 GHz, 8 Gb RAM, Microsoft Windows 10 x64.

5.4. Описание логической структуры

Программа логически разделена на две части: режим обучения и вспомогательные функции.

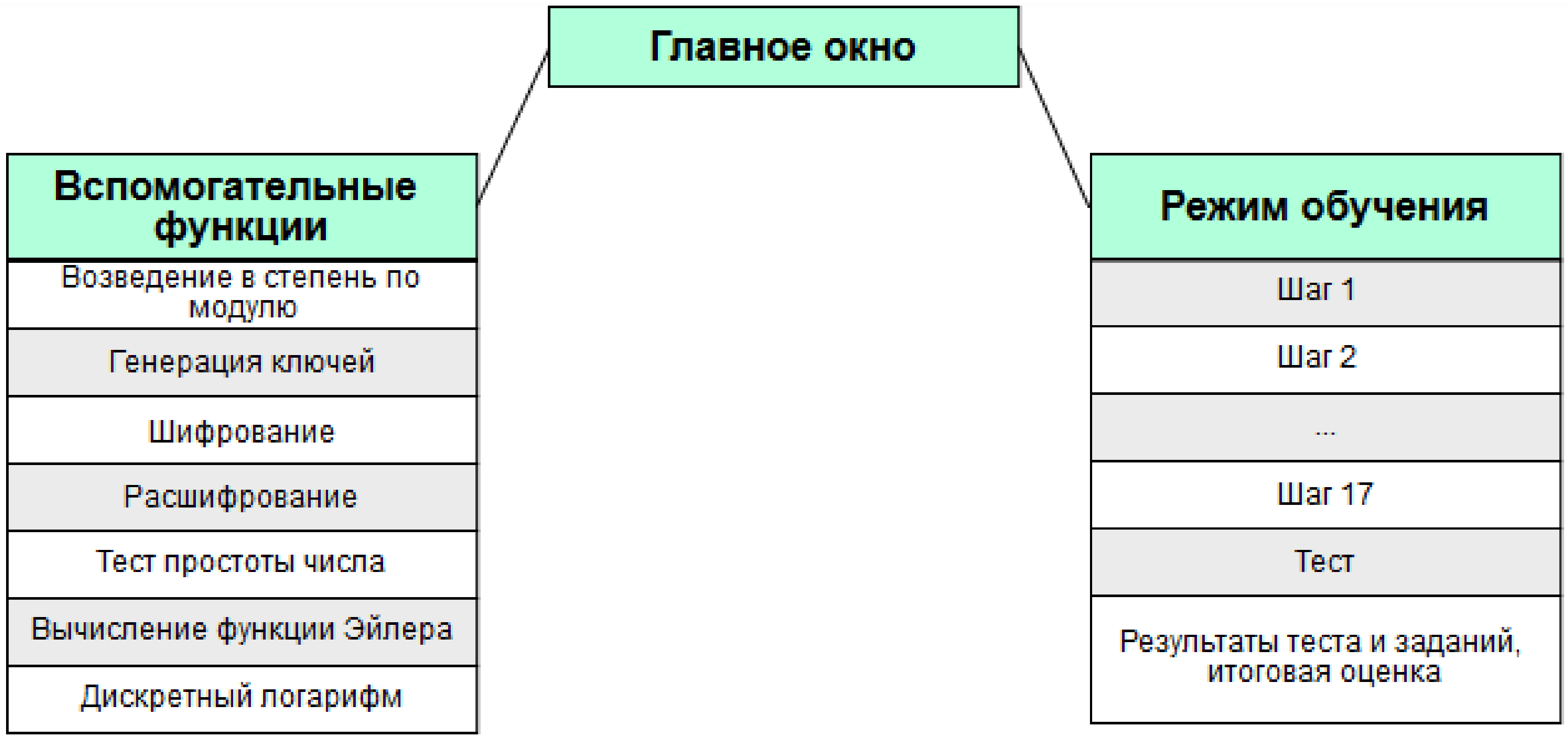

Рис. 11: Блок-схема программы "El-Gamal_Tutor"

В режиме обучения рассматриваются математические основы, на которых базируется криптосистема Эль-Гамаля, алгоритмы генерации ключей, шифрования и расшифрования, а также основы криптоанализа системы и некоторые алгоритмы дискретного логарифмирования. Дополнительный функционал включает в себя различные вычислительные возможности, так или иначе связанные с криптосистемой Эль-Гамаля. Они могут использоваться как в совокупности с обучением криптосистеме, так и отдельно от него.

5.5. Описание алгоритма

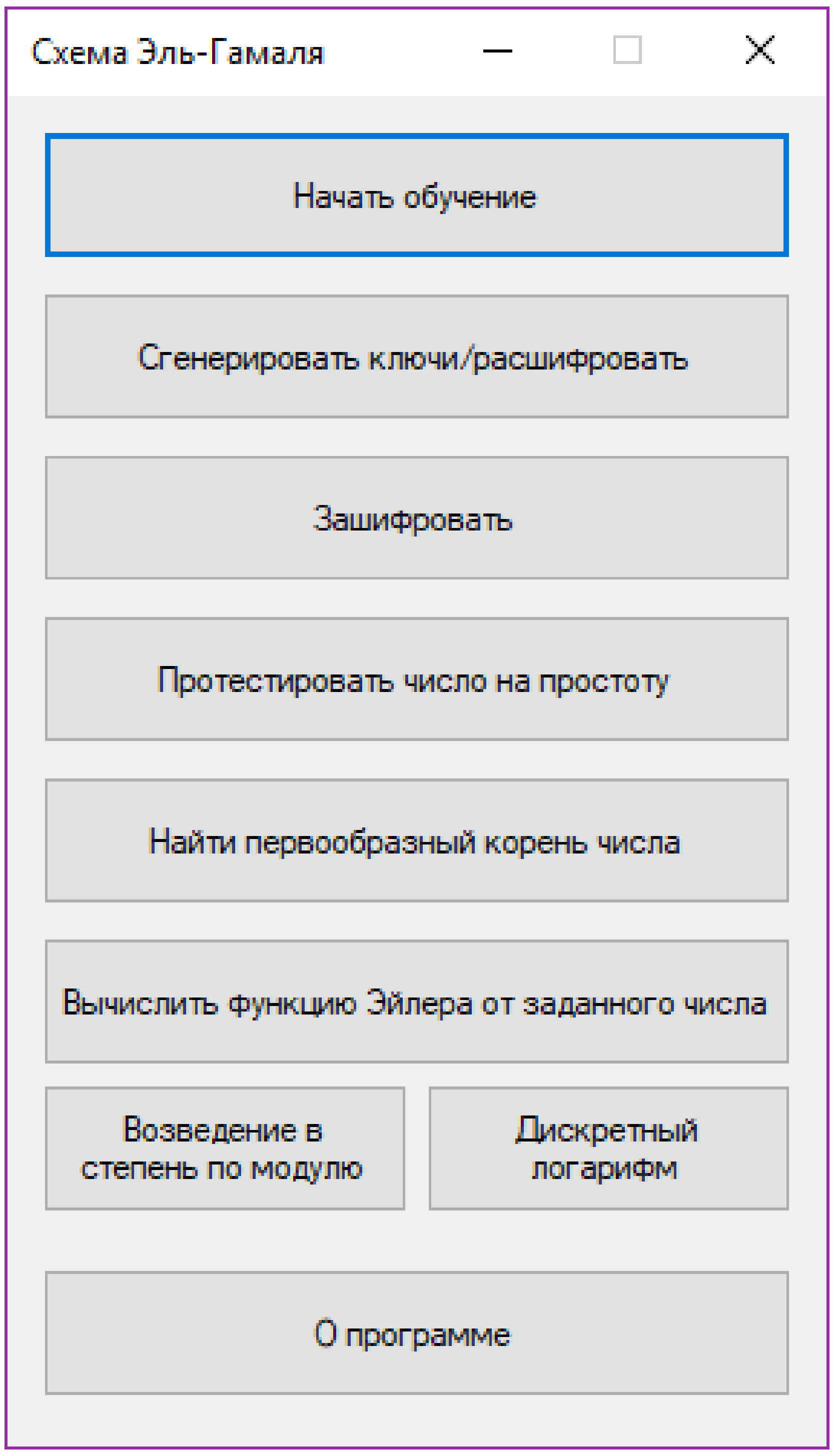

Рис. 12: Основное меню программы "El-Gamal_Tutor"

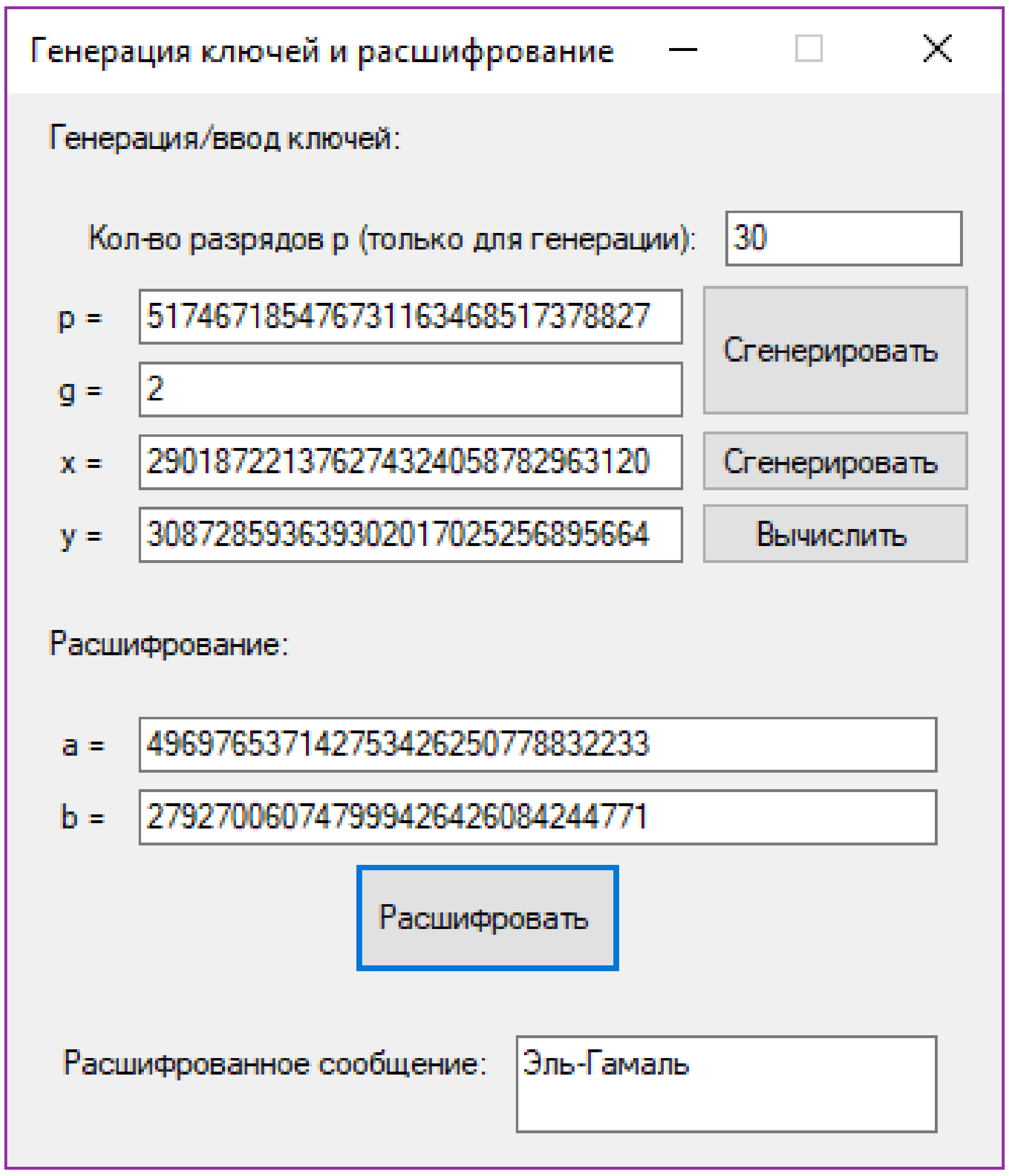

Рис. 13: Генерация ключей и расшифрование

В режиме генерации ключей мы можем сгенерировать ключи для криптосистемы Эль-Гамаля, а также расшифровать необходимую фразу из шифротекста с использованием этих ключей.

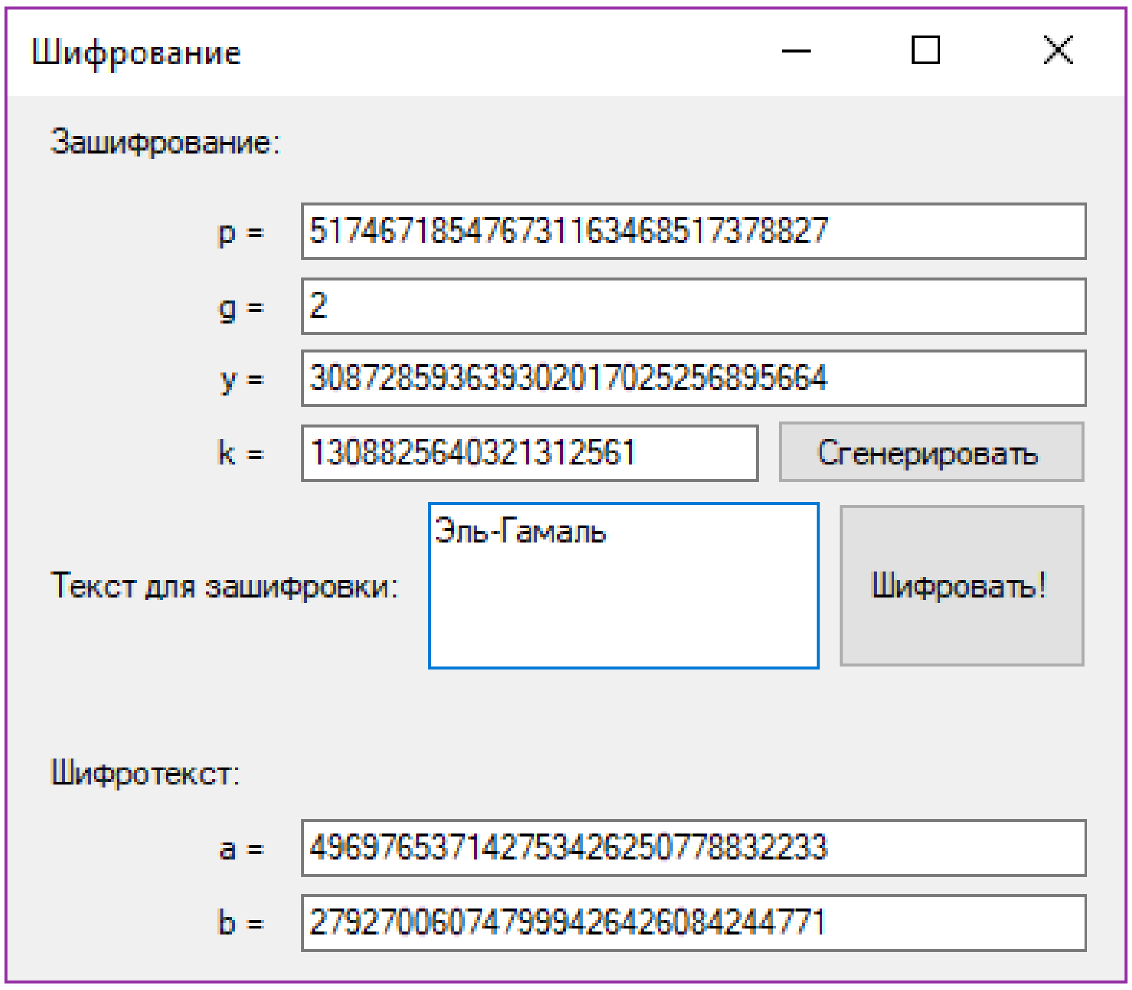

Рис. 14: Результат шифрования

Режим шифрования позволяет зашифровать сообщение пользователя с помощью введённого открытого ключа.

