ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

Кафедра «Гидравлика, гидропневмоавтоматика и тепловые процессы»

Учебное пособие

по дисциплине

«Системы автоматизированного управления проектами»

Авторы
Кожухова А.В.,
Полешкин М.С.

Ростов-на-Дону, 2016

Аннотация

Учебное пособие по дисциплине «Системы автоматизированного управления проектами» состоит из теоретического курса, цикла практических работ для закрепления полученных знаний и вопросов для самостоятельной подготовки, используемых при изучении дисциплины и при выполнении курсовых и выпускных работ студентами направления 27.03.05 «Инноватика».

Пособие рекомендовано студентам 2,3,4 курсов направления 27.03.05 «Инноватика» очной и заочной форм обучения.

Оглавление

ВВЕДЕНИЕ

CALS-технология – это технология комплексной компьютеризации сфер промышленного производства, цель которой – унификация и стандартизация спецификаций промышленной продукции на всех этапах ее жизненного цикла. Основные спецификации представлены проектной, технологической, производственной, маркетинговой, эксплуатационной документацией.

В CALS-системах предусмотрены хранение, обработка и передача информации в компьютерных средах, оперативный доступ к данным в нужное время и в нужном месте. Соответствующие системы автоматизации назвали автоматизированными логистическими системами или CALS (Computer Aided Logistic Systems). Поскольку под логистикой обычно понимают дисциплину, посвященную вопросам снабжения и управления запасами, а функции CALS намного шире и связаны со всеми этапами жизненного цикла промышленныхизделий, то применяют и более соответствующую предмету расшифровку аббревиатуры CALS – Continuous Acquisition and LifeCycle Support [1].

Применение CALS позволяет существенно сократить объемы проектных работ, так как описания многих составных частей оборудования, машин и систем, проектировавшихся ранее, хранятся в базах данных сетевых серверов, доступных любому пользователю технологии CALS. Существенно облегчается решение проблем ремонтопригодности, интеграции продукциив различного рода системы и среды, адаптации к меняющимся условиям эксплуатации, специализации проектных организаций и т. п. Ожидается, что ус-пех на рынке сложной технической продукции будет немыслим вне технологии CALS. Развитие CALS-технологии должно привести к появлению так называемых виртуальных производств, в которых процесс создания спецификаций с информацией для программно управляемого технологического оборудования, достаточной для изготовления изделия, может быть распределен вовремени и пространстве между многими организационно автономными проектными студиями. Среди несомненных достижений CALS-технологии следует отметить легкость распространения передовых проектных решений, возможность многократного воспроизведения частей

проекта в новых разработках и др.

Одним из важных компонентов CALS-систем являются программные средства шифрования данных для осуществления электронной обработки и передачи данных о проекте. Данное учебное пособие содержит методические рекомендации по выполнению практических работ с использованием самых распространенных программных средств шифрования данных.

ПРАКТИЧЕСКАЯ РАБОТА №1

Алгоритм асимметричного шифрования RSA

1. Цель работы

1.1 Изучить назначение, особенности и структуру алгоритма RSA, области его применения в информационных процессах предприятий.

1.2 Приобретение практических навыков по разработке, составлению и реализации алгоритмов RSA в программной среде C++ Visual Studio 2008.

1.3 Закрепление теоретических знаний по разделу «Шифрование и ЭЦП».

2. Краткие теоретические сведения

Алгоритм RSA предложили в 1978 г. три автора: Роналдьд Райвест (Ronald Rivest), Ади Шамир (Adi Shamir) и Леонард Адльман (Leonard Adlman). Алгоритма получил свое название по первым буквам фамилий их авторов. Алгоритм RSA стал первым полноценным алгоритмом с открытым ключом, который может работать как в режиме шифрования данных, так и в режиме электронной цифровой подписи [2].

Надежность алгоритма основывается на трудности факторизации большихз чисел и трудности вычисления дискретных логарифмов.

Первый этап любого асимметричного алгоритма – создание пары ключей – состоит для схемы RSA из следующих операций.

1. Выбираются два больших простых числа p и q (простым называется число, делящееся на единицу и на само себя).

2. Вычисляется n, равное ($p \times q$).

3. Выбирается произвольное число e ($e < n$), такое,

что наибольший общий делитель НОД (e, (p-1) × (q-1)) = 1, т. е. должно быть взаимно простым с числом (p-1) × (q-1).

4. Методом Евклида решается в целых числах уравнение e × d + (p-1) × (q-1) × y = 1. Здесь неизвестными являются переменные d и y – метод Евклида как раз и находит множество пар (d, y), каждая из которых является решением уравнения в целых числах.

5. Пара чисел (e, n) – публикуется как открытый ключ. Число d хранится в строжайшем секрете – это и есть закрытый ключ, который позволит читать исе послания, зашифрованные с помощью пары ключей (e, n).

Второй этап – собственно шифрование с помощью открытого ключа.

1. Отправитель разбивает свое сообщение на блоки, равные $k = [\log_2 (n)]$, где квадратные обозначают взятие целой части от дробного числа. Подобный блок может быть интерпретирован как число из диапазона $(0 : 2^k – 1)$.

2. Для каждого такого числа (назовем его m_i) вычисляется выражение $c_i = ((m_i)^e) \bmod n$. Блоки c_i и есть зашифрованное сообщение. Их можно без опасения передавать по открытому каналу, поскольку операция возведения в степень по модулю простого числа является трудноразрешимой математической задачей.

Третий этап – дешифрование послания с помощью секретного ключа. Частный случай теоремы Эйлера утверждает, что если число n может быть представлено в виде произведения двух простых чисел p и q, то для любого x имеет место равенство:

$$(x^{(p-1) \times (q-1)}) \bmod n = 1.$$

Для дешифрования RSA – сообщений воспользуемся этой формулой. Возведем обе ее части в степень (-y): $((x^{(-y)} \times (p-1) \times (q-1)}) \bmod n = 1^{(-y)} = x$. После умножения обеих частей равенства на x получим:

$$(x^{(-y) \times (p-1) \times (q-1)}) \bmod n = 1 \times = x.$$

Далее вернемся к созданию открытого и закрытого ключей. Величина d была подобрана с помощью алгоритма Евклида так, что:

$$e \times d + (p-1) \times (q-1) \times y = 1, \text{ т. е.}$$
$$e \times d = 1 + (-y) \times (p-1) \times (q-1).$$

Следовательно, в последнем выражении предыдущего абзаца мы можем заменить показатель степени на число (e×d). Получаем выражение:

$$(x^{(e \times d)}) \bmod n = (x^{(-y) \times (p-1) \times (q-1)+1}) \bmod = m_i$$

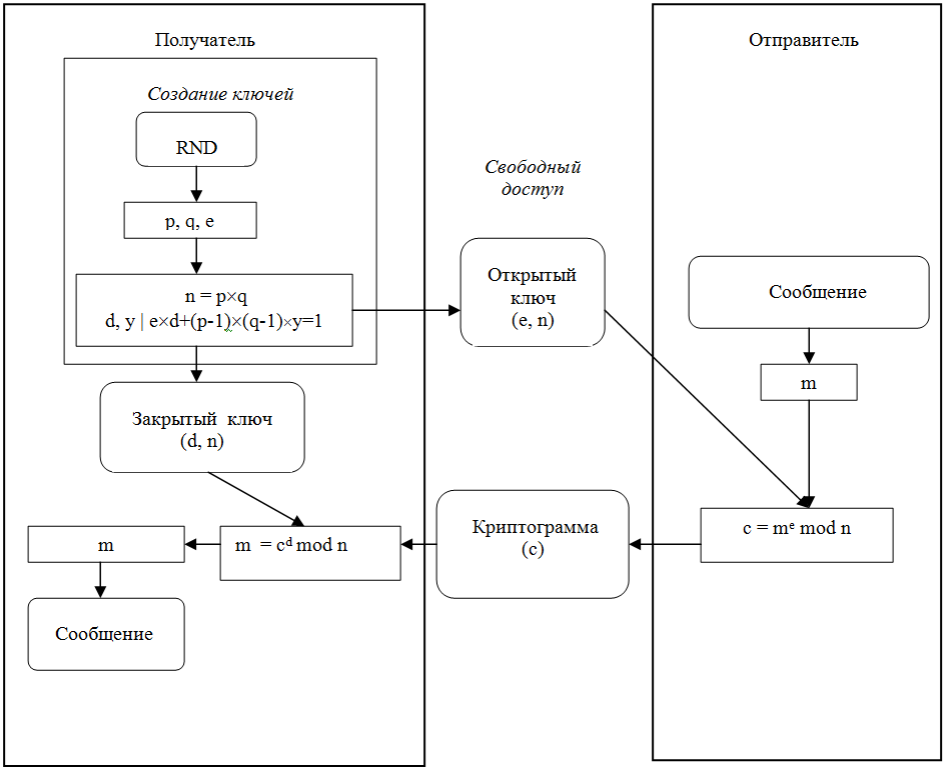


Рис. 1.1. Структура криптосистемы RSA

Общий вид криптосистемы RSA приведен на рис. 1.1.

Рассмотрим работу схемы RSA на примерах шифрования небольших чисел. Небольшие числа используются для простоты (на практике применяются числа, которые намного больше).

3. Порядок выполнения работы

Работа выполняется, решая две представленные задачи, а затем проверяя при помощи компьютерного программы правильность их решения.

Задание 1. Числовое шифрование. Пусть p = 5, а q = 11, тогда значение n = 55. В качестве открытого ключа e выберем число 7, таким образом, весь открытый ключ имеет вид (e=7, n=55).

Вычислим закрытый ключ d: уравнение $e \times d + (p-1) \times (q-1) \times y = 1$ приобретает вид

$7 \times d + 40 \times y = 1$ и имеет в целых числах решение d = 23, y = - 4. Таким образом, закрытым ключом являются числа (23, 55).

Пусть произвольный отправитель хочет передать абоненту комбинацию бит 100111_2 , ее числовой эквивалент 39_{10} . Возводим 39 в степень открытого ключа e = 7 по модулю n = 55: $(39^7 \bmod 55) = 19$. Число 39 является шифрограммой и передается по каналу связи. Получатель по приходу сообщения возводит его в степень d = 23: $(19^{23} \bmod 55) = 39$. Исходное значение восстановлено.

Задание 2. Зашифруем сообщение “CAB”.

1. Выберем p=3 и q=11.
2. Определим n=3*11=33.
3. Найдем n=(p-1)(q-1)=20. Выберем в качестве d, число взаимно простое с 20, например, d = 3. Взаимно простые числа делятся только на 1 и на само себя.
4. Выберем число e. В качестве такого числа может быть взято любое число, для которого

удовлетворяется соотношение (e×3) (mod 20) = 1, например 7.

5. Представим шифруемое сообщение как последовательность целых чисел с помощью отображения: A→1, B→2, C→3. Тогда сообщение принимает вид (3,1,2). Зашифруем сообщение с помощью ключа {7,33}.
   ШТ1 = (3^7) (mod 33) = 2187 (mod 33) = 9,
   ШТ2 = (1^7) (mod 33) = 1 (mod 33) = 1,
   ШТ3 = (2^7) (mod 33) = 128 (mod 33) = 29.
6. Расшифруем полученное зашифрованное сообщение (9,1,29) на основе закрытого ключа {3,33}:

ИТ1 = (9^3) (mod 33) = 729 (mod 33) = 3,
ИТ2= (1^3) (mod 33) = 1 (mod 33) = 1,
ИТ3 = (29^3) (mod 33) = 24389 (mod 33) = 2.
Здесь ШТ – шифротекст, ИТ – исходный текст.

Итак, в реальных системах алгоритм RSA реализуется следующим образом: каждый пользователь выбирает два больших простых числа, и в соответствии с описанным выше алгоритмом выбирает два простых числа e и d. Как результат умножения первых двух чисел (p и q) устанавливается n.

{e,n} образует открытый ключ, а {d,n} - закрытый (хотя можно взять и наоборот).

Открытый ключ публикуется и доступен каждому, кто желает послать владельцу ключа сообщение, которое зашифровывается указанным алгоритмом. После шифрования, сообщение невозможно раскрыть с помощью открытого ключа. Владелец же закрытого ключа без труда может расшифровать принятое сообщение.

Скорость шифрования, обеспечиваемая двухключевыми (асимметричными) шифрами, на несколько порядков ниже скорости, которой обладают одноключевые (симметричные) криптосистемы. Поэтому наиболее эффективны гибридные криптосистемы, в которых информация шифруется с помощью одноключевых шифров, а распределение сеансовых ключей осуществляется по

открытому каналу с помощью двухключевых шифров. Например, используя криптосистему RSA , можно легко обменяться сеансовым ключом с любым абонентом, зашифровав сеансовый ключ с помощью его открытого ключа. Зашифрованный сеансовый ключ можно безопасно передать по открытому каналу связи, поскольку необходимым для дешифрования секретным ключом обладает только абонент, открытый ключ которого был использован для зашифрования.

Для непосредственного засекречивания информации двухключевые шифры находят ограниченное применение.

4. Экспериментальные исследования

Экспериментальные исследования алгоритмов шифрования, производятся в специальных программах и в следующей последовательности:

1) Программно реализовать алгоритм шифрования и дешифрования с помощью открытого ключа - алгоритм RSA для любых типов файлов.

Размер генерируемого ключа должен быть 32 (или 64, или 128) байт, или соответственно 256 (или 512, или 1024) бит.

2) Реализовать создание цифровой подписи с использованием RSA для любых типов файлов.

Выполнение задачи складывается из двух частей. Первая часть задачи *«Генерация ключей» RSA* наиболее сложная. Вторая часть наиболее легкая – это *кодирование и декодирование.*

Что бы приступить к любой из частей задачи, у Вас должен быть готов программный аппарат для **«длинной арифметики»**. Причем целесообразно максимальную длину (32 или 64, или 128 байт) задать константой, чтобы можно было легко перестроить программу на другую длину пакета.

4.1 Генерация открытого и закрытого ключей в алгоритме

RSA

Цель: найти три длинных числа – **n, e, d** (модуль, открытая и закрытая экспоненты)! Их длины должны быть равны заданной длине пакета. Прежде всего, находятся длинные простые числа **p** и **q** (их длина равна половине длины пакета).

Даже самый легкий и быстрый тест на простоту – «Тест Рабина-Миллера» (одноименные кнопки предназначены для каждого из чисел) занимает длительное время у современного компьютера (рис.1.2).

Для скорости последующих вычислений желательно получить длинные числа с большим количеством нулевых битов. Обратите внимание, на картинке **p** и **q**представленные в 16-ричной форме, а если 16-ричная цифра равна 0, то все 4 бита этой цифры, также будут равны нулям.

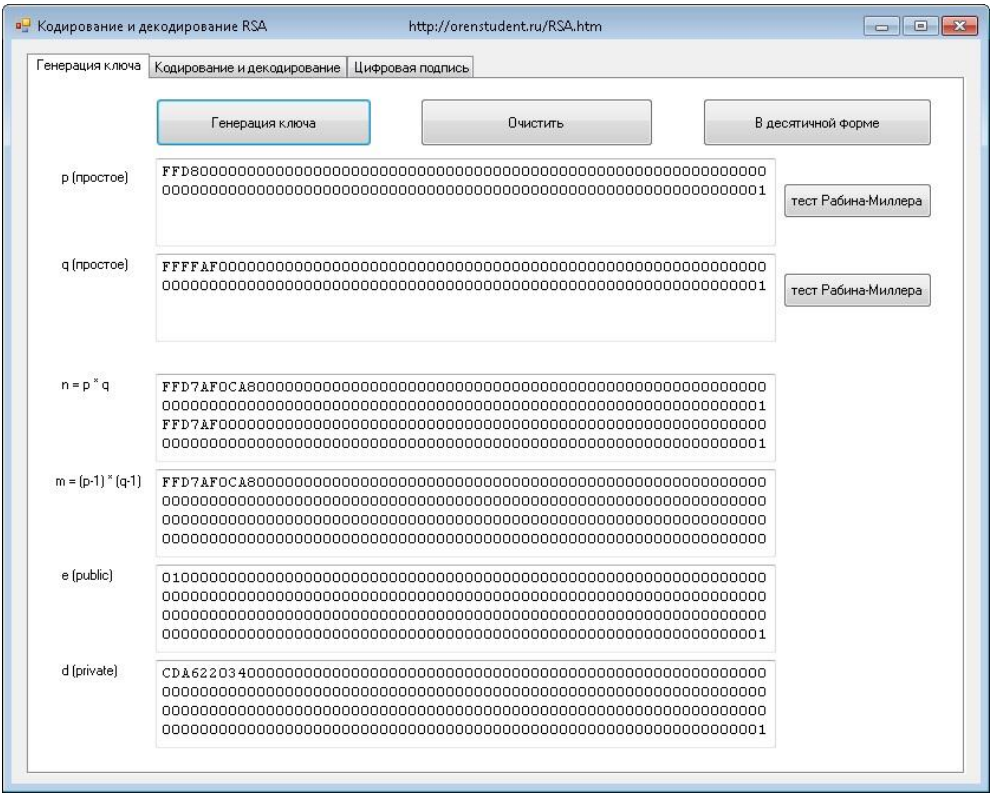


Рис.1.2. Окно программы по изучению криптосистемы RSA - генерация ключа

Далее (когда простые числа получены и видны на форме) необходимо щелкнуть кнопку **«Генерация ключа»** для автоматического заполнения пустых полей. Это уже легко и быстро: **n** (для нахождения остатка по модулю) получаем перемножением **p** и **q**.

Функцию Эйлера **m** тоже без труда получаем перемножением **p-1** и **q-1**. Открытая экспонента **e** выбирается большего по возможности размера, но с максимальным количеством нулевых битов (для скорости возведения в степень). Если текстовое поле для **e** пустое, то программа сама подберет значение

открытой экспоненты. Но если Вы пожелали ввести в поле для **e** свое длинное число до щелчка по кнопке **«Генерация ключа»**, то программа его примет по возможности. А вот закрытую экспоненту **d**, мультипликативно обратную к числу **e** по модулю **m**, вычисляет специальная функция:

*CLongRsaNum *Get_d(CLongRsaNum *m_op1, CLongRsaNum *e_op2);*

Эта функция входит в модуль **ArifmeticRSA.cpp** , как и сам класс длинных чисел **CLongRsaNum**. Видно, что, получая в виде параметров **m** и **e**, функция в результате вернет указатель на длинное число **d**, именно, мультипликативно обратное к числу **e** по модулю **m**…

4.2 Кодирование и декодирование по алгоритму RSA

Процесс **RSA кодирования**: файл рубится на пакеты заданной длины и вне зависимости, что находилось в файле (текст, музыка, изображение), каждый пакет рассматривается, как последовательность бит, а значит как длинное число.

Внимание! Кодирование производится открытым ключом того респондента, кому предназначено послание. **Zp = Op^e mod n** – закрытый пакет, длинное число, полученное путем взятия остатка по модулю **n** от возведения в степень открытой экспоненты **e** открытого пакета **Op**. Понятно, что длины пакетов **Op** и **Zp** будут совпадать.

Только человек, обладающий соответствующим закрытым ключом **(d и n)** сможет расшифровать пакеты послания. **Декодирование RSA** (обратная операция) **Op=Zp^d mod n**. Вот поэтому я и говорил, что вторая часть наиболее простая! И кодирование, и декодирование выполняется одной функцией (кодеком):

*void Main_codec(unsigned char *pack, int sizpack, CLongRsaNum *de, CLongRsaNum *n);*

Первый параметр – указатель на пакет (открытый или закрытый)

Второй параметр – размер пакета

Третий параметр – экспонента (открытая или закрытая)

Четвертый параметр – модуль для взятия остатка.

Здесь еще можно поговорить о том, почему вычисления ускоряются, если в экспоненте много нулевых битов. Давайте предположим, что экспонента равна 129 (или в двоичной форме 10000001)=128+1.

Тогда $\mathbf{Zp^{129} = Zp^{128} * Zp^{1}}$

Как видите, нам необходимо вычислять сомножители, возводя пакет в степень только тех степеней двойки (2^7 и 2^0), где в экспоненте расположены **1**. А все нулевые биты можно смело игнорировать… Хочу еще добавить, что возводить пакет в степень степени двойки очень легко.

$Zp^2 = Zp * Zp;$

$Zp^4 = Zp^2 * Zp^2;$

$Zp^8 = Zp^4 * Zp^4$ и так далее….

В цикле перемножаем само на себя значение, полученное на предыдущем шаге (на предыдущей итерации).

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Объясните структуру алгоритма шифрования RSA.
2. К какому типу криптоалгоритма (с точки зрения его устойчивости к взлому) и почему относится алгоритм RSA?
3. Какая трудноразрешимая математическая задача лежит в основе стойкости алгоритма RSA?
4. Назовите основные этапы данного шифрования;
5. Как осуществляется защита от несанкционированного доступа при данном шифровании?

ПРАКТИЧЕСКАЯ РАБОТА №2

АЛГОРИТМ ОБМЕНА КЛЮЧАМИ ДИФФИ-ХЕЛЛМАНА

1. Цель работы

1.1 Изучить назначение, особенности и структуру алгоритма Диффи–Хеллмана, области его применения в информационных процессах предприятий.

1.2 Приобретение практических навыков по разработке, составлению и реализации алгоритмов Диффи–Хеллмана в программной среде C++ Visual Studio 2008.

1.3 Закрепление теоретических знаний по разделу «Шифрование и ЭЦП».

2. Краткие теоретические сведения

В 1976 г. Диффи и Хеллман опубликовали статью, которая ознаменовала собой рождении асимметричной криптографии и привела к сильному росту числа открытых исследований в области криптографии. Она содержала ошеломляющий результат: возможно построение практически стойких секретных систем, которые не требуют передачи секретного ключа. Диффи и Хеллман ввели понятие односторонней функции с потайным ходом. Под односторонней функцией f понимается функция f(x), которая легко вычислима для любого значения аргумента x из области определения, однако для данного y из области ее значений вычислительно сложно нахождение значения аргумента x, для которого f(x) = y. Применение таких функций для защиты входа в вычислительную систему

путем одностороннего преобразования паролей было известно. Но как применить одностороннюю функцию в криптографических системах, когда даже законный получатель не сможет выполнить дешифрования? Для шифрования была предложена односторонняя функция с потайным ходом (секретом)[2[.

Под односторонней функцией с потайным ходом понимается семейство обратимых функций fz с параметром z , таких, что для данного z можно найти алгоритмы Ez и Dz , позволяющие легко вычислить значение fz(x) для всех x из области определения, а также вычислить значение fz-1(y) для всех y из области значений, однако практически для всех значений параметра z и практически для всех значений y из области значений fz нахождение fz-1(y) вычислительно неосуществимо даже при известном Ez .

В качестве односторонней функции Диффи и Хеллман предложили функцию дискретного возведения в степень:

$f(x) = g^x \pmod n$,

где x – целое число, $1 \le x \le n - 1$,

n – k-битовое простое число.

