УДК 681.511

Стасюк А.И., Корченко А.Г., Литвинова Т.В.

ПОВЫШЕНИЕ СКОРОСТИ ВЫПОЛНЕНИЯ СИММЕТРИЧНЫХ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ ПРИ ИХ ПРОГРАММНОЙ РЕАЛИЗАЦИИ

Актуальность проблемы информационной безопасности постоянно возрастает с процессами проникновения технических средств обработки и передачи данных, прежде всего вычислительных систем, практически во все сферы деятельности общества. Также об актуальности проблемы свидетельствует обширный перечень возможных способов компьютерных преступлений. Объектами посягательств могут быть сами технические средства как материальные объекты, программное обеспечение, базы данных, электронная документация. Наиболее распространенной угрозой безопасности указанных систем является несанкционированный доступ (НСД). В общем случае, действия, порождаемые НСД к компонентам вычислительных сетей, можно классифицировать по следующим видам [1]:

- интерраптация (*Interrupt* - прерывание);
- интерсептация (*Intercept* - перехват);
- модификация (*Modification* - изменение);
- фальсификация (*Falsification* - подделка).

Одним из эффективных методов защиты информации от НСД является криптографическое преобразование данных. Этот метод защиты информации является фрагментарным, поскольку не распространяется на всю систему в целом, и относится к «добавленной» защите.

Основные направления использования криптографических методов:

- установление подлинности пересылаемых сообщений;
- хранение конфиденциальной информации (документов, баз данных) на носителях в зашифрованном виде;
- передача по телекоммуникационным каналам общего пользования (аналоговым или цифровым), передача через Internet (например, электронная почта, FTP) и др.

В мировой практике к алгоритмам шифрования предъявляются следующие требования [2]:

1. Высокий уровень защиты данных против дешифрования и возможной модификации.
2. Защищенность информации должна основываться только на знании ключа и не зависеть от того, известен ли алгоритм шифрования или нет.
3. Незначительное изменение исходного текста или ключа должно приводить к существенному изменению шифрованного текста.
4. Длина шифрованного текста должна быть равна длине исходного текста.
5. Мощность множества ключей должна исключить расшифрование данных путем тотального перебора.
6. Стоимость расшифрования данных без ключа должна превышать стоимость самих данных.
7. Экономичность реализации при достаточной скорости выполнения алгоритма.

В разработанные ранее криптографические алгоритмы, которые нашли наибольшее распространение и применение в качестве государственных стандартов (DES – стандарт шифрования данных США, ГОСТ 28147-89(ГОСТ) [3] – отечественный стандарт, IDEA – Швейцарский международный алгоритм и т.д.), изначально закладывались первые пять пунктов вышеуказанных требований. Оценка стоимости данных (шестое требование) обычно является прерогативой их владельца (юридического или физического лица). Обеспечение седьмого пункта требований является одной из актуальных проблем, поскольку криптографически стойкие алгоритмы в настоящее время недостаточно обеспечивают необходимое быстродействие шифрования данных, подготовленных для передачи по скоростным линиям связи.

Алгоритмы шифрования составляют две основные группы:

- симметричные криптосистемы (криптосистемы с секретным ключом);
- асимметричные криптосистемы (криптосистемы с открытым ключом).

В свою очередь, симметричные методы шифрования делятся также на:

- потоковые (шифрование потока данных) (с одноразовым или бесконечным ключом, с конечным ключом (система Вермана), на основе генератора псевдослучайных чисел (ПСЧ));
- блочные (шифрование данных поблочно) (шифры перестановки (Р - блоки), шифры замены (S - блоки), составные).

Для обеспечения эквивалентной криптостойкости в симметричных криптосистемах по отношению к асимметричным используются ключи меньшей длины, что существенно повышает быстродействие (приблизительно на 3-4 порядка). Брюс Шнейер в работе "Прикладная криптография: протоколы, алгоритмы и исходный текст на С" приводит следующие данные об эквивалентных длинах ключей, которые представлены в табл.1.

Таблица 1

Эквивалентные длины ключей в симметричных и асимметричных криптосистемах

Длина ключа в симметричных криптосистемах	Длина ключа в асимметричных криптосистемах
56 бит	384 бита
64 бита	512 бит
80 бит	768 бит
112 бит	1792 бита
128 бит	2304 бита

Если отсутствует проблема рассылки ключей, целесообразно применять симметричные алгоритмы шифрования.