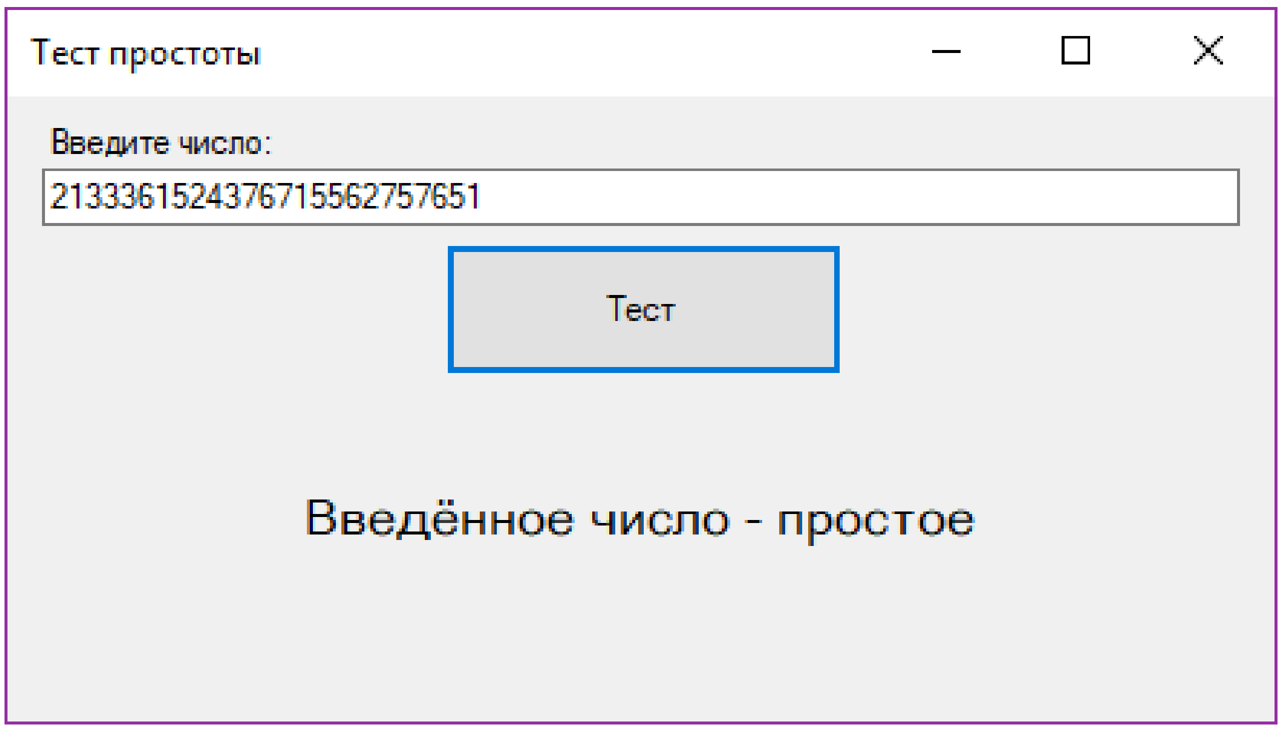

Рис. 15: Тест простоты произвольного числа

Режим теста простоты позволяет проверить, является ли введённое целое неотрицательное число простым или составным. Для определения простоты числа в программе используется вероятностный тест Миллера-Рабина, количество «свидетелей простоты» - 4000.

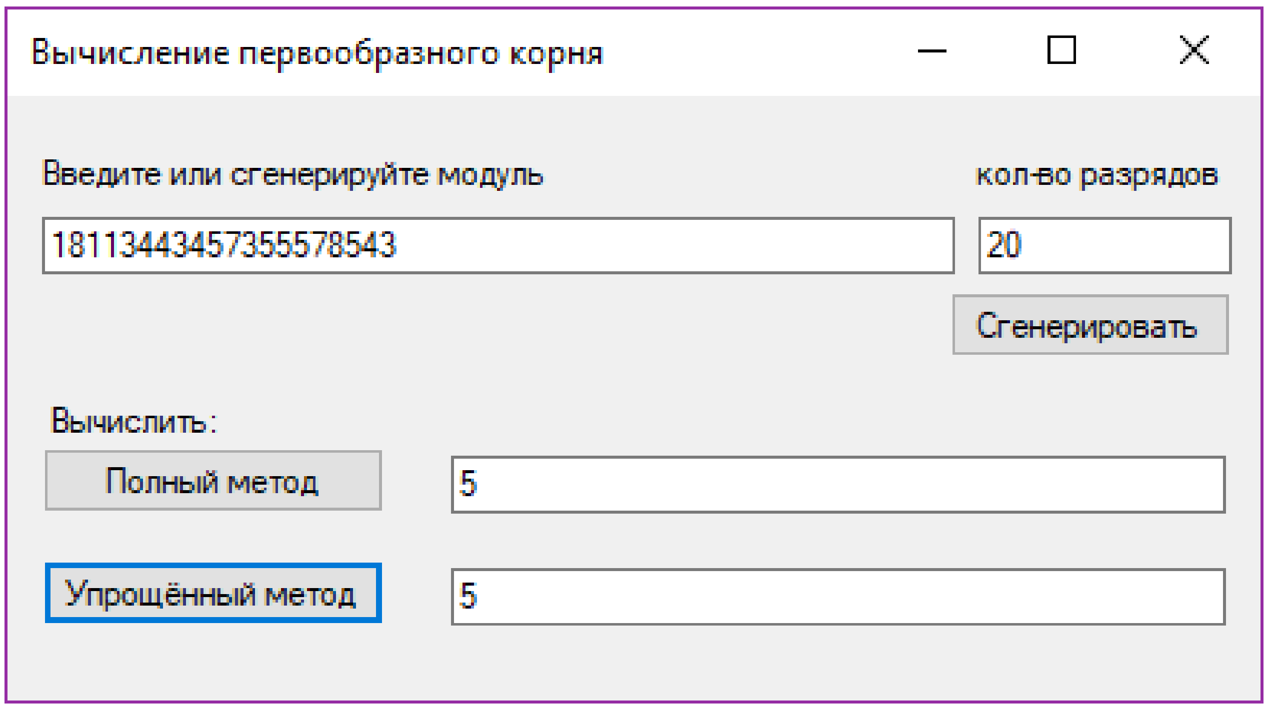

Рис. 16: Вычисление перевообразного корня по заданному модулю

Режим вычисления первообразного корня позволяет вычислить первообразный (или примитивный) корень для большого числа. Поскольку полный метод вычисления первообразного корня очень медленен для больших чисел, в программе предусмотрена возможность вычисления первообразного корня по «упрощённому» методу, который даёт ответ, верный только с некоторой вероятностью.

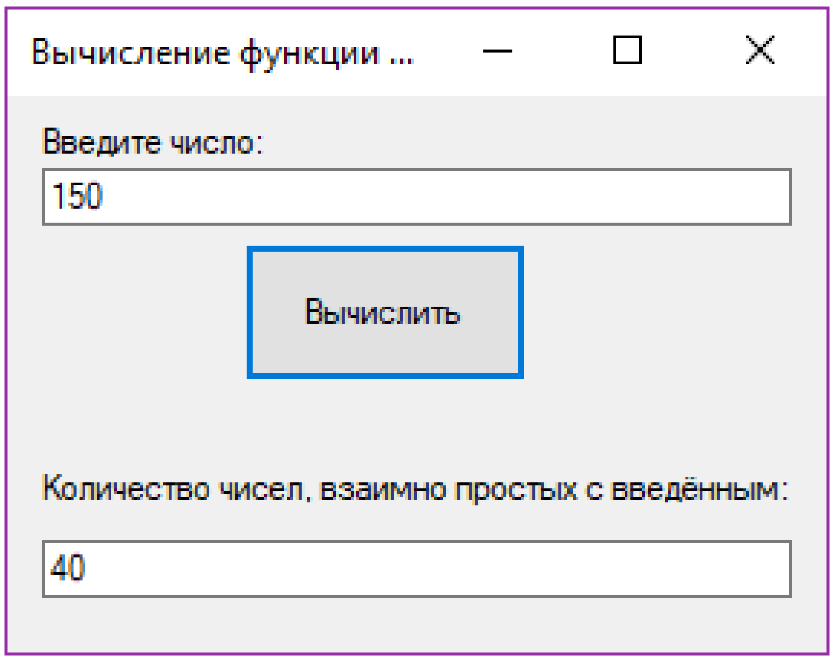

Рис. 17: Вычисление функции Эйлера

Режим вычисления функции Эйлера позволяет вычислить количество чисел, взаимно простых с заданным.

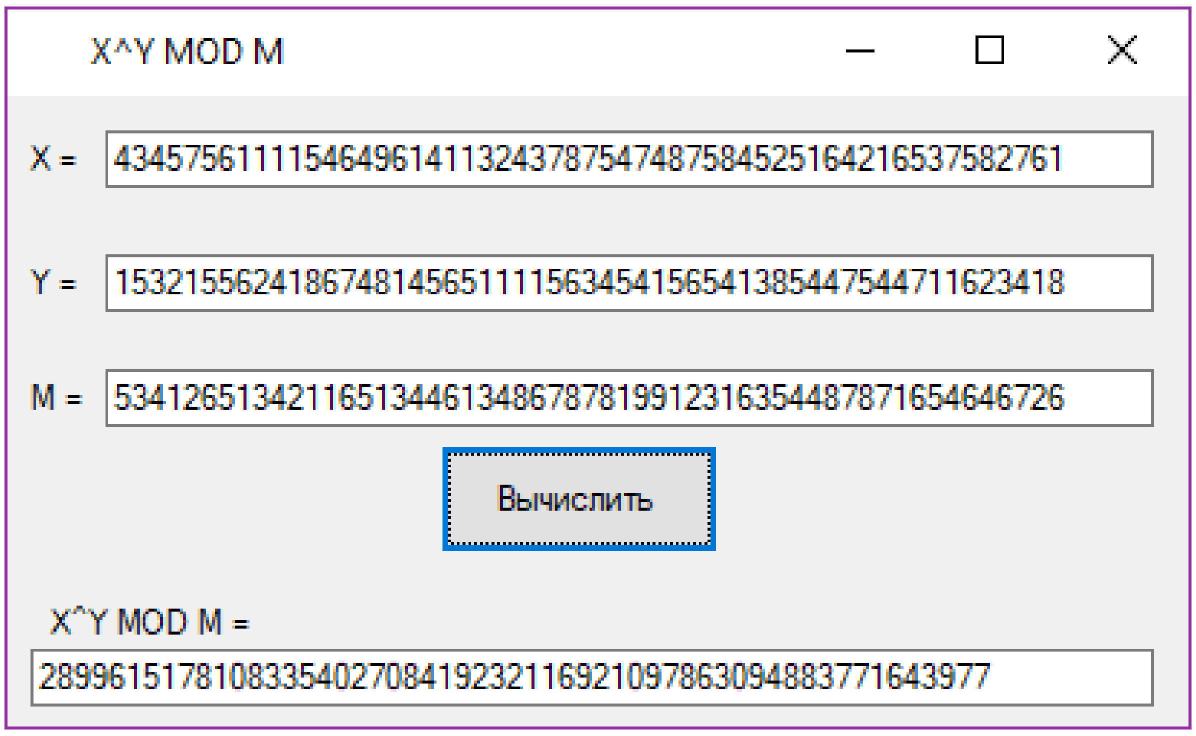

Рис. 18: Возведение в степень по модулю

Режим возведения в степень по модулю представляет собой калькулятор заданных степеней произвольных чисел по заданному модулю.

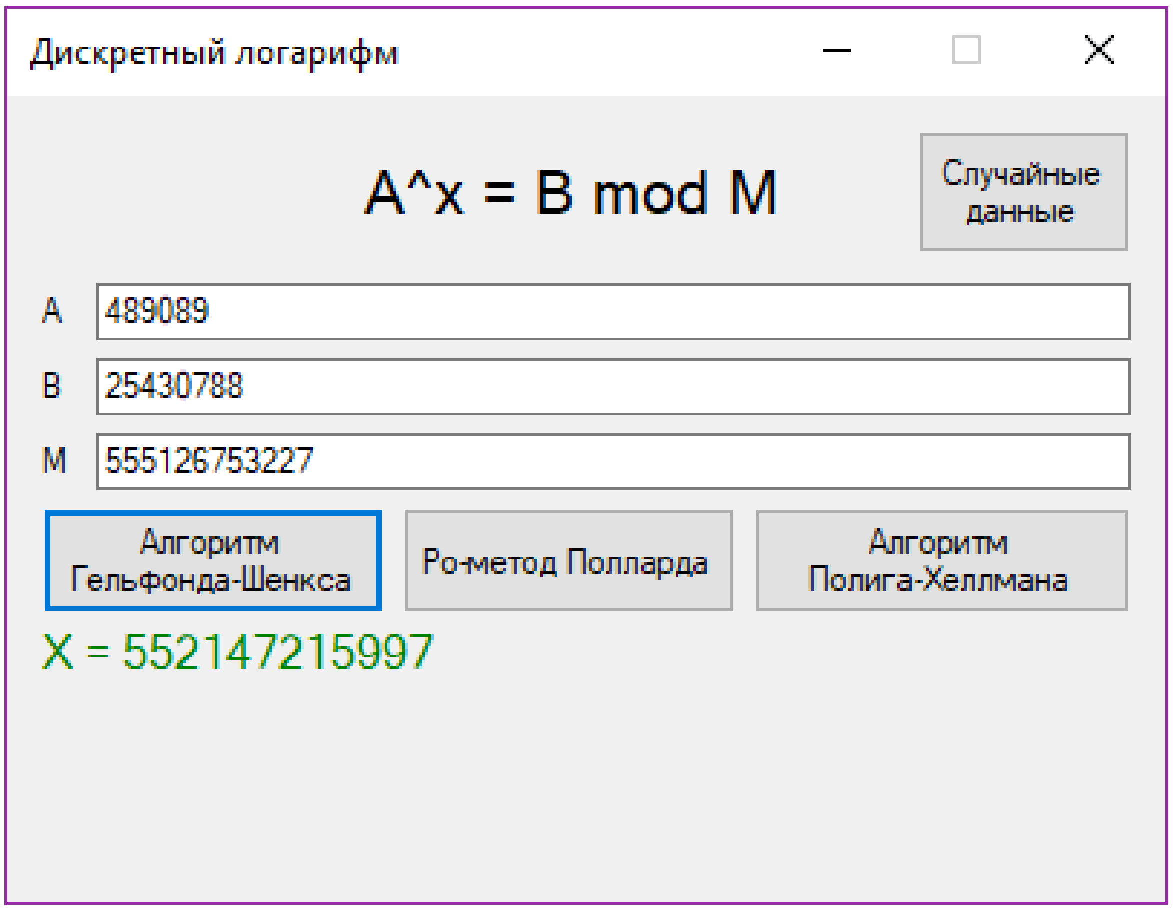

Рис. 19: Дискретное логарифмирование

Режим дискретного логарифмирования позволяет произвести поиск решения уравнения $A^X = B \mod M$ для произвольных целых чисел A и B и простого числа M. Для поиска решения пользователю предлагается использовать три алгоритма дискретного логарифмирования: алгоритм Гельфонда-Шенкса, ρ-метод Полларда и алгоритм Полига-Хеллмана.

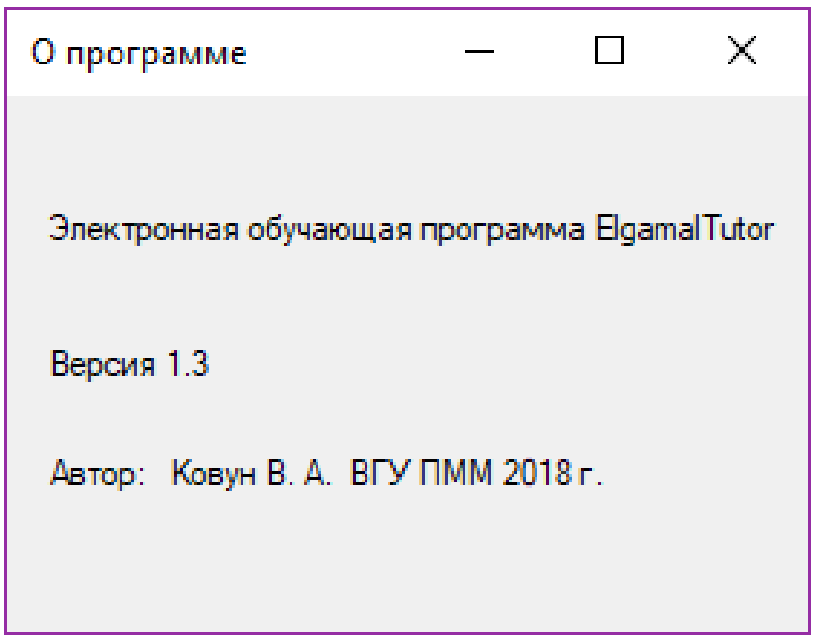

Рис. 20: О программе

В окне «О программе» мы можем увидеть информацию о приложении El-Gamal_Tutor.

Теперь перейдем к режиму обучения. Он состоит из нескольких шагов.