Причем выбирается такое число g < n , степени которого по модулю n представляют собой упорядоченное множество чисел {g1, g2, … , gn-1}, являющееся некоторой перестановкой чисел {1, 2, … , n-1}. (Такое число g называется первообразным корнем по модулю n.)

Даже для очень больших модулей n (например, при k = 1024 бит) для данного x легко вычислить значение этой функции. Процедура вычисления этой функции называется дискретным возведением в степень. Для выполнения этой процедуры достаточно выполнение около 2log2n операций умножения k-битовых чисел (или log2n умножений и log2n делений 2k-битовых чисел на k-битовые). Процедура дискретного возведения в степень основана на предварительном вычислений значений (по модулю n)

Обратной к функции дискретного возведения в степень является функция f -1(y), которая ставит в соответствие заданному значению y такое значение x, для

которого выполняется условие gx = y (mod n). Задача нахождения такого x называется задачей дискретного логарифмирования (нахождения дискретных логарифмов). Дискретные логарифмы сложно вычисляются, когда число n-1 содержит один большой простой множитель, например, когда оно представимо в виде n-1 = 2n′ , где n′ - простое число. При этом условии трудоемкость задачи нахождения дискретного логарифма равна примерно √n умножений по модулю n. Решение такой задачи является вычислительно неосуществимым при больших значениях k (например, при k≥ 512), а следовательно при указанных условиях, накладываемых на выбор чисел n и g , функция дискретного возведения в степень является односторонней.

Методом открытого распространения ключей Диффи – Хеллмана называется следующий способ использования дискретного возведения в степень для обмена секретными ключами между пользователями сети с применением только открытых сообщений. Выбирается большое простое число n и соответствующий ему первообразный корень g < n. (Для обеспечения стойкости рассматриваемой системы открытого шифрования на число n накладывается следующее условие: разложение числа n-1 на множители должно содержать по крайней мере один большой простой множитель; размер числа n должен быть не менее 512 бит.)

Механизм распределения секретных ключей по открытому каналу состоит в следующем. Каждый абонент выбирает случайный секретный ключ x и вырабатывает открытый ключ y, соответствующий выбранному секретному ключу, в соответствии с формулой

$$y = gx \pmod{n}.$$

Для любого значения x легко вычислить y, однако при размере числа n, равном 512 бит и более, вычислительно неосуществимо выполнение дискретного логарифмирования, а следовательно и определение числа x, для которого значение gx mod n равно заданному значению y.

Все абоненты размещают свои открытые ключи в

общедоступном справочнике. Данный справочник должен быть заверен специально созданным доверительным центром, чтобы исключить возможные нападения путем подмены открытых ключей или навязывания ложных открытых ключей. Если два абонента Боб и Алиса хотят установить секретную связь, то они поступают следующим образом.

Протокол обмена ключами Diffie-Hellman легко можно расширил» на случай с тремя и более участниками. В приводимом примере Алиса, Боб и Кэрол вместе генерируют секретный ключ.

(1) Алиса выбирает случайное большое целое число x и вычисляет

$$X = g^x \bmod n$$

(2) Боб выбирает случайное большое целое число y и посылает Кэрол

$$Y = g^y \bmod n$$

(3) Кэрол выбирает случайное большое целое число z и посылает Алисе

$$Z = g^z \bmod n$$

(4) Алиса посылает Бобу

$$Z' = Z^x \bmod n$$

(5) Боб посылает Кэрол *

$$X' = X^y \bmod n$$

(6) Кэрол посылает Алисе

$$Y' = Y^z \bmod n$$

(7) Алиса вычисляет

$$k = Y'^x \bmod n$$

(8) Боб вычисляет

$$k = Z'^{y} \bmod n$$

(9) Кэрол вычисляет

$$k = X'^{z} \bmod n$$

Секреты и ключ k равен $g^{xyz} \bmod n$, и никто из подслушивающих каналы связи не сможет вычислить это значение. Протокол можно легко расширить для четверых и более участников, просто добавляются участники и этапы вычислений.

Общий секретный ключ может использоваться абонентами для шифрования сеансовых секретных ключей, а последние – для шифрования сообщений с использованием симметричных методов шифрования. Решение задачи дискретного логарифмирования существует, но оно вычислительно неосуществимо. Таким образом, стойкость метода Диффи – Хеллмана основана на сложности дискретного логарифмирования.

В симметричных криптосистемах существуют две принципиальные проблемы:

распределение секретных ключей по защищенному каналу;

аутентификация секретного ключа.

Под аутентификацией понимается проведение процедуры, которая позволяет удостовериться получателю, что секретный ключ принадлежит законному отправителю (например центру распределения ключей). Система открытого распределения ключей решает первую проблему, т.е. она позволяет обойтись без защищенного канала для распределения секретных ключей. Однако она не устраняет необходимость аутентификации.

3. Порядок выполнения работы

Порядок выполнения практической работы заключается в следующем:

1) ознакомиться с разделами методических указаний к данной лабораторной работе;

2) получить у преподавателя вариант (варианты) заданий на исследование описанных выше шифров;

3) составить контрольный пример;

4) разработать и реализовать заданный(е) алгоритм(ы) шифрования/дешифрования или криптоатаки;

5) на контрольном примере проверить правильность работы алгоритмов шифрования и дешифрования;

6) составить отчет о проделанной работе.

Примечание. Разнообразие вариантов заданий определяется заданным вариантом симметричного криптографического алгоритма, длиной ключа, образцами зашифрованного сообщения для алгоритма криптоатаки и т. д.

4. Экспериментальные исследования

Прежде всего, необходимо создать (сгенерировать) свой открытый ключ, который можно бы было отправлять без всякой опаски своим респондентам. Заметьте, это Боб хочет написать Алисе первое закрытое сообщение, но не может, пока Алиса не пришлет ему свой открытый ключ.

Алиса - ввела свой ключ Medichi фамилия моей матери до 1912 года, сгенерировала открытый ключ, сохранила в файл 123.key и отправила Бобу. Если хотите вставлять ключ в окно формы из буфера обмена, то щелкните по окну правой кнопкой мыши.

Ключ – это фраза, которую Алиса не сможет забыть практически никогда и которая не вызовет особого интереса у шпионов, даже если будет написана в ее ежедневнике.

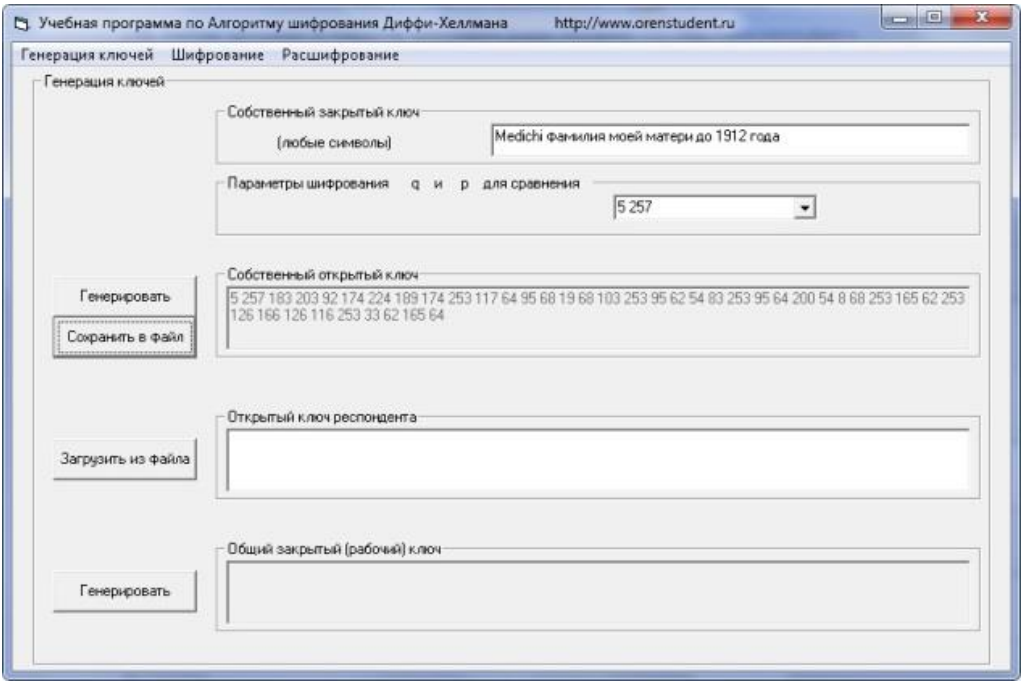


Рис.2.1 Окно программы по изучению алгоритма Diffie-Hellman: генерация ключей Алиса

А открытый ключ – это фраза, практически не поддающаяся запоминанию. И даже переписывать ее без оргтехники очень долго и чревато совершением ошибок.

Теперь Боб – Ввел свой ключ «Люблю свою охотничью собаку *** Jerry ***», сгенерировал открытый, сохранил в файл 124.key, загрузил Алисин открытый ключ из файла 123.key и сгенерировал рабочий ключ. Боб готов к работе.

Открыв вкладку Расшифровка (так удобнее писать и править небольшой черновик) и пишет в нижнем окне текст…. Сохраняет в файле откр_текст_1.txt (или создает сообщение в любом текстовом редакторе).

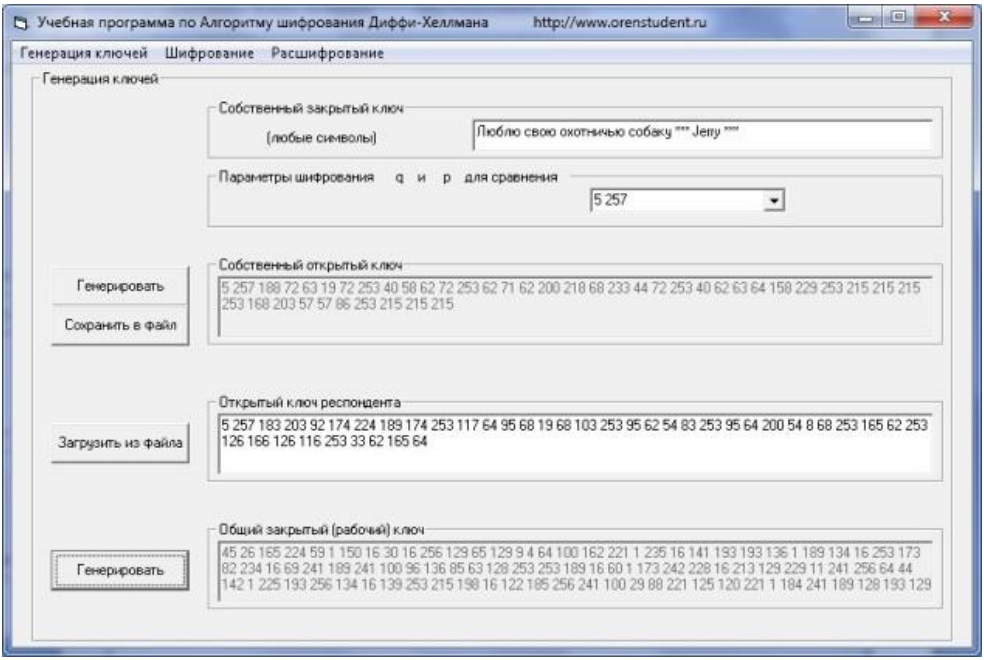


Рис.2.2 Окно программы по изучению алгоритма Diffie-Hellman: генерация ключей Боб

Боб - открывает вкладку Шифрование и загружает свой текст из файла (или набирает с клавиатуры). Шифрует. Сохраняет в файл закр_текст_1.txt.

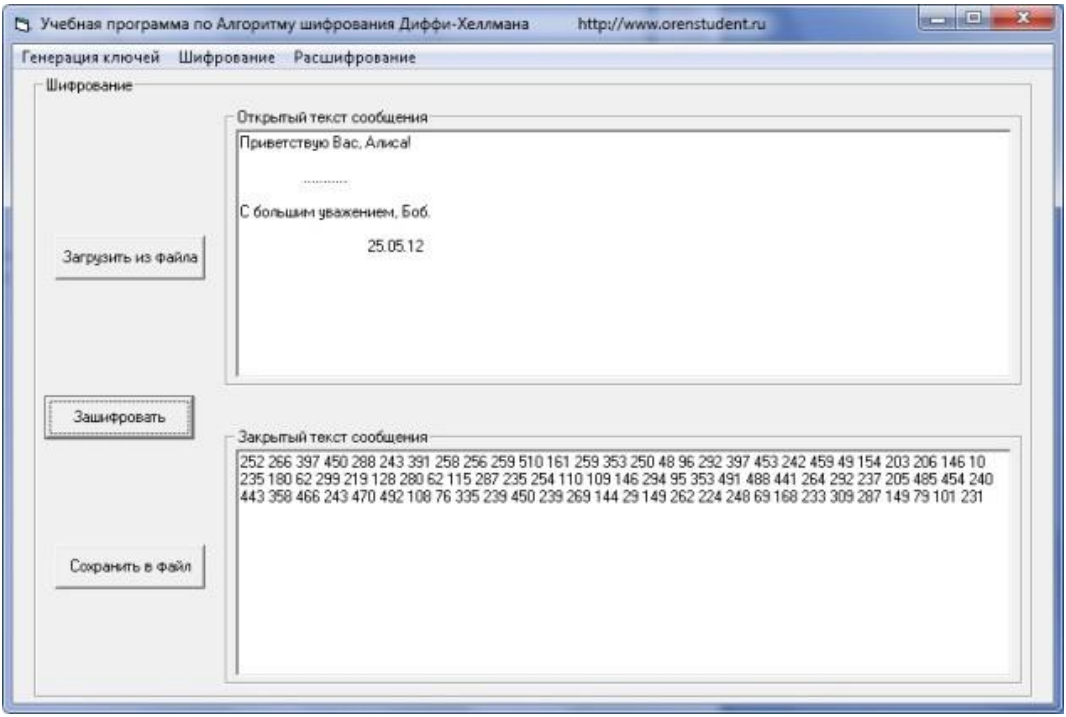


Рис.2.3 Окно программы по изучению алгоритма Diffie-Hellman: шифрование ключей Боб

Далее оба файла закр_текст_1.txt и 124.key отправляем Алисе.

Алиса - загрузила ключ Боба, сгенерировала рабочий ключ и… Может работать с информацией.

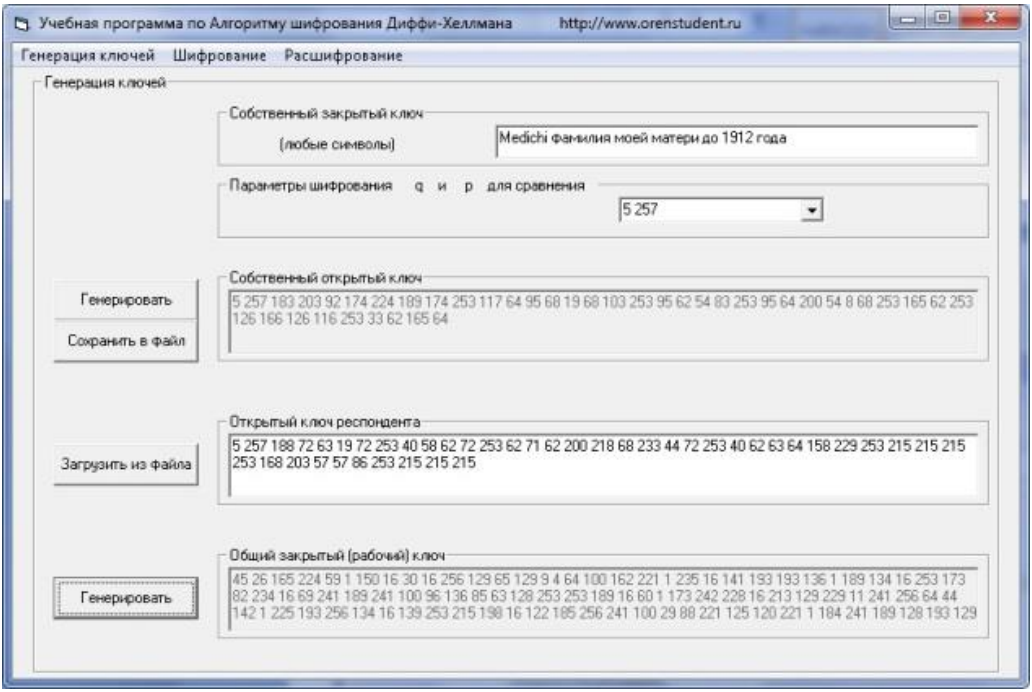


Рис.2.4 Окно программы по изучению алгоритма Diffie-Hellman: генерация рабочего ключа Боба

Теперь выполните расшифровку. Раз ключ общий (рабочий), то они могут расшифровывать и свои сообщения, и респондента. Это симметричное шифрование, в отличии от ассиметричного (когда свое сообщение зашифровать можешь, а расшифровать нет).

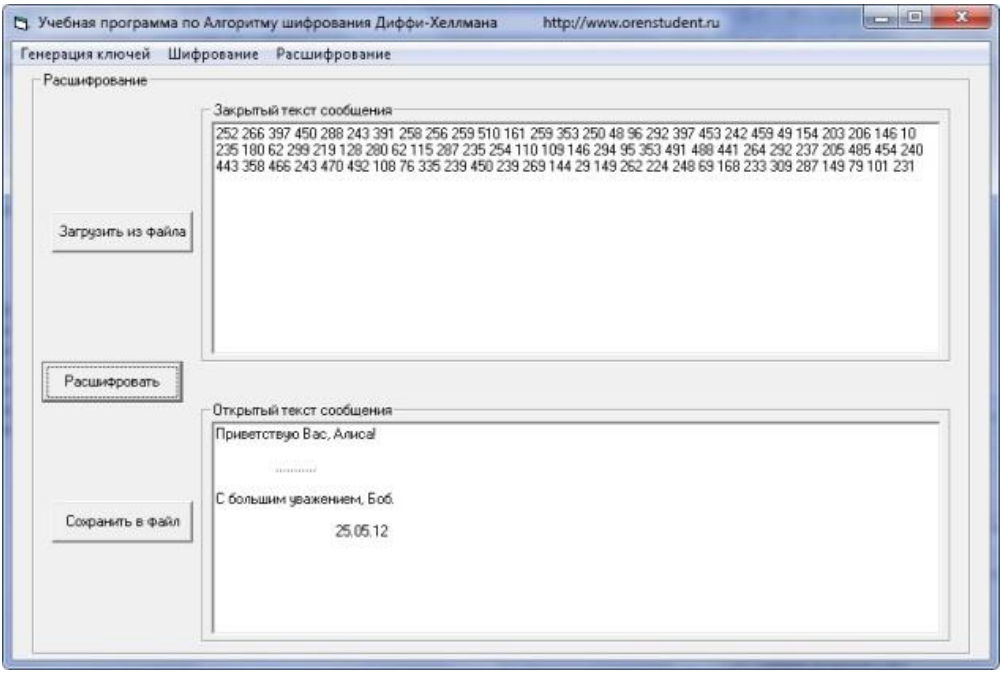


Рис.2.5 Окно программы по изучению алгоритма Diffie-Hellman: расшифрование рабочего ключа Боба

Суть алгоритма и программы - дать возможность обоим респондентам сгенерировать одинаковые (симметричного шифрования) общие рабочие ключи. Для этого необходимо наличие файла открытого ключа другого респондента (где в первых двух группах передаются параметры шифрования) и, конечно, помнить свой секретный ключ, на основе которого сгенерирован и передан респонденту Ваш открытый ключ.

Контрольные вопросы

1.Объясните структуру алгоритма шифрования Диффи–Хеллмана.

2.Назовите основные этапы данного шифрования;

3. Как осуществляется защита от несанкционированного доступа при данном алгоритме шифровании?

ПРАКТИЧЕСКАЯ РАБОТА №3

АЛГОРИТМ АСИММЕТРИЧНОГО ШИФРОВАНИЯ ЭЛЬ ГАМАЛЬ

1. Цель работы

1.1 Изучить назначение, особенности и структуру алгоритма Эль Гамаля, области его применения в информационных процессах предприятий.

1.2 Приобретение практических навыков по разработке, составлению и реализации алгоритмов Эль Гамаля в программной среде C++ Visual Studio.

1.3 Закрепление теоретических знаний по разделу «Шифрование и ЭЦП».

2. Краткие теоретические сведения

Асимметричная схема Эль Гамаль, предложенная автором (El Gamal), использует операцию возведения в степень по модулю простого числа. При этом трудноразрешимой задачей для злоумышленника является отыскание не числа, которое возведено в степень, а то, в какую степень возведено известное число. Эта задача носит название проблемы дискретного логарифма[2].

На этапе выработки ключей должно производиться следующее:

1. Выбирается произвольное (правда достаточно большое) простое число p.

2. Для этого простого числа определяется любой образующий элемент (англ. primitive root) – т. е. такое число a, при многократном возведении которого в степень по модулю p (a1 mod p, a2 mod p, …) , будут перебираться (в произвольном порядке, но обязательно по одному разу) все числа от 1 до (p-1) включительно.

3. Генерируется произвольное случайное число x (0<x<p) – это и есть

закрытый ключ.

4. Вычисляется значение b = ax mod p – комбинация (a, p, b) представляет собой открытый ключ получателя.

На этапе шифрования:

1. Отправитель генерирует произвольное случайное число y (0<y<p).

2. Помещает в начале шифрограммы число (ay mod p) .

3. Вычисляет величину k = (by mod p) = ((ax mod p)y mod p).

4. Используя некоторую, заранее оговоренную в данной реализации, часть k в качестве симметричного ключа для любого блочного шифра шифрует отправляемое сообщение.

5. Надежно стирает числа y и k из оперативной памяти и других мест, куда они могли случайно попасть.

На этапе дешифрования:

1. По приходу зашифрованного сообщения получатель отделяет от пакета величину (ay mod p) и вычисляет на ее основе ((ay mod p)x mod p) – математика доказывает, что полученное число будет равно тому самому k, которое вычислил отправитель, так как в данной формуле операнды x и y можно менять местами.

2. Выделив из k туже самую часть, что и отправитель, получатель дешифрует весь идущий далее пакет симметричным алгоритмом.

Схема алгоритма Эль Гамаль приведена на рис. 1.2. Проблема дискретного логарифма состоит в том, что, зная основание степени и получившийся после возведения

результат по модулю простого числа, невозможно за обозримое время определить, в какую именно степень было возведено основание. В схеме Эль Гамаль потенциальный злоумышленник может получить значения a, p, (ax mod p) и (ay mod p). Однако из-за сложности определения чисел x и y "в чистом виде" у него не оказывается возможности вычислить значение k = (axy mod p), которое так необходимо для прочтения шифровки.