Наиболее известными симметричными алгоритмами шифрования данных являются следующие:

- Lucipher – алгоритм фирмы IBM, США;
- DES (Data Encryption Standart) – американский стандарт шифрования данных;
- FEAL-1 – японский стандарт шифрования данных;
- B-Crypt – алгоритм компании Telecom, Великобритания;
- IDEA – швейцарский международный алгоритм;
- ГОСТ 28147-89 – советский стандарт шифрования данных.

В данной статье речь пойдет о блочных алгоритмах симметричной криптографии. На диаграмме (см. рис.1) представлена сравнительная характеристика симметричных алгоритмов шифрования. Здесь в качестве критерия сравнения выбран размер ключа в битах, от длины которого зависит криптостойкость соответствующего алгоритма.

Видно, что относительно выбранного критерия наибольшим преимуществом обладает алгоритм ГОСТ, который может быть реализован как аппаратно, так и программно. Если рассматривать использование ГОСТ с юридической точки зрения, то это не противоречит действующему законодательству Украины, претензий со стороны авторов ГОСТ быть не может, так как юридические права на алгоритм ни за кем не закреплены.

Наиболее известные аппаратные средства, выполняющие обработку по данному алгоритму – это криптографические вычислители серии “Криптон”. Характеристика данных плат приведена в табл. 2.

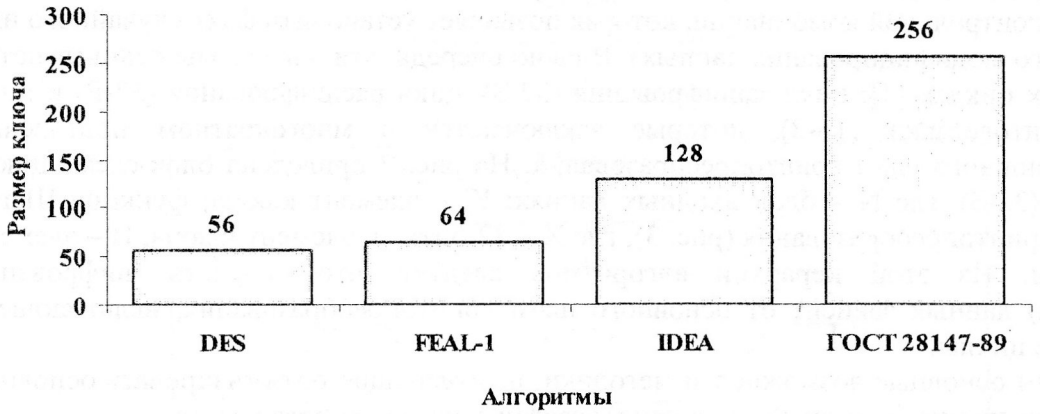

Рис. 1. Сравнительные характеристики симметричных алгоритмов

Таблица 2
Характеристика плат Криптон

N	Плата Криптон	Скорость шифрования, Кб/с
1	Криптон – 3, для шины ISA	300
2	Криптон – 4, для шины ISA	400
3	Криптон – 4, для шины PCI	1100
4	Криптон – 8, для шины PCI	8500
5	Криптон – 9, для шины PCI	10000

Недостатком аппаратной реализации является наличие соответствующего дополнительного оборудования, от которого фактически и зависит значение скорости шифрования данных для определенной реализации. С появлением новых быстродействующих СБИС для повышения быстродействия криптографических преобразований необходимо будет заново осуществлять синтез и изготавливать криптовычислитель. К достоинствам аппаратной реализации можно отнести тот факт, что битовые перестановки, которые крайне неэффективно реализуются на современных процессорах, достаточно просто реализовать путем разводки проводников в кристалле или на плате. Создание специальных методик, позволяющих осуществлять оптимальную программную реализацию, позволит увеличивать скорость шифрования данных в соответствии с разработкой и применением более быстродействующих процессоров.

Для построения программных средств, позволяющих с высокой скоростью производить шифрование информации на основе DES-подобных криптографических систем, разработана специальная методика. Ее основные положения отображены в следующем примере, реализующем алгоритм ГОСТ. Табл. 3 содержит результаты программной реализации алгоритма с применением описанной ниже методики на современных компьютерах.