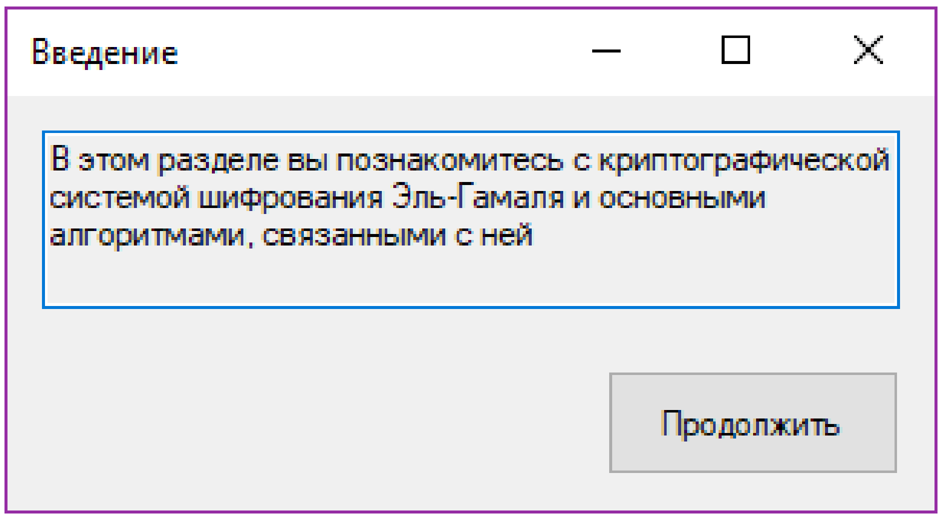

Рис. 21: Введение

На первом шаге рассказывается про операцию возведения в степень по модулю и предлагается решить три примера. Условия заданий генерируются случайным образом.

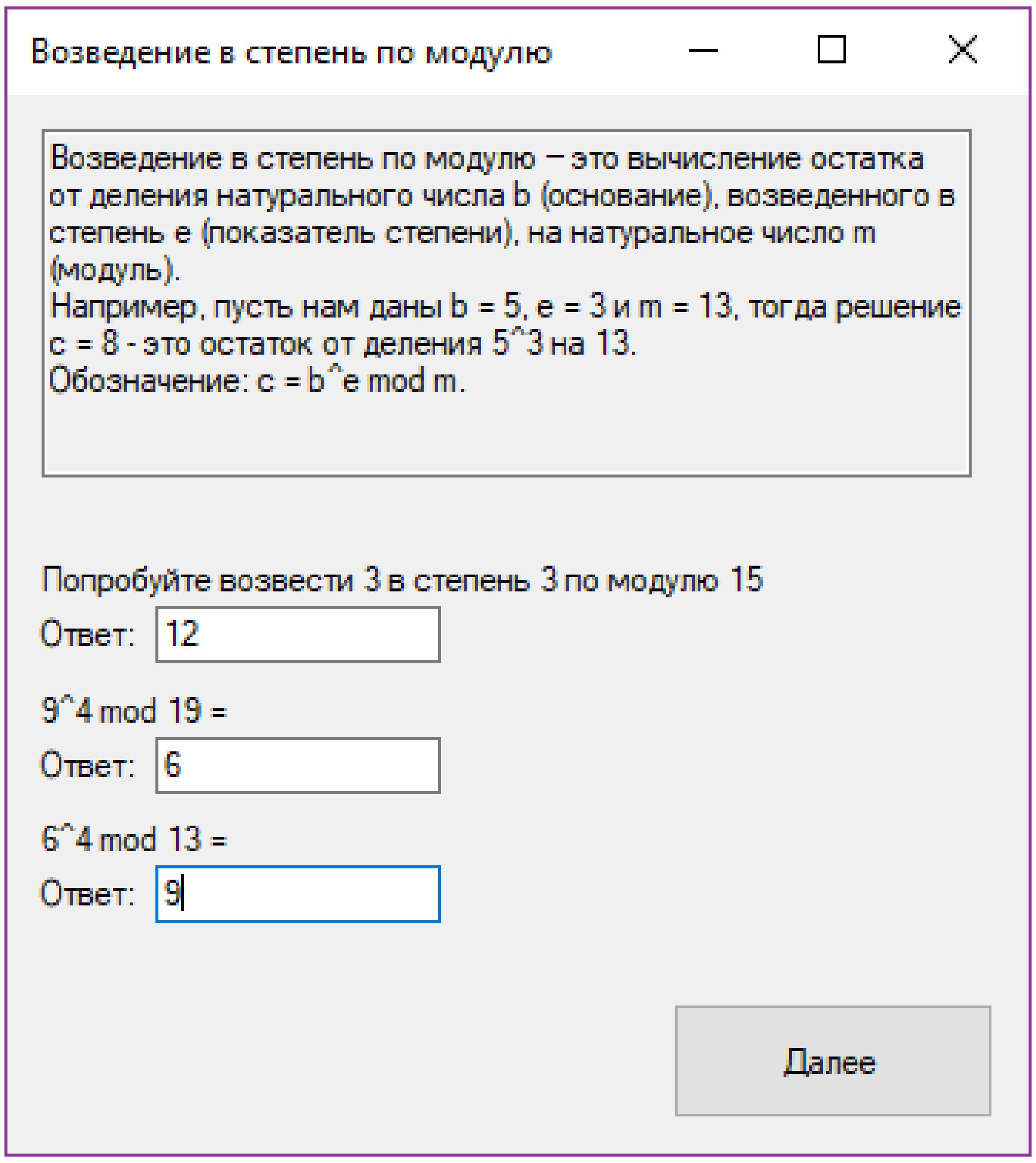

Рис. 22: Возведение в степень по модулю

На втором шаге рассказывается про функцию Эйлера и предлагается решить три примера. Условия заданий так же генерируются случайным образом.

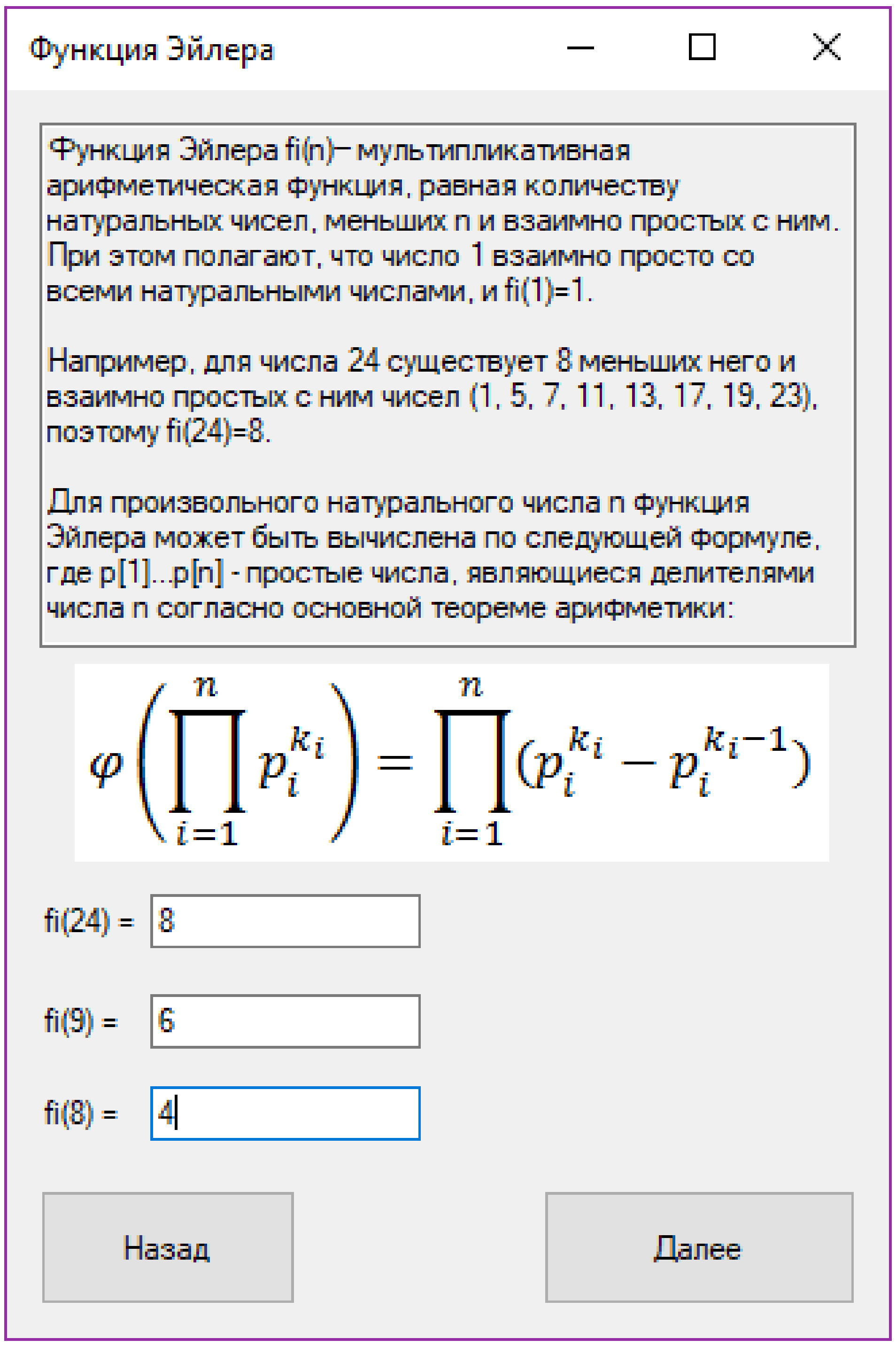

Рис. 23: Функция Эйлера

На третьем шаге объясняется операция нахождения обратного по модулю числа, и предлагается найти два таких числа для сгенерированных условий.

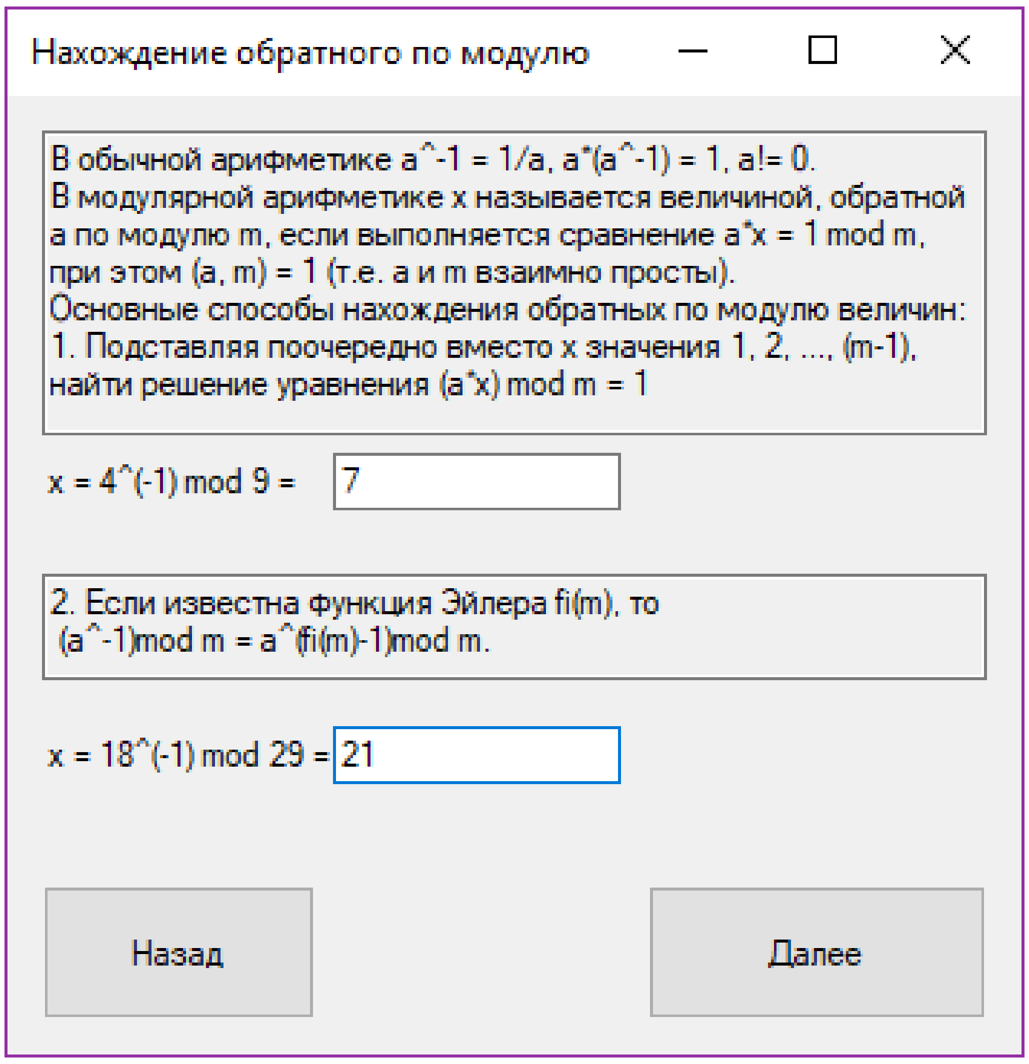

Рис. 24: Обратное по модулю число

Четвёртый шаг рассказывает о Тахере Эль-Гамале.

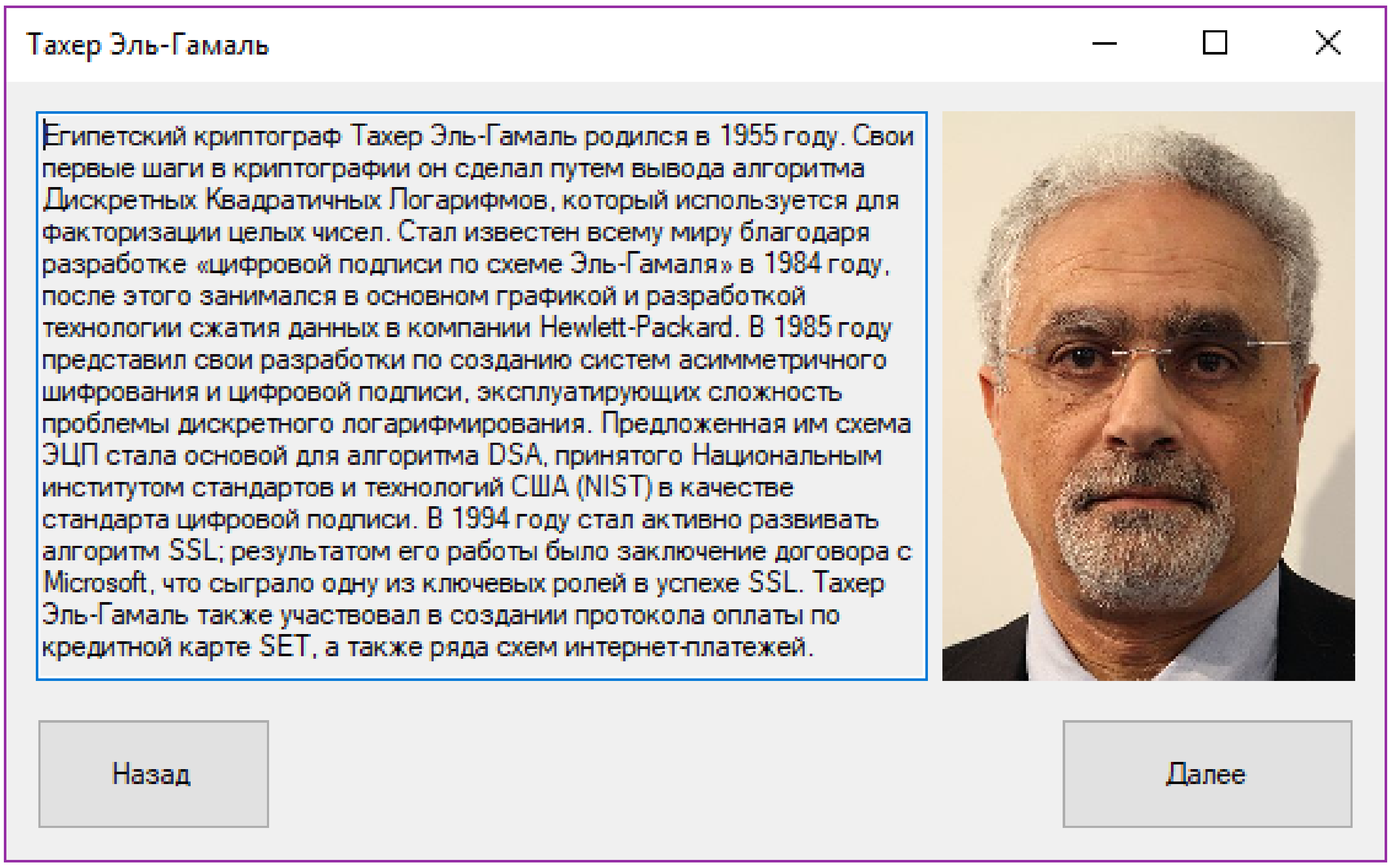

Рис. 25: Тахер Эль-Гамаль

Пятый шаг рассказывает общую информацию о схеме Эль-Гамаля и основных стандартах где она использовалась или используется.

