Терехов А. Н., Богданов В. Л.

Предложение по реализации основных криптографических алгоритмов в заказных кристаллах

Введение

Проблема использования криптографических методов в информационных системах стала в настоящий момент особенно актуальна. С одной стороны, расширилось использование компьютерных сетей, в частности, глобальной сети Интернет, по которым передаются большие объёмы информации государственного, военного, коммерческого и частного характера, не допускающего возможность доступа к ней посторонних лиц. С другой стороны, появление новых мощных компьютеров, технологий сетевых и нейронных вычислений сделало возможным дискредитацию криптографических систем, ещё недавно считавшихся практически не раскрываемыми.

Сейчас большинство компаний имеют локальные сети, подключенные к Internet. Это позволяет воспользоваться огромным количеством ресурсов Internet, но вместе с тем может привести к проникновению посторонними в хранилища важной для компании информации.

Таким образом, возникают следующие задачи защиты информации:

- шифрование информации, передаваемой по открытому каналу с тем, чтобы она могла быть расшифрована только получателем;
- контроль трафика между локальной и глобальной сетями с использованием брандмауэров (firewall) и proxy-серверов.

Для решения первой задачи в ГП “ТЕРКОМ” в 1995 г. была создана своя криптографическая система под названием “ТерКрипт”, реализующая следующие отечественные криптографические стандарты:

- **ГОСТ 28147-89** "Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования" в области симметричного шифрования;
- **ГОСТ Р 34.10-94** "Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма";
- **ГОСТ Р 34.11-94** "Функция хэширования".

Последние два стандарта объединены под общим заголовком "Информационная технология. Криптографическая защита информации". Поскольку данные алгоритмы обладают большой вычислительной емкостью, их программные реализации работают достаточно медленно. Для ускорения работы конструируют специализированные жесткие кристаллы, реализующие функции, наиболее часто используемые в данных алгоритмах.

Для нахождения таких критических по времени участков кода было проведено профилирование системы “ТерКрипт”. Данная система не имеет сертификата, но она и не является коммерческим продуктом, а носит исследовательский характер и, несомненно, может быть применена для такого профилирования.

Исследование проводилось в следующих условиях:

- операционная система: Microsoft Windows NT 4.0;
- система программирования: Microsoft Visual C++ 5.0.

Профилирование алгоритма ГОСТ 28147-89

Основные определения

Данный алгоритм достаточно прост с точки зрения поиска критичных по времени участков кода, поскольку представляет собой лишь несколько вложенных циклов, поэтому для их нахождения был проанализировано только описание алгоритма.

В оригинале ГОСТа 28147–89 содержится описание алгоритмов нескольких уровней. На самом верхнем находятся практические алгоритмы, предназначенные для шифрования массивов данных и выработки для них имитовставки. Стандарт предусматривает три следующих режима шифрования:

- простая замена,
- гаммирование,
- гаммирование с обратной связью,

и один дополнительный режим выработки имитовставки.

Все они опираются на три алгоритма низшего уровня, называемые в тексте ГОСТа ***циклами***. Эти фундаментальные алгоритмы упоминаются в данной статье как ***базовые циклы***, чтобы отличать их от всех прочих циклов. Они имеют следующие названия и обозначения, последние приведены в скобках:

- **цикл зашифрования (32-З);**
- **цикл расшифрования (32-Р);**
- **цикл выработки имитовставки (16-З).**

В свою очередь, каждый из базовых циклов представляет собой многократное повторение одной единственной процедуры, называемой для определенности далее в настоящей работе ***основным шагом криптопреобразования***.

В ГОСТе 28147–89 ключевая информация состоит из двух структур данных. Помимо собственно ***ключа***, необходимого для всех шифров, она содержит еще и ***таблицу замен***. Ниже приведены основные характеристики ключевых структур ГОСТа.

1. ***Ключ*** является массивом из восьми 32-битных элементов кода. В ГОСТе элементы ключа используются как 32-разрядные целые числа без знака. Таким образом, размер ключа составляет 32·8=256 бит или 32 байта.