Таблица 3
Результаты программной реализации алгоритма

N	Процессор	ОЗУ, Кб	Скорость шифрования, Кб/с
1	Pentium3, 448 MHz	256	1800
2	Intel Celeron 1.70 GHz	256	2820
3	Intel Celeron 2.4 GHz	256	4310
4	AMD Athlon Processor 3200+ 1.0GHz	1 Гб	11830

ГОСТ предусматривает три режима шифрования данных и режим выработки имитовставки (контрольной комбинации, которая позволяет установить факт случайного или преднамеренного модифицирования данных). В свою очередь, эти алгоритмы основываются на трех базовых циклах [3]: цикл зашифрования (32-З), цикл расшифрования (32-Р) и цикл выработки имитовставки (16-З), которые заключаются в многократном повторении выполнения основного шага криптопреобразования. На рис. 2 приведена блок-схема цикла зашифрования (32-З), где N – блок входных даннях; Kj – элемент ключа; функция Шаг – основной шаг криптопреобразования (рис. 3), где X – 32-битный элемент ключа; H – элемент таблицы замен. Из этой иерархии алгоритмов следует, что скорость шифрования (дешифрования) данных зависит от основного шага криптопреобразования, включающего вышеуказанные циклы.

Рассмотрим основные возможности методики, позволяющие оптимизировать основной шаг с целью повышения быстродействия криптографического преобразования.

1. Преобразование по таблице замен.

Таблица замен состоит из 8 строк и 16 столбцов, которые содержат 4-битные элементы замен. Обрабатываемый 32-битный блок данных разбивается на 8 блоков по 4 бита. В качестве замены для значения блока выбирается элемент из таблицы замен с номером строки, равным номеру заменяемого блока, и номером столбца, равным значению заменяемого блока как 4-битового целого неотрицательного числа. Учитывая архитектуру современных процессоров, имеющих разрядность 32 и выше, производить замену блоками по 4 бита довольно не эффективно. Для этого необходимо 32-разрядное слово разбивать на восемь 4-битных блоков, затем 8 раз производить замену, а затем опять составлять 32-разрядное слово.

В основном шаге также выполняется поблочная замена 32-битового значения, которое интерпретируется как массив из восьми 4-битовых блоков кода. Ни в языках высокого уровня (C++, Паскаль), ни в ассемблере нет команды выделения полубайтов из байтов, слов, двойных слов, а также нет команды замены полубайтов. В этом случае производить замену блоками по 4 бита не эффективно. Если осуществлять замену побайтно, то:

- за одну команду выполняются сразу две замены;
- исчезает необходимость отделять полубайты из двойных слов для выполнения замены и затем вновь формировать двойное слово.

Этим достигается значительное увеличение скорости выполнения, но появляется необходимость преобразования таблицы замен. Каждая из четырех пар 4-разрядных узлов замен заменяется одним 8-разрядным узлом, который представляет собой прямое произведение узлов, входящих в указанную пару. Пара 4-разрядных узлов требует для своего представления 16 байтов, один 8-разрядный – 256 байтов. Таким образом, размер таблицы замен, которая должна храниться в памяти компьютера, увеличивается до $4 \cdot 256 = 1024$ байтов, или до одного килобайта.

Еще больше повысить быстродействие можно, производя замену по 16 бит. По аналогии преобразования первоначальной таблицы замен производится обработка ранее преобразованной таблицы в новую, которая будет представлять матрицу 2×65536 с 16-разрядными элементами. Размер таблицы замен, которая должна будет храниться в памяти компьютера, возрастает до $2 \cdot 65536 \cdot 2 = 262144б = 256Кб$. Такое увеличение хранимой в памяти информации оправдывается в случае необходимости высокоскоростного шифрования большого количества данных.