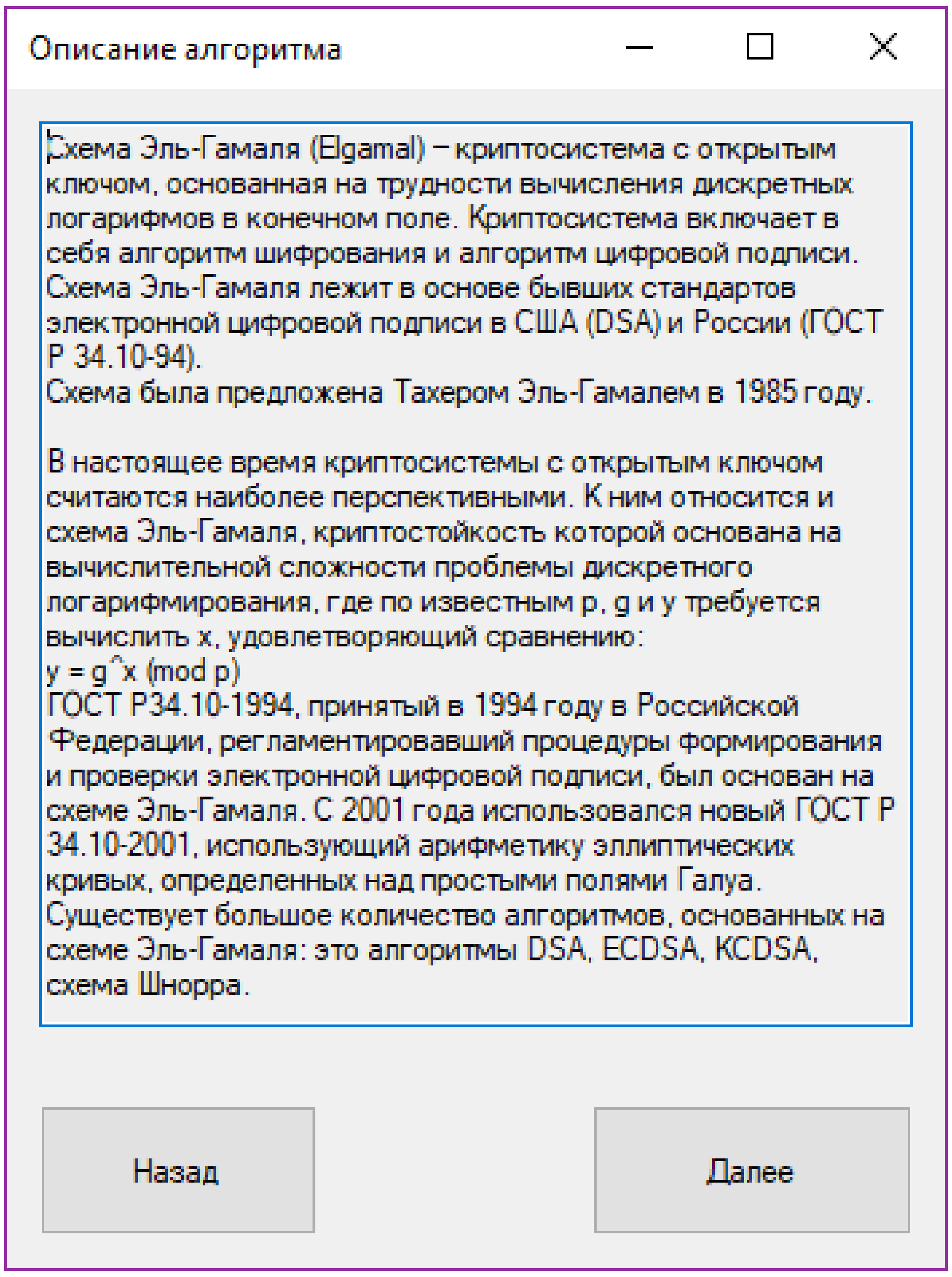

Рис. 26: Общая информация о криптосистеме

На пятом шаге мы видим конкретный пример генерирования ключей криптосистемы. Числа p и x можно как вводить с клавиатуры, так и случайно сгенерировать.

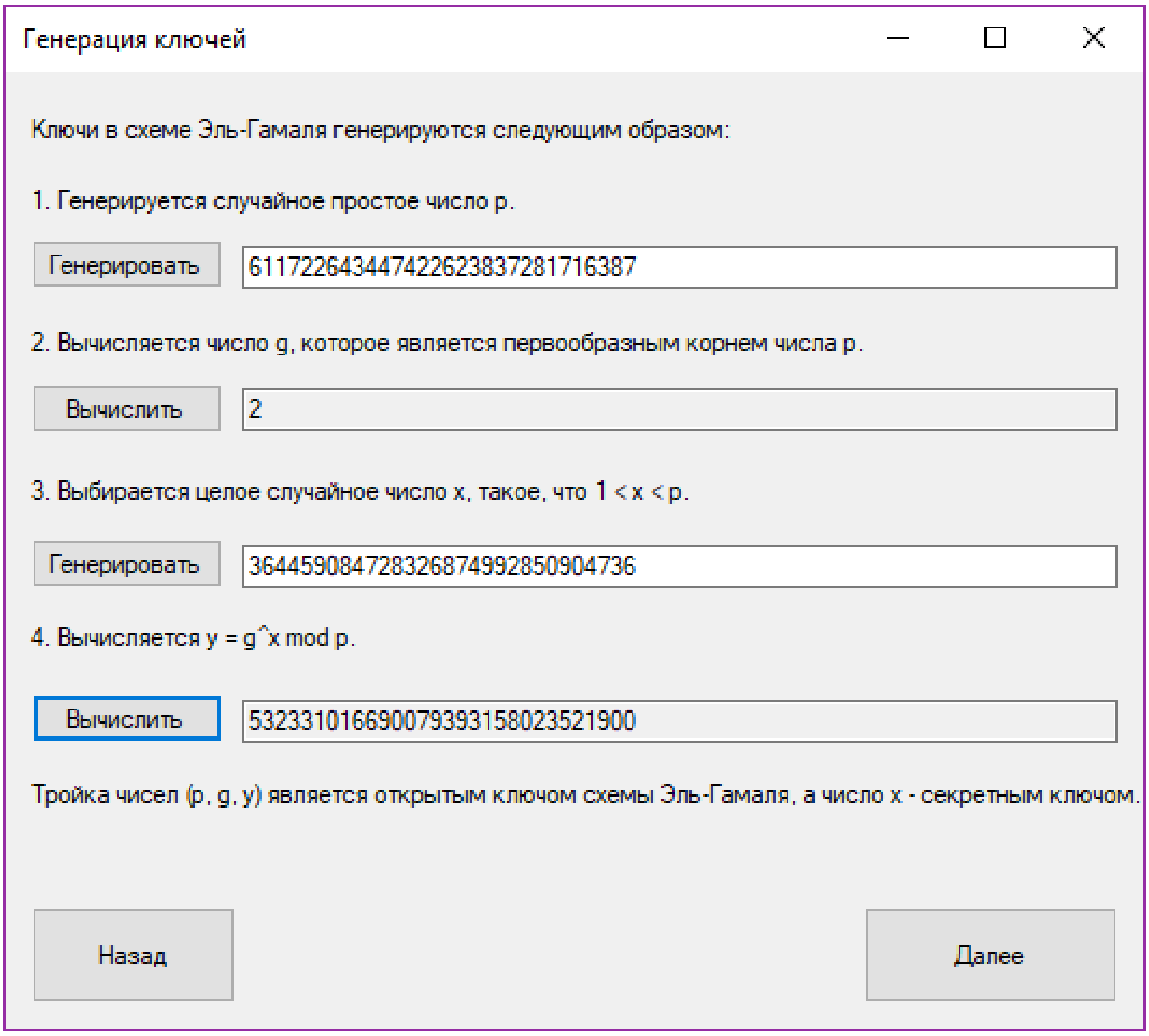

Рис. 27: Генерация ключей

На следующем шаге объясняется алгоритм шифрования по схеме Эль-Гамаля.

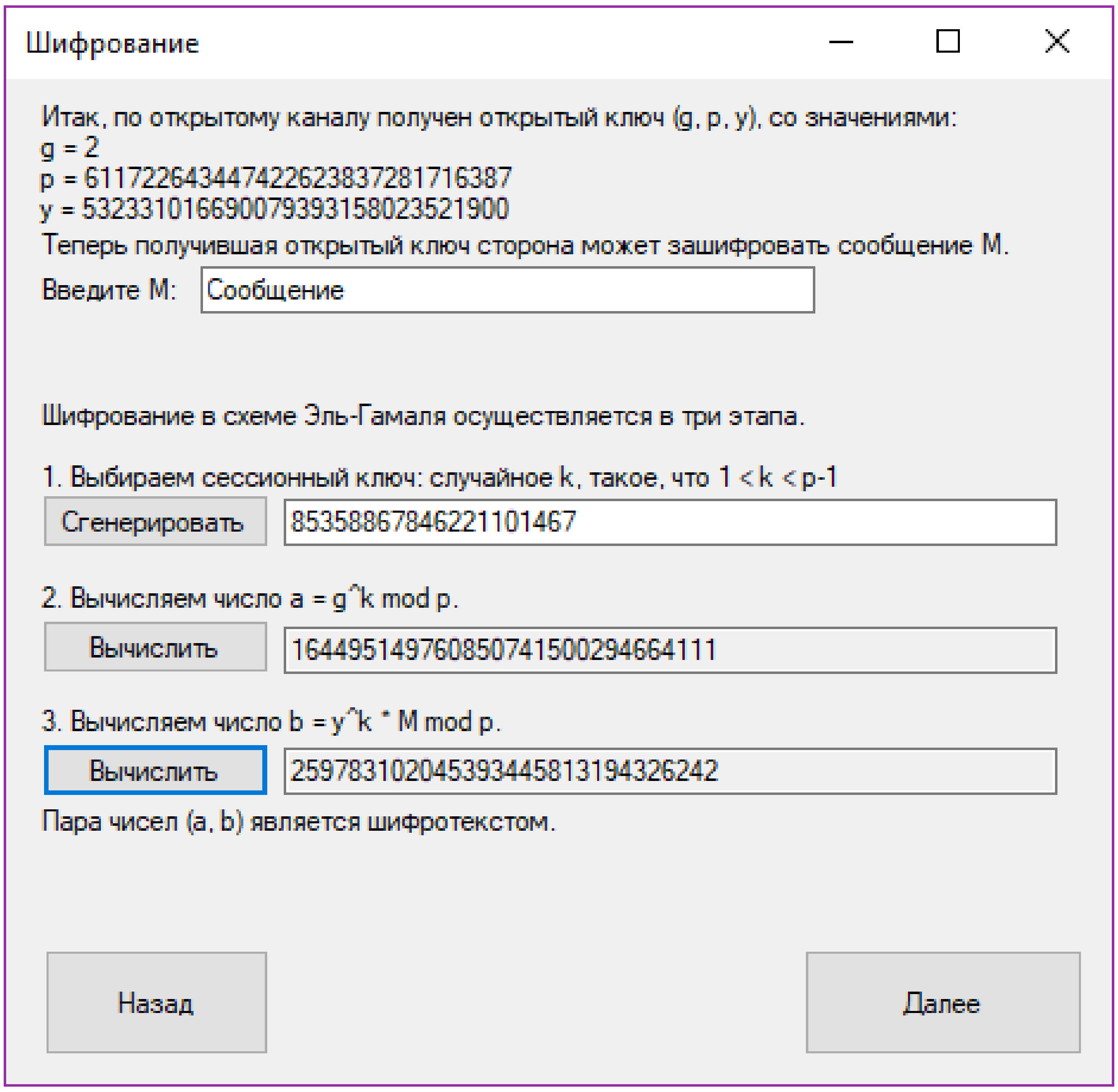

Рис. 28: Шифрование

Следующий шаг показывает, как с помощью открытого и секретного ключей расшифровать сообщение, введённое и зашифрованное на предыдущем шаге.

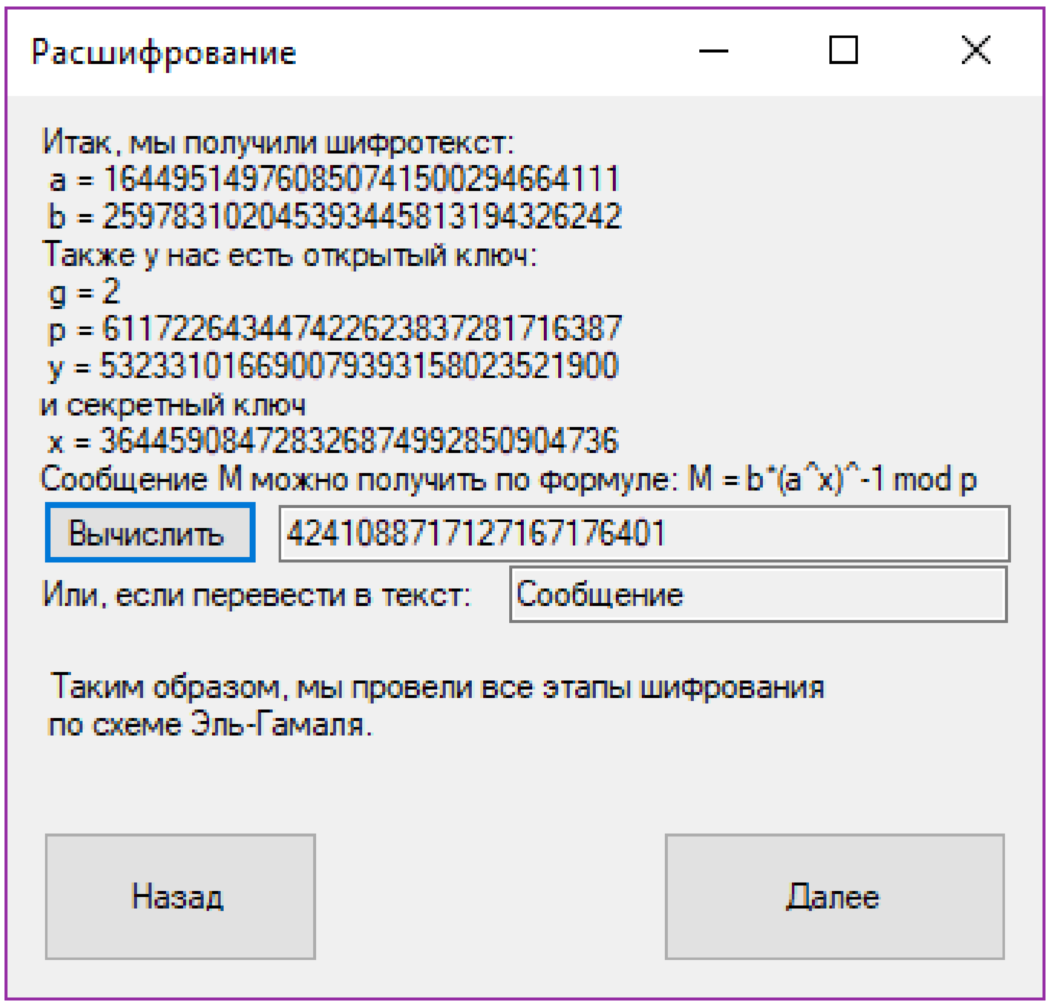

Рис. 29: Расшифрование

На следующем шаге рассказывается о математической задаче дискретного логарифмирования, её связи с криптоанализом схемы Эль-Гамаля, а также предлагается вручную решить два задания. Задания на этом шаге генерируются случайным образом.