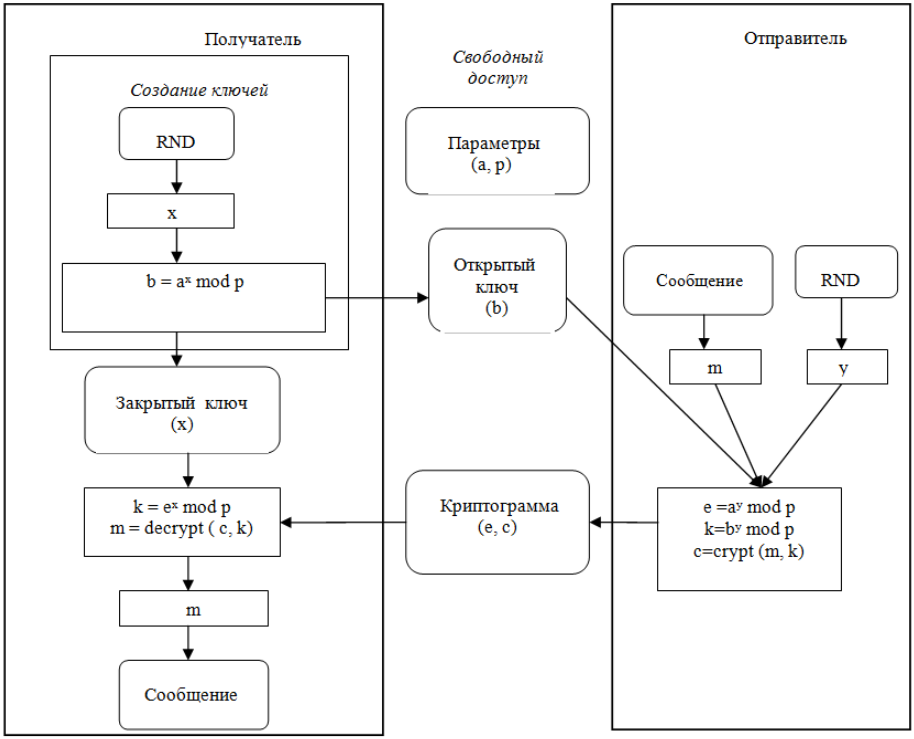


Рис. 3.1.Структура криптосистемы Эль Гамаль

По криптостойкости в схеме Эль Гамаль 512-битное число p приравнивается к 56-битному симметричному ключу, размер которого в настоящее время недостаточен для надежного шифрования. Поэтому на практике применяются p длиной в 768, 1024 и 1536 бит.

Пример 3. В качестве простого числа, порождающего циклическую группу, выберем p = 11, за образующий элемент примем число a = 7 (при возведении 7 в степень 1, 2, 3 и т. д. по модулю 11 последовательно проходят все 10 значений [7, 5, 2, 3, 10, 4, 6, 9, 8, 1]). Секретным ключом x выберем 6, параметр b принимает значение b = (ax mod p) = (76 mod 11) = 4. В целом ключ принимает вид (a = 7, p = 11, b = 4).

Предположим, что некий абонент хочет передать сообщение. Он выбрал случайное число, не превосходящее p, например, y = 9. В начало шифрограммы помещается число (ay mod p) = 79 mod 11) = 8. Кроме того, на основе y и открытого ключа отправитель вычисляет k = by mod p = 49 mod 11 = 3. Выбрав значение 3 или какие-либо его биты в качестве симметричного ключа, отправитель шифрует передаваемые данные и стирает величины 9 и 3 со своих накопителей.

Получатель по приходу пакета для вычисления k = (ay mod p)x mod p возводит число 8 из заголовка шифрограммы в степень секретного ключа и получает k = 86 mod 11 = 3 – то же самое значение, которое использовал отправитель, шифруя собственно данные.

Схемы на основе эллиптических кривых. Асимметричное шифрование, использующее эллиптические кривые, не является отдельной схемой. Она представляет собой модификацию других схем, увеличивающую скорость работы алгоритмов и одновременно уменьшающую размеры ключей. Асимметричная криптография на эллиптических кривых очень похожа на проблему дискретного логарифма. В связи с этим асимметричная криптография на

эллиптических кривых больше всего напоминает криптосистему Эль Гамаль.

Эллиптической кривой, используемой в данной схеме, является выражение вида $y2 = (x2 + a \times x + b) \bmod p$, где p – большое простое число. Обе координаты x и y, а также параметры a и b являются натуральными числами из диапазона [0; p-1], т. е. все вычисления производятся по модулю p. Пары чисел (x, y) удовлетворяющие приведенному равенству называются точками эллиптической кривой.

Над точками определена операция сложения следующим образом. Если абсциссы точек Q1 (x1, y1) и Q2 (x2,y2) различимы, то точка S = Q1 + Q2 имеет координаты (xs, ys), определяемые формулами:

$$k = ((y2 - y1) / (x2 - x1)) \bmod p$$
$$xs = (k2 - x1 - x2) \bmod p$$
$$ys = (k \times (x1-x2) - e1) \bmod p .$$

Если же точки Q1 и Q2 совпадают, т. е. речь идет о об "удвоении точки" то применяются следующие формулы:

$$k = ((3 \times x\ 12 + a) / (2 \times y1)) \bmod p$$
$$xs = k2 - 2\times x1\ ) \bmod p$$
$$ys = (k \times (x1 - xs) - y1) \bmod p.$$

Подобные формулы позволяют ввести над точками эллиптической кривой операцию умножения на число: $R = n \times P$ – n кратное сложение точки P с самой собой. Данная операция по свойствам тождественна операции возведения в степень в конечном поле простого числа. Само умножение (шифрование) характеризуется полиномиальной скоростью вычислений, а вот попытка по известным P и R определить число n уже не укладывается в полиномиальные рамки.

3. Порядок выполнения работы

Порядок выполнения лабораторной работы

заключается в следующем:

1) ознакомиться с разделами методических указаний к данной лабораторной работе;

2) получить у преподавателя вариант (варианты) заданий на исследование описанных выше шифров;

3) составить контрольный пример;

4) разработать и реализовать заданный(е) алгоритм(ы) шифрования/дешифрования или криптоатаки;

5) на контрольном примере проверить правильность работы алгоритмов шифрования и дешифрования;

6) составить отчет о проделанной работе.

Примечание. Разнообразие вариантов заданий определяется заданным вариантом симметричного криптографического алгоритма, длиной ключа, образцами зашифрованного сообщения для алгоритма криптоатаки и т. д.

4. Экспериментальные исследования

Экспериментальные исследования модели заключаются в проверке каждого этапа ее работы и результатов деятельности каждого ее механизма. Необходимо зашифровать следующий текст «Я буду в семь» на ключе 6901, получить зашифрованный текст: 5451900 3891160 4810500 5195340 4874640 5195340 3891160 4831880 3891160 5152580 4896020 5045680 5387760, и расшифровать его в исходное сообщение.

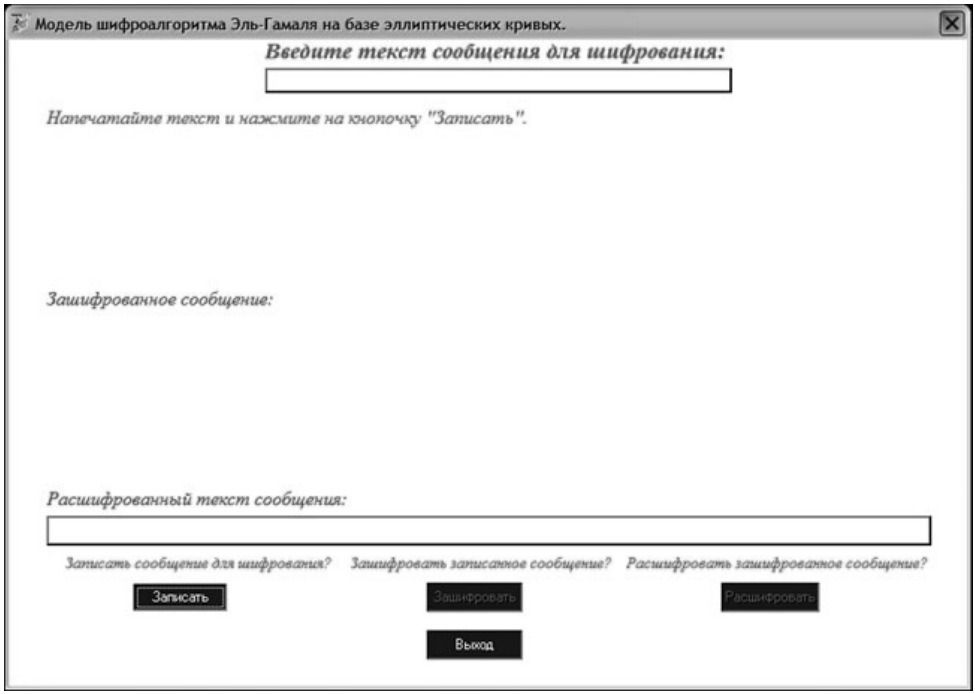


Рис.3.2. Окно программы по изучению алгоритма Эль Гамаля: основной интерфейс

Для этого, следует ввести текст в верхнее окошко, чуть ниже его будут появляться подсказки по мере выполнения программы. Текст можно ввести любой, по заданию преподавателя, но нужно помнить, что программа разработана только для букв русского алфавита и пробела.

Замечание: поскольку ASCII кодировка заглавных и прописных букв различна, во избежание проблем, невозможно ввести заглавные буквы.

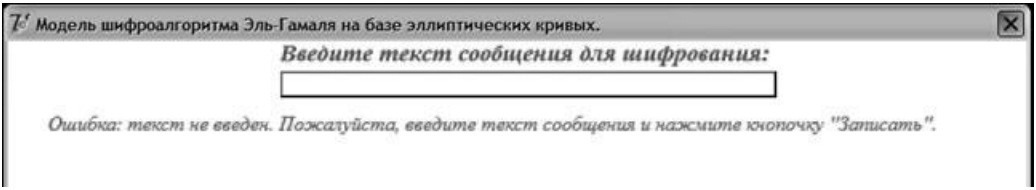


Рис.3.3. Окно программы по изучению алгоритма Эль Гамаля: проверка на наличие текста

После ввода текста и нажатия на кнопку «Записать» пользователю выдается сообщение, о том что текст записан, и становится доступна кнопочка «Зашифровать» (рис. 3).

Рис.3.4. Окно программы по изучению алгоритма Эль Гамаля: запись текста сообщения для шифрования

После появления надписи «Сообщение успешно записано», это сообщение можно зашифровать, о чем подсказывает вторая надпись. При нажатии на кнопку «Зашифровать», пользователю будет предложено ввести целое число в заданном диапазоне. Если число в данный диапазон не попадает, то появляется соответствующее

сообщение.

После чего пользователь может повторно нажать на кнопку «Зашифровать» и ввести нужное число. Если число было введено правильно, то появиться сообщение, о том, что записанный текст успешно зашифрован и после надписи «Зашифрованное сообщение» пользователь увидит некоторое количество чисел.

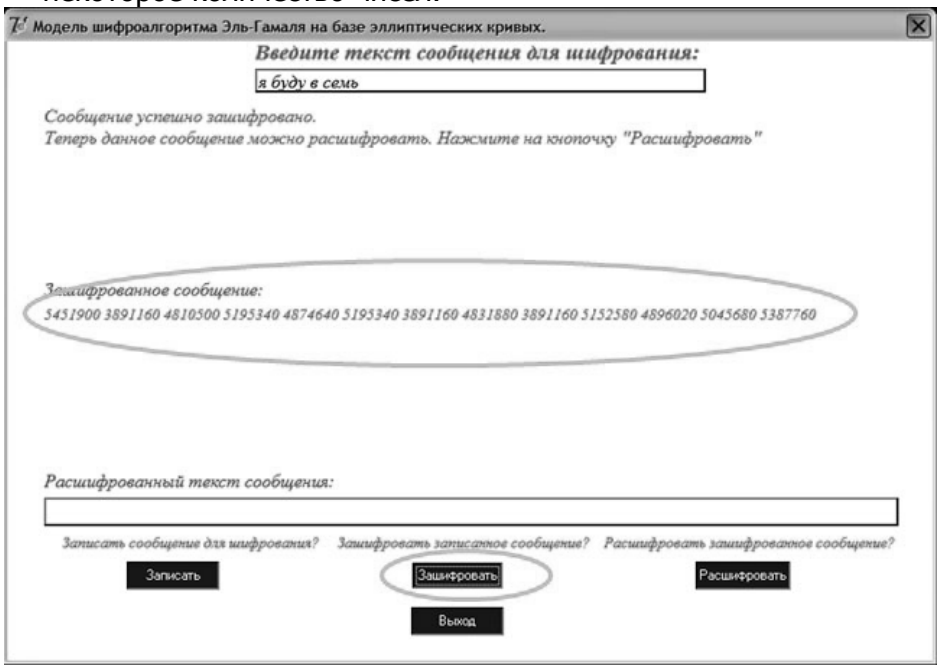


Рис.3.5.Окно программы по изучению алгоритма Эль Гамаля : зашифрованное сообщение.

Для того чтобы наглядно показать, что шифруется каждый символ, числа разделены пробелом, следовательно количество чисел должно совпадать с количеством символов (включая пробел) исходного текста сообщения (рис. 4). На самом деле числа никакими знаками не отделяются, в связи с чем зашифрованное сообщение представляет собой просто набор цифр.

После завершения шифрования, появится

соответствующее сообщение, подсказка о дальнейших действиях и станет доступна последняя кнопочка «Расшифровать».

Нажатие кнопки «Расшифровать» сопровождается появлением соответствующего сообщения и в нижнем окошечке будет отображаться расшифрованный текст (рис. 5). Исходный текст в верхнем окошке и расшифрованный в нижнем должны совпадать.

Замечание: в идеале зашифрованный текст представляет собой сплошной набор чисел. Но определение числа, которое соответствует тому или иному символу не является сложной задачей, поскольку все числа имеют одинаковый порядок, то есть одинаковое количество цифр.

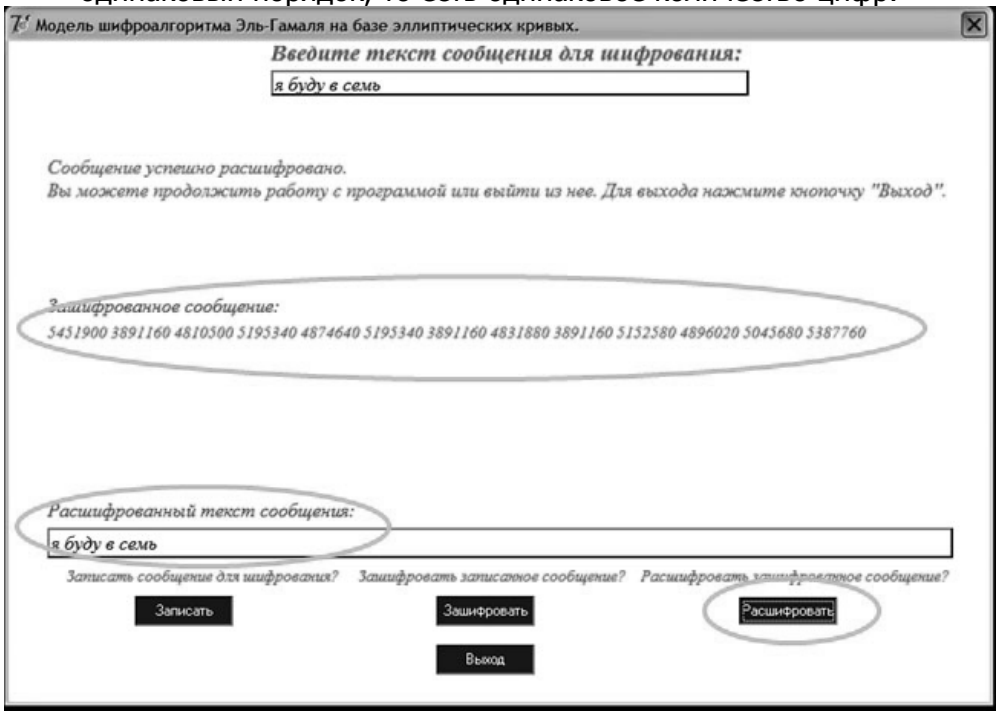


Рис.3.6 Окно программы по изучению алгоритма Эль Гамаля : расшифрованное сообщение.

Пограмма позволяет завершить работу приложения на

любом этапе работы, либо зашифровать уже записанный текст сообщения повторно, например, с другим ключом, либо он может ввести новый текст и пройти все этапы заново. Если зашифровать одно и тоже сообщение разными ключами, все равно должен получится исходный текст.

Выполните выполните проверку вашего шифрования. Если все выполненно верно, то при вводе недопустимых значений, шифрование и расшифрование текста разными ключами, прviедет к недопустимым действиям.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какая процедура является более производительной – асимметричное шифрование (дешифрование) или симметричное шифрование (дешифрование)?

4. Какая трудноразрешимая математическая задача лежит в основе стойкости алгоритма Эль Гамаль?

5. В чем заключается проблема дискретного логарифма?

6. В чем заключаются проблемы разложения больших чисел на простые множители и вычисления корней алгебраических уравнений?

ПРАКТИЧЕСКАЯ РАБОТА №4

АЛГОРИТМЫ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ

1. Цель работы

1.1 Изучить назначение, особенности и структуру алгоритма электронной цифровой подписи, области его применения в информационных процессах предприятий.

1.2 Приобретение практических навыков по разработке, составлению и исследованию алгоритмов электронной цифровой подписи в программной среде C++ Visual Studio 2008.

1.3 Закрепление теоретических знаний по разделу «Шифрование и ЭЦП».

1. Краткие теоретические сведения

2.1 Электронная цифровая подпись

Электронная цифровая подпись – некоторая дополнительная информация, соответствующая данному электронному документу (сообщению), которая могла быть сформирована только владельцем некоторого секрета – закрытого ключа и которая позволяет с использованием специального алгоритма установить факт соответствия подписи закрытому ключу подписывающего. Под электронной цифровой подписью (ЭЦП) понимается также криптографическая система (совокупность алгоритмов и правил), позволяющая подписывать цифровые сообщения и проверять правильность формируемых цифровых подписей [1].

Для формирования цифровой подписи документа обычно создается так называемый дайджест сообщения

(message digest), который представляет собой свертку исходного сообщения с помощью специальной хэш-функции (англ. hash – мелко измельчать и перемешивать). Длина дайджеста с одной стороны намного меньше, чем возможные исходные сообщения, а с другой стороны такова, что полный перебор возможных значений является практически невыполнимым. Например, длина дайджеста, порождаемого алгоритмами Ривеста – MD2, MD4,MD5, равняется 128 битам, а алгоритмом SHA – 160 битам [3].

Хэш-функция должна удовлетворять следующим условиям:

а) на вход алгоритма преобразования может поступать двоичный блок данных произвольной длины;

б) на выходе алгоритма получается двоичный блок данных фиксированной длины;

в) значения на выходе алгоритма распределяются по равномерному закону по всему диапазону возможных результатов;

г) восстановить аргумент по значению с вычислительной точки зрения практически невозможно;

д) при изменении хотя бы одного бита на входе алгоритма его выход значительно меняется: в идеальном случае инвертируется половина бит.

Хэш-функция называется криптографически стойкой, если в дополнение к перечисленным свойствам она удовлетворяет еще двум требованиям:

1) зная результат хэш-функции, невозможно подобрать, кроме как полным перебором, какой-либо входной блок данных, дающий такое же значение на выходе;

2) невозможно подобрать, кроме как полным перебором, пару различных входных блоков, дающих на выходе произвольный, но одинаковый результат.

Разрядность в 128 или 160 бит гарантирует, что на сегодняшнее время в мире не существует двух разных документов, имеющих одинаковую хэш-сумму.

Получающееся таким образом "практически"

однозначное соответствие между документом и хэш-суммой позволяет защищать целостность не самого документа, а только лишь 16-байтового блока данных, т. е. с помощью хэш-функций проблема защиты большого блока данных сводится к проблеме защиты маленького блока данных заранее известной длины. В системах электронной цифровой подписи (подписи) подписывается не сам документ, а только его хэш-сумма. Если на приемной стороне с помощью ЭЦП проверена целостность хэш-суммы, а вычисленное получателем самостоятельно хэш-сумма документа совпадает с присланным, то и весь документ признается аутентичным – внести в него изменения, не изменив значение хэш-суммы, было невозможно.

Общепринятым принципом построения хэш-функций является итеративная последовательная схема. По этой методике ядром алгоритма является преобразование k бит в n бит. Величина n – разрядность результата хэш-функции, а k – произвольное число, больше n. Базовое преобразование должно обладать всеми свойствами криптостойкой хэш-функции, т. е. необратимостью и невозможностью инвариантного изменения входных данных.

Хэширование производится с помощью промежуточной вспомогательной переменной разрядностью в n бит. В качестве ее начального значения выбирается произвольное известное всем сторонам значение, например, 0. Входные данные разбиваются на блоки по (k-n) бит. На каждой итерации хэширования со значением промежуточной величины, полученной на предыдущей итерации, объединяется очередная (k-n) – битная порция входных данных, и над получившимся k-битным блоком производится базовое преобразование. В результате весь входной текст оказывается "перемешанным" с начальным значением вспомогательной функции. Из-за характера преобразования базовую функцию еще часто называют сжимающей. Значение вспомогательной величины после финальной итерации поступает на выход хэш-функции (рис.

3.1.).

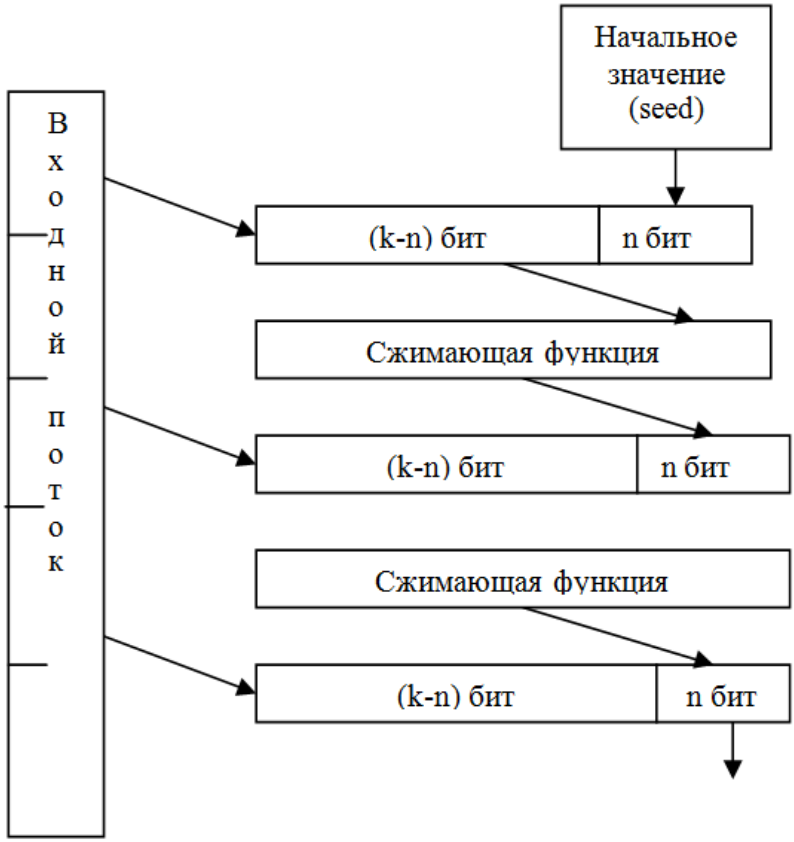


Рис. 4.1 . Итеративные хэш-функции (сжимающая функция)

На сегодняшний день наибольшее распространение получили два алгоритма криптостойкого хэширования: алгоритм MD5 (Message digest № 5), разработанный

Рональдом Ривестом (Ronald Rivest), и алгоритм SHA-1 (Sequre Hash Algorithm), предложенный Институтом Стандартизации США NIST как стандарт хэширования в гражданской криптографии.