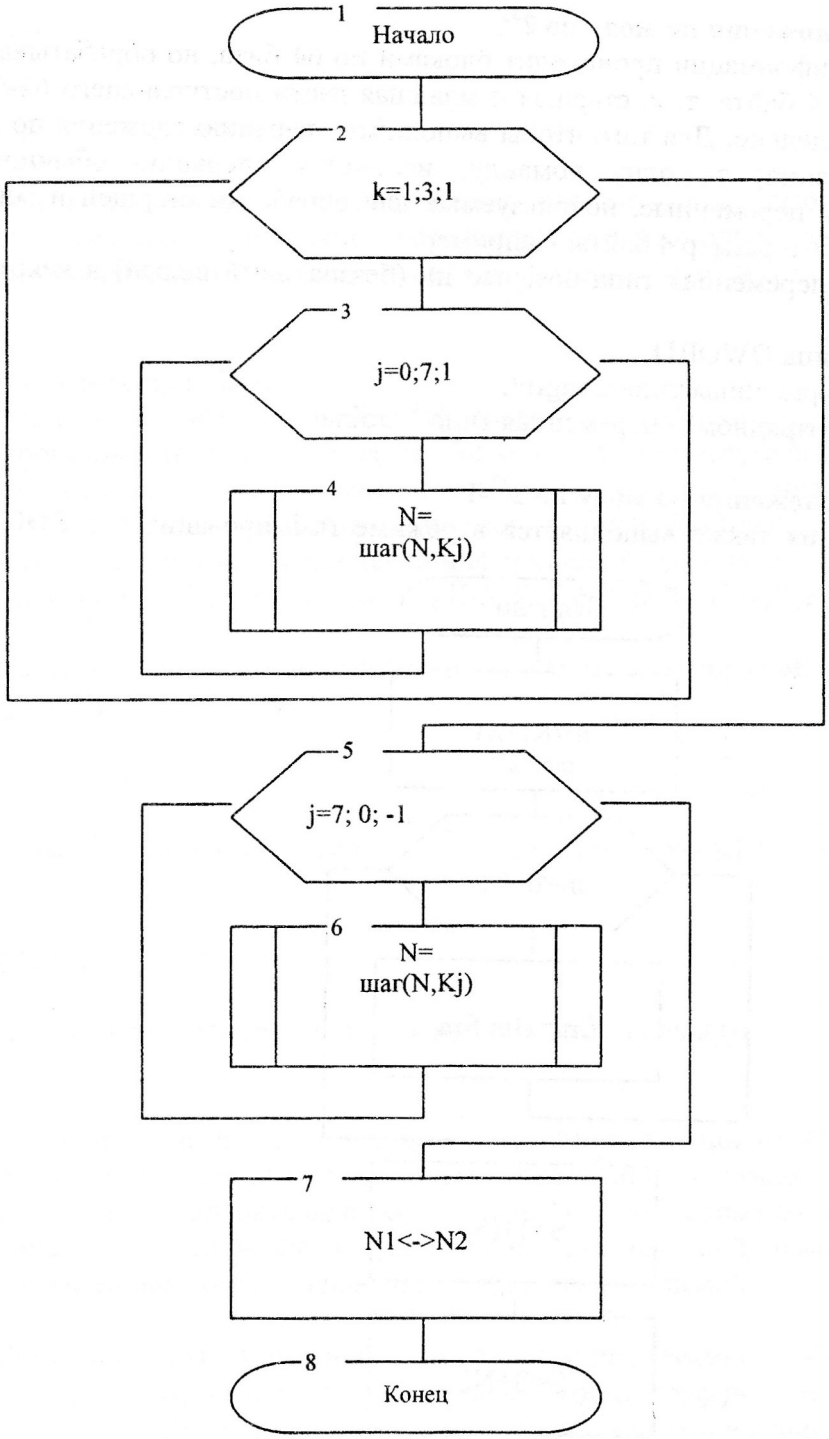

Рис. 2. Схема цикла 32-3

2. Циклический сдвиг влево

Циклический сдвиг влево в основном шаге предусматривает вращение блока данных на 11 разрядов в сторону старших, т.е. старшие биты становятся на место младших, и наоборот. В языках высокого уровня (C, C++, Паскаль и т.д.) нет функции, реализующей циклический сдвиг. Это можно сделать следующим образом: 11 раз сдвигать блок данных на 1 разряд, запоминать его и ставить на место младшего. Для повышения эффективности выполнения сдвига следует использовать ассемблерную вставку, содержащую команду ROL, второй операнд которой задает число сдвигаемых разрядов.