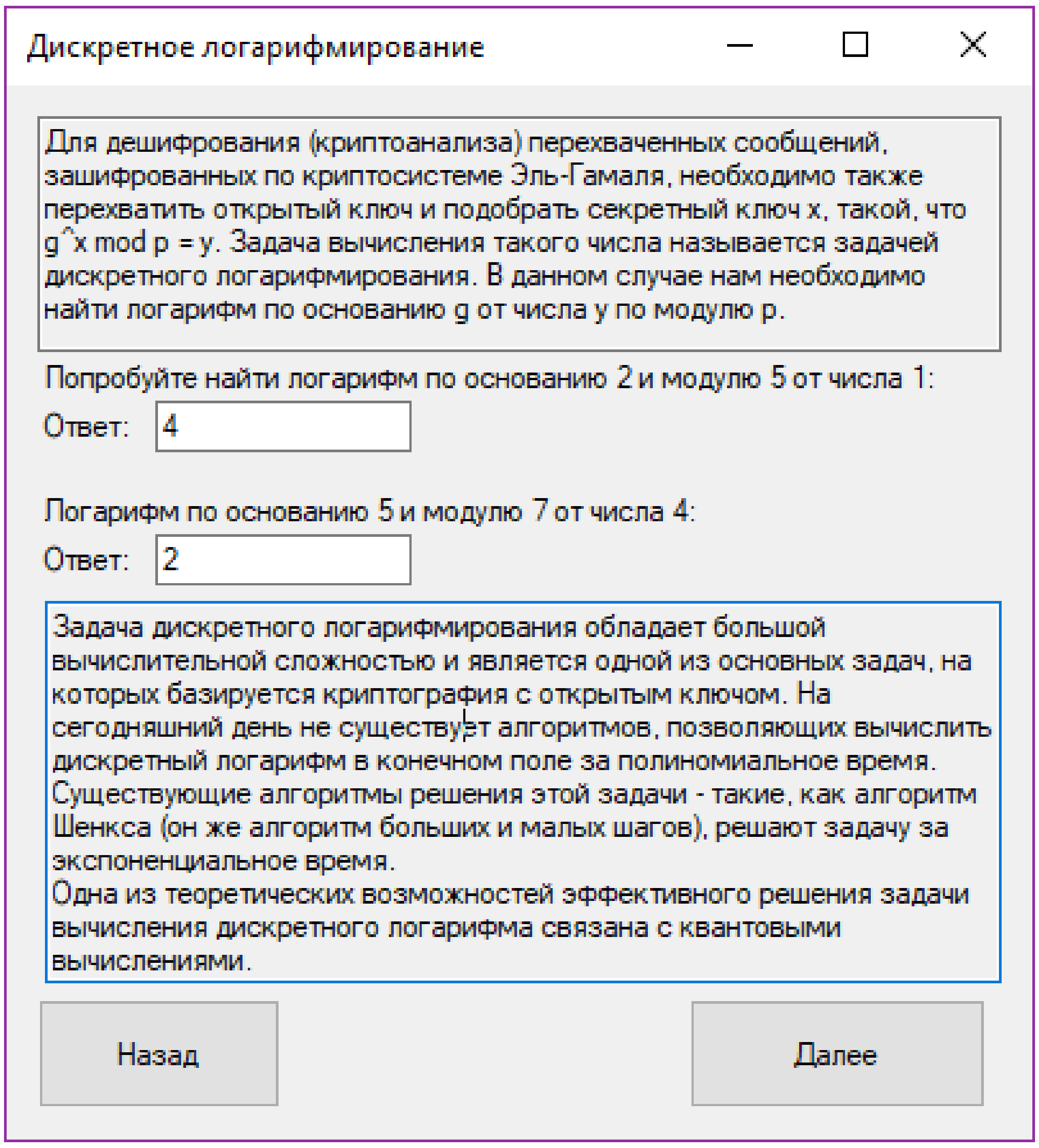

Рис. 30: Дискретный логарифм

На следующем шаге рассказывается о существующих экспоненциальных алгоритмах нахождения дискретного логарифма. Для наглядной демонстрации вычислительной сложности дискретного логарифмирования пользователю предлагается реализовать алгоритм полного перебора для дискретного логарифмирования и убедиться в полной непригодности этого метода даже для сравнительно небольших модулей.

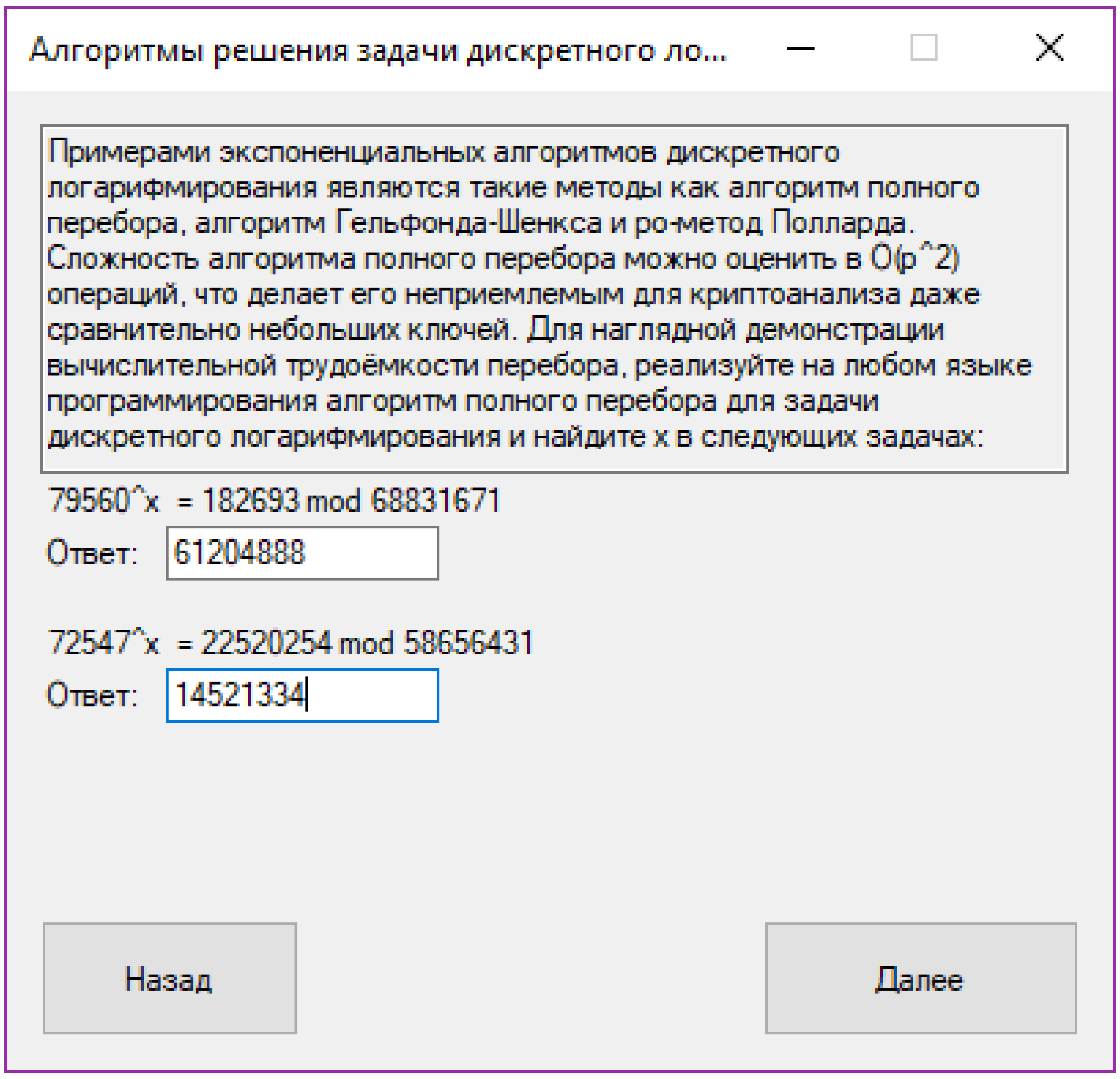

Рис. 31: Алгоритмы дискретного логарифмирования

На следующих трёх шагах рассказывается об одном из алгоритмов дискретного логарифмирования - алгоритме Гельфонда-Шенкса, также известном как алгоритм больших и малых шагов. На первом из этих шагов пользователь получает общую информацию об алгоритме Гельфонда-Шенкса.

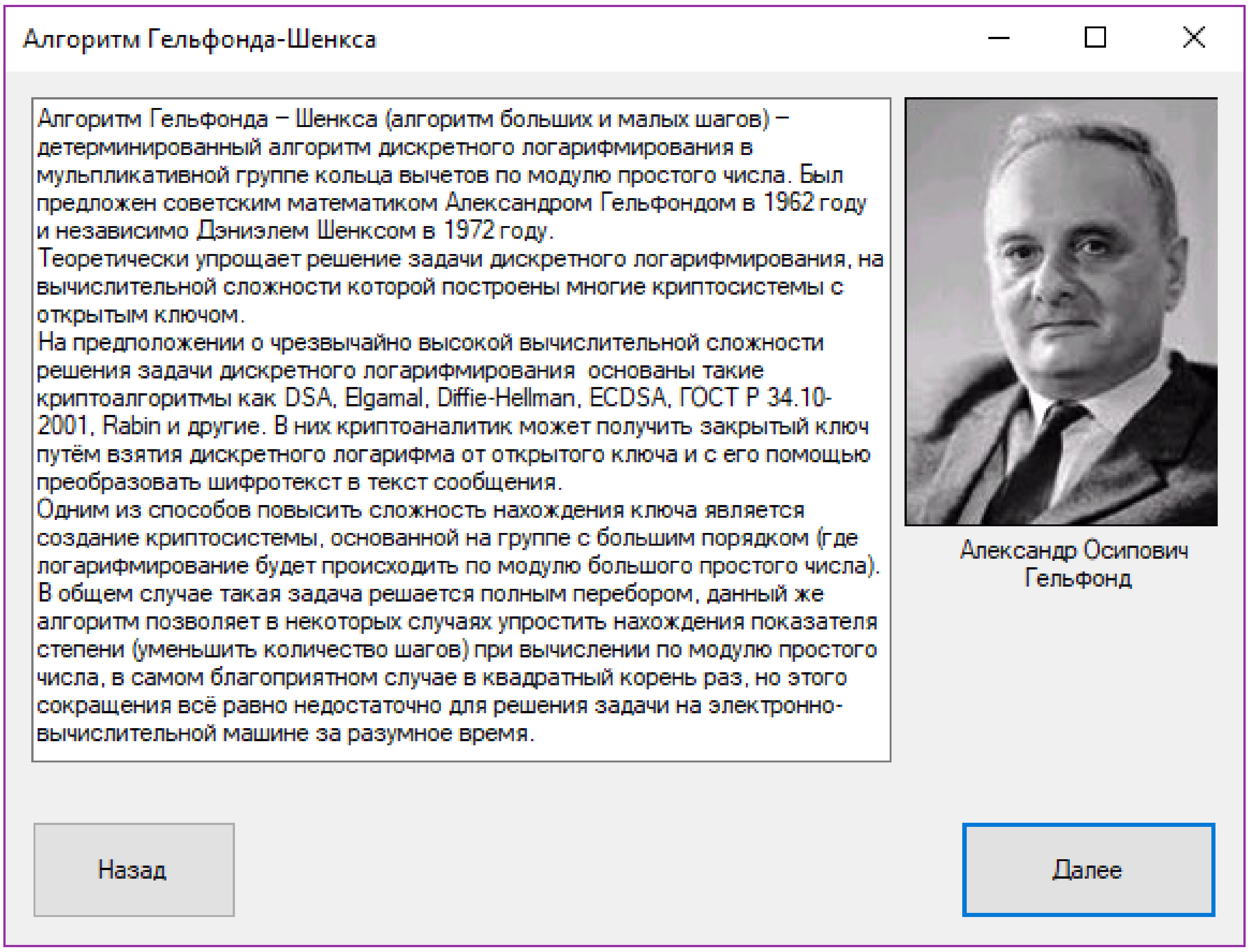

Рис. 32: Общая информация об алгоритме Гельфонда-Шенкса

На следующем шаге пользователь знакомится с математическим обоснованием алгоритма Гельфонда-Шенкса и, на ещё одном шаге пользователю демонстрируются шаги алгоритма Гельфонда-Шенкса, записанные предвокодом.