Размер значения хэш-функции, вычисленной по алгоритма MD5 равен 128 битам, а значение хэш-функции, вычисленной по алгоритму SHA-1 равен 160 битам, что дает дополнительный запас стойкости.

2.2 Алгоритм цифровой подписи RSA

В основе алгоритма цифровой подписи RSA лежит инверсия ассиметричного алгоритма шифрования RSA. Для формирования электронной подписи отправитель выполняет над контрольной суммой документа h те же самые действия, что и при шифровании, но использует не открытый ключ получателя, а свой собственный закрытый ключ, т. е. signi = (hid mod n). Открытый и закрытый ключи просто меняются сместами. На приемной стороне получатель возводит подпись в степень открытого ключа e отправителя и получает (signie mod n) = (hide mod n) = hi (согласно тем же формулам, что и в асимметричном шифровании RSA).

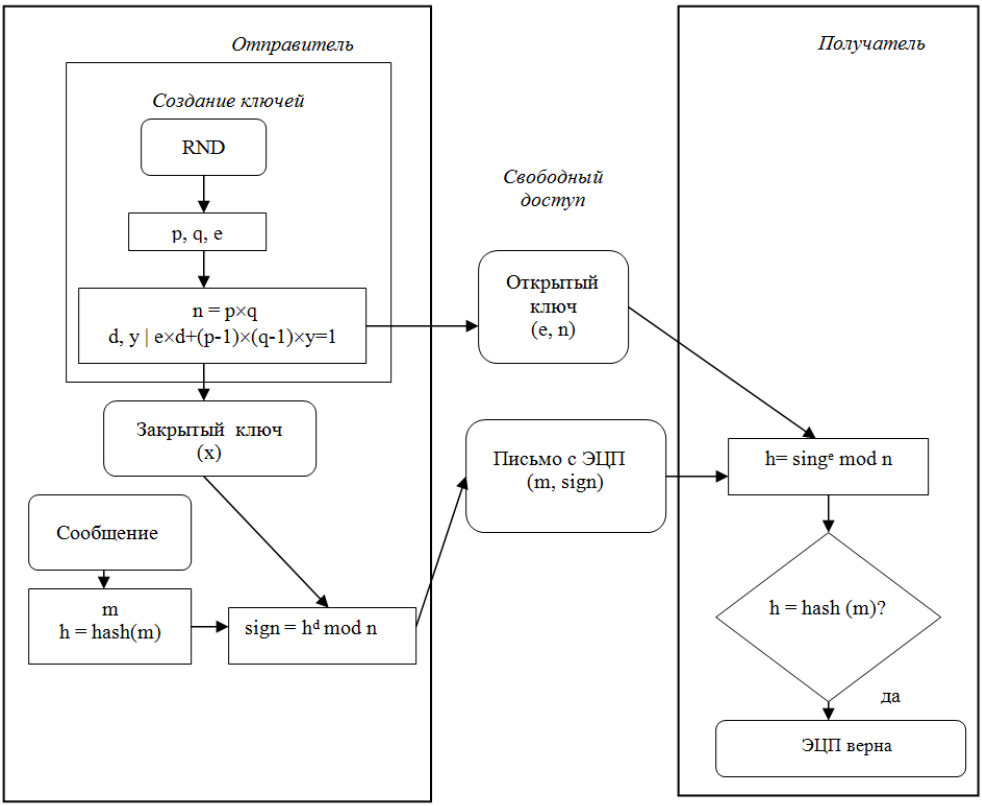


Рис. 4.2. Алгоритм электронной цифровой подписи RSA

Если получившееся после возведения в степень значение совпадает с вычисленной независимо на приемной стороне контрольной суммой документа, то проверка считается выполненной, а документ – подлинным. Никто, кроме отправителя, не зная d, не сможет вычислить такую подпись signi , чтобы возведение ее в степень открытого ключа e дало требуемую контрольную сумму – это та же самая трудноразрешимая задача, что и в асимметричном шифровании RSA. Следовательно снабдить документ такой подписью signi мог только истинный

владелец закрытого ключа. Схема ЭЦП приведена на рис. 3.2.

2.3 Алгоритм цифровой подписи Эль Гамаля

Схема Эль-Гамаль является одной из самых распространенных схем ЭЦП. Этому послужило, во-первых то, что при надлежащей и достаточно хорошо проверенной стойкости система имеет хорошую скорость вычисления. Во-вторых, схема имеет достаточно много модификаций, что в принципе мало свойственно асимметричным шифрам. В схеме Эль Гамаль абонент, подписывающий документ, доказывает все желающим проверить подпись, что знает секретный ключ x – степень, в которую был возведен образующий элемент a, чтобы получить открытый ключ b. Явным образом продемонстрировать x, естественно нельзя, так как любой участник информационного обмена начнет подписывать им документы. Поэтому приходится демонстрировать не само письмо, а результат некоторой математической формулы с участием x. При проверке само число x ни на каком этапе не раскрывается, но проверяющий на основе этой формулы удостоверяется, что отправитель сообщения действительно знает x.

Базовый элемент ЭЦП по схеме Эль Гамаля выглядит следующим образом. На этапе подписания отправитель:

1. Генерирует случайное число k, уникальное для каждого подписываемого документа, взаимно простое с числом (p-1).
2. Вычисляет r = (ak mod p).
3. Вычисляет обратный элемент поля к числу k – его обозначают (k-1mod (p-1)) или 1/k , в дальнейшем операция "деление на k" равносильно умножению на 1/k.
4. Вычисляет s=((h-x×r)/k) mod (p-1), где h – контрольная сумма (значение хэш-функции) подписываемого документа.

Пара чисел (r,s) является цифровой подписью для документа, имеющего контрольную сумму h. На приемной

стороне получатель:

1. Вычисляет u=((br) × (rs) mod p).
2. Вычисляет v=(ah mod p).
3. Проверяет равенство значений u=v (если равенство выполняется, то подпись верна, в противном случае документ сфальсифицирован).

В случае корректности ЭЦП данное равенство является тождеством. Произведем несложные преобразования над u (подразумеваем, что все действия выполняются по модулю числа p):

u = (br) × (rs) =(((ax)r) × ((ak)s) = a(xr + h-xr) = ah = v.

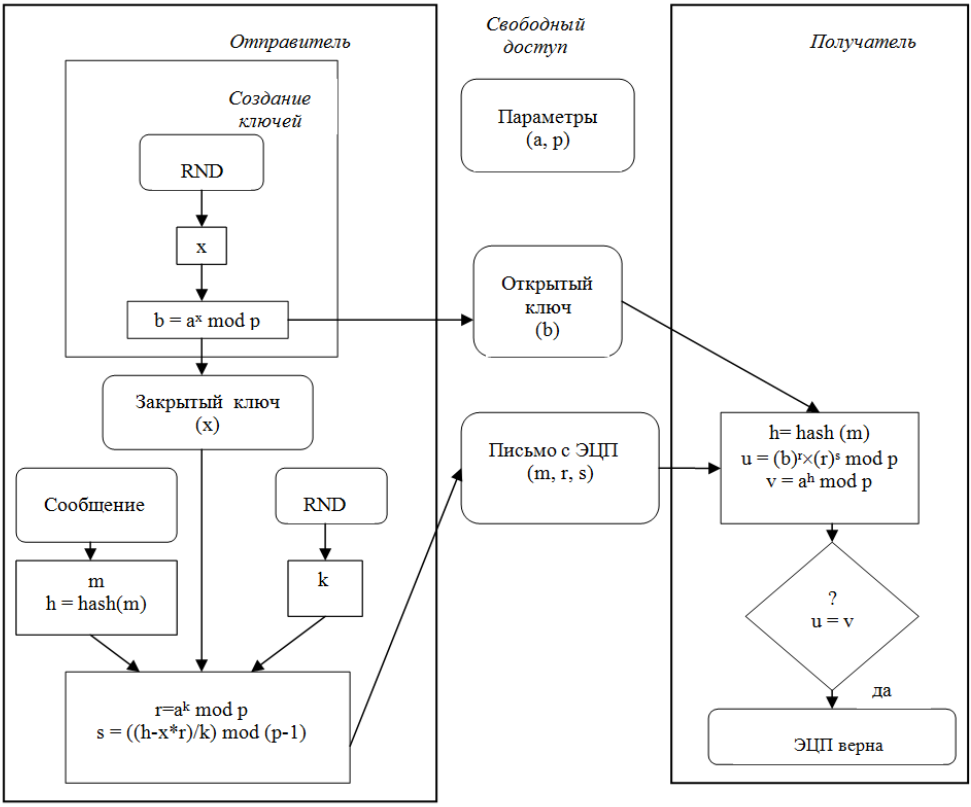


Рис. 4.3. Алгоритм электронной цифровой подписи Эль Гамаля

Как видим, во-первых, получилось тождество, а, во-вторых, ни на одном этапе преобразований (а все они доступны проверяющему) число x не фигурировало в открытом виде – его как бы ”экранируют” две величины r и k, причем вторая неизвестна получателю. Схема ЭЦП Эль Гамаля приведена на рис. 3.3.

В качестве особенностей ЭЦП схемы Эль Гамаля необходимо упомянуть требование проверять на приемной стороне неравенство (r<p). Если такой проверки не делать, то злоумышленник сможет подделать подпись, имея перехваченным хотя бы один документ, подписанный легальным пользователем. Правда по этому алгоритму значение r получается достаточно большим, что и позволяет защищаться от него сравнением r с простым числом p. Вторым требованием является наличие у числа (p-1) большого простого делителя. Обычно первоначально выбирается большое простое число q на один бит меньше требуемой, а затем p вычисляется как $p = 2 \times q + 1$ и проверяется на простоту.

2.4 Стандарт цифровой подписи DSS

Стандарт электронной цифровой подписи США DSS (Digital Signature Standart), принятый в 1992 году, является одной из модификаций схемы Эль Гамаля и приведен на рис. 3.4.

Стандарт DSS определяет Digital Signature Algorithm (DSA), согласно которому цифровая подпись представляет собой пару больших чисел. Цифровая подпись вычисляется на основе правил (т.е. алгоритма DSA) и набора параметров, которые могут быть использованы для проверки идентичности подлинника и целостности данных. DSA включает генерацию подписи и ее проверку. Генерация

использует секретный (private) ключ для получения цифровой подписи. Проверка подписи использует открытый (public) ключ, который соответствует секретному ключу, использованному при генерации подписи, но не равный ему.

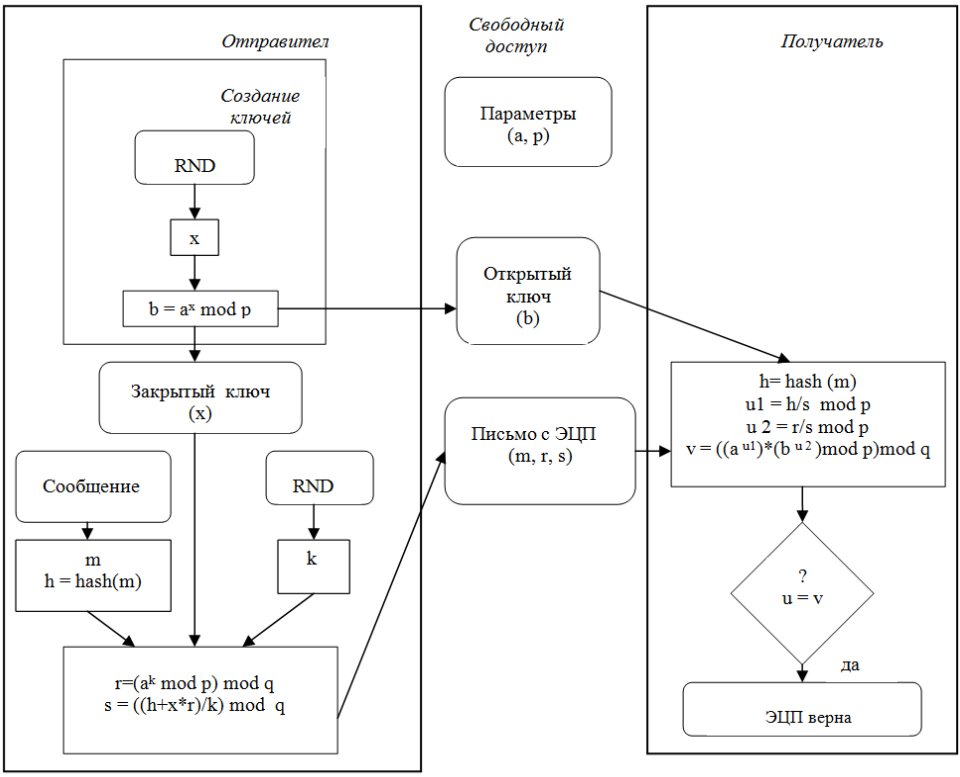


Рис. 4.4. Алгоритм электронной цифровой подписи DSS

Каждый пользователь обладает парой ключей: секретным и открытым. Предполагается, что открытые ключи известны всем членам группы пользователей, либо вообще доступны всем. Секретные ключи должны знать

только их создатели (владельцы). Любой может проверить подпись пользователя с помощью применения его открытого ключа. Генерация подписи может быть осуществлена только владельцем секретного ключа.

При генерации подписи для получения сжатой версии данных, называемой сверткой сообщения (message digest) используется хэш-функция. Свертка сообщения подписывается. Цифровая подпись отсылается получателю вместе с подписанными данными (часто называемыми сообщением). Получатель сообщения и подписи проверяет подпись, используя открытый ключ отправителя. В процессе проверки подписи должна использоваться аналогичная хэш-функция, что и при подписании.

DSA может иметь программное, микропрограммное и аппаратное обеспечение. Стойкость системы цифровой подписи в значительной степени зависит от защищенности пользовательских секретных ключей. Поэтому пользователи должны хорошо защищать свои секретные ключи от неавторизованного доступа к ним.

В стандарте цифровой подписи DSS определено, что длина числа p должна составлять от 512 бит до 1024 бит, длина числа q – 160 бит. Эти требования связаны с тем, что при таких длинах достигается достаточная вычислительная стойкость. Но алгоритм цифровой подписи будет работать, даже если используются числа p и q другой длины.

Стандарт DSS определяет, что в качестве хэш-функции следует использовать SHA (Secure Hash Algorithm).

2.5 Стандарт цифровой подписи ГОСТ Р 34.10 – 2001

Алгоритм Государственного стандарта РФ по ЭЦП (полное название «Процессы формирования и проверки электронной цифровой подписи») является переложением схемы Эль Гамаль в область эллиптических кривых. Общеизвестными параметрами являются сама эллиптическая кривая и точка P на ней, превращающаяся в после q-кратного сложения с самой собой в «нулевую»

точку кривой. Секретным ключом отправителя d является случайным образом сгенерированное число, а открытым – точка Q. определяемая по формуле (Q = d x P).

На этапе выработки подписи отправитель:

1. Генерирует случайное число k (0<k<q).

2. Вычисляет x-координату точки (k x P), назовем это число r.

3. Вычисляет значение s = (rd + kh) mod q, где h – контрольная сумма подписываемого сообщения.

Подписью является пара чисел (r, s).

На этапе проверки выполняются следующие действия:

1. Вычисляются значения (z1 = (s x (h-1)) mod q) и (z2 = ((-r) x (h-1) mod q).

2. Определяется x- координата точки (z1 x P = z2 x Q)), назовем это числоR.

3. Проверяется равенство (r = R) (если оно верно, подпись корректна).

Разрядность числа q фиксируется Стандартом в 255 бит. Это означает, что при заданной стойкости открытый и закрытый ключи имеют гораздо меньшую длину и требуют при вычислениях меньших затрат ресурсов – все это достоинства эллиптических кривых. Схема ЭЦП по ГОСТ Р 34.10 – 2001 приведена на рис. 3.6.

3. Порядок выполнения работы

Порядок выполнения практической работы заключается в следующем:

1) ознакомиться с разделами методических указаний к данной лабораторной работе;

2) получить у преподавателя вариант (варианты) заданий на исследование описанных выше шифров;

3) составить контрольный пример;

4) разработать и реализовать заданный(е) алгоритм(ы) цифровой подписи;

5) на контрольном примере проверить правильность работы алгоритмов верификации цифровой подписи;

6) составить отчет о проделанной работе.

4. Экспериментальные исследования

4.1 Создание цифровой подписи файла, используя алгоритм RSA

Суть цифровой подписи – это гарантия того, что подписанный файл не был изменен с момента создания цифровой подписи автором файла. Используя ранее полученные данные из практической работы №1 «Алгоритм ассиметричного шифрования RSA» (в программе «Кодирование и декодирование RCA»), по заданию преподавателя на файл с информацией (Notebook.txt), устанавливается цифровая подпись.

В реальности, данный процесс выглядит следующим образом:

автор файла с очень важными данными перед его отправкой щелкает кнопку «Получить цифровую подпись» (предварительно выбрав ее, т.е. указать сам файл в верхнем окне).

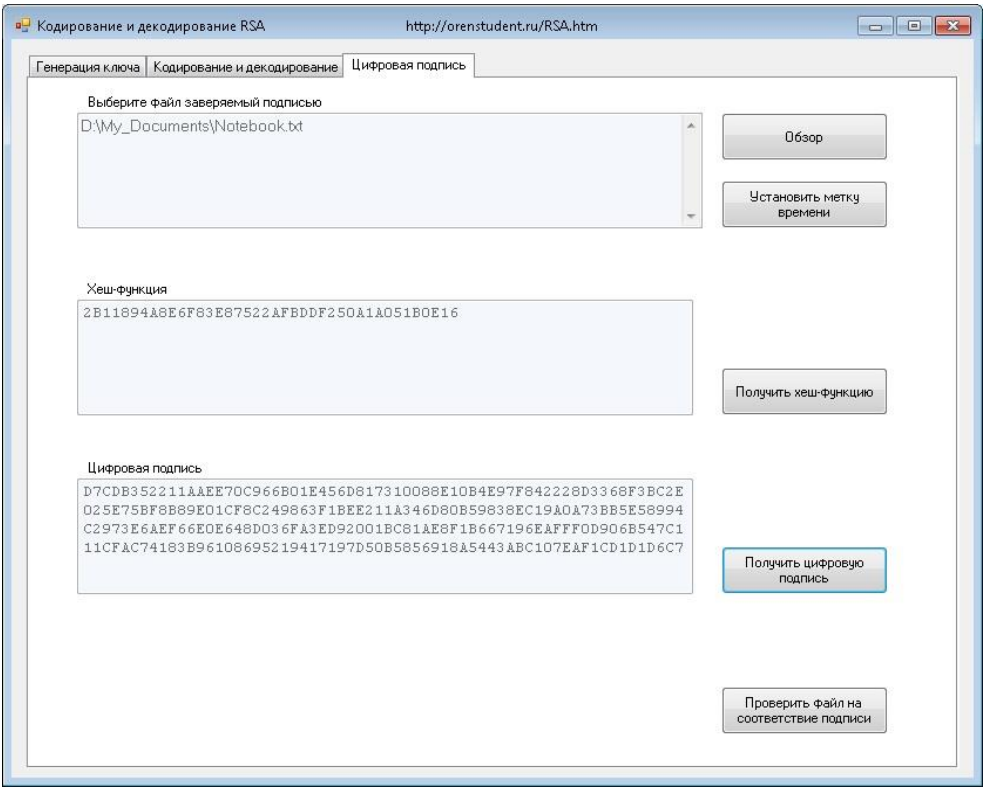


Рис.4.5. Окно программы по изучению криптосистемы RSA: цифровая подпись

В результате в той же папке создается небольшой файл цифровой подписи с расширением *.dsf (по размеру равен одному пакету). В процессе генерации цифровой подписи используется закрытый ключ автора (d и n). Вот сейчас можно отправлять оба файла респонденту, который имеет открытый ключ автора и может всегда проверить (обнаружить искажения) соответствующего файла.

Подписать документ может только автор (обладатель закрытого ключа), а убедиться, что файл не был случайно или злонамеренно искажен, может любой заинтересованный респондент, который имеет открытый (публичный) ключ автор. Повреждение или утеря файла цифровой подписи (*.dsf) ведет к утрате гарантии подлинности данных в документе.

4.2 Генерации цифровой подписи

Для генерации цифровой подписи, прежде всего, файл хешируется или, другими словами, вычисляется его хеш-функция. В данной работе требуется использовать SHA-1 (Secure Hash Algorithm).

Secure Hash Algorithm 1 — алгоритм криптографического хеширования. Для входного сообщения произвольной длины (максимум бит, что примерно равно 2 эксабайта) алгоритм генерирует 160-битное хеш-значение, называемое также дайджестом сообщения. Используется во многих криптографических приложениях и протоколах. Также рекомендован в качестве основного для государственных учреждений в США. Принципы, положенные в основу SHA-1, аналогичны тем, которые использовались Рональдом Ривестом при проектировании MD4.

Поэтому, какого бы большого размера не был исходный файл, результатом работы SHA-1 будет хеш-функция размером 32 байта (160 бит).

Даже если Вы незначительно измените содержимое файла (добавите один пробел в конце или замените одну маленькую букву на заглавную), новая хеш-функция будет очень сильно отличаться от первоначальной. Подобрать специально два разных текста, которые бы имели одинаковые хеш-функции – трудновыполнимая задача.

Если полученную хеш-функцию закрыть секретным ключом автора, то и получится цифровая подпись данного файла. Размер ее всегда будет равен размеру пакета, т.к. берется остаток по модулю n.

Обратим внимание, что никто кроме автора не может создать цифровую подпись, но все обладатели открытого ключа автора, могут из цифровой подписи получить хеш-функцию подписанного (начального, не искаженного) файла.