3. Операция сложения по модулю 2^{32}

Шифрование информации происходит блоками по 64 бита, но обрабатываемая единица данных составляет 4 байта, т. к. старшая и младшая части поступающего 64-битного слова обрабатываются отдельно. Для того чтобы выполнить операцию сложения по модулю 2^{32} (в режиме гаммирования) за одну команду, используя операцию обычного сложения, необходимо, чтобы переменные, используемые для обработки старшей и младшей частей блока, имели в памяти размер 4 байта. Например:

- в C и C++ - переменная типа unsigned int (беззнаковый целый) в некоторых версиях компиляторов;
- переменная типа DWORD;
- в Паскале - переменная типа LongInt;
- в Delphi (32-разрядном) - переменная типа Cardinal.

4. Операция сложения по модулю $2^{32}-1$

Данная операция также выполняется в режиме гаммирования и в ГОСТе определена

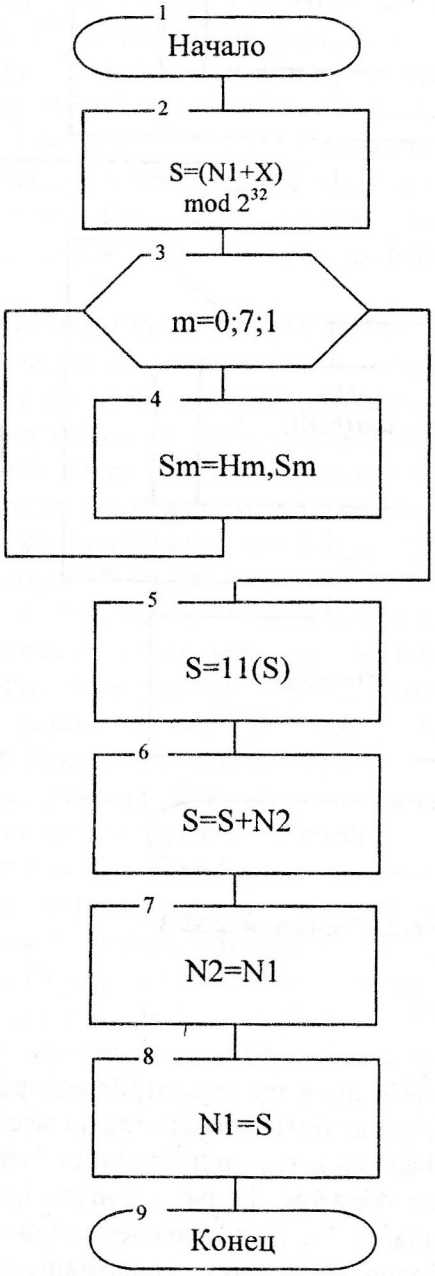

Рис. 3. Схема основного шага криптопреобразования

следующим образом:

$a+b = a+b$, если $a+b < 2^{32}$,
$a+b = a+b - 2^{32} + 1$, если $a+b \geq 2^{32}$.

Эффективнее всего данную операцию можно реализовать с помощью ассемблерной команды ADC, которая выполняет целочисленное сложение двух операндов с использованием содержимого флага переноса (CF), который прибавляется к младшему биту результата.

5. Вложенные циклы в базовых

В базовых циклах зашифрования, расшифрования и выработки имитовставки основной шаг криптопреобразования используется 32; 32 и 16 раз соответственно для каждого 64-битного блока данных с различными значениями ключевой информации, которая разбивается на 8 элементов по 4 байта. Например, в цикле 32-З выполняемого алгоритма последовательность использования ключевой информации следующая:

$K_1K_2K_3K_4K_5K_6K_7K_8\ K_1K_2K_3K_4K_5K_6K_7K_8\ K_1K_2K_3K_4K_5K_6K_7K_8\ K_8K_7K_6K_5K_4K_3K_2K_1$.

Реализовать такое использование ключа можно следующим образом:

```
for(j = 1; j≤3; j++)
{
  for(i = 1; i≤8; i++)
  {
   //….вызвать функцию основного шага с параметром ключа K[i]
   }
}
for(i = 8; i≥1; i--)
{
 //….вызвать функцию основного шага с параметром ключа K[i].
}
```

Как видно, для этого необходимо использовать вложенные циклы. С целью повышения быстродействия следует заменить вложенные циклы многократным вызовом функции основного шага или предварительно сформировать необходимую последовательность ключевых данных. Это приведёт к увеличению программной памяти, но учитывая возможности современных компьютеров, это не является проблемой.