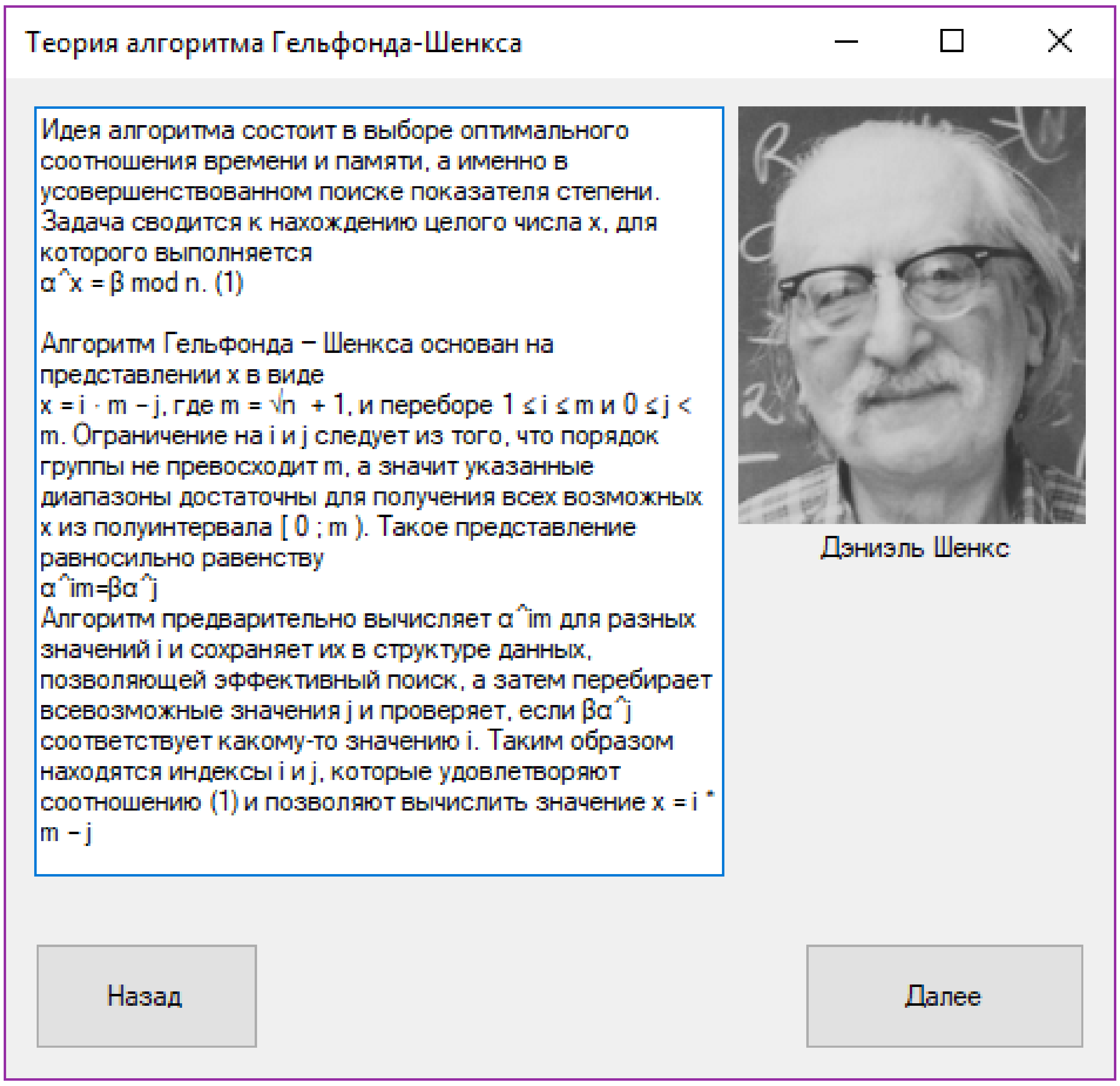

Рис. 33: Математическое обоснование алгоритма Гельфонда-Шенкса

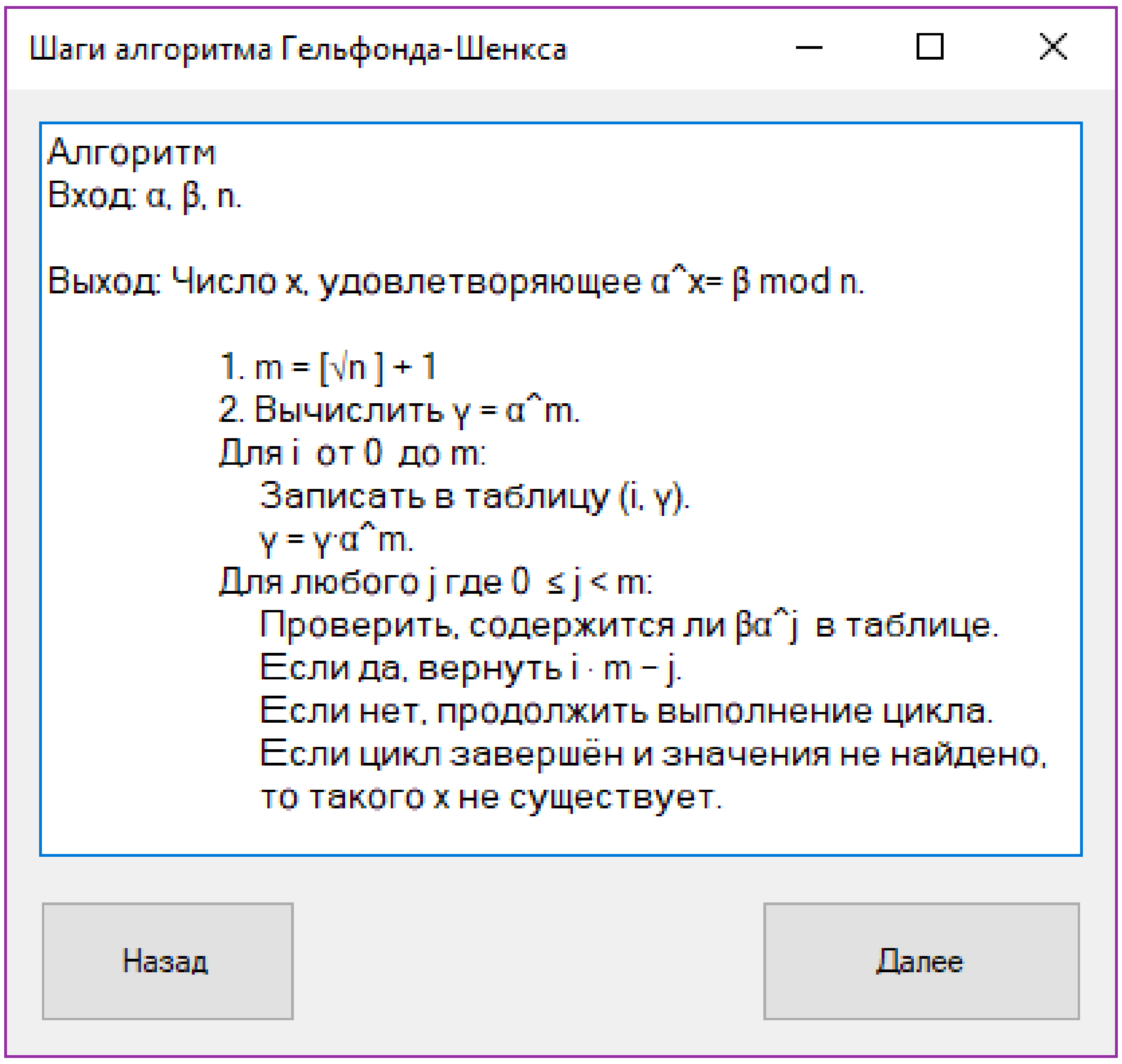

Рис. 34: Псевдокод алгоритма Гельфонда-Шенкса

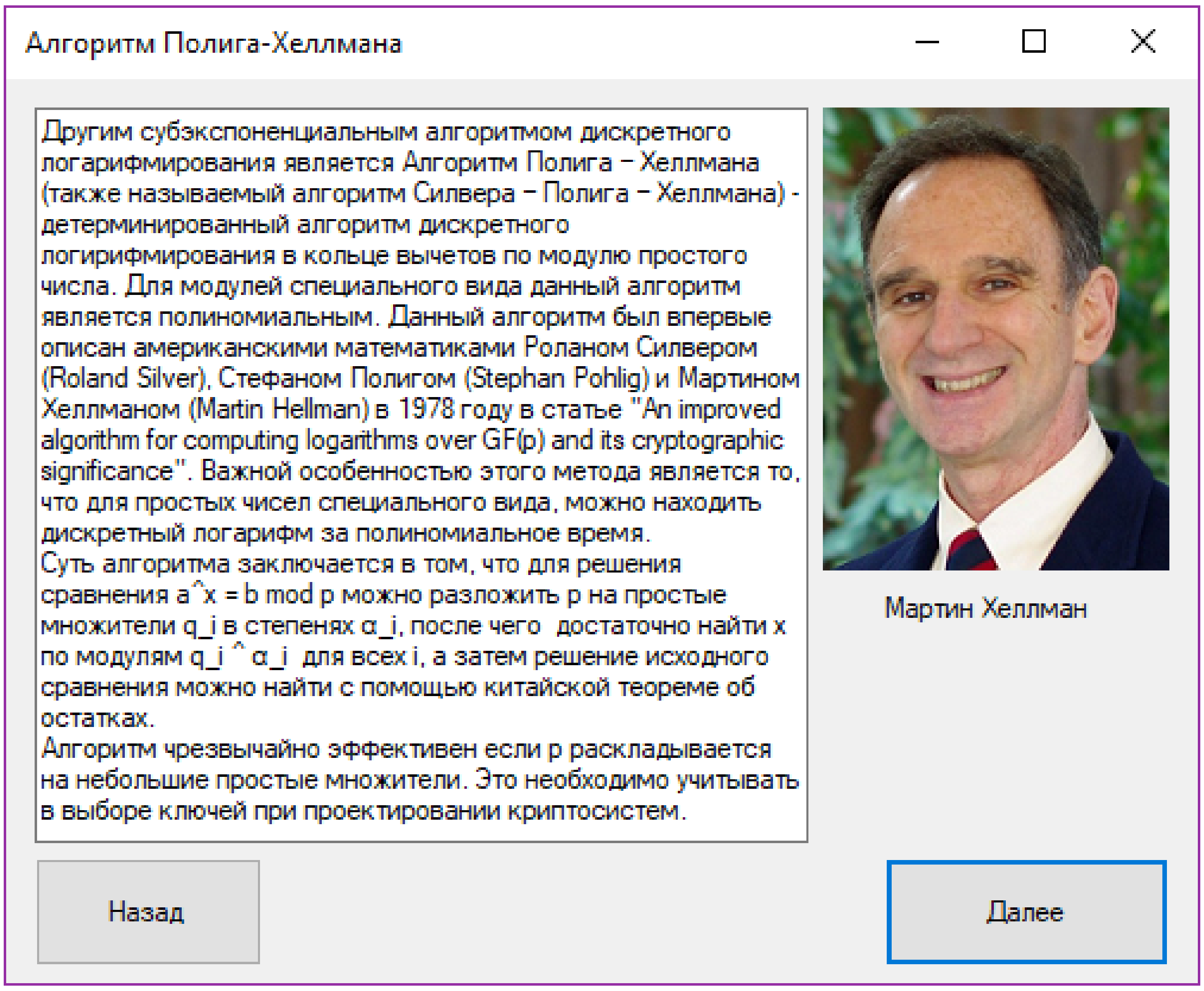

Рис. 35: Общая информация об алгоритме Полига-Хеллмана

На следующих трёх шагах рассказывается о другом алгоритме дискретного логарифмирования - алгоритме Сильвера-Полига-Хеллмана. На первом из этих шагов пользователь получает общую информацию об этом алгоритме.

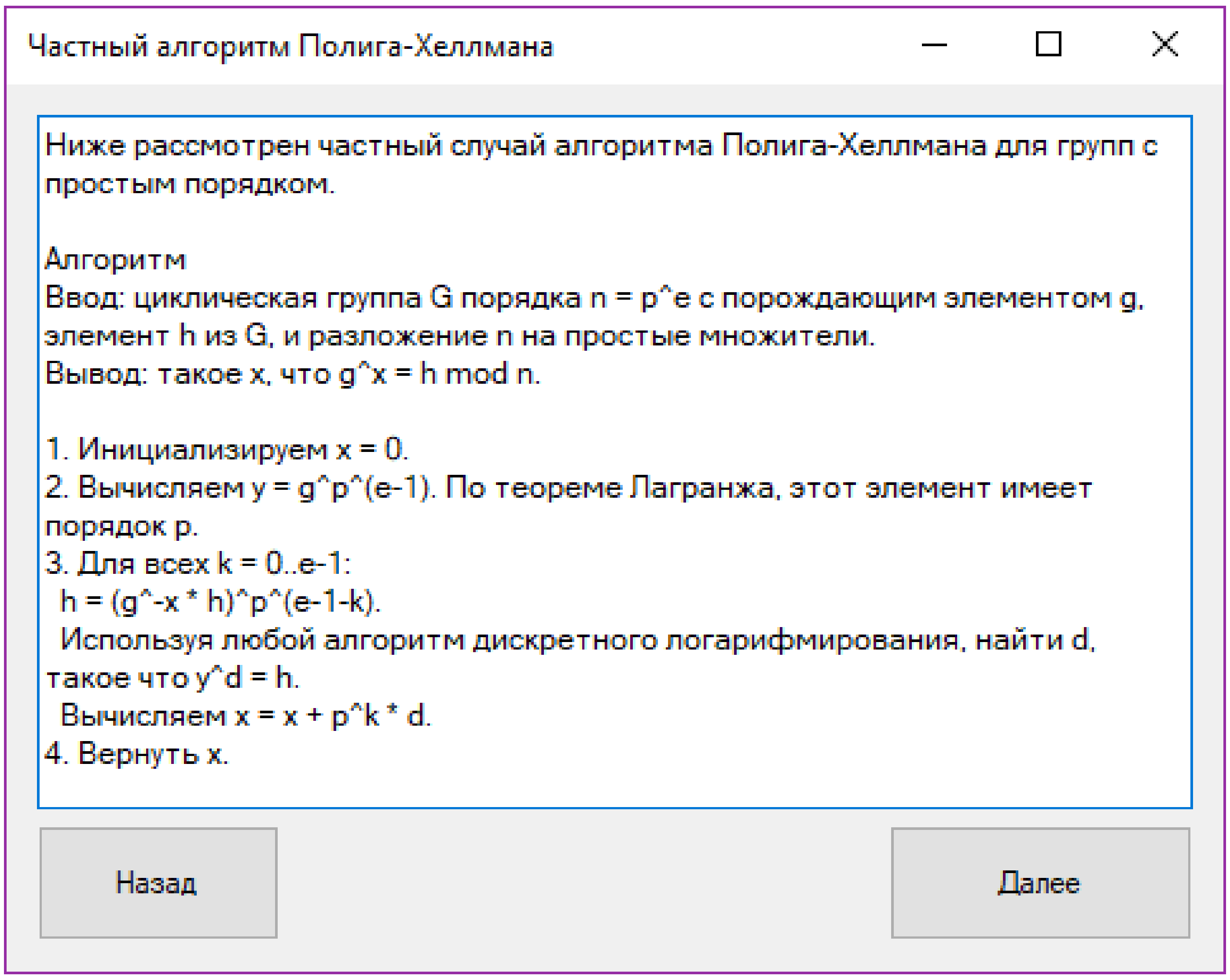

Рис. 36: Частный случай алгоритма Полига-Хеллмана

На втором из этих шагов обучения демонстрируется псевдокод частного случая алгоритма Полига-Хеллмана, применимого только для групп с простым порядком. Этот алгоритм используется в общем алгоритме Сильвера-Полига-Хеллмана.

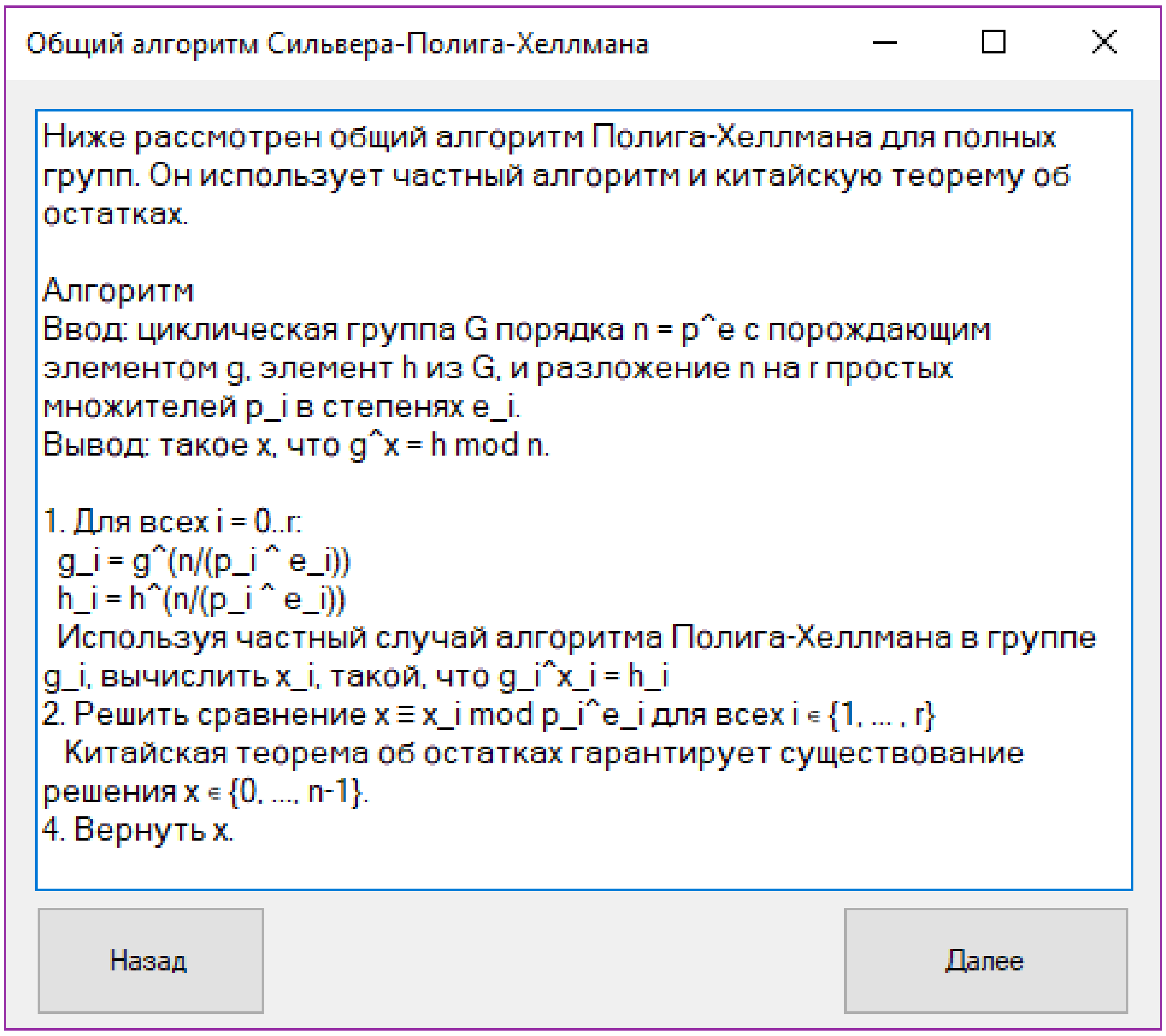

Рис. 37: Общий случай алгоритма Полига-Хеллмана

На третьем из этих шагов обучения демонстрируется псевдокод общего случая алгоритма Полига-Хеллмана, использующего рассмотренный выше частный алгоритм Сильвера-Полига-Хеллмана и китайскую теорему об остатках.