4.3 Проверка файла на соответствие подписи

Задача – выполнить проверку файла ранее подписанного файла с информацией, при помощи программы «Кодирование и декадирование RCA» на соответствие электронной цифровой подписи.

Поскольку SHA-1 является общедоступной, то респондент, желающий убедиться в подлинности файла, хеширует

его и сравнивает с хеш-функцией полученной путем декодирования из цифровой подписи. Если совпадают, то файл не изменялся и его данным можно доверять. Для этого следует выполнить действия:

1. Протестировать алгоритм - создав цифровую подпись, для этого нажмите «Проверить файл на соответствие подписи», получите ответ о соответствии.

2. Измените файл (сохранив изменения, пусть самые незначительные) и еще раз нажмите «Проверить файл на соответствие подписи».

3. Убедиться, что полученный ответ при проверке будет отрицательным, что подтверждает надежность такого метода шифрования при защите данных.

Контрольные вопросы

1. Что называется электронной цифровой подписью?
2. Какие варианты цифровой подписи Вы знаете, в чем заключаются их особенности?
3. Для чего используется электронная цифровая подпись?
4. Что такое хэш-функция?
5. Что такое дайджест сообщения?

ПРАКТИЧЕСКАЯ РАБОТА №5

ПРИМЕНЕНИЕ ПРОЦЕССНОГО ПОДХОДА К АНАЛИЗУ БИЗНЕС-ПРОЦЕССОВ НА ПРЕДПРИЯТИИ

1. Цель работы

1.1 Изучение алгоритма составления вербального описания бизнес-процесса «Анализа рынка и потребности потребителей».

1.2 Изучение алгоритма составления вербального описания бизнес-процесса «Подготовка и оформление заявки на товар».

1.3 Изучение алгоритма составления вербального описания вербального описания бизнес-процесса «Приемка товара».

2. Краткие теоретические сведения

Современные предприятия и организации для эффективной своей деятельности, должны уметь управлять бизнес-процессами в соответствии с требованиями международного стандарта ИСО 9000:2000.

Организация должна:

выявлять необходимые для системы менеджмента качества процессы и области их применения по всей организации;

определять последовательность и взаимодействие этих процессов;

определять требуемые критерии и методы, позволяющие гарантировать, что функ-ционирование и контроль этих процессов эффективны;

обеспечивать наличие ресурсов и информации, необходимых, чтобы поддерживать функционирование и мониторинг этих процессов;

вести мониторинг, измерять и анализировать эти процессы, а также предпринимать необходимые действия с целью достичь запланированных результатов и непрерывного совершенствования этих процессов.

Рассмотрим основные термины, применяемые при реализации процессного подхода.

Процессный подход основывается на концепции, согласно которой управление — это непрерывная взаимосвязь действий и функций. Таким образом, процессный подход в управлении — это процесс формирования целей и способов их достижений, деятельность, ограниченная в пространстве и во времени, требующаяся для реализации комплекса управленческих ресурсов. К ресурсам, необходимым для осуществления процессов управления социально-экономическими системами, относятся информация, человеческие ресурсы, финансы, время, организационно-административные и материально-технические ресурсы.

Бизнес-процесс — это совокупность взаимосвязанных мероприятий или задач, направленных на создание определенного продукта или услуги для потребителей. Для наглядности бизнес-процессы визуализируют при помощи блок-схемы бизнес-процессов [4].

Понятие и классификация бизнес-процессов. Существуют три вида бизнес-процессов:

Управляющие — бизнес-процессы, которые управляют функционированием системы. Примером управляющего процесса может служить Корпоративное управление и Стратегический менеджмент.

Операционные — бизнес-процессы, которые составляют основной бизнес компании и создают основной поток доходов. Примерами операционных бизнес-процессов являются Снабжение, Производство, Маркетинг и Продажи.

Поддерживающие — бизнес-процессы, которые обслуживают основной бизнес. Например, Бухгалтерский учет, Подбор персонала, Техническая поддержка, АХО.

Для изучения и закрепления теоретического материала, по применению процессного подхода к анализу информационных бизнес-процессов на предприятии, рассмотрим основные задачи их вербального описания.

Задание 1. Разработать вербальное описание бизнес-процесса процесса «Анализировать рынок и потребности потребителей».

Бизнес-процесс «Анализ рынка и потребностей потребителей» включает себя след функции: определение потребностей потребителей, осуществление мониторинга внешней среды, определение концепции бизнеса и стратегию

организации, разработка продукта/услуги, продажа продукта/услуги.

На шаге 01 *«Определение потребности и пожелания потребителей»* проводим интервьюирование потребителей, анализ фокус-групп, подготовим и проведем инспекции, что позволит нам выполнить количественную и качественную оценку, а также спрогнозировать покупательский спрос потребителей. Для измерения удовлетворения потребителей мы должны провести:

Мониторинг удовлетворенности продуктами и услугами;

Мониторинг удовлетворенности потребителей при разрешении жалоб;

Мониторинг удовлетворенности потребителей от общения.

Как правило, при осуществлении мониторинга изменений на рынке или в ожиданиях потребителей мы:

Определяем слабые стороны в предложении продуктов/услуг;

Идентифицируем новые инновации, которые обеспечивают потребности потребителей;

Определяем реакцию потребителей на конкурирующие предложения.

На шаге 02 *«Разработка видения и стратегии».* На этом шаге мы проводим мониторинг внешней среды, определяем концепцию бизнеса и стратегию бизнеса, также разрабатываем организационную структуру и систему взаимоотношений между организационными единицами, а также проводим ранжирование и разработку цели организации.

Для осуществления мониторинга внешней среды необходимо:

- анализировать и выявить причины конкуренции;
- определить экономические тренды;
-идентифицировать политические и правовые вопросы;
- оценить новые технологические инновации;
-провести анализ демографии;
- идентифицировать социальные и культурные изменения;
-провести анализ экологических проблем;

Также для определения концепцию бизнеса и стратегию бизнеса мы выбираем релевантные рынки, формулируем стратегию бизнес – единиц, разрабатываем всеобщую формулировку миссии организации.

На шаге 03 *«Разработка продуктов и услуг»* мы в

первую очередь разрабатываем концепцию и план продукта/услуги, далее разрабатываем, создаем и оцениваем прототипы продуктов и услуг, в- третьих, совершенствуем существующие продукты/услуги, а также проводим тестирование эффективности новых или измененных продуктов/услуг и управляем процессом разработки продукта/услуги.

На шаге 04 *«Продажа продуктов/услуг»* мы позиционируем наши продукты/услуги на сегментах потребительского рынка, обрабатываем заказы потребителей.

Позиционирование продуктов/услуг включает в себя:

- разработку ценовой, рекламной стратегии;

-разработку маркетинговых слоганов;

-оценку возможности рекламы и требования по ее финансированию;

-идентификацию выделенных целевых потребителей и их потребности;

-разработку прогноза продаж;

-продажу продуктов/услуг;

-ведение переговоров об условиях поставки.

Задание 2. Разработать вербальное описание бизнес-процесса «Подготовка и оформление заявки на товар»

Описание бизнес-процесса «Подготовка и оформление заявки на товар»:

Процесс «Подготовка и оформление заявки на товар» включает в себя задачи по определению потребности в товаре, ведение справочника товаров и создание, просмотр, редактирование заявок на основании плана потребностей в товарах на определенный период.

При выполнении шага 01 «Определение потребности в товарах» происходит определение потребности в товаре исходя из анализа исторических данных о продажах, анализа рыночной ситуации и прогноза спроса. В результате определяются базовые характеристики требуемого товара, необходимое его количество и сроки поставки. На основании данной информации составляется проект заявки на товар и передается в отдел закупок.

На шаге 02 «Проверка наличия товара в справочнике» необходимо проверить наличие товаров, которые нужно заказать в общем справочнике. Если товар не найден, необходимо перейти к шагу 03 «Заведение нового товара в справочник», иначе - к шагу 04 «Подготовка заявки на товар».

На шаге 03 «Заведение нового товара в справочник» необходимо занести информацию по новому товару в общий справочник. Нужно внести в систему следующие данные:

- название;
- единица измерения;
- цена единицы;
- себестоимость единицы;
- наименование поставщика.

При выполнении шага 04 «Подготовка заявки на товар» на основании проекта заявки на товар составляется список товаров, которые необходимо приобрести. В заявке необходимо указать:

- код товара;
- количество товара;
- дату заказа.

На шаге 05 «Согласование заявки на товар» исполнитель, подготовивший заявку на товар, согласует ее содержание с руководителем функционального отдела. Если в процессе согласования потребовалось внести изменения, исполнитель проводит повторную подготовку и согласование заявки на товар. В этом случае необходимо перейти к шагу 04 «Подготовка заявки на товар», иначе - к шагу 06 «Оформление заявки на товар».

На шаге 06 «Оформление заявки на товар» согласованную заявку необходимо оформить в соответствии с внутренними правилами компании.

Задание 3. Разработать вербальное описание бизнес-процесса «Приемка товара».

Описание бизнес-процесса «Приемка товара»: В рамках данного процесса ответственный за приемку товаров получает документы основания (в данном случае зарегистрированные заказы на покупку), которые служат источником для формирования приходных накладных и заданий на размещение товаров работникам склада.

При выполнении шага 01 «Получение документов оснований» ответственный за прием товаров получает список выпущенных заказов на покупку. После этого, на шаге 02 «Формирование приходной накладной», ответственный формирует приходную накладную, в которой должна содержаться следующая информация:

Заголовок документа:

- номер документа;
- информация по складу;

- дата учета.

Строки документа:

- ссылка на документ-основание;
- описание товара;
- количество для получения;
- принятое количество;
- дата выполнения.

Далее, на шаге 03 «Формирование инструкции на размещение», ответственный за приемку товара формирует инструкцию работнику склада на размещение товара, в которой указывает следующую информацию:

Заголовок документа:

- номер документа;
- информация по складу.

Строки документа:

- ссылка на документ-основание;
- информация по товару;
- количество для размещения;
- указание по перемещению товара.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем особенности вербального описания бизнес-процесса «Анализа рынка и потребности потребителей»?
2. В чем особенности вербального описания бизнес-процесса «Подготовка и оформление заявки на товар»?
3. В чем особенности вербального описания бизнес-процесса «Приемка товара»?
4. Что такое процессный подход?
5. Дать определение бизнес-процесса.
6. Что описывают регламенты бизнес-процесса?
7. На какие цели ориентированы основные бизнес-процессы организации?

ПРАКТИЧЕСКАЯ РАБОТА №6

СОЗДАНИЕ ИНФОРМАЦИОННОЙ ЭЛЕКТРОННОЙ ПОДДЕРЖКИ ПРОЕКТА В ПРОГРАММЕ TG BULDER

1. Цель работы:

1. Изучение и приобритение навыков работы в TG Bulder, путем построения простейшей структуру программы;
2. Изучение порядка действий при вычислении выражений;
3. Получение навыков построения простейших форм приложений по вводу-выводу данных.

2. Краткие теоретические сведения

TG Builder — это программный комплекс, предназначенный для разработки, сопровождения, изменения и публикации технической документации. Его функциональные возможности позволяют разрабатывать документацию соответствующую требованиям, как отечественных стандартов в области технической документации (ГОСТ 2.051-2006, ГОСТ 2.601-2006, ГОСТ 2.602-95, ГОСТ 2.610-2006, ГОСТ 2.611-2011), так и требованиям международной спецификации S1000D.

Утилита TG Update входит в стандартный комплект поставки TG Builder, однако оно поставляется как отдельное приложения и к нему нельзя получить доступ из основного меню TG Builder. Это связано с тем, что данная утилита используется только для обновления документации у пользователя документации. Т.е. она может быть передана пользователю документации при поставке документа, чтобы пользователь мог самостоятельно актуализировать документ [4].

TG WebServer является отдельным приложением и не входит в стандартный комплект поставки TG Builder. Данное

приложение предназначено для предоставления удаленного доступа к интерактивной технической документации через интернет. TG WebServer будет посвящена отдельная статья.

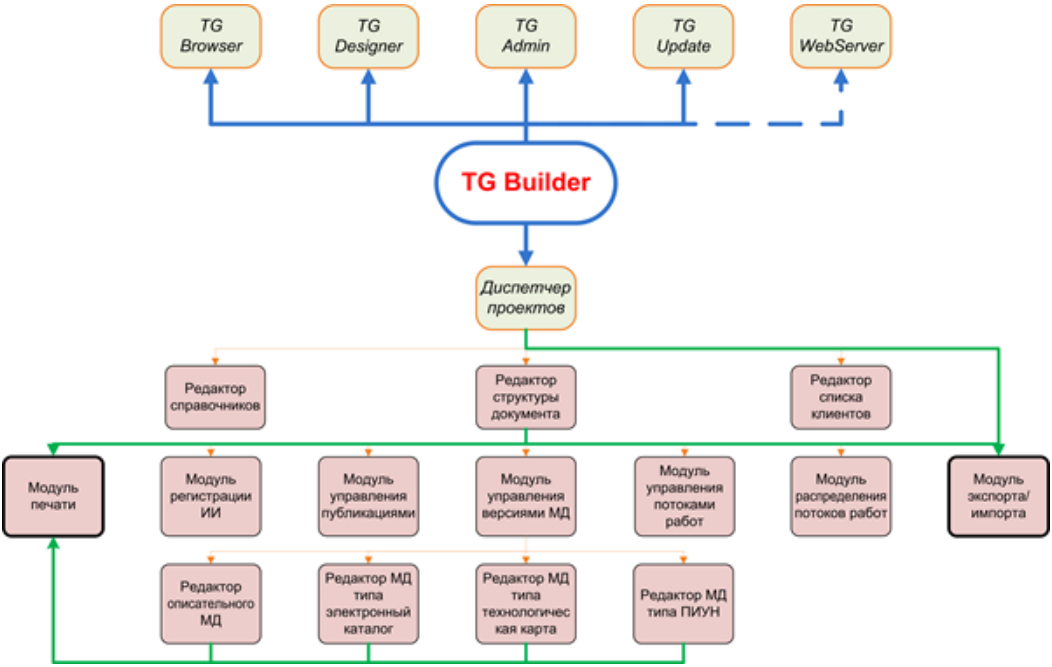


Рис.6.1.Структурная схема TG Builder

Структурно TG Builder состоит из 4 основных программных модулей — это Диспетчер проектов, TGDesigner, TG Browser и TG Admin.

Представленная структура (рис.6.1) охватывает все этапы жизненного цикла эксплуатационного документа, в том числе такой сложный этап, с точки зрения его реализации, как внесение изменений у пользователя документации.

Как видно из рис. 2 наиболее структурно сложным элементом TG Builder является программный модуль. Диспетчер проектов, предназначенный для создания,

удаления и редактирования проектов по разработке технической документации. Более подробную информацию по остальным модулям, можно почерпнуть из соответствующих источников [5].

3. Порядок выполнения работы

Для составления простейшей программы, необходимо придерживаться следующей последовательности:

3.1 Изучить теоретический материал, представленнный в разделе краткая теория: интерфейс, структуру и принципы работы в программе TG Builder.

3.2 Составить блок-схему и написать программу решения выражения согласно варианту, заданного преподавателем.

3.3 Использовать компоненты: Label, Edit, Button, Memo и вкладки Standard среды Си++Builder, при составлении программы.

3.4 Вывести на печать 5 вариантов исходных данных и результатов выполнения программы.

3.5 Ответить на контрольные вопросы по соответствующей работе.

4. Экспериментальные исследования

Вычислить арифметическое выражение $y = x^2 - 2^{mx} \sin 5x + a^{3,2}$, если переменные имеют следующие значения: a=0.2; x=0.4; m=2.

Порядок составления программы для вычисления математического выражения состоит из нескольких задач, каждая из которых задействует функционал программы TG

Builder, приведен далее.

1. Алгоритм решения задачи представляем на блок схеме. Алгоритм - линейный (рис.6.2).

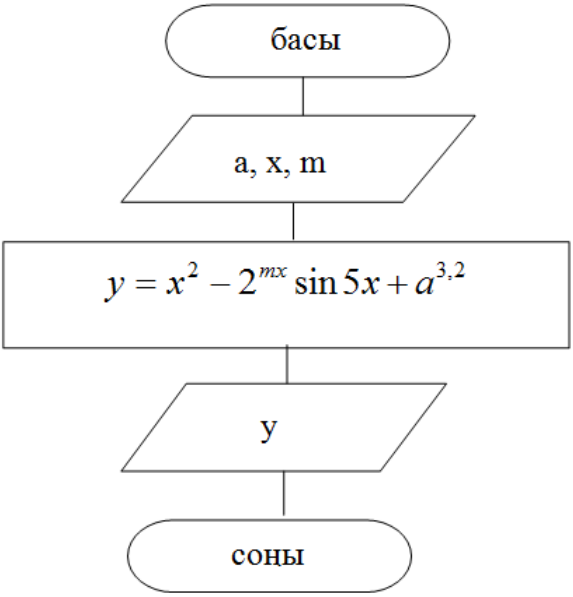


Рис.6.2. Блок-схема алгоритма программы

2. На форму из палитры компонентов вкладки Standard среды Си++Builder установить следующие компоненты: пять - Label, три - Edit, два – Button, задаем свойства. Для компоненты Form1 установить BorderStyle – формат формы, Width – ширину формы; Height – высоту (рис.6.3).

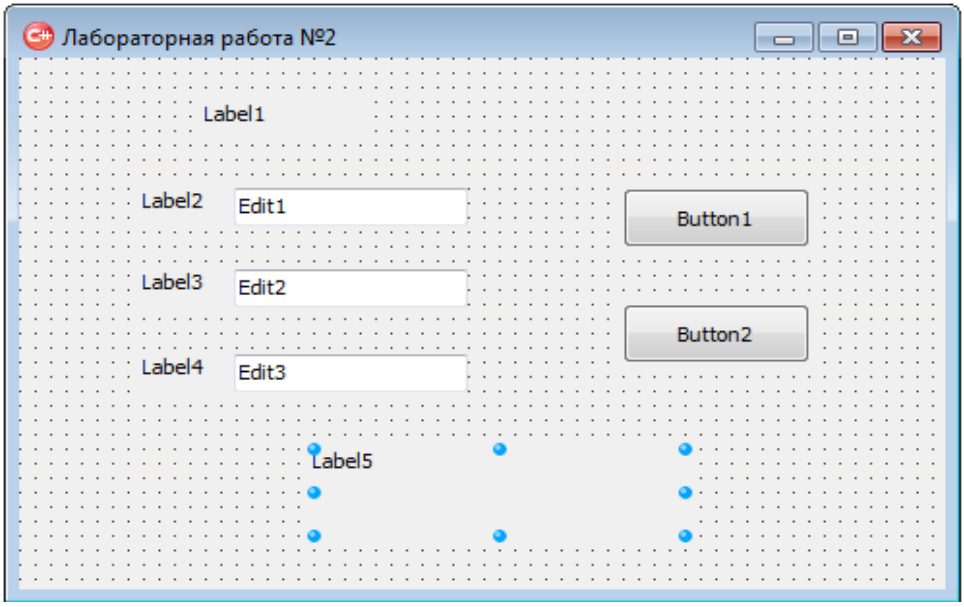


Рис.6.3. Заготовка окна интерфейса программы в TG Bulder

Для компонентов Label установить свойства Caption, Left, Top, для двух командных кнопок Button – в свойстве Caption отразить выполняемые действия. Поле Edit предназначено для ввода значений переменных, поэтому необходимо его свойство Text очистить. Для этих целей можно воспользоваться возможностями страницы Properties окна инспектора объектов (Object Inspector) Си++Builder.

Таблица 1. Свойства компонента Form1

Компонент	Свойства	Значение
Form1	BorderStyle	bsSizeable
	Caption	Лабораторная_работа№2
	Width	495
	Height	332

Таблица 2. Свойства компонентов Label, Edit, Button

Компонент	Caption	Text	Left	Top
Label1	Задача №1		96	24
Label2	a =		64	72
Label3	x =		64	117
Label4	m =		64	164
Label5	-		152	216
Edit1		-	111	72
Edit2		-	111	117
Edit3		-	111	164
Button1	Вычислить			
Button2	Очистить			

4. Для использования математических функций необходимо подключить файл math.h с помощью директивы препроцессора #include <math.h>.

5. Для вычисления выражения поставленной задачи вызываем событие OnClick компоненты Button1: имя метода состоит из имени компоненты и имени события: Button1+Click=Button1Click. В результате страка в программе выглядит следующе:

void __fastcall TForm1::Button1Click(Sender: TObject);

6. Необходимо объявить переменные, согласно заданию: m - целое, a, и x - вещественное, результат y – число вещественного типа: int m; double a,x,y;

7. Считывание данных

a=StrToFloat(Edit1.Text);

x=StrToFloat(Edit2.Text); //перевод строкового типа данных в вещественные

m=StrToInt(Edit3.Text);

8. Принимая во внимание правило записи арифметических выражений, вычисляем значение переменной y:

y = pow(x,2)-pow(2, m*x)*sin(5*x)+pow(a, 3.2);

9. Для вывода результатов вычислений , используем

ствойство Caption компоненты Label5 :

Label5->Caption="Y = "+FloatToStrF (y,ffFixed,5,2);

10. С помощью функциональной клавишы F12 или кнопки  панели иснтрументов переходим с окна кода редактора программы на окно формы .

11. Теперь редактируем событие кнопки Button2:

void __fastcall TForm1::Button2Click(Sender: TObject);

12. Для обновления данных в поле компоненты Edit необходимо с помощью функции Clear () очистить его:

Edit1->Clear(); Edit2->Clear();Edit3->Clear();

13. Повторяем пункт 9

14. При сохранении проекта с помощью команды File→Save Project As основного меню среда Си++Builder предлагает два диалоговых окна: Save Unit As – сохранение модуля ; Save Project As – сохранение проекта. В обоих окнах в поле Имя файла необходимо ввести имена модуля и проекта.