Описанный подход можно применить при создании программного обеспечения, которое реализует и другие известные симметричные криптографические алгоритмы, т.к. практически все алгоримы используют вложенные циклы, операции сдвига и перестановки.

Список литературы

1. *Корченко А.Г.* Несанкционированный доступ к компьютерным системам и методы защиты. Учебное пособие. Киев: КМУГА, 1998. -116 с.

2. *Новосельский А.* Алгоритмы шифрования//Компьютеры+Программы.- 1996. - №5. -С. 70-71.

3. ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. Введ. 01.07.90.

4. *Игнатенко Ю.И.* Как сделать так, чтобы…? // Мир ПК. - 1994. - №8. - С. 52-54.

5. *Спесивцев А. В., Вегнер В. А.* Защита информации в персональных ЭВМ. -М.: Радио и связь, 1992.- 192 с.

6. *Винокуров А.* «ГОСТ не прост, а очень прост». -М.: «Монитор », 1995, -№1, с. 60-73.

7. *Абель Питер.* Язык ассемблера для IBM PC и программирования. -М.: Высшая школа, 1992. - 192 с.

8. *Мешков А.В., Тихомиров Ю.В.* Visual C++ и MFC. -СПб.: БХВ-Петербург, 2003. –1040 с.

9. *RSA* Laboratories' Frequently Asked Questions About Today's Cryptography, v4.0

10. http://www.ancud.ru/catalog/crypton.htm (Устройства криптографической защиты данных (УКЗД) серии КРИПТОН.)

Поступила 19.04.2006

УДК 004.31 **Журавель Т.Н.**

ИСПОЛЬЗОВАНИЕ АППАРАТНО-ПРОГРАММНЫХ КОМПЛЕКСОВ ВОССТАНОВЛЕНИЯ ПРОТОКОЛОВ СВЯЗИ ЦИФРОВЫХ ТЕЛЕФОННЫХ АППАРАТОВ ДЛЯ ЗАЩИТЫ ТЕЛЕФОННЫХ ЛИНИЙ

Защита информации в телефонных линиях связи является неотъемлемой частью комплексной защиты объекта информационной деятельности.

Актуальность проблемы съема информации с использованием телефонных линий (ТЛ) ни у кого в наше время не вызывает сомнений. Как правило, ни один объект информационной деятельности (ОИД), в том числе тот, на котором циркулирует информация с ограниченным доступом (ИсОД), не обходится без ТЛ. В настоящее время широкое распространение получили цифровые автоматические телефонные станции (ЦАТС) импортного производства, которые полностью удовлетворяют потребность в комфортности и качестве связи как крупных учреждений и ведомств, так и более мелких организаций.

Проведя сравнительный анализ существующих методов и средств съема информации с ТЛ, можно считать цифровую телефонную связь более надежной, но ввиду некоторых особенностей и реализуемых функций существуют и недостатки.

Основным недостатком можно считать возможность дистанционного контроля акустической обстановки в помещении с использованием цифрового телефонного аппарата (ЦТА). Данная функция ЦТА может быть как заявленной производителем, так и принудительной (в случае несанкционированного прослушивания). Штатные режимы ЦАТС, реализующие удаленное прослушивание, сопровождаются звуковыми либо визуальными (на дисплее ЦТА) сигналами, каких-либо реальных гарантий относительно нештатных (или необъявленных) режимов ЦАТС импортного производства не существует.

Таким образом, четко прослеживается необходимость применения средств технической защиты информации. В данном случае к таким средствам выдвигаются следующие требования:

- блокирование возможности дистанционного контроля акустической обстановки в помещении, то есть гарантированная передача речевой информации от ЦТА к ЦАТС только в штатном режиме разговора;
- исключение возможности утечки преобразованной речевой информации через вспомогательные цифровые каналы ЦТА во всех режимах работы ЦТА;
- сохранение комфортности и качества выполнения основных заявленных производителем функций.

Проанализировав возможные пути реализации таких тренований, можно прийти к выводу, что существует необходимость вмешательства со стороны средств технической защиты в цифровые сигналы обмена между ЦАТС и ЦТА. Необходимо в то же время