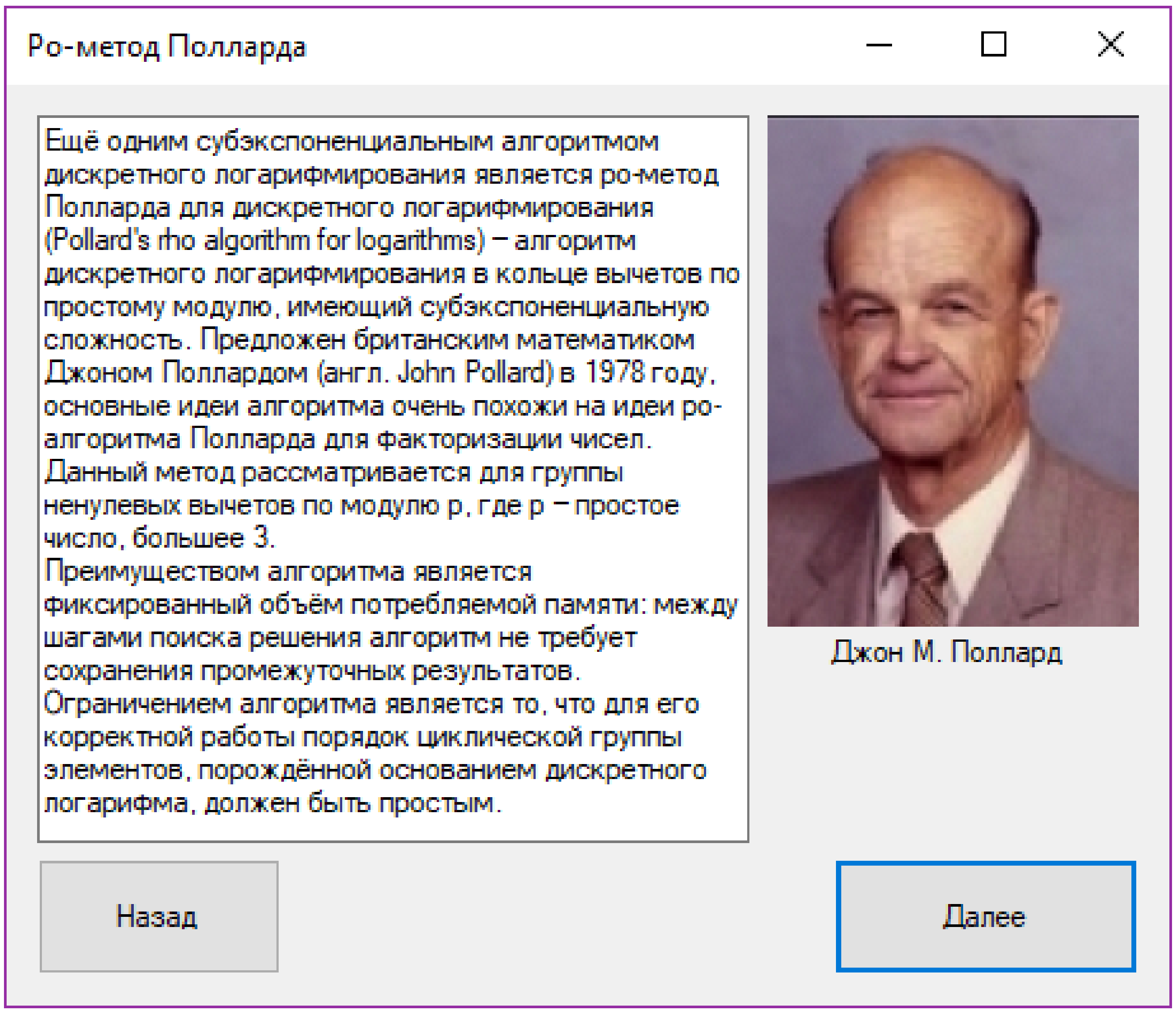

Рис. 38: Общая информация о ρ-методе Полларда

На следующем шаге обучения пользователь знакомится с общей информацией об ещё одном алгоритме дискретного логарифмирования - ρ-методе Полларда, его преимуществами и ограничениями.

На следующих пяти шагах пользователю предлагается пройти небольшое теоретическое по пройденным темам. Тестирование состоит из десяти вопросов с четырьмя вариантами ответов для каждого из них.

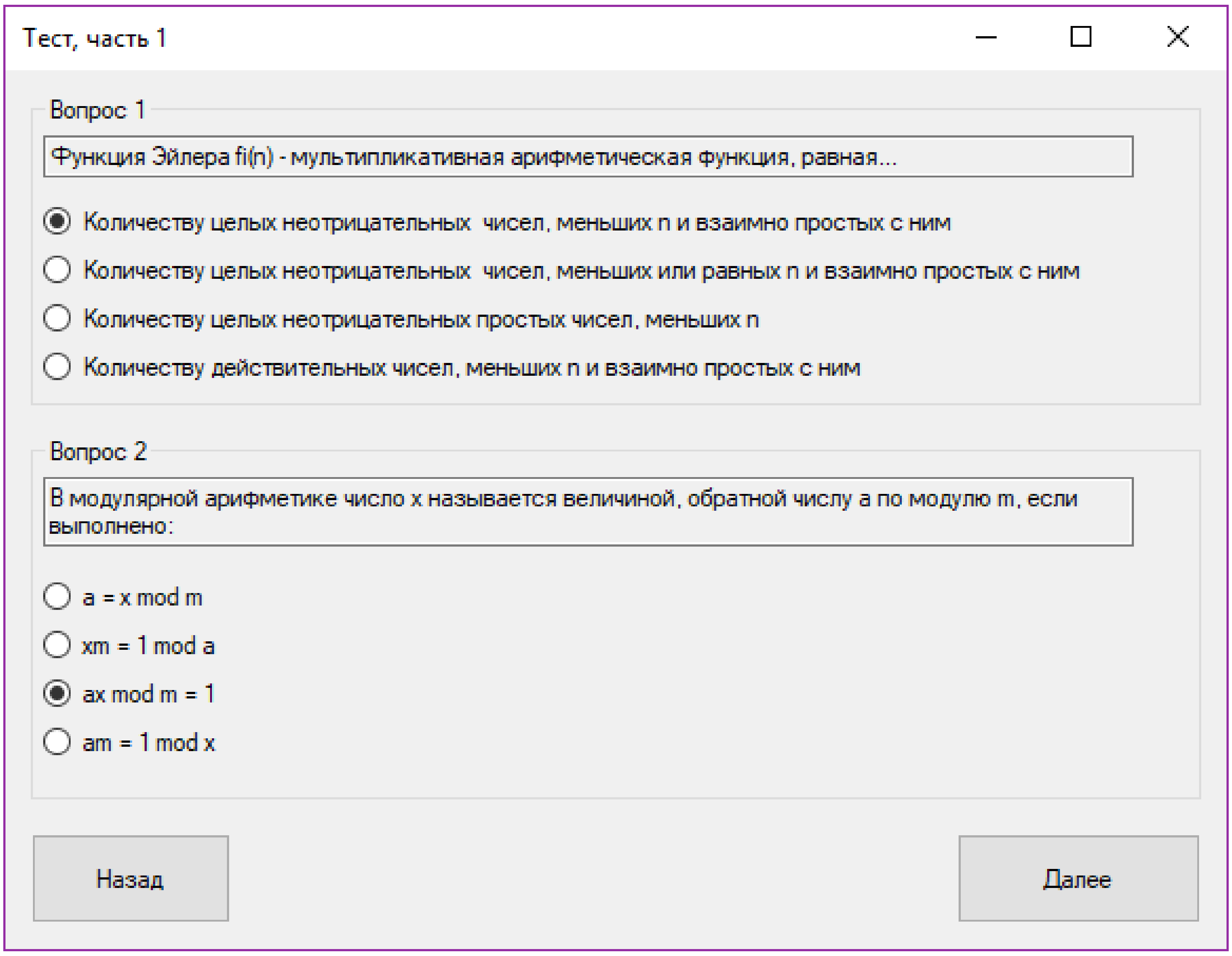

Рис. 39: Тест, часть 1

Первая часть тестирования включает в себя вопросы, касающиеся определения функции Эйлера и определения обратного по модулю числа.

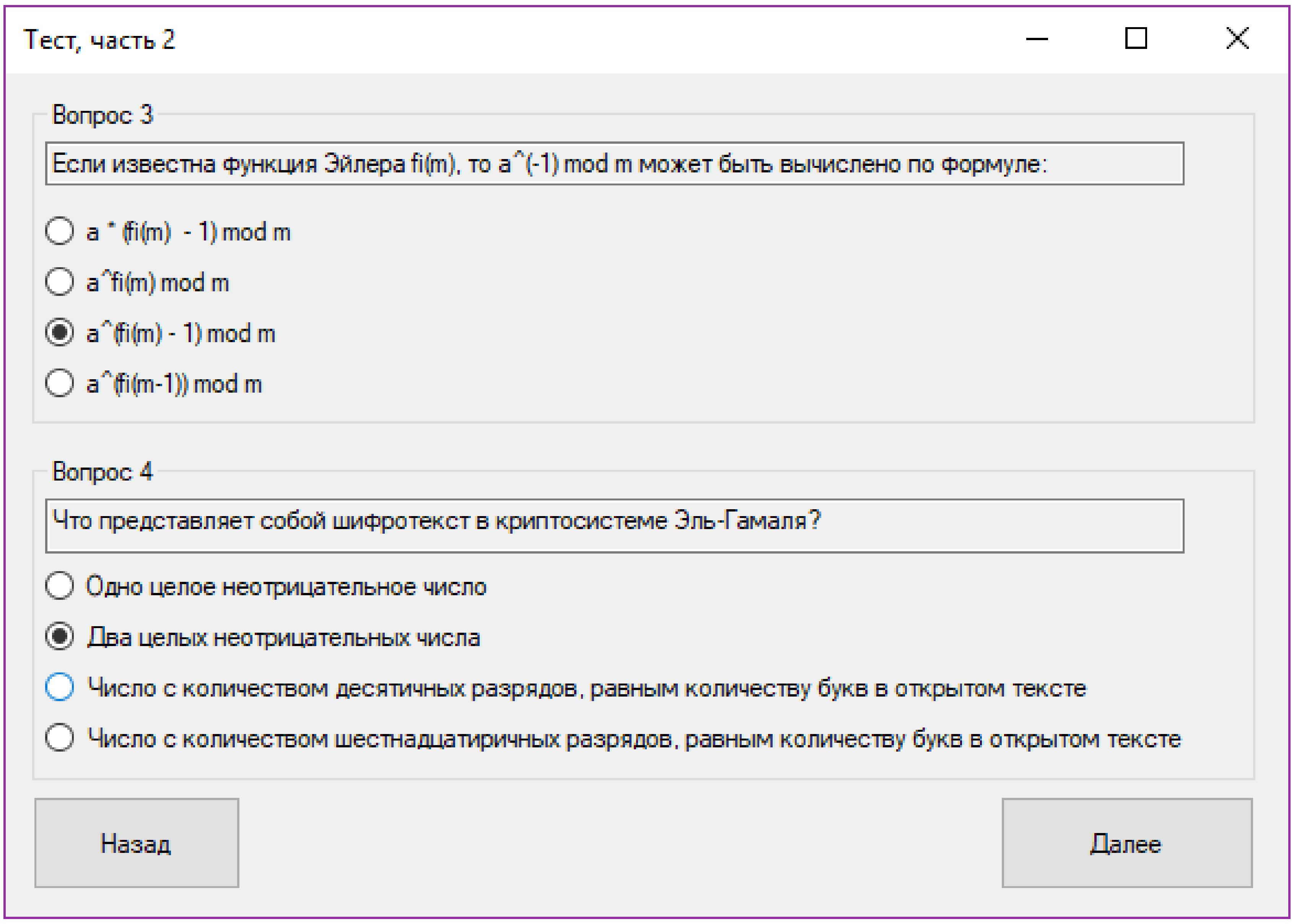

Рис. 40: Тест, часть 2

Вторая часть тестирования включает в себя два вопроса. Первый вопрос касается вычисления обратного числа по модулю с помощью функции Эйлера, второй вопрос касается вида широтекста в криптосистеме Эль-Гамаля.

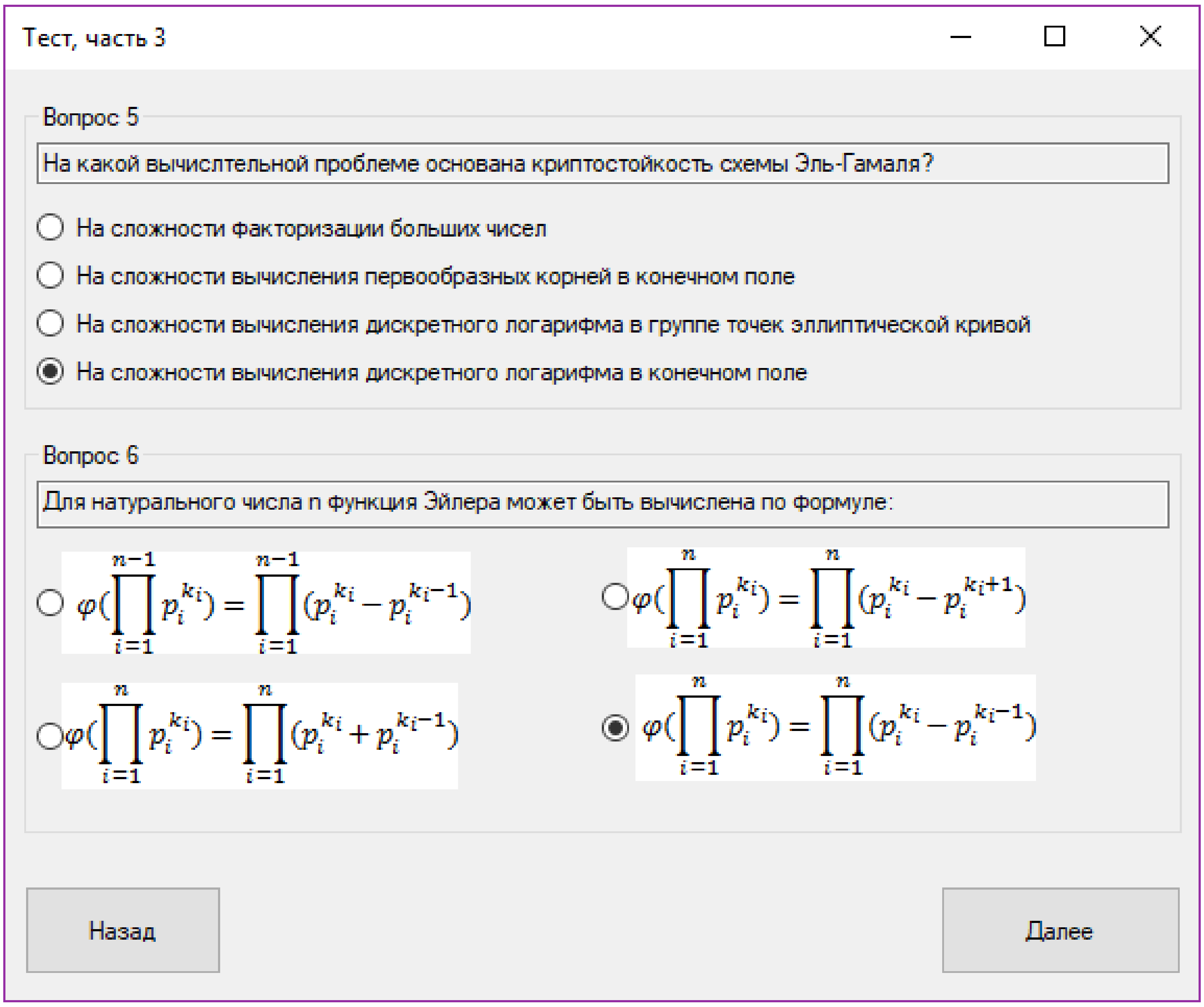

Рис. 41: Тест, часть 3

Следующие два вопроса касаются оснований криптографической стойкости криптосистемы Эль-Гамаля и общей формулы вычисления функции Эйлера.

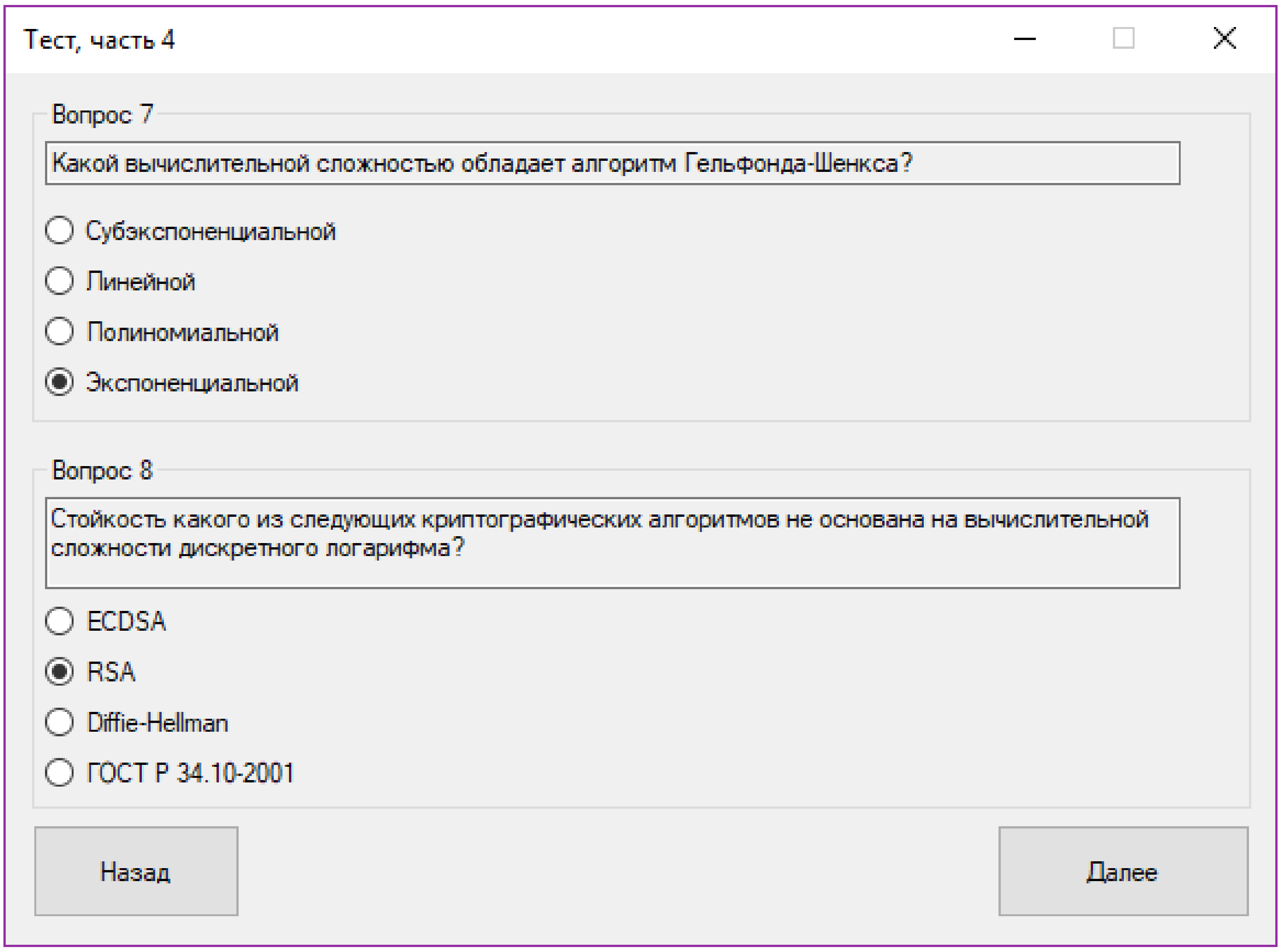

Рис. 42: Тест, часть 4

Седьмой вопрос посвящён вычислительной сложности алгоритма Гельфонда-Шенкса. Восьмой вопрос посвящён другим криптосистемам, основанным на криптографической стойкости дискретного логарифмирования.

