Рассмотрим основные процедуры и функции, обеспечивающие работу модулей системы генерации ключей и обмена конфиденциальной информацией. Основная часть этих функций реализована в модуле Unit1.

Переменные, используемые в рассматриваемом модуле именованы согласно математической модели. Так для работы алгоритма Эль-Гамаля используются переменные целого типа (integer)

P, Q - простые числа для реализации алгоритма ELGAMAL

N - модуль, по которому выполняется шифрование сообщения

fn - функция Эйлера

Ka,Kb - открытый и личный ключи пользователя системы

Возможности, предоставляемые пользователям системы клиентским приложением, определены в модуле как элементы массива ClCmds

ClCmds:array[1..3] of string = ('-msg','-RSA','-sig');

Здесь

'-msg' - передача открытого сообщения,

'-ELGAMAL' - организация процесса шифрования и передачи сообщения,

'-sig' – реализация цифровой подписи для сообщения.

Процедура ChooseCommand(RecText :string) отвечает за обработку комманды сервера и выбор нужного действия

Центральное место в реализации алгоритмов шифрования и цифровой подписи занимает функция нахождения НОД - iNOD

```
function iNOD(a,b:real):real;
var
i:real;
xnod:real;
begin
if a<b then xnod:=a
else xnod:=b;
i:=xnod;
while i>0 do
begin
if ((iMod(a,i)=0) and (iMod(b,i) = 0))
then
begin
xnod:=i;
i:=0;
end
else
i:=i-1;
end;
iNOD:=xnod;
end;
```

Генерация ключей реализована в процедуре ELGAMAL_keys

```
procedure ELGAMAL_keys(P,Q:integer; var N,fn,Ka,Kb:integer);
begin
```

```
N:=P*Q;
fn:=(P-1)*(Q-1);
randomize;
repeat
begin
repeat
Ka:=(Random(fn-2)+2)
until (iNod(fn,ka)=1);
repeat
Kb:=(Random(fn-2)+2)
until (((Ka*Kb) mod fn=1));
end
until ((Ka<>Kb) and (iNod(Kb,N)=1));
end;
```

Ключи, сгенерированные в процедуре ELGAMAL_keys используются в **функции шифрования сообщения методом ELGAMAL - shifrD.** В зависимости от того, какое значение Ka или Kb было присвоено формальному параметру Kab, будет выполнятся либо шифрование сообщения либо его расшифрование.

```
function shifrD(Kab,N:integer;input:string):string;
var
i,codeS:integer;
code,cod:real;
res,tmp:string;
iCode,iKab,jN,iRes:TFGInt;
begin
res:='';
Base10StringToFGInt(FloatToStr(Kab),iKab);
Base10StringToFGInt(IntToStr(N),jN);
for i := 1 to length(input) do
begin
code:=ord(input[i]);
Base10StringToFGInt(FloatToStr(code),iCode);
FGIntModExp(iCode,iKab,jN,iRes);
FGIntToBase10String(iRes,tmp);
codeS:=StrToInt(tmp);
res:=res+ chr(codes);
end;
shifrD:=res;
end;
```

Для организации цифровой подписи было принято решение использовать хеш-функцию, которая обрабатывает блоки данных в 64 бит и, используя операцию сложения по модулю 2, на выходе выдает дайджест длиной в 8 бит. Поэтому весь передаваемый текст необходимо перевести в двоичные коды. Для этого использовалась **функция function ch8to64b(InputText:string):string, которая переводит 8 символов таблицы ASCII кодов в 64-битный блок**:

```
function ch8to64b(InputText:string):string;
```

```
var
i,j:integer;
BinText,tmp:string;
smb,tsmb:array[1..8] of string;
begin
for i:=1 to 8 do
begin
tsmb[i]:=IntToBin(ord(InputText[i]));
for j:=1 to 8 do
begin
tmp:=tsmb[i];
smb[i]:=smb[i]+tmp[j+24];
end;
end;
for i:= 1 to 8 do
BinText:=BinText+smb[i];
ch8to64b:=BinText; end;
```

Функция function xHash(inp: string):string хеширует n-ое количество символов, используя в своей работе функцию хеширования двух символов.

```
function xHash(inp: string):string;
var
tmp,res,inp2:string;
i:integer;
begin
inp2:=inp;
while (length(inp) mod 8 <>0) do
begin
inp:=inp+'a';
end;
tmp:='';
res:='';
for i:=1 to length(inp) do
begin
tmp:=tmp+inp[i];
if (i mod 8 = 0) then
begin
res:= res+iHash(tmp);
tmp:='';
end;
end;
xHash:=res;
end;
```

Выполнив хеширование сообщения, необходимо организовать подписание полученного дайджеста личным ключом пользователя системы, другими словами необходимо зашифровать дайджест. Шифрование дайджеста происходит и

использованием функции shifr_hash. В зависимости от того, какое значение Ka или Kb было присвоено формальному параметру Keyb, будет выполняться либо шифрование дайджеста сообщения, либо его расшифровывание.

```
function shifr_hash(res1:string;Keyb,n:real):string;
var
i,fx,fcode:integer;
res2,tmp:string;
iKab,jN,ifx,iRes:TFGInt;
begin
res2:='';
Base10StringToFGInt(FloatToStr(Keyb),iKab);
Base10StringToFGInt(FloatToStr(N),jN);
for i:=1 to length(res1) do
begin
fx:=ord(res1[i]);
Base10StringToFGInt(FloatToStr(fx),ifx);
FGIntModExp(ifx,iKab,jN,iRes);
FGIntToBase10String(iRes,tmp);
fcode:=StrToInt(tmp);
res2:=res2+chr(fcode);
end;
shifr_hash:=res2;
end.
```

Для передачи подписанного сообщения необходимо присоединить дайджест. Было принято решение о том, что дайджест должен передаваться в виде текста для удобства контроля целостности, поэтому переход от бинарного представления дайджеста к символьному (в ASCII-символах), был выполнен через функцию перевода в десятичную систему счисления **BinToInt**.

```
Function BinToInt(binText:string):longint;
var
bin,mult:longint;
i:integer;
begin
mult:=1;
bin:=0;
for i:=length(binText) downto 1 do
begin
if binText[i]='1' then bin:=bin+mult;
mult:=mult shl 1;
end;
BinToInt:=bin;
End.
```