15. С помощью функциональной клавиши F9 или кнопки  панели инструментов произвести компиляцию проекта

16. Ввести значения переменных в поле компонентов Edit1, Edit2, Edit3

17. Текст программы:

```
//---------------------------------------------------
----------------------

#include <vcl.h>

#include <math.h>

#pragma hdrstop
```

```
#include "Unit1.h"
//---------------------------------------------------
---------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------
---------------------
__fastcall TForm1::TForm1(TComponent*
Owner)
: TForm(Owner)
{
}
//---------------------------------------------------
---------------------
void __fastcall
TForm1::Button1Click(TObject *Sender)
{
int m; float a,x,y;
a= StrToFloat(Edit1->Text);
```

```
        x= StrToFloat(Edit2->Text);
        m= StrToInt(Edit3->Text);
        y = pow(x,2)-pow(2, m*x)*sin(5*x)+pow(a,
3.2);
        Label5->Caption = "Y = " +
FloatToStrF(y,ffFixed,5,2);
        }
        //---------------------------------------------
---------------------
        void __fastcall
TForm1::Button2Click(TObject *Sender)
        {
        Edit1->Clear(); Edit2->Clear(); Edit3-
>Clear();
        }
        //---------------------------------------------
---------------------
```

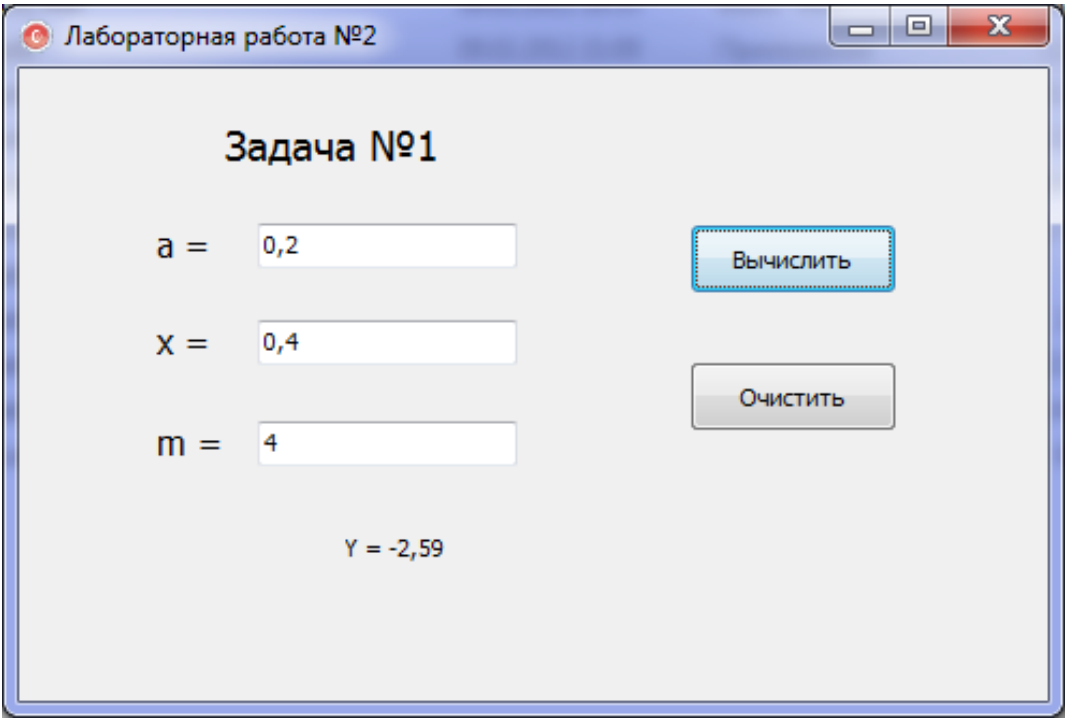


Рис.6.4. Окно готовой программы в TG Bulder.

В результате выполненной работы, готовая программа должна выглядеть в виде окна с интерфейсом представленным на рис.6.4, и производить функции расчета согласно заданному алгоритму.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем особенности заключаются особенности работы программного комплекса TG Bulder?
2. Объясните структуру программного комплекса TG

Bulder?

3. Какие возможности для автоматизированного управления проектами предоставляет программа TG Bulder?

4. Опишите основные этапы построения программы в среде TG Bulder.

ПРАКТИЧЕСКАЯ РАБОТА №7

АВТОМАТИЗАЦИЯ ПРОЦЕССОВ В СИСТЕМЕ MASTERSCADA - ФОРМИРОВАНИЕ ИСХОДНЫХ ДАННЫХ ПРОЕКТА

1. Цель работы

1.1 Изучить основные понятия, структуру и назначение отдельных элементов SCADA системы программы MasterScada.

1.2 Изучить порядок работы по созданию автоматизированной системы управления технологическими процессами с помощью редакторов SCADA системы.

1.3 Разработать проект учебной автоматизированной системы управления технологическими процессами SCADA системы в программе MasterScada.

2. Краткие теоретические сведения

MasterSCADA — самый современный, инновационный мощный и удобный инструмент для быстрой и качественной разработки систем. В нем реализованы средства и методы разработки проектов, обеспечивающие резкое сокращение трудозатрат и повышение надежности создаваемой системы. Разрабатывать проекты в Master SCADA легко и приятно - это первая в нашей стране система, в которой реализован объектный подход к разработке систем управления, учета или диспетчеризации [5].

В основе построения MasterSCADA лежит объектно-ориентированный подход, который фактически стал единственным и общепризнанным как для построения систем программирования, так и программ пользователей. Кроме того, объектно--ориентированный подход является естественным для технолога, разрабатывающего SCADA для

своего техпроцесса, поскольку объект в SCADA проекте однозначно соответствует контролируемому (управляемому) технологическому объекту.

В MasterSCADA проект создается как отображение взаимосвязи модели технологического объекта, представленной в виде дерева технологической иерархии, и модели системы контроля и управления — в виде дерева иерархии технических средств системы (рис.7.1).

Предположим, к примеру, что на заводе имеются цеха, которые подразделяются на участки, на участках имеются технологические аппараты, обвязанные исполнительными механизмами и датчиками. Это технологический объект. АСУ ТП объединяет операторские станции, к которым по каналам связи подключены контроллеры, состоящие из модулей ввода-вывода, содержащих входы и выходы. Свяжем датчик в дереве объекта с входом модуля контроллера в дереве системы и получим взаимосвязь этих моделей. Связь устанавливается простым перетаскиванием с помощью "мышки" одного элемента на другой в любом направлении. Можно представить это и как взаимосвязь логического и физического уровней представления АСУ ТП.

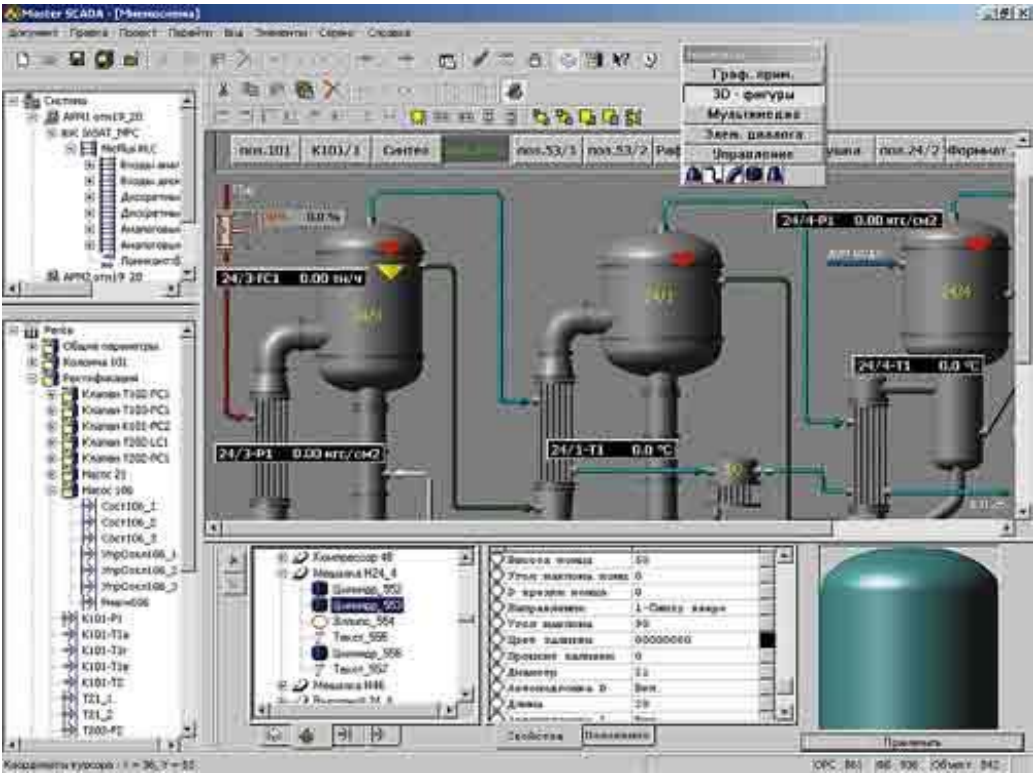


Рис.7.1 Графический интерфейс программы Master-Scada

Существенно и то, что проект разрабатывается не для отдельного рабочего места, а для всей АСУ ТП в целом, включая все операторские станции и все PC-совместимые контроллеры. При этом никакой настройки сетевых связей как при разработке, так и при переносе системы на целевую платформу не требуется. Единственная настройка — это выбор для объекта его базового компьютера (или контроллера) в дереве системы. При старте системы все узлы находят друг друга по проектным именам, не имеющим никакого отношения к сетевым именам узлов.

Реализованная в пакете концепция “всё в одном” обеспечила “бесшовное” объединение всех

функциональных модулей в едином пользовательском интерфейсе, который пользователь с легкостью воспринимает даже без предварительного обучения на курсах или изучения документации.

Выбор объекта в качестве центральной единицы разработки проекта позволил все переменные рассматривать как элементы его структуры, а документы — как его свойства. В MasterSCADA нет нужды редактировать отдельно список тегов, отдельно — мнемосхемы, отдельно — тренды и каждый раз держать в памяти их взаимосвязи. Выбрав объект, мы можем быть уверены, что на закладках его свойств в пользовательском интерфейсе MasterSCADA есть все необходимое.

3. Порядок выполнения работы

Для разработки автоматизированной системы управления технологическими процессами в SCADA системе, необходимо придерживаться следующей последовательности:

1) Ознакомиться с возможностями и структурой системы MasterScada используя информацию в презентации.

2) В программе MasterScada изучить основные функции интерфейса и подготовиться к разработке учебного проекта.

3) Используя инструкцию и рекомендации в учебном пособии, выполнить проект и сохранить его для дальнейшей работы.

4) Подготовить отчет о выполненной работе и защитить его.

Рекомендации по выполнению практической работы:

1. Начало работы в системе MasterScada.

1.1 Работу по созданию первого проекта в системе Master-Scada следует начинать с открытия окна программы. При этом автоматически появиться меню *Создание проекта*, в окне которого следует ввести *имя проекта* и утвердить его, нажав *ОК*.

Если Вы хотите защитить проект паролем, то возможно использовать эту опцию, вписав пароль, в противном случае нажимаете *ОК*.

2. Настройка дерева системы.

2.1 В результате действий п.1.1, появиться окно программы в виде представленном на рис.7.2. Интерфейс рабочего окна программы MasterScada содержит: «Панель Меню»; «Дерево объектов»; «Панель свойств»; «Палитру функциональных блоков». Рассмотрим каждый элемент программы и его функциональное назначение.

Добавим в окно Дерево системы, новый объект – Компьютер (используя правую кнопку мыши и выбор в контекстном меню), он будет является базисом нашего проекта.

2.2 Связь от устройств (приборов) обеспечивающих автоматический контроль, управление и др. функции осуществляется через OPC DA сервер, который требуется так же добавить в дерево системы.

Используя команду *Поиск OCP DA серверов*, выбрать в окне на рисунке, сервер *InSAT Modbus OPC*, затем добавить его в *Компьютер*, командой «Вставить» *OPC Сервер* через контекстное меню.

Сервер *InSAT Modbus OPC*, является эмулятором связи от приборов автоматизации и в реальном проекте, может быть другим, например для подключения к контроллеру Siemens S7-1200.

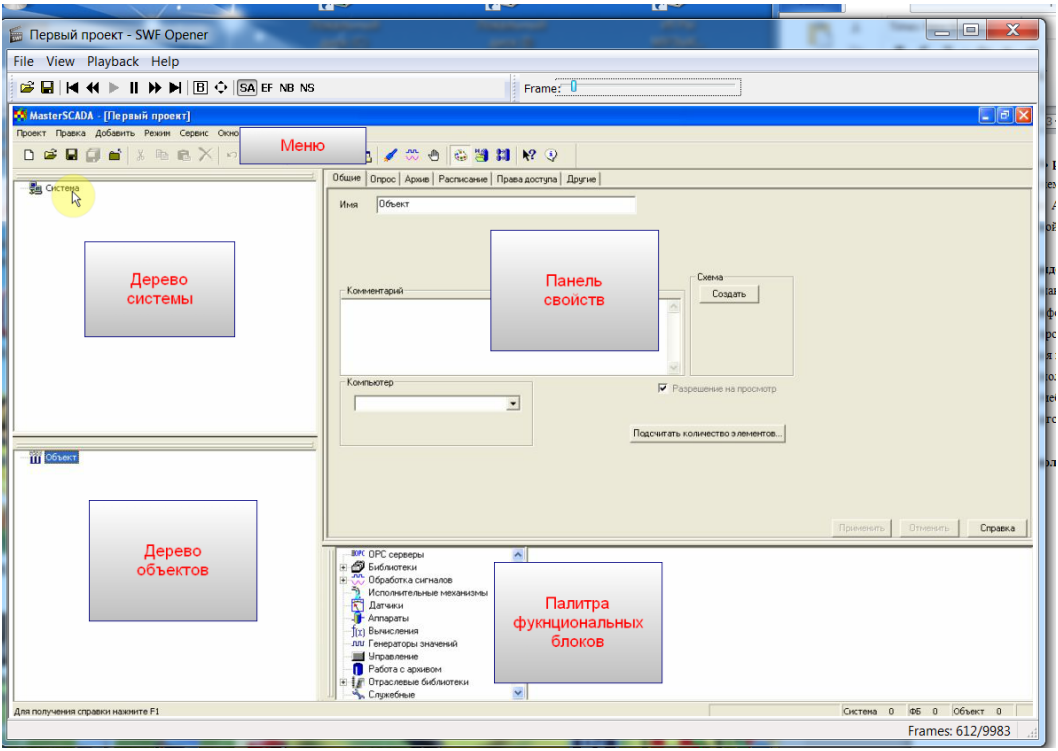


Рис.7.2 Интерфейс программы MasterScada – главное окно

Modbus Universal Master OPC Server сочетает в себе возможности OPC-сервера наиболее распространенного промышленного протокола передачи Modbus RTU/ASCII/TCP, а также инструментария для разработки новых OPC-серверов, как для поддержки специализированных расширений Modbus, так и для поддержки любых иных протоколов.

MasterOPC реализует два набора OPC-интерфейсов – DA (Data Access – текущие данные) и HDA (Historical Data Access – архивные данные). Для организации хранения архивов опрашиваемых переменных Master OPC использует встроенный SQL-сервер.

OPC-сервер имеет в своем составе поддержку

простого сценарного языка, что позволяет проводить предварительную обработку данных после их считывания из внешних устройств, а также перед записью в них (рис.7.3).

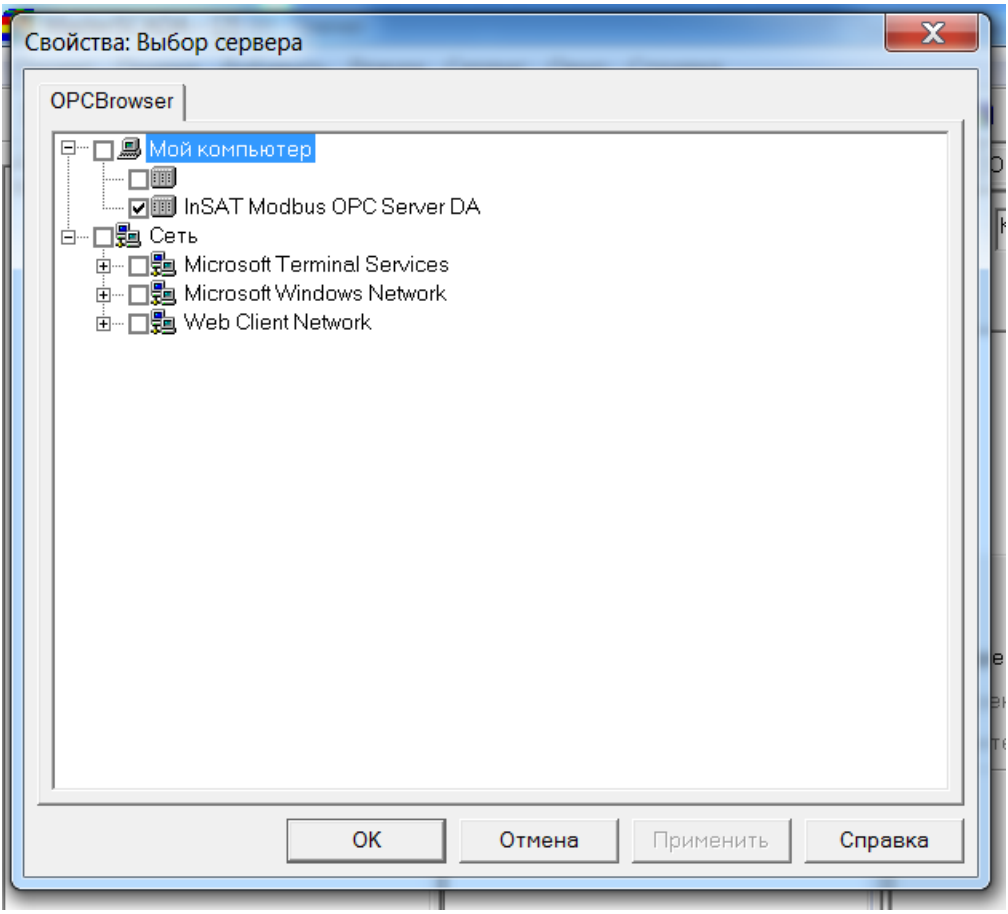


Рис.7.3 Окно выбора сервера программы MasterScada

Возможно использование сценариев для написания новых драйверов (как для протоколов, построенных на транспорте Modbus, так и любых других), сохранения архивов в SQL-сервере, написания имитаторов сигналов,

вычисления косвенных параметров, работы с признаками качества и т.п. Сценарии могут использоваться на уровне коммуникационных узлов, устройств и подустройств, отдельных тегов. Встроенный редактор обеспечивает стандартный сервис - подсветку ключевых слов, удобную работу с тегами и библиотеками. Ниже прилагается документация по разработке собственных протоколов – на примере DCON, Rnet, расширенный Modbus, а также по работе с архивами.

Сервер содержит встроенные средства типовых обработок: автоматическое преобразование типа значения, перевод в реальные единицы измерения, перестановку байтов в любом порядке (слова длиной до 8 байтов), выделение битов и т.п.

Для облегчения тиражирования OPC также поддерживает возможность экспорта и импорта конфигураций устройств. В поставку OPC включены все приборы фирм ОВЕН, ICP DAS и Delta Electronics работающих по протоколу Modbus (список готовых конфигураций различных устройств). Пользователь может создавать, сохранять и распространять собственные библиотеки устройств.

Master OPC также поддерживает работу по каналам GSM или иной модемной связи, что позволит использовать его в системах диспетчеризации и удаленного сбора данных. Для работы в радиосетях и иных сетях, требующих дополнительной адресации устройства передачи, возможно использование лидирующего префикса перед кадрами Modbus.

4. Экспериментальные исследования

В данной практической работе, будем рассматривать систему сбора данных, построенную из электрических приборов: вольтметра и амперметра.

4.1 Настройка дерева объектов.

3.1 Для реализации системы сбора данных необходимо добавить те переменные, которые хотим оследить и управлять. Выполняя команду Вставить - OPC переменные, добавим из списка все, которые находиться там в папке прибор: Напряжение, Ток, Угол (рис.7.4).

Для редактирования структуры программы, необходимо осуществить добавление каких либо объектов (приборов) в дерево объектов.

Объект – элемент программы, предназначенный для размещения в нем других элементов, переменных и функциональтных блоков. Объекты добавляются через контектное меню в левом нижнем углу окна программы. Командой Вставить – Объект, при этом на закладке Свойства – Общие, зададим ему имя – «Установка».

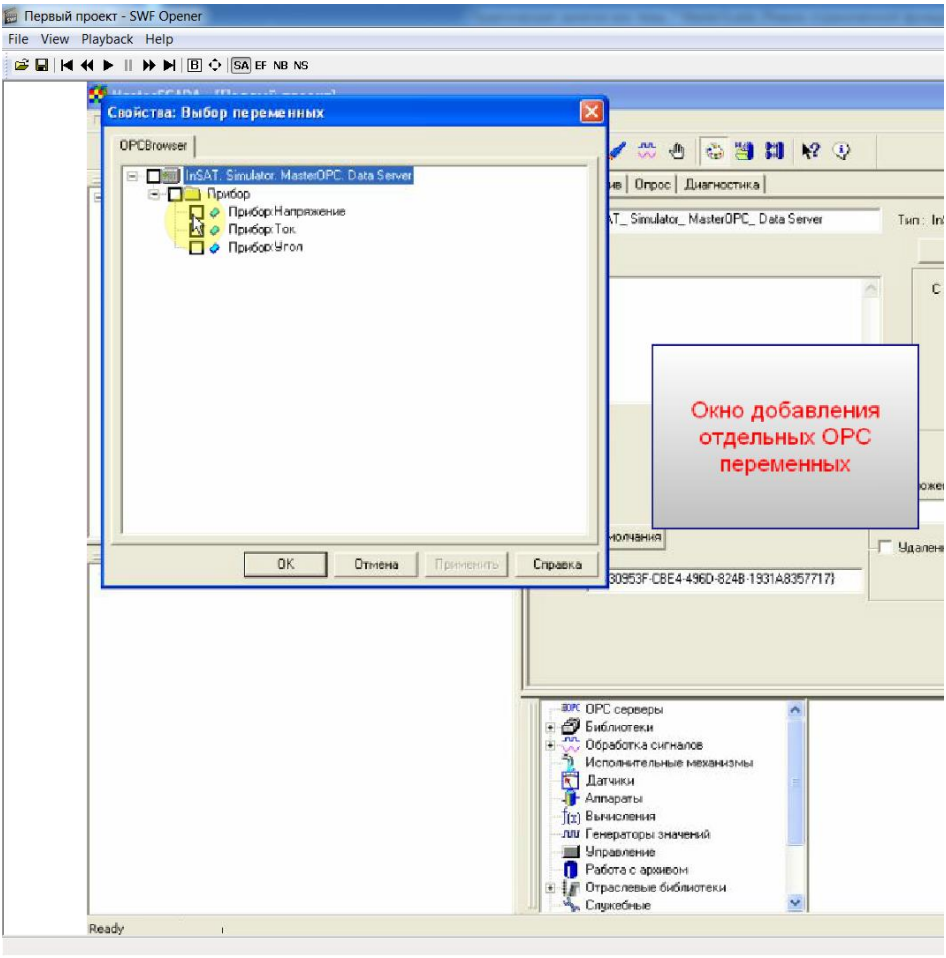


Рис.7.4 Окно выбора выбора переменных программы MasterScada

В дереве объектов, в пункте «Установка», добавим еще объект и назовем его «Реактивная», по функциональному назначению. В нем мы будем считать реактивную мощность, по следующему вражению:

$$Q = U \cdot I \cdot \sin\varphi$$

Для этого воспользуемся уже готовыми формами выражений, библиотека которых находиться в правом

нижнем углу. Перетащив функциональный блок «sin» в созданный объект, раскроем его структуру и присвоим входные параметры: «Период», «Диапазон», «Аргумент» и выходной: «Синус».