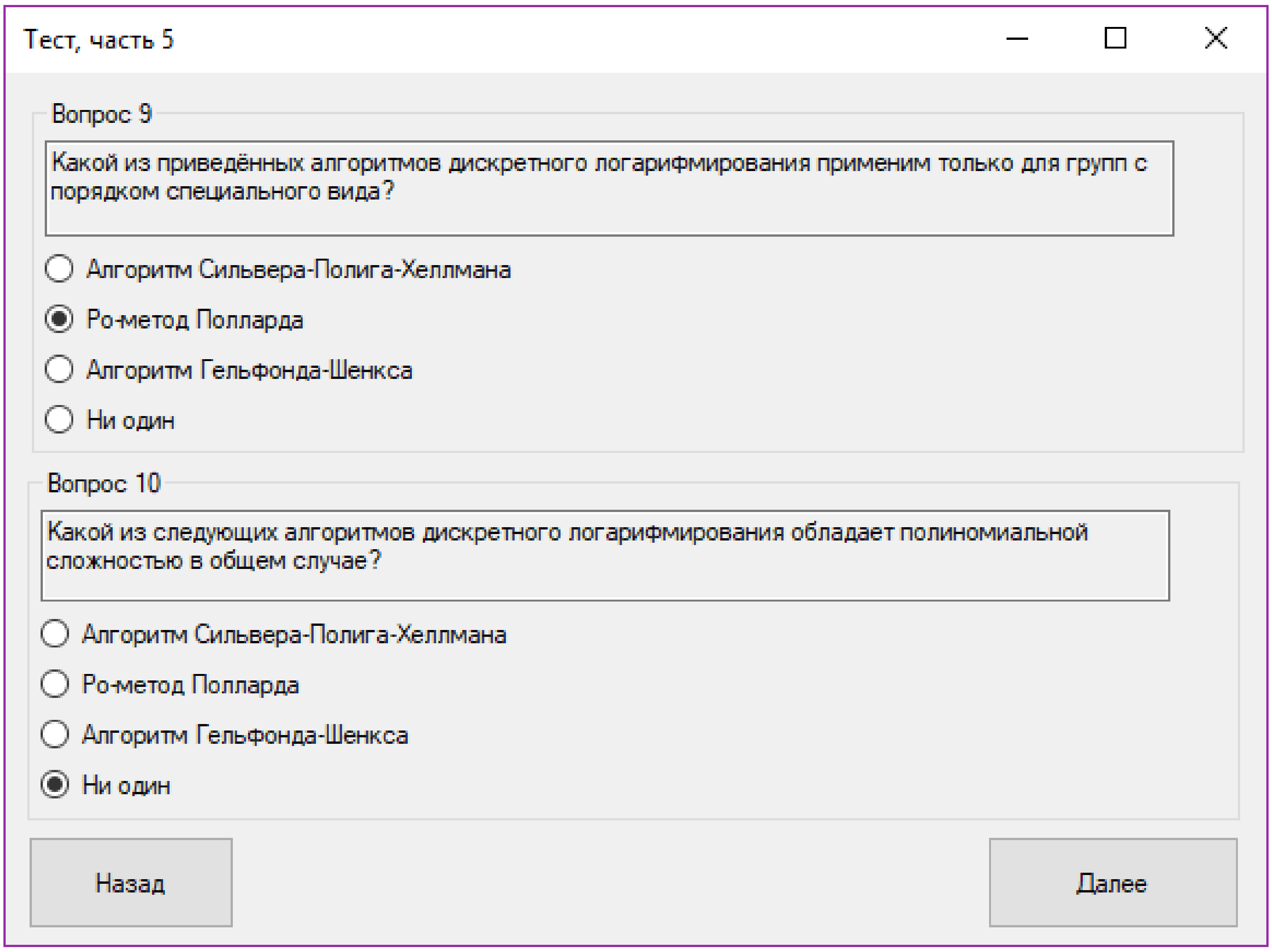

Рис. 43: Тест, часть 5

Наконец, пятая часть теста посвящена границам применимости и вычислительной сложности алгоритмов дискретного логарифмирования.

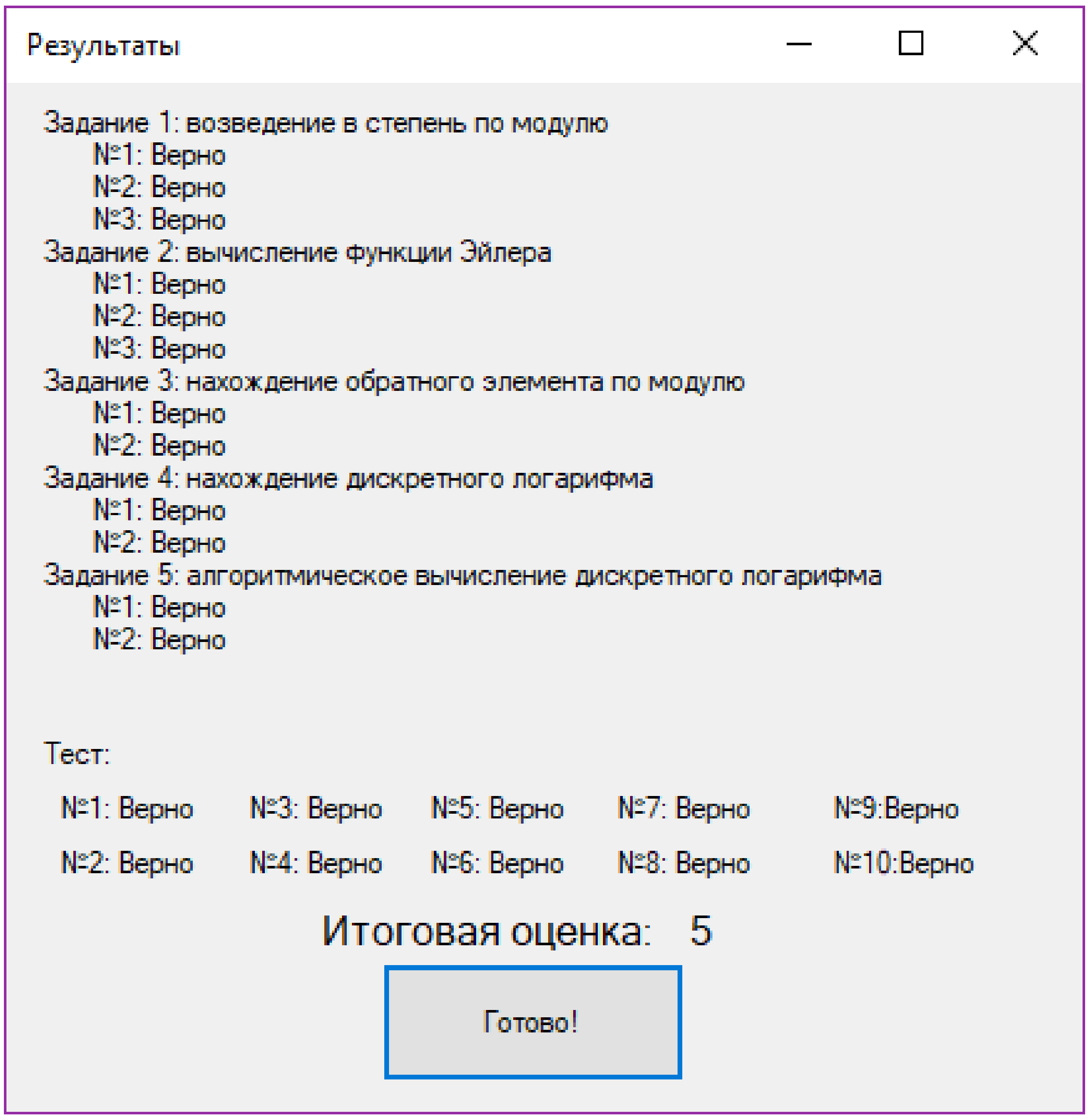

Рис. 44: Результаты проверки знаний

На завершающем шаге обучения показаны результаты выполненных заданий и ответов на вопросы теста, и пользователь получает оценку своих знаний.

5.6. Вызов и загрузка

Вызов программы осуществляется стандартными средствами системы Windows с установленной платформой Microsoft .NET Framework 4.5. Имя загрузочного модуля – ElgamalTutor.exe.

5.7. Входные и выходные данные

Входные данные – ответы на поставленные вопросы, выходные данные – результаты тестирования и примеры работы криптосистемы.

6. ОПИСАНИЕ СЦЕНАРИЯ ЛАБОРАТОРНОЙ РАБОТЫ

6.1. Постановка задачи

1. Ознакомиться с обучающей компьютерной программой El-Gamal_Tutor.

2. Изучить и привести описание алгоритма Эль Гамаля (в соответствии с обозначениями из [**?**]) с доказательством корректности алгоритма, его достоинствами и недостатками.

3. Зафиксировать (для отчета) последовательность этапов обучения в программе El-Gamal_Tutor.

4. Провести тестирование программы El-Gamal_Tutor с целью выявления ошибок и недочетов.

5. С помощью математического пакета прикладных программ произвести шифрование и расшифрование сообщения, заданного в виде одного блока открытого текста. При этом длина ключей должна удовлетворять условиям: $|p|, |\delta| \geq 80$, $|r|, |\alpha| \geq 40$. Ключи описываются соотношениями: $K = (p, \alpha, \beta, \delta) : \alpha^{\delta} \equiv \beta (mod\ p)$, где $K = (k_O; k_S)$; $k_O = (p, \alpha, \beta)$ – открытый ключ; $k_S = (\delta)$ – закрытый ключ.

6. Сформулировать и обосновать принципы работы алгоритма Эль Гамаля.

7. Одним из методов решения задачи дискретного логарифмирования осуществить криптоанализ заданного шифрованного текста на основе известных составляющих открытого ключа (p, α, β).

8. Ответить на контрольные вопросы.

9. Составить и защитить отчет о проделанной работе.

6.2. Содержание отчета о выполнении лабораторной работы

1. Постановка задачи
2. Описание криптосистемы Эль Гамаля.
3. Последовательность этапов и результаты обучения с использованием программы El-Gamal_Tutor.
4. Выявление ошибок и недочетов в обучающей программе El-Gamal_Tutor.
5. Результаты шифрования и расшифрования с использованием ППП Maple.
6. Принципы работы алгоритма Эль Гамаля.
7. Последовательность этапов и результаты криптоанализа.
8. Ответы на контрольные вопросы.
9. Выводы
10. Библиография

ЗАКЛЮЧЕНИЕ

Практически в каждой коммерческой, военной и государственной отрасли требуется наличие системы, позволяющей шифровать данные, делая информацию, заложенную в эти данные, невозможной для воспроизведения обычным пользователем. Для получения нужных для этого кадров необходим эффективный процесс обучения специалистов в данной отрасли.

В обучении будущих специалистов в любой области немаловажную роль играет использование наглядных средств обучения, эксперимент (в том числе вычислительный) и проверка полученных знаний на практике. В современной образовательной системе успешно используются обучающие программы, лучшие из которых, как правило, сочетают в себе все эти три фактора. В настоящей работе был предложен вариант лабораторной работы по защите информации, специально для которой была разработана обучающая программа, проектировавшаяся в первую очередь исходя из всех вышеуказанных принципов.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. **Гилбарг, Д.** Эллиптические дифференциальные уравнения с частными производными второго порядка / Д. Гилбарг, П. Трудингер. — М. : Наука, 1989. — 464 с.

2. **Ильин, В. А.** О рядах Фурье по фундаментальным системам функций оператора Бельтрами/В. А. Ильин // Дифференц. уравнения. — 1969. — Т. 5, № 11. — С. 1940–1978.

3. **Ильин, В. А.** Некоторые свойства регулярного решения уравнения Гельмгольца в плоской области / В. А. Ильин // Мат. заметки. — 1974. — Т. 15, № 6. — С. 885–890.

4. **Ильин, В. А.** Об одном обобщении формулы среднего значения для регулярного решения уравнения Шредингера / В. А. Ильин, Е. И. Моисеев // ИПМ АН СССР, 1977. — С. 157–166.

5. **Ильин, В. А.** Формула среднего значения для присоединенных функций опера- тора Лапласа / В. А. Ильин, Е. И. Моисеев // Дифференц. уравнения. — 1981. — Т. 17, № 10. — С. 1908–1910.

6. **Моисеев, Е. И.** Формула среднего для собственных функций эллиптического са- мосопряженного оператора второго порядка / Е. И. Моисеев // Докл. АН СССР. — 1971. — Т. 197, № 3. — С. 524–525.

7. **Моисеев, Е. И.** Асимптотическая формула среднего значения для регулярного решения дифференциального уравнения / Е. И. Моисеев // Дифференц. уравнения. — 1980. — Т. 16, № 5. — С. 827–844.

8. **Хелгасон, С.** Дифференциальная геометрия и симметрические пространства / С. Хелгасон. — М. : Мир, 1964. — 534 с.

9. **Иванов, Л. А.** О некоторых свойствах оператора Бельтрами в римановой метри- ке / Л. А. Иванов, И. П. Половинкин // Докл. РАН. — 1999. — Т. 365, № 3. — С. 306–309.

10. **Йон, Ф.** Плоские волны и сферические средние / Ф. Йон. — М. : Иностр. лит., 1958. — 158 с.

11. **Бицадзе, А. В.** К теории уравнений смешанного типа в многомерных областях / А. В. Бицадзе, А. М. Нахушев // Дифференц. уравнения. — 1974. — Т. 10, № 12. — С. 2184–2191.

12. **Гельфанд, И. М.** Обобщенные функции и действия над ними / И. М. Гельфанд, Г. Е. Шилов. — М. : Физматлит, 1958. — 440 с.

13. **Хермандер, Л.** Анализ линейных дифференциальных операторов с частными производными. Т. 1 / Л. Хермандер. — М. : Мир, 1986. — 464 с.

14. **Мешков, В. З.** К свойствам решений линейных уравнений в частных производных / В. З. Мешков, И. П. Половинкин // Черноземный альманах научных исследований. — Сер. Фундамент. математика. — 2007. — Вып. 1(5). — С. 3–11.

15. **Мешков, В. З.** Разностная формула среднего значения для двумерного линейного гиперболического уравнения третьего порядка / В. З. Мешков, И. П. Половинкин, М. В. Половинкина, Ю. Д. Ермакова, С. А. Рабееах // ВЕСТНИК ВГУ. СЕРИЯ: ФИЗИКА. МАТЕМАТИКА. — 2015. —
№ 3

16. **Половинкин, И.П.** К свойствам решений линейных уравнений в частных производных / Половинкин И.П. // Вестник Челябинского государственного университета. Математика. Механика. Информатика. Выпуск 12. — 2010. — № 23(204) — С. 59–66.

17. **Половинкин, И.П.** О получении новых формул среднего значения для линейных дифференциальных уравнений с постоянными коэффициентами / Половинкин И.П., Мешков В.З. // Дифференциальные уравнения. — 2011 — - Т. 47, № 12 — С. 1724–1791.

18. **Половинкин, И.П.** Дополнения к свойствам средних значений решений линейных дифференциальных уравнений с постоянными коэффициентами / Половинкин И.П., Мешков В.З. // Дифференциальные уравнения. — 2011 — Т. 47, № 11 — С. 1669 – 1671.