2. ***Таблица замен*** является матрицей 8×16, содержащей 4-битовые элементы, которые можно представить в виде целых чисел от 0 до 15. Строки ***таблицы замен*** называются ***узлами замен***, они должны содержать различные значения, то есть каждый ***узел замен*** должен содержать 16 различных чисел от 0 до 15 в произвольном порядке. Таблица замен зафиксирована стандартом ГОСТ 28147–89 и не является секретной.

Основной шаг криптопреобразования

Основной шаг криптопреобразования по своей сути является оператором, определяющим преобразование 64-битового блока данных. Дополнительным параметром этого оператора является 32-битовый блок, в качестве которого используется какой-либо элемент ключа. Также в нем используется таблица замен. Ниже представлено описание алгоритма *основного шага криптопреобразования* на языке C++:

```
// основной шаг криптопреобразования
void gost_symm::basic_step(unsigned& r1, unsigned& r2, const unsigned subkey)
{
    // сохранение рабочего подблока
    unsigned save_r1 = r1;

    // суммирование рабочего подблока с подключом
    r1 += subkey;

    // преобразование рабочего подблока в узле замены
    unsigned temp = r1; r1 = 0;
    for (int i=0, sh=0; i<8; i++, sh+=4)
        r1 |= (unsigned) sbox [i][(temp>>sh) & 0xF] << sh;

    // сдвиг рабочего подблока
    r1 = (r1<<11) | (r1>>(32-11));

    // логическое суммирование подблоков
    r1 ^= r2;
    r2 = save_r1;
}
```

Входные параметры:

- sbox – *таблица замен;*
- r1 – младшая половина 64-битового блока данных;
- r2 – старшая половина 64-битового блока данных;
- subkey – 32-битовый элемент ключа.

Выходные параметры:

- r1 – младшая половина преобразованного блока данных;
- r2 – старшая половина преобразованного блока данных.

Базовые циклы

Базовые циклы предназначены для криптографического преобразования одного блока информации, состоящего из 64 бит, и заключаются в многократном выполнении *основного шага* с использованием разных элементов ключа и отличаются друг от друга только числом повторения шага и порядком использования ключевых элементов. Для циклов зашифрования и расшифрования *основной шаг* выполняется 32 раза, а для цикла выработки имитовставки – 16 раз.

Базовые циклы зашифрования и расшифрования выглядят одинаково:

```
// базовые циклы зашифрования и расшифрования
void gost_symm::subst_block ()
{
    // разбиение исходного блока на подблоки
    unsigned r1 = * ((unsigned*) (&work_block[0]));
    unsigned r2 = * ((unsigned*) (&work_block[4]));

    // итеративная обработка текущего блока
    for (int iter=0; iter<32; iter++)
        basic_step ( r1, r2, key [select_subkey [iter]] );

    // перестановка подблоков
    unsigned temp = r1;
    r1 = r2;
    r2 = temp;

    // объединение подблоков в результирующий блок
    * ((unsigned*) (&work_block[0])) = r1;
    * ((unsigned*) (&work_block[4])) = r2;
};
```

Входные параметры:

- work_block – 64-битовый блок данных;
- key – 256-битовый ключ;
- select_subkey – последовательность выборки элементов ключа, состоящая из 32 чисел в интервале от 0 до 7.

Выходные параметры:

- work_block – преобразованный блок данных.

Цикл выработки имитовставки отличается только меньшим числом повторений, а именно 16 вместо 32, и отсутствием перестановки подблоков.

Результаты

Для оценки потенциального выигрыша в случае использования специализированного кристалла было проведено профилирование системы “ТерКрипт”, результаты которого приведены в таблице.

функция	процент использования процессорного времени в функции	процент использования процессорного времени в функции и в вызываемых функциях
Основной шаг криптопреобразования	88%	88%
базовые циклы	8%	96%
Шифрование одного блока в режиме гаммирования	1%	97%

Предлагается в специализированный кристалл включить следующие функции и данные (в порядке убывания важности):

1. *Основной шаг криптопреобразования* и *таблицу замен*.

Входные данные:

- блок данных, подлежащих шифрованию, – 64 бит;
- элемент ключа – 32 бит.

