Оптимизация схемной реализации потокового шифра ZUC

С. О. Супрунюк, Е. А. Курганов
(МГУ им. М. В. Ломоносова, Москва)

В работе говорится о глубине аппаратной (схемной) реализации потокового шифра ZUC и способах ее минимизации. Сначала приводится простая реализация алгоритма. После этого показываются способы оптимизации данной реализации.

Ключевые слова: потоковые шифры, оптимизация глубины схем.

Введение

ZUC — это потоковый шифр, который входит в состав мобильного стандарта 4G под названием LTE (Long Term Evolution).

В докладе рассматривается аппаратная реализация алгоритма ZUC. В частности, исследуется глубина схемы, реализующей данный алгоритм. Под глубиной понимается длина максимального простого пути схемы. Рассматривается базис из элементов конъюнкции, дизъюнкции, отрицания и задержки. При этом отрицание игнорируется при вычислении глубины. Результаты, связанные с оптимизацией глубины других алгоритмов в данном базисе можно найти, например, в работе [**?**].

Описание шифра ZUC

Потоковый шифр ZUC принимает на вход 128-битовый ключ и 128-битовый вектор инициализации (IV), а на выход идет ключевой поток из 32-битовых слов, который используется для шифрования. Подробное описание алгоритма ZUC может быть найдено в документе [**?**]. Алгоритм ZUC состоит из следующих частей (см рис. **??**):

1) регистр сдвига с обратной линейной связью LFSR (Linear Feedback Shift Register);

2) реорганизация битов BR (Bit Reorganization);
3) нелинейная функция F.

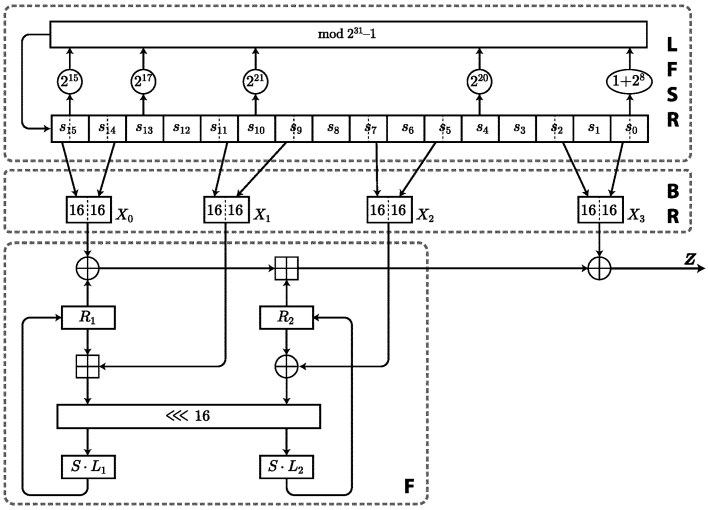

Рис. 1. Потоковый шифр ZUC.

Данный регистр сдвига имеет 2 режима работы: режим инициализации и рабочий режим. В первом режиме он получает на вход 31-битовое слово u, которое получается удалением старшего бита из 32-битового выхода W нелинейной функции F, ($u = W \gg 1$). Более подробно, на стадии инициализации регистр сдвига работает так:

LFSRWithInitialisationMode(u):

1) $v = 2^{15}s_{15} + 2^{17}s_{13} + 2^{21}s_{10} + 2^{20}s_4 + (1 + 2^8)s_0 \mod (2^{31} - 1)$;
2) $s_{16} = (v + u) \mod (2^{31} - 1)$;
3) Если $s_{16} = 0$, то определить $s_{16} = 2^{31} - 1$;
4) $(s_1, s_2, \ldots, s_{15}, s_{16}) \to (s_0, s_1, \ldots, s_{14}, s_{15})$.

В рабочем режиме регистр сдвига работает немного иначе: он ничего не получает на вход и выполняет шаги 1, 3, 4 работы в режиме инициализации.

В процессе реорганизации битов из восьми клеток LFSR s_0, s_2, s_5, s_7, s_9, s_{11}, s_{14}, s_{15} формируется 4 32-битовых слова X_0, X_1, X_2, X_3 по следующему правилу:

$$X_0 = s_{15H}||s_{14L},\ X_1 = s_{11L}||s_{9H},\ X_2 = s_{7L}||s_{5H}, X_3 = s_{2L}||s_{0H},$$

где s_{iH} означает биты $30 \dots 15$ и s_{iL} — биты $15 \dots 0$ клетки s_i соответственно, а $||$ означает конкатенацию.

Нелинейная функция F получает на вход 3 32-битовых слова X_0, X_1 и X_2. В состав функции входит две 32-битовых клетки памяти R_1 и R_2. На выход функция подает 32-битовое слово W. Более подробно, функция F имеет следующую структуру:

F (X_0, X_1, X_2):

1) $W = (X_0 \oplus R_1) + R_2$;
2) $W_1 = R_1 + X_1$;
3) $W_2 = R_2 \oplus X_2$;
4) $R_1 = S(L_1(W_{1L}||W_{2H}))$;
5) $R_2 = S(L_2(W_{2L}||W_{1H}))$.

где S — это S-блок размером 32×32 бита, а L_1, L_2 — линейные преобразования.

S-блок S размером 32×32 состоит из четырех S-блоков размера 8×8 бит, то есть $S = (S_0, S_1, S_2, S_3)$, где $S_0 = S_2, S_1 = S_3$. Описание данных S-блоков можно найти в [**?**].

Оценка глубины

Сложение по модулю $2^{31} - 1$

Существует множество разных сумматоров по модулю $2^{31}-1$. Минимальный по глубине сумматор описан в работе [**?**]. Для данного сумматора верна следующая оценка

Утверждение 1. *Используемый сумматор вычисляет сумму по модулю* $2^{31} - 1$ *с глубиной* $\leqslant 13$.