Объект рассчитывает значение по следующей формуле:

$$Синус = \frac{Диапазон}{2} \sin\left(\frac{2 \times \pi \times Аргумент}{Период}\right)$$

В соотвествии с выражением выше, присвоим значения переменным: Диапазон=2, Период=360. Для этого в меню период, отроем вкладку опрос и введем константу=360, для диапазона и нажмем кнопку «Применить».

Для работы средства измерения, теперь следует создать связь между прибором и нашим виртуальным объектом, для этого из окна «Дерево системы», «Прибор», «Угол», его следует перетащить в объект «Аргумент» (рис.7.5).

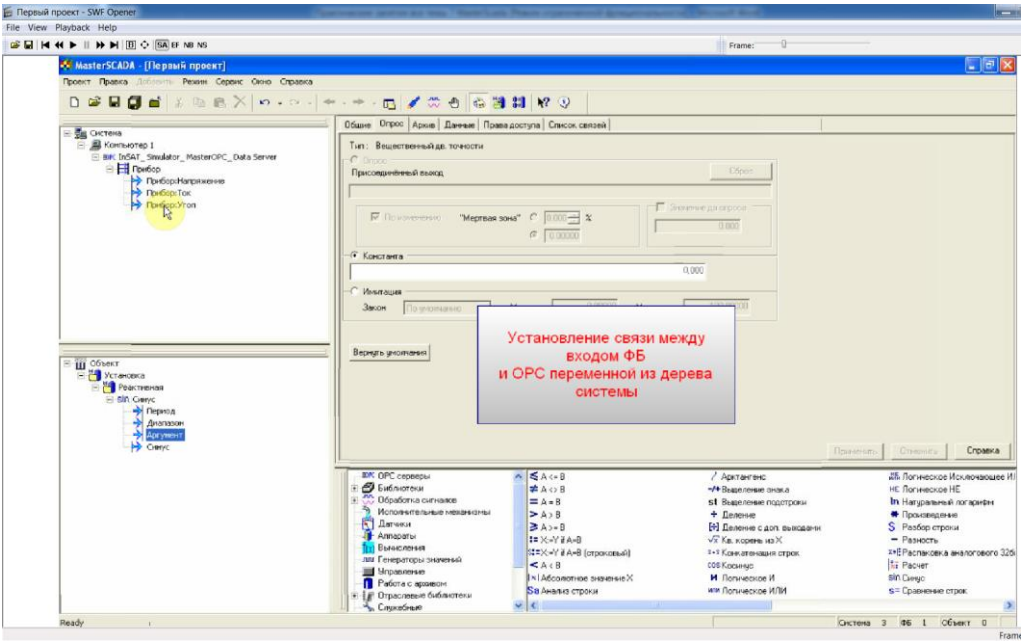


Рис.7.5 Установление связей переменных программы MasterScada

Для подсчета трех компонентов формулы реактивной мощности, воспользуемся действием «Произведение» из библиотеки программы. Как и любой объект, умножение необходимо настроить, поскольку количество множителей равно 3-м, то в правом окне, во вкладке настройке, установите число 3. Далее вновь выполним связь: «Множитель 0» – «Напряжение»; «Множитель 1» – «Ток», «Множитель 2» – «sin».

4.2 Создание дополнительных объектов

В качестве дополнительного объекта создадим имитацию активной мощности, для этого добавим объект с именем «Активное», через контекстное меню (рис.7.6). Активная энергия рассчитывается по формуле:

$$P = U \cdot I \cdot \cos\beta$$

Аналогичным образом зададим значения и связи объектов для расчета активной мощности. Поскольку используются действия умножения с тем же количеством множителей, то можно упростить создание объекта, простым копированием из «Дерева объектов» объекта «Произведение», и создать сиротствующую связь.

4.3 Создание расчетной формулы полной мощности.

Полная мощность представляет собой произведение напряжения и силы тока и находится по выражению:

$$S = U \cdot I$$

Воспользуемся для этого модулем Расчета, предварительно вставив его в объект. Расчет выполняется при помощи формулы, которая находиться на одноименной закладке.

Для этого, выделив предварительно, переместим элементы дерева объектов в таблицу исходных данных расчета (рис.7.7). Затем необходимо сформировать формулу, дважды нажав на пункт с исходными данными, выбрав параметр «Напряжение» и «Сила тока» (рис.7.8). Завершить работу с исходными данными, применив сделанные изменения.

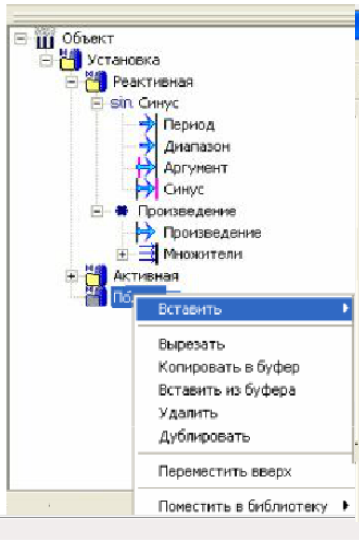


Рис.7.6 Окно выбора обьектов

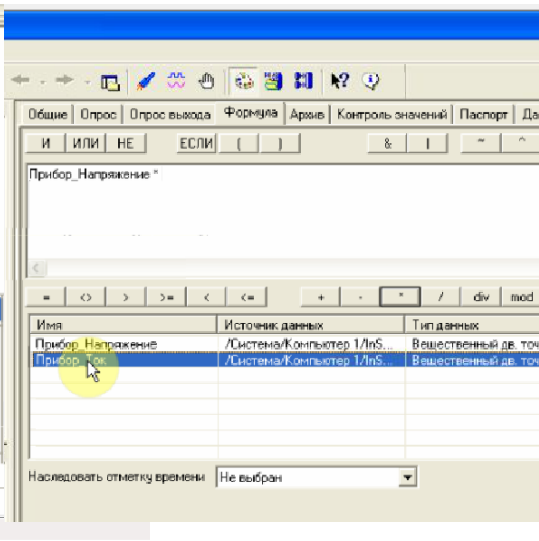


Рис.7.7 Окно свойств обьекта-прибора

Добавим в дерево объектов параметры сети: создадим объект с названием «Параметры», перетащим группу «Прибор» из OPC сервера в данный объект (находится внизу справа).

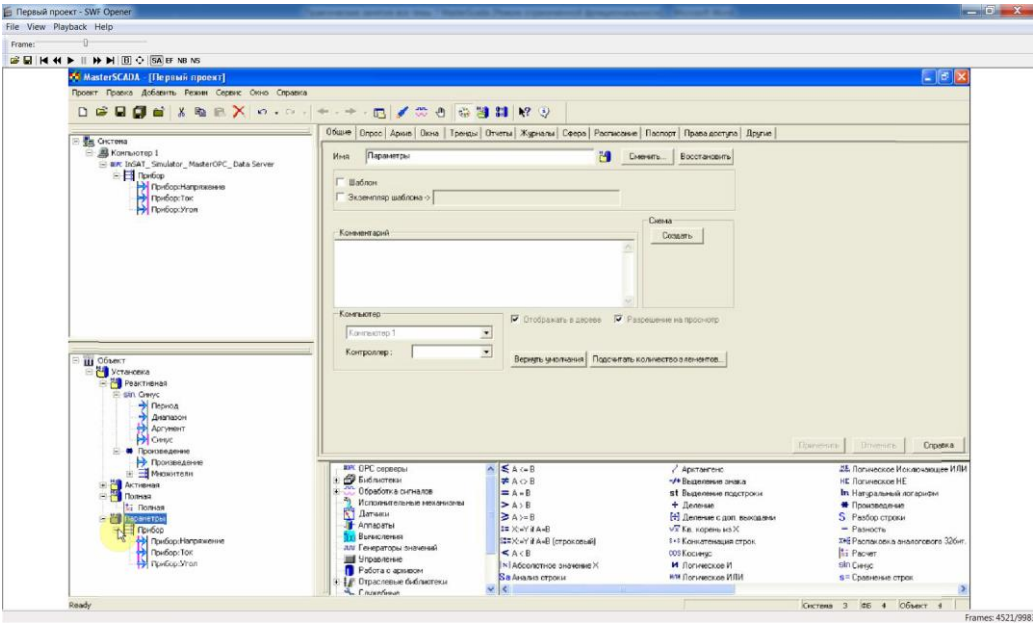


Рис.7.8 Окно выбора формулы объекта

В результате выполненной работы, должна быть подготовлена вычислительная подпрограмма, позволяющая снимать показания с эмулятора контрольно-измерительного прибора и выполнить последующий расчет его параметров. Созданное «Дерево объектов» с соответствующей структурой (см.рис.7.7), является подготовительным этапом к разработке и настройке мнемосхемы.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие возможности при управлении проектами дает Scada системы;
2. Поясните структуру типовой Scada системы на примере MasterScada;
3. В чем заключается функции «Дерева объектов»?
4. Что такое OPC-сервер и его функциональные возможности.

5. Какие интерфейсы применяются для подключения приборов контроля и регулирования с системой Master-Scada? Их преимущества и недостатки.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №8

АВТОМАТИЗАЦИЯ ПРОЦЕССОВ В СИСТЕМЕ MASTERSCADA - РАЗРАБОТКА И НАСТРОЙКА МНЕМОСХЕМЫ ПРОЕКТА

1. Цель работы

1.1 Изучить назначение и основные элементы мнемосхем в SCADA системах

1.2 Приобретение навыков по созданию мнемосхем типовых проектов.

1.3 Разработать мнемосхему, позволяющую визуализировать процессы АСУ ТП для проекта в программе MasterScada.

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2. Краткие теоретические сведения

Основным средством взаимодействия с оператором являются мнемосхемы – окна с представлением информации в графическом виде. На мнемосхемах отображаются состояние исполнительных механизмов и аппаратов, значения параметров системы, аварии и т.д.

Мнемосхемы в системе MasterSCADA принадлежат объекту. Число мнемосхем в проекте не ограничено. Число элементов на мнемосхеме также не ограничено. Библиотеки стандартных элементов содержат множество элементов, включая объемные элементы со встроенным индикатором заполнения, элементы для создания пользовательских диалогов, элементы, воспроизводящие полный комплект приборов щитового контроля и управления (рис.8.1).

Имеется встроенный редактор для создания

мультфильмов (с регулируемой прозрачностью изображения) с различными законами трансформации исходных графических файлов (покадровый показ, прокрутка в любом направлении, изменение резкости или размера и т.п.) [5].

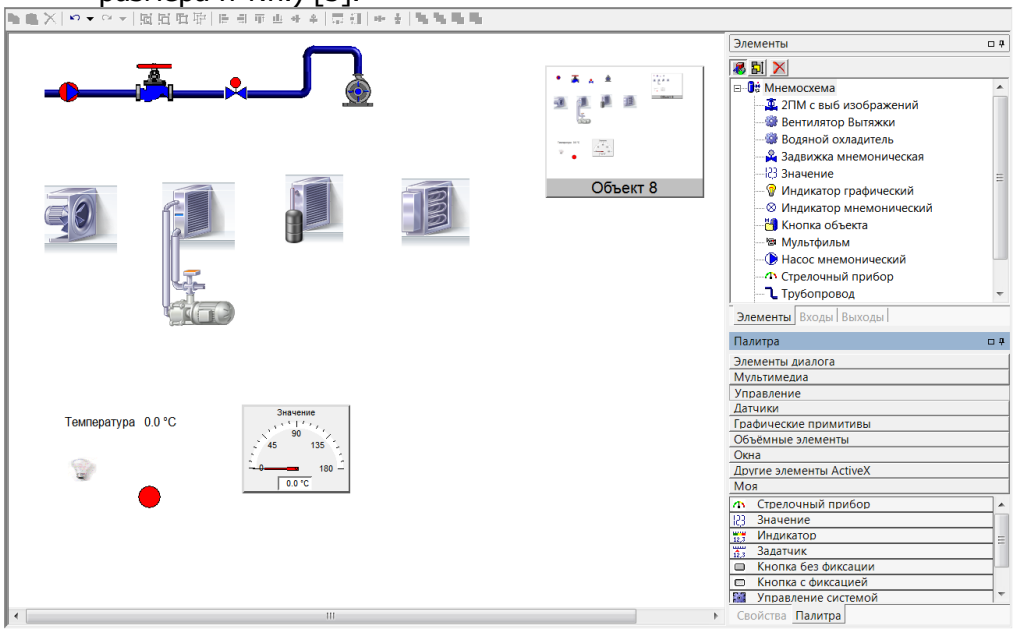


Рис.8.1 Пример интерфейса окна мнемосхемы в программе MasterScada

Среди файлов поддерживаются вставка основных мультимедийных форматов: avi, gif, jpg, png, gif, tiff, bmp.

Основной способ создания мнемосхем - перетаскивание элементов из дерева объектов: как включенных в проект из библиотек визуальных функциональных блоков и переменных, уже обладающих всей необходимой функциональностью (динамизированное изображение, окно управления и т.п.), так и созданных пользователем объектов со своими изображениями и окнами управления. Например, для переменной может быть выбран способ отображения: в виде текстового значения

или в виде одного из типовых приборов щитового монтажа.

Объект может быть представлен на мнемосхеме в виде кнопки с текстом, либо уменьшенным изображением его мнемосхемы (рис.8.2), также существует возможность создания изображений объекта – который можно иначе назвать «виджетом», т.е. элементом способным не только открывать мнемосхемы объектом, но и отображать актуальную информацию о состоянии объекта (например – наличие аварий, значение критических параметров, состояние главных исполнительных механизмов).

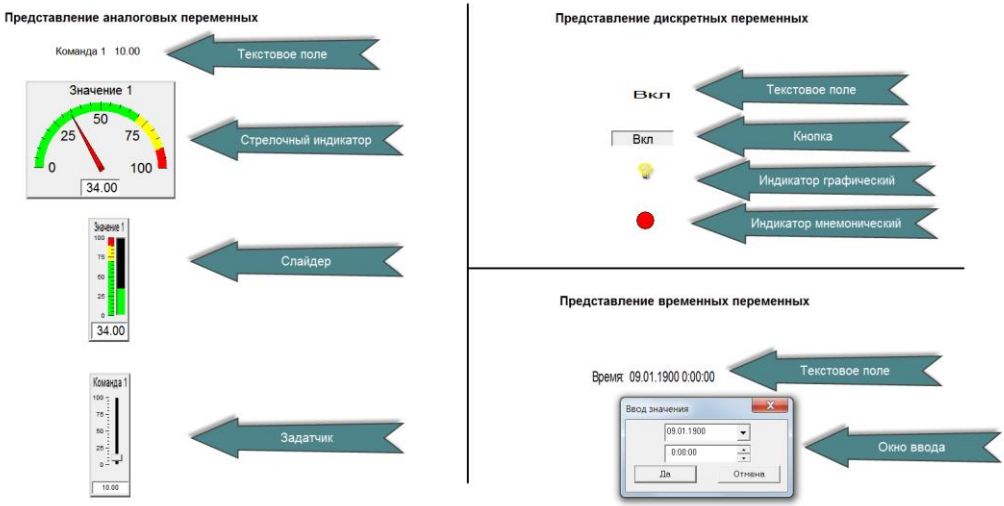


Рис.8.2 Пример отображения интерактивных объектов мнемосхемы

Также в качестве элемента мнемосхемы может быть использован любой элемент управления ActiveX, с возможностью динамизации любого его свойства, а также одного из стандартных свойств (положения, размера, отрисовки, мигания и т.п.). Связь с деревом проекта в этом случае осуществляется через переменные динамизации [8].

3. Порядок выполнения работы

Для разработки мнемосхемы автоматизированной системы управления технологическими процессами в SCADA системе, необходимо придерживаться следующей последовательности:

1) Ознакомиться с возможностями построения и структурой мнемосхем в системе MasterScada.

2) Используя инструкцию и рекомендации в учебном пособии, составить пнемосхему для учебного проекта разработанного в предыдущей практической работе.

3) Подготовить отчет о выполненной работе и защитить его.

Используя проект разработанный на предыдущем практическом занятии, требуется добавить к нему мнемосхему для визуализации отображения значений переменных и улучшенной их визуализации.

3.1 Создание мнемосхемы.

Практическое задание выполняется в соответствии с установленной последовательностью действий приведенных далее.

3.1.1 Вызовите левой кнопкой мыши на объекте «Установка», в дереве объектов, далее на вкладке «Окна», «Создать» (рис.8.3). В результате этих действий появиться чистое пространство - рабочее поле, предназначенное для размещения элементов мнемосхем.

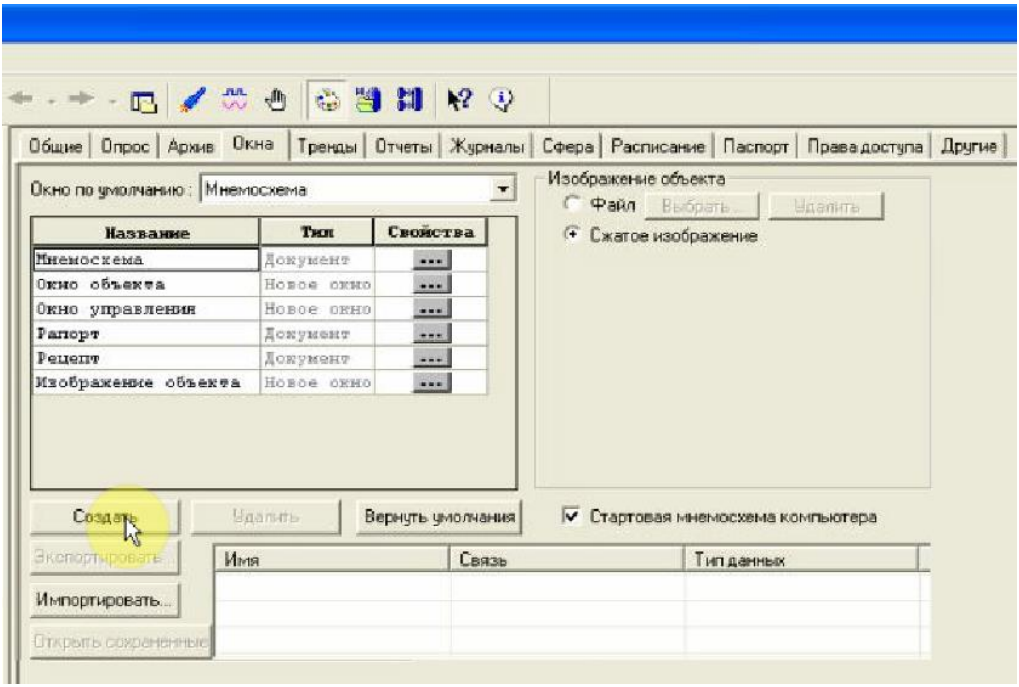


Рис.8.3 Окно выбора мнемосхемы проекта

3.1.2 Переместим из дерева объектов значение напряжения на рабочее поле мнемосхемы. При этом можно выбрать следующие варианты отображения информации: «значение» – отображает числовое значение; «индикатор» – показывает визуально информацию; «Стрелочный прибор» – используя заготовки приборов, выводит их на панель мнемосхемы и график – отображает график изменения параметров (рис.8.5).

При построении мнемосхемы, для отображения информации очень важно верно осуществить, выбор компонентов, в зависимости от контролируемых параметров. Поэтому, для визуализации электрических значений проекта, выбираем – «Стрелочный прибор».

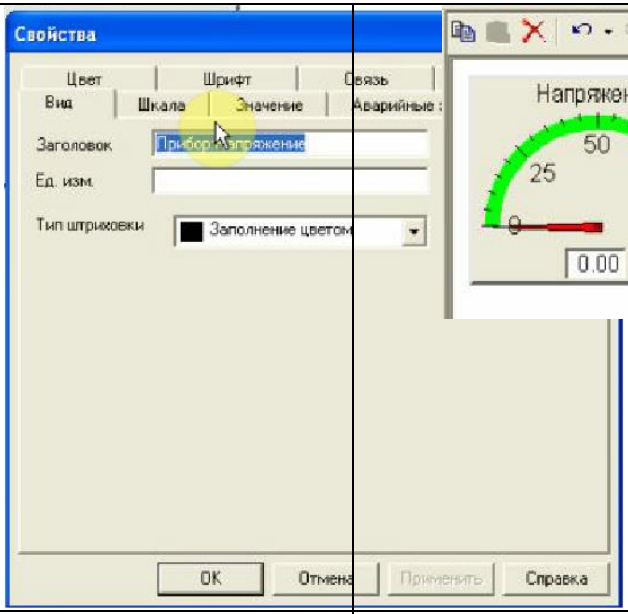


Рис.8.4 Окно свойств прибора

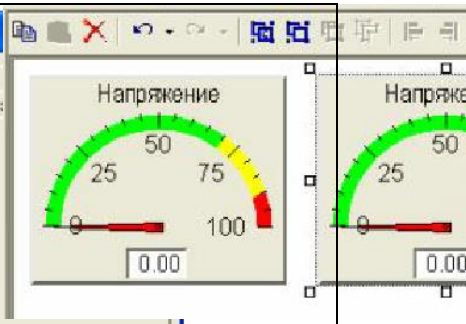


Рис.8.5 Внешний вид приборов

По умолчанию, заготовки отображаются в стандартном размере, для удобства использования настройте размеры (простым растягиванием окна объекта) и исправьте его название, щелкнув правой кнопкой мыши и выбрав – Свойства. Введите название – Напряжение (рис.8.4).

Необходимо создать для каждого исходного и резулитирующего сигнала по прибору. Для удобства построения мнемосхем, скопируйте элемент и вставьте его на поле.

3.1.3 Установка связей с мнемосхемой. Для этого перетащите переменную Ток в окошко Свойсво элемента. После этого связь с прибором будет установлена. Сделайте то же самое и для остальных выходных параметров: «Тока», «Реактивной мощности», «Активной мощности» и «Полной мощности».