Используемые данные: таблица замен (постоянная) – 512 бит.

Выходные данные: 64 бит.

2. *Базовые циклы криптопреобразования* – цикл зашифрования, цикл расшифрования и цикл выработки имитовставки.

Входные данные:

- блок данных, подлежащих шифрованию, – 64 бит;
- ключ – 256 бит.

Выходные данные – 64 бит.

Используемые данные: последовательность выборки элементов ключа – 2·96 бит для циклов зашифрования и расшифрования, а также 48 бит для цикла выработки имитовставки.

Включение в кристалл остальных функций (шифрования простой заменой, гаммированием, гаммированием с обратной связью и выработки имитовставки) представляется нецелесообразным.

Профилирование алгоритмов ГОСТ Р 34.10-94 и ГОСТ Р 34.11-94

Основные обозначения

Стандарт ГОСТ Р 34.10-94 "Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма" основан на стандарте ГОСТ Р 34.11-94 "Функция хэширования". То есть, как формирование, так и проверка цифровой подписи начинаются с вычисления функции хэширования для заданного набора данных. От длины данных зависит только время вычисления этой функции хэширования, поэтому для нахождения критичных по времени участков кода внимание необходимо сосредоточить именно на ней.

В описании стандарта используются следующие обозначения:

- $V_k(2)$ – множество всех бинарных слов длины k;
- β^* - множество всех конечных слов в алфавите $\beta = \{0,1\}$;
- || - операция конкатенации;
- $\oplus$ - побитовое сложение слов одинаковой длины по модулю 2;
- A^k - конкатенация k экземпляров слова A ($A \in \beta^*$);
- M – последовательность двоичных символов, подлежащих хэшированию, ($M \in \beta^*$).

Под хэш-функцией h в стандарте понимается зависящее от параметра (стартового вектора хэширования $H \in V_k(2)$) отображение

$$h : \beta^* \to V_{256}(2)$$

При ее вычислении используются, в частности, следующие функции:

- функция шифрования в режиме простой замены по ГОСТ 28147-89;
- функция ψ;
- шаговая функция хэширования.

Рассмотрим эти функции более подробно.

Функция шифрования одного блока в режиме простой замены совпадает с *базовым циклом криптопреобразования*, описанным выше.

Функция ψ

Функция ψ: $V_{256}(2) \to V_{256}(2)$ преобразует слово

$$\eta_{16}||\ldots||\eta_1,\ \eta_i \in V_{16}(2),\ i=1..16$$

в слово

$$\eta_1 \oplus \eta_2 \oplus \eta_3 \oplus \eta_4 \oplus \eta_{13} \oplus \eta_{16} || \eta_{16} || \ldots || \eta_2$$

Шаговая функция хэширования

Шаговая функция хэширования – это отображение

$$V_{256}(2) \times V_{256}(2) \to V_{256}(2)$$

Ее вычисление включает три части:

- генерацию ключей – слов длины 256 бит;
- шифрующее преобразование – зашифрование 64-битовых подслов стартового слова H на ключах K_i (i = 1, 2, 3, 4) с использованием алгоритма по ГОСТ в режиме простой замены;
- перемешивающее преобразование результата шифрования.

Рассмотрим **алгоритм генерации ключей**.

Пусть

$X = (b_{256}, b_{255}, \ldots, b_1) = x_4 \,||\, x_3 \,||\, x_2 \,||\, x_1 = \eta_{16} \,||\, \eta_{15} \,||\, \ldots \,||\, \eta_1 = \xi_{32} \,||\, \xi_{31} \,||\, \ldots\, \xi_1$,

где $x_i = (b_{i*64}, \ldots b_{(i-1)*64+1}) \in V_{64}(2)$, i = 1 … 4;

$\eta_j = (b_{j*16}, \ldots b_{(j-1)*16+1}) \in V_{16}(2)$, i = 1 … 16;

$\xi_k = (b_{k*8}, \ldots b_{(k-1)*8+1}) \in V_8(2)$, i = 1 … 32.

Обозначим $A(X) = (x_1 \oplus x_2) \,||\, x_4 \,||\, x_3 \,||\, x2$.