Заметим, что в LFSR 6 или 7 слагаемых в зависимости от режима, в котором находится алгоритм. Мы можем уменьшить глубину сложения, сделав из трех слагаемых два следующим образом:

$$A_1 + A_2 + A_3 \mod (2^{31} - 1) = A + B \mod (2^{31} - 1), \text{где}$$
$$A = A_1 \oplus A_2 \oplus A_3,$$
$$B = [b_{30}, b_{29}, \dots, b_0], b_i = a_{1,i-1}a_{2,i-1} \vee a_{2,i-1}a_{3,i-1} \vee a_{1,i-1}a_{3,i-1},$$

причем $a_{i,-1} = a_{i,30}$, $i = 1, 2, 3$. Тогда A вычисляется на глубине 4, B вычисляется на глубине 3.

Другие операции

Сложение по модулю 2^{32}. Для того, чтобы вычислить сумму двух 32-битовых векторов по модулю 2^{32}, можно воспользоваться методом золотого сечения, описанным в [**?**]. Тогда данную операцию можно реализовать с глубиной 11.

S-блок. Любой S-блок размером 8×8 бит можно реализовать, воспользовавшись СДНФ [**?**]. Глубина такой реализации равна 10.

Линейные преобразования. Линейные преобразования алгоритма ZUC имеют следующий вид:

$$L_1(X) = X \oplus (X \lll_{32} 2) \oplus (X \lll_{32} 10) \oplus (X \lll_{32} 18) \oplus (X \lll_{32} 24),$$
$$L_2(X) = X \oplus (X \lll_{32} 8) \oplus (X \lll_{32} 14) \oplus (X \lll_{32} 22) \oplus (X \lll_{32} 30).$$

Легко видеть, что каждое линейное преобразование L_i, $i = 1, 2$ можно реализовать с глубиной 6.

Общая оценка

Учитывая приведенные выше оценки, получается, что один раунд шифра ZUC имеет глубину $\leqslant 36$.

Оптимизирующие преобразования

Сложение по модулю $2^{31} - 1$

Так как в поле $GF(2^{31} - 1)$ ноль может быть представлен с помощью 31 бита как 0 и как $2^{31} - 1$, сумматор может представить ноль, полученный после сложения двух чисел по модулю $2^{31} - 1$, любым из этих способов. Поэтому в дальнейшем будет удобно пользоваться данным утверждением

Утверждение 2. *Выход схемы используемого сумматора по модулю* $2^{31} - 1$ *равен* 0 *тогда и только тогда, когда оба слагаемых равны* 0.

Следовательно, вместо того, чтобы проверять, равен ли нулю результат, можно проверить отдельно каждое слагаемое. Учитывая строение алгоритма ZUC, можно доказать следующее

Утверждение 3. *Во время работы алгоритма* ZUC *всегда хотя бы одно из слагаемых в сумматоре по модулю* $2^{31} - 1$ *не равно нулю.*

Оптимизация нелинейной функции F

Заметим, что при применении $L_i(A \boxplus B)$ мы можем сократить глубину на 2 путем представления в следующем виде: $L_i(A \boxplus B) = L_i(A \oplus B \oplus C) = L_i(A \oplus B) \oplus L_i(C)$, где $C = c_0, c_1, \ldots, c_{31}$ — биты переноса из сложения по модулю 2^{32}. Глубина нахождения c_{31} равна 9, $L_i(A \oplus B)$ считается схемой с глубиной 8. Следовательно, $L_i(A \boxplus B)$ можно посчитать на глубине 15.

Модификация сложения по модулю 2^{32}

Можно увидеть, что

$$(A \boxplus B) \gg 1 = (A \gg 1) + (B \gg 1) + \varepsilon \mod (2^{31} - 1),$$

где A и B — 32-битовые числа, а $\varepsilon \in \{-1, 0, 1\}$.

Существующие исследования

В настоящее время ведутся работы по оптимизации аппаратной реализации алгоритма ZUC. Среди всех них можно выделить работу [**?**]. Глубина одного раунда приведенной там реализации равна 31.

Результат

После применения всех оптимизирующих преобразований, указанных выше, удалось получить реализацию, для которой глубина одного раунда равна 26.

Список литературы

[1] Болотов А. А., Галатенко А. В., Гринчук М. И., Золотых А. А., Иванович Л. Методы оптимизации глубины реализации хэш-функций // Интеллектуальные системы. — 2013. — Т. 17, вып. 1–4. — С. 224–228.

[2] Specification of he 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification. Version: 1.6. — 2011.

[3] Specification of he 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 4: Design and Evaluation Report. Version: 2.0. — 2011.

[4] Dimitrakopoulos G., Vergos H. T., Nikolos D., Efstathiou C. A systematic methodology for designing area-time efficient parallel-prefix modulo 2^n-1 adders // Proc. IEEE Int. Symp. on Circuits and Systems. — 2003. — P. 225–228.

[5] Гашков С. Б., Гринчук М. И., Сергеев И. С. О построении схем сумматоров малой глубины // Дискретн. анализ и исслед. опер., сер. 1. — 2007. — № 14: 1. — С. 19–25.

[6] Яблонский С. В., Гаврилов Г. П., Кудрявцев В. Б. Функции алгебры логики и классы Поста. — М.: Наука, 1966.

[7] Lingchen Z., Luning X., Zongbin L., Jiwu J., Yuan M. Evaluating the optimized implementations of SNOW 3G and ZUC on FPGA // Proceedings of IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). — 25–27 June 2012. — P. 436–442.