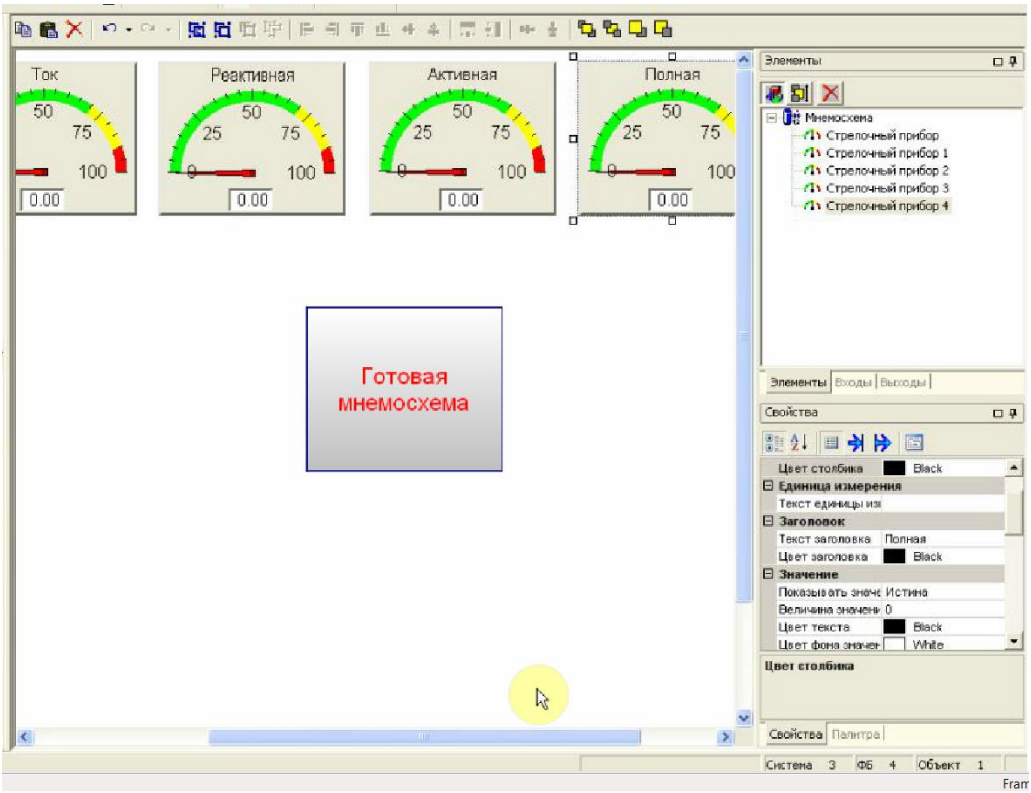


Рис.8.6 Окно рабочего поля мнемосхемы

В результате, дожно получиться окно с 5 иденичными приборами, имеющмим только с разные названия (рис.8.6). Все устройства добавленные вами должны появиться справа в дереве мнемосхемы, аналогично их объектам.

3.1.4 Задание истинных параметров приборов.

Все приборы имеют диапазон измерение 0-100, что не подходит по параметры наших измеряемых величин, кроме того отсутствуют единицы измерения. Для расширения диапазона измерений, можно отредактировать вручную каждую шкалу, однако это не всегда удобно, особенно если у нас есть множество приборов с одинаковым диапазоном измерений. Поэтому используем функцию

задания Шкалы, с одноименной закладки при выборе Системы из дерева системы.

Добавим шакалу и назовем ее Напряжение с диапазоном 0-250, выберим единицу измерения, В. Возможности программы MasterScada, позволяют создавать ограничение на получаемый сигнал с прибора и разным способом сигнализировать о его превышение (рис.8.7-8.8). Для этого необходимо указать аварийные и предупоетидительные границы. Если измеренное значение попадет в установленный диапазон, будет сформированно сообщение, а сама величина выделенна цветом.

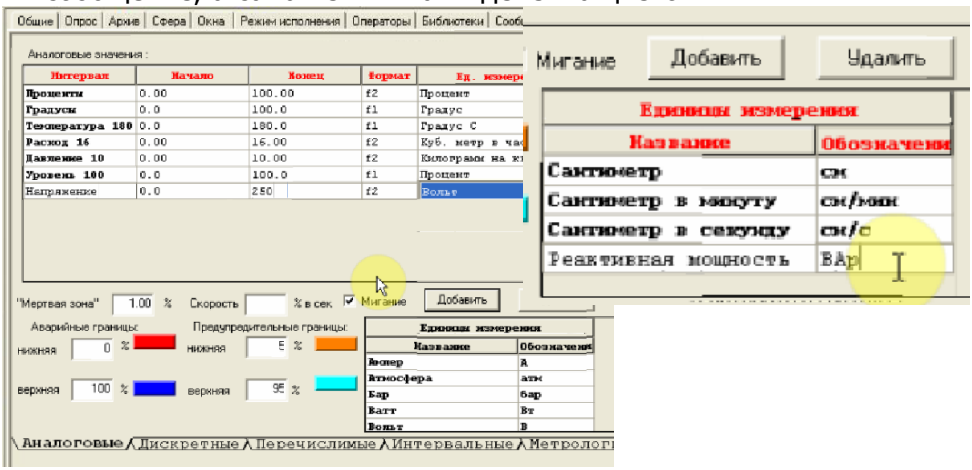


Рис.8.7 Окно настройки свойств мнемосхемы

Рис.8.8 Настройка отображения единиц измерения приборов

Для напряжения, аварию при значениях нижних границ уберем, т.к. отсутсвие напряжения в нашем случае не увляется аварийной ситуацией. Оставим только верхнюю границу. Аналогично создадим шкалы для: Тока (предел измерений 10 А, аварийные и предупредительные границы отсутсвуют); Реактивной мощности (предел измерений 2500 Вт, значения); Активной мощности (рис.8.9).

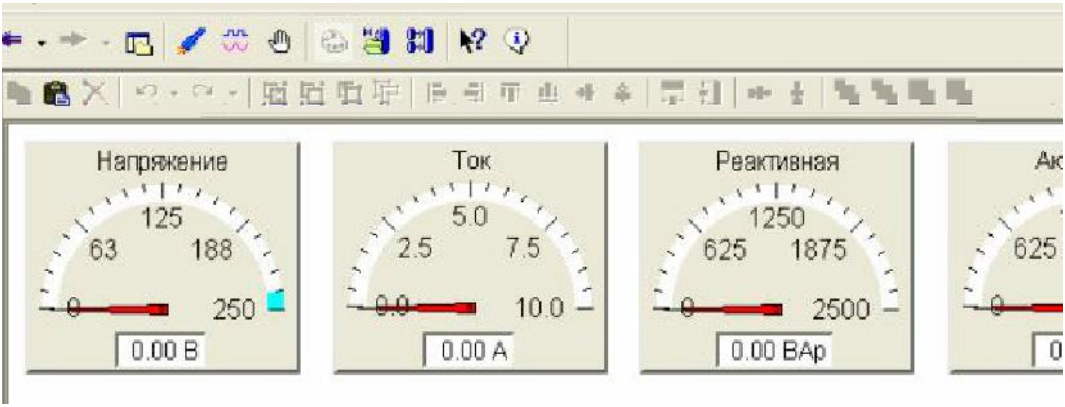


Рис.8.9 Приборная панель с измененными свойствами

Посокльку вы программе заложены лишь самые распространенные единицы, то для реактивной мощности, единыцы измрения которой ВАр следует их добавить вручную, с помощью кнопки Добавить. Так же добавить и единицы измерения полной мощности ВА. Теперь можно присвоить шкалам созданные единицы измерения.

3.1.5 Присвоение параметрам шкал значений.

Выбрать в дереве системы, «Прибор» – «Напряжение», вкладку «Общие». В диапазоне измерений укажте соотвествующие единицы, В. Таким же образом зададите единицы и для всех остальных переменных, включая результаты вычислений.

Переменные напряжение и ток в дереве объекто автоматически выставляются на значения переменных OPC сервера. Откроем Мнемосхему и убедимся, что индикаторы изменили свою шкалу согласно заданным значениям.

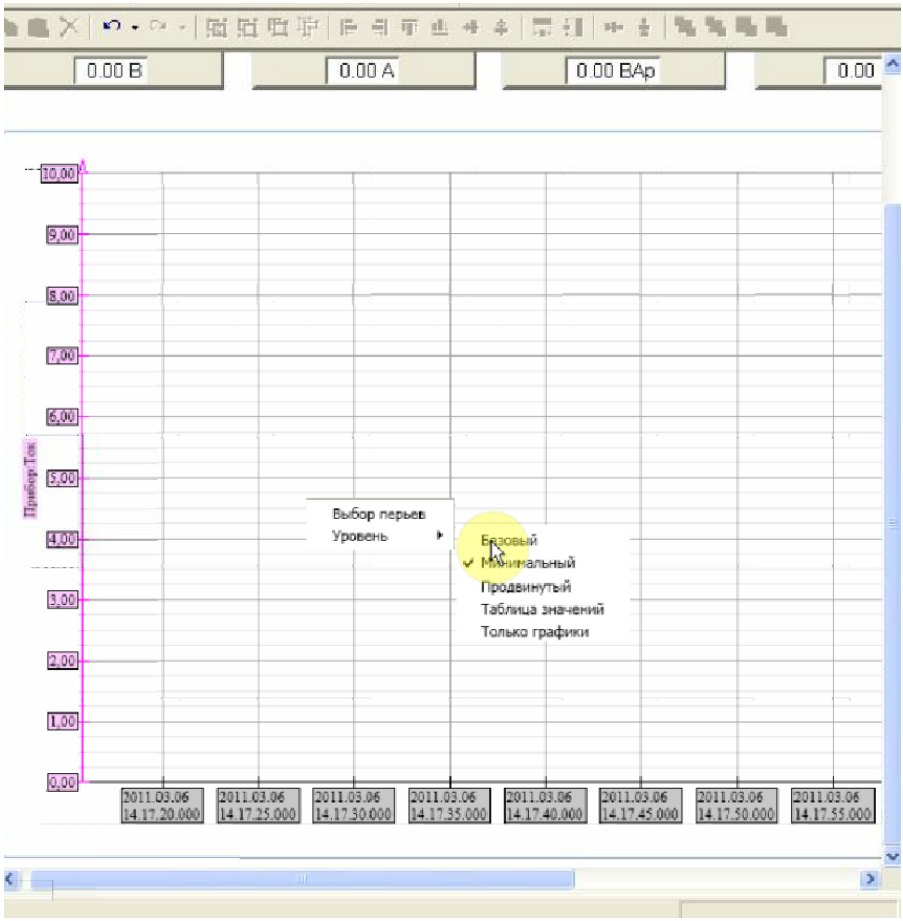


Рис.8.10 Элемент тренд системы MasterScada

3.1.6 Создание тренда.

При необходимости отображения параметров на графике, следуют воспользоватся элементов Тренд. Добавим его на мнемосхему, из вкладки Палитра.

Скорректируем размеры ренда, что бы он удобно отобрадался на мнемосхеме, для этого переместите его и растяните мышкой. Добавим резульирующие параметры в тренд, для этого переместим их из дерева объектов на поле графика (возможно групповое перемещение, выделив сразу

несколько переменных).

Подкорректируем уровень представления ппараметров, используя контектное меню: Уровень – Базовый. После этого появяться дополнительные кнопки и панели. Откроем панель Легенда и откоретируем названия переменных в ней на истинные (рис.8.10).

После задания имен переменных, сохраните и закройте тренд. Для правильного отображения параметров на графике они должны архивироваться (рис.8.11).

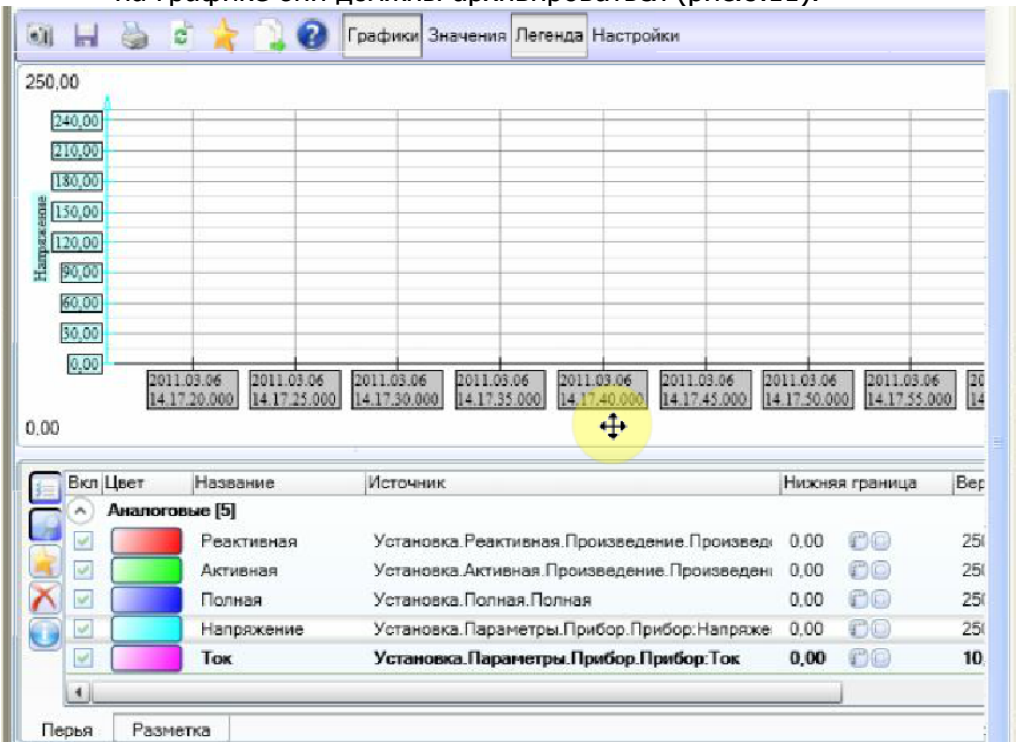


Рис.8.11 Корректировка отображения тренда системы MasterScada

Настроим режим архивации, щелкнув по параметрам приборов в дереве объектов и вкладку Архив, выставленные по умолчанию настройки для данного проекта нас удовлетворяют.

3.1.7. Выполнение программы.

Поскольку программа может работать в сети с несколькими компьютерами, для расчета и выполнения программы, требуется назначить исполнительный компьютер. Для этого выберите объект, и во вкладке общие, назначьте Компьютер (в нашем проекте один компьютер участвует в обработке сигналов).

Сделаем, так же что бы при запуске у нас автоматически открывалась мнемосхема: закладка Окна, отметим Стартовая мнемосхема на компьютере, применить.

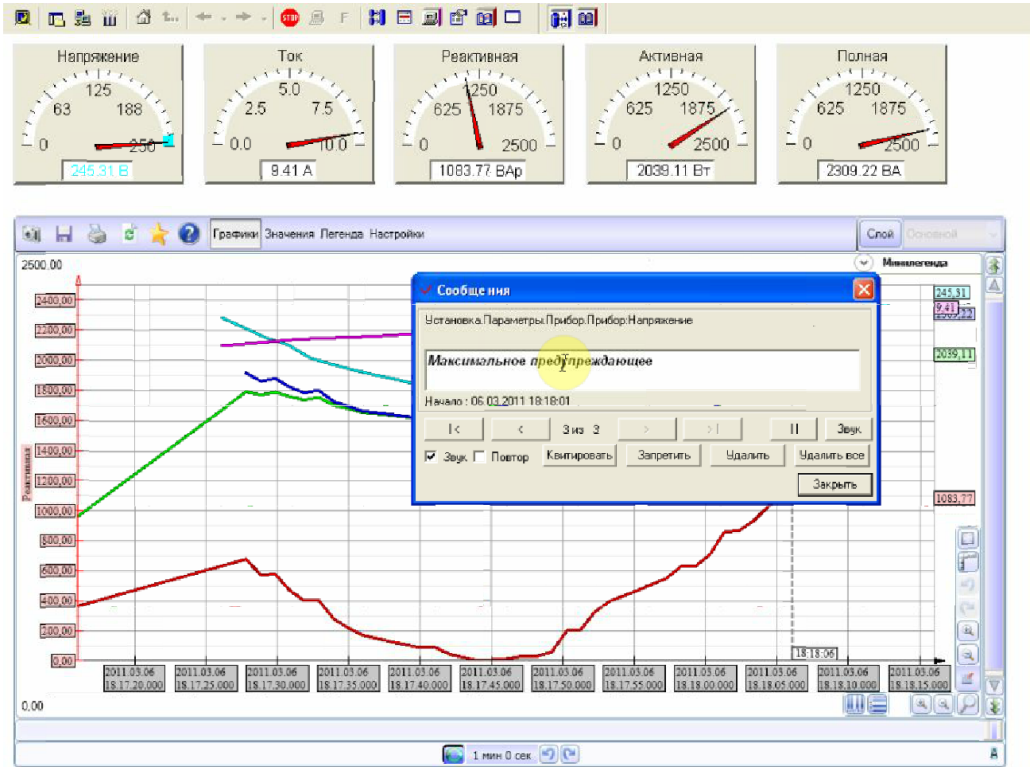


Рис.8.12 Результаты работы мнемосхемы системы MasterScada

Создание проета завершенно, запустим режим

исполнения проекта, для этого нажмем на кнопку «Ракета». После загрузки появиться окно пользователя с паролем (если его задавали вначале) и все обхекты мнемосхемы.

Если какой либо из параметров попадет в аварийную зону, то в окне сообщений появиться предупреждение. Есди информация еще актуальна, то сообщение выделяется жирным курсивом (рис.8.12).Проект завершен!

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое мнемосхема, для каких целей используется в Scada системе;
2. Какие возможности предоставляет введение мнемосхемы?;
3. В чем заключается функции тренд системы MasterScada?
4. Какие возможности для автоматизации проекта в процессе его работы реализует мнемосхема?
5. Какие возможности для обеспечения безопасности функционирования проекта реализует мнемосхема?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Инновационные технологии автоматизированного проектирования Кожухова А.В., Плеханов С.А. Учебное пособие / Ростов-на-Дону, 2012.

2. http://wiki.itorum.ru/2012/09/obzor-technical-guide-builder/

3.http://www.owen.ru/catalog/universalnij_izmeritel_regulyator_temperaturi_davleniya_owen_trm_138/opisanie

4. http://www.masterscada.ru/products/?category=1279

5. http://www.masterscada.ru/products/?category=1280

ПРИЛОЖЕНИЕ 1

ОПИСАНИЕ И УПРАВЛЕНИЕ УНИВЕРСАЛЬНОГО ИЗМЕРИТЕЛЯ-РЕГУЛЯТОРА

ОВЕН ТРМ138

На лицевой панели прибора расположены цифровые и единичные светодиодные индикаторы, служащие для отображения текущей информации о параметрах и режимах работы прибора; а также шесть кнопок, предназначенных для управления прибором.

1 - Четырехразрядный цифровой индикатор ЦИ-1 отображает измеренное или вычисленное значение параметра в выбранном канале контроля; при аварии индикатор отображает порядковый номер неисправного датчика. Возможны два режима индикации:

статический режим – выбор канала индикации производится оператором при помощи кнопок управления, расположенных на лицевой панели прибора, и контролируется по засветке соответствующего светодиода «КАНАЛ»;

циклический режим – информация о каждом канале контроля выводится по замкнутому циклу на заданное пользователем время.

2 - Четырехразрядный цифровой индикатор ЦИ-2 отображает уставку выводимого на индикацию канала контроля; при аварии индикатор отображает причину неисправности датчика в символьном виде.

3 - Двухразрядный цифровой индикатор ЦИ-3 отображает информацию о подключенном к данному каналу

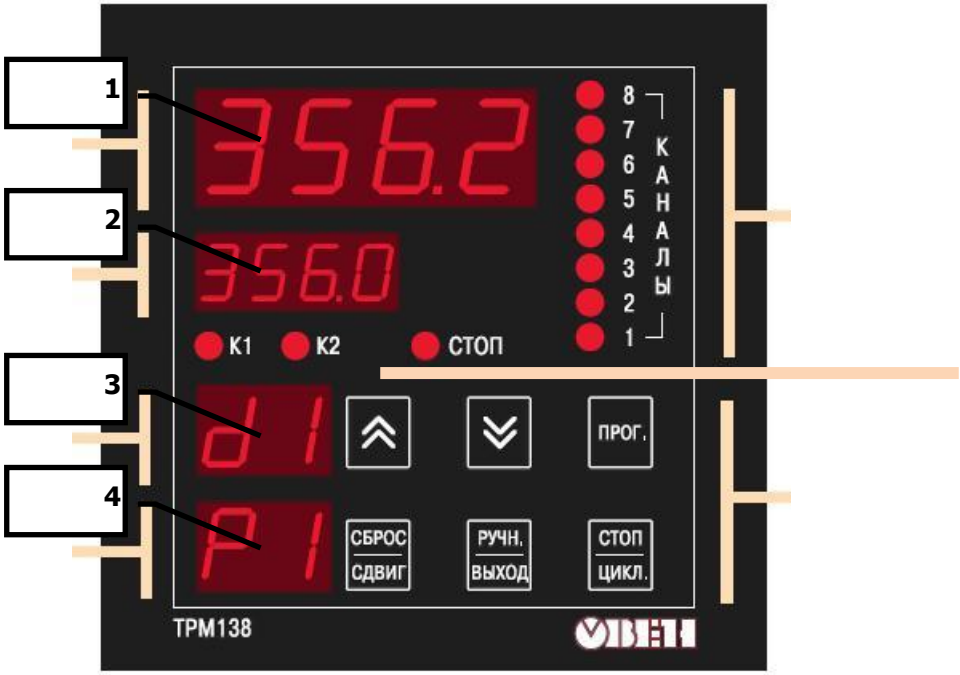
входном параметре (например, датчик «d1»).

4 - Двухразрядный цифровой индикатор ЦИ-4 цифровой индикатор № 4
отображает в режиме РАБОТА номер подключенного к данному каналу выходного устройства; мигающей засветкой сигнализирует о переводе ВУ в режим РУЧНОЕ УПРАВЛЕНИЕ.

5 - Светодиоды «КАНАЛ 1…8» постоянной засветкой показывают номер ЛУ, параметры которого в данный момент выводятся на индикацию, мигающей засветкой сигнализируют о возникновении аварийной ситуации в данном канале контроля или срабатывании в нем предупредительной сигнализации.

6 - Светодиод «К1» («К2») засвечивается при включении ВУ канала контроля, выводимого на индикацию (только для ключевых ВУ). Светодиод «СТОП» светится при работе в статическом режиме индикации.

ПРИЛОЖЕНИЕ 2 (ПРОДОЛЖЕНИЕ)



7 - Кнопки  и  служат для выбора канала индикации в статическом режиме работы, а также для управления ВУ в ручном режиме.

Кнопка  предназначена для перевода прибора в режим ПРОГРАММИРОВАНИЕ.

Кнопка 

предназначена для остановки работы аварийного ВУ, а также для сдвига информации на верхнем индикаторе при его переполнении.

Кнопка 

предназначена для перевода выбранного оператором ЛУ в режим «РУЧНОЕ УПРАВЛЕНИЕ», а также для возврата прибора из режима ПРОГРАММИРОВАНИЕ в режим РАБОТА.

Кнопка 

предназначена для переключения режима индикации прибора со статического на циклический, и обратно.