Используем преобразование $P : V_{256}(2) \to V_{256}(2)$

$$P(\xi_{32} \,||\, \ldots\, \xi_1) = \xi_{\varphi(32)} \,||\, \ldots\, \xi_{\varphi(1)}, \text{ где}$$

$$\varphi(i+1+4*(k-1)) = 8i+k,\ i = 0 \ldots 3,\ k = 1 \ldots 8$$

Для генерации ключей необходимо использовать следующие данные:

- слова $H, M \in V_{256}(2)$;
- параметры: слова C_i (i = 2, 3, 4), имеющие значения $C_2 = C_4 = 0^{256}$ и $C_3 = 1^8 0^8 1^{16} 0^{24} 1^{16} 0^8 (0^8 1^8)^2 1^8 0^8 (0^8 1^8)^4 (1^8 0^8)^4$.

При вычислении ключей реализуется следующий алгоритм:

1. Присвоить значения i = 1, U = H, V = M.
2. Выполнить вычисление $W = U \oplus V$, $K_1 = P(W)$.
3. Присвоить i = i + 1.
4. Проверить условие i = 5. При положительном исходе перейти к шагу 7. При отрицательном перейти к шагу 5.
5. Выполнить вычисление $U = A(U) \oplus C_i$, $V = A(A(V))$, $W = U \oplus V$, $K_i = P(W)$.
6. Перейти к шагу 3.
7. Конец работы алгоритма.

На этапе **шифрующего преобразования** осуществляется зашифрование 64-битных подслов слова H на ключах K_i (i = 1, 2, 3, 4).

Для шифрующего преобразования необходимо использовать следующие исходные данные:

- $H = h_4 \,||\, h_3 \,||\, h_2 \,||\, h_1$, $h_i \in V_{64}(2)$, i = 1 … 4;
- набор ключей K_1, K_2, K_3, K_4.

Реализуя алгоритм шифрования, получаем слова

$$s_i = E_{Ki}(h_i), \text{ где } i = 1, 2, 3, 4.$$

В результате данного этапа образуется последовательность $S = s_4 \,||\, s_3 \,||\, s_2 \,||\, s_1$.

На этапе **перемешивающего преобразования** осуществляется перемешивание полученной последовательности с применением регистра сдвига.

Исходными данными являются:

- слова H, M $\in V_{256}(2)$;
- слово S $\in V_{256}(2)$.

В качестве значения шаговой функции хэширования принимается слово

$$\psi^{61}(H \oplus \psi(M \oplus \psi^{12}(S))),$$

где ψ^i – i-я степень преобразования ψ.

Результаты

Результаты профилирования системы “ТерКрипт” в режиме формирования и проверки цифровой подписи по стандарту ГОСТ приведены в таблице.

Функция	Процент использования процессорного времени в функции	процент использования процессорного времени в функции и в вызываемых функциях
Шифрования в режиме простой замены	25%	25%
Функция ψ	14%	50%
Шаговая функция хэширования	4%	95%

Предлагается в специализированный кристалл включить следующие функции и данные (в порядке убывания важности):

1. Функцию шифрования в режиме простой замены, совпадающую для одного 64-битового блока с базовым циклом зашифрования, описанным выше.
2. Функцию ψ.

   Входные данные: 256 бит.

   Выходные данные: 256 бит.
3. Шаговую функцию хэширования.

   Входные данные:
   - стартовое слово H – 256 бит;
   - блок данных M – 256 бит;

   Используемые данные:
   - слова C_i (постоянные) – $3 \cdot 256$ бит.

   Выходные данные: 256 бит.

По нашим оценкам объем кристалла, реализующего предлагаемые функции, составит 10000 вентилей. Ориентировочная стоимость одного такого кристалла – 250$.

Список литературы

1. Андрей Винокуров “Алгоритм шифрования ГОСТ 28147-89, его использование и реализация для компьютеров платформы Intel x86”
2. ГОСТ Р 34.11-94 “Информационная технология. Криптографическая защита информации. Функция хэширования”
3. ГОСТ Р 34.10-94 “Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма”