АЛГОРИТМ ШИФРОВАНИЯ ICE

С. П. Панасенко , начальник отдела разработки программного обеспечения фирмы «АНКАД», к. т. н.

ИСТОРИЯ СОЗДАНИЯ АЛГОРИТМА

ICE (Information Concealment Engine — «механизм сокрытия информации») — один из многих алгоритмов шифрования, позиционируемых авторами как возможная замена стандарта шифрования США DES (Data Encryption Standard). Автор алгоритма — известный австралийский криптолог Мэтью Кван (Matthew Kwan). Алгоритм разработан в 1997 году.

Для облегчения возможной замены DES на ICE во многих приложениях автор алгоритма придал ему схожие с DES параметры:

- размер блока шифруемых данных — 64 бита;
- размер ключа шифрования — 64 бита (в отличие от 64-битного ключа DES, в котором значащими являются всего 56 бит, все биты ключа алгоритма ICE являются значащими).

Как известно, алгоритм ICE не стал заменой алгоритму DES и не получил широкого распространения. Судя по информации, изложенной в спецификации алгоритма (Kwan M. The Design of the ICE Encryption Algorithm http://www.darkside.com.au — 1997), вопреки мнению большинства экспертов, Мэтью Кван считал основной проблемой алгоритма DES не короткий ключ, а восприимчивость к линейному и дифференциальному криптоанализу, а также наличие слабых ключей.

64-битный ключ алгоритма ICE всего в 256 раз длиннее ключа DES, то есть тоже подвержен проблеме полного перебора ключей. Кроме того, 64-битный размер блока также стал считаться недостаточным — уже через 2 года после разработки алгоритма ICE начался конкурс AES по выбору нового стандарта шифрования США, который устанавливал принципиально иные характеристики нового стандарта:

- 128-битный размер блока шифруемых данных;
- 3 фиксированные длины ключа шифрования: 128, 192 и 256 бит.

Так что можно утверждать, что алгоритм ICE появился уже несколько устаревшим. Кроме того, в алгоритме были обнаружены уязвимости, которые также не способствовали его популярности.

СТРУКТУРА АЛГОРИТМА

Алгоритм ICE представляет собой сеть Фейстеля. Это наиболее распространенная структура алгоритмов блочного симметричного шифрования, которая подразумевает разбиение блока шифруемых данных на два (или более) субблока. Один из них обрабатывается определенным образом (с участием ключа раунда, который вычисляется из ключа шифрования алгоритма — данная процедура описана ниже) и накладывается на необработанный субблок, после чего они меняются местами. Такая обработка выполняется в несколько раундов (рис. 1).

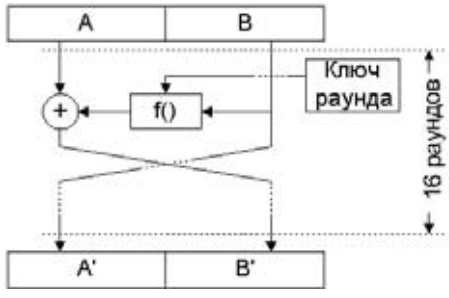

Рис. 1. Структура алгоритма ICE

В алгоритме ICE проводится 16 раундов, в каждом из которых над обрабатываемым 32-битным субблоком выполняются следующие операции (рис. 2).

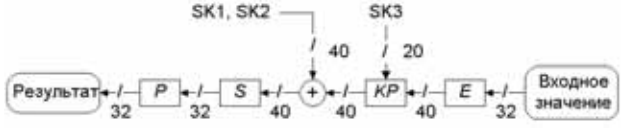

Рис. 2. Функция ***f***

1. Функция ***E***, выполняющая расширение субблока до четырех 10-битных величин $E_1 \ldots E_4$ следующим образом:

$E_1 = P_1, P_0, P_{31}, P_{30}, P_{29}, P_{28}, P_{27}, P_{26}, P_{25}, P_{24}$;
$E_2 = P_{25}, P_{24}, P_{23}, P_{22}, P_{21}, P_{20}, P_{19}, P_{18}, P_{17}, P_{16}$;
$E_3 = P_{17}, P_{16}, P_{15}, P_{14}, P_{13}, P_{12}, P_{11}, P_{10}, P_9, P_8$;
$E_4 = P_9, P_8, P_7, P_6, P_5, P_4, P_3, P_2, P_1, P_0$,

где P_n — ***n***-й бит обрабатываемого субблока (считая справа налево).

2. Зависящая от значения первых 20 бит ключа раунда (K_i, где ***i*** — номер текущего раунда) перестановка ***KP***, которая выполняется по следующим правилам:

если ***n***-й бит ($0 \le n \le 9$) фрагмента ключа раунда имеет значение 1, то меняются местами ***n***-е биты E_2 и E_4;

если ($10 + n$)-й бит фрагмента ключа раунда имеет значение 1, то меняются местами ***n***-е биты E_1 и E_3.

3. Наложение материала ключа — выполнение побитовой логической операции «исключающее или» (XOR) над каждым битом следующих 40 бит ключа раунда (ключ раунда имеет размер 60 бит) и соответствующим битом последовательности $E_1 \ldots E_4$.

4. Табличная замена S: после наложения ключа $E_1 \ldots E_4$ «прогоняются» через таблицы замен $S_1 \ldots S_4$ соответственно. Каждая из этих таблиц заменяет входное 10-битное значение 8-битным. Таблицы замен работают следующим образом:

• 9 и 0-й биты входного значения формируют значение переменной V, значение остальных бит входного значения обозначается как Y;

• выходное значение Z вычисляется следующим образом:

$$Z = (Y + O_X)^{7} \bmod P_V,$$

где O_X — константа, определяемая следующим образом:

	O_0	O_1	O_2	O_3
S_1	83	85	9B	CD
S_2	CC	A7	AD	41
S_3	4B	2E	D4	33
S_4	EA	CD	2E	04

P_V — неприводимый в 8-битном поле Галуа многочлен, коэффициенты которого определяются двоичной последовательностью m_V, представленной в десятичном виде в следующей таблице:

	m_0	m_1	m_2	m_3
S_1	333	313	505	369
S_2	379	375	319	391
S_3	361	445	451	397
S_4	397	425	395	505

5. Функция P, выполняющая битовую перестановку данных согласно следующей таблице:

$S\,1_7$	$S\,4_7$	$S\,3_7$	$S\,2_7$	$S\,2_6$	$S\,3_6$	$S\,1_6$	$S\,4_6$
$S\,3_5$	$S\,2_5$	$S\,4_5$	$S\,1_5$	$S\,4_4$	$S\,1_4$	$S\,2_4$	$S\,3_4$
$S\,2_3$	$S\,3_3$	$S\,4_3$	$S\,1_3$	$S\,1_2$	$S\,4_2$	$S\,2_2$	$S\,3_2$
$S\,4_1$	$S\,1_1$	$S\,3_1$	$S\,2_1$	$S\,3_0$	$S\,2_0$	$S\,1_0$	$S\,4_0$

где Sa_b — b-й выходной бит таблицы замен Sa; таблица обозначает, что бит $S1_7$ становится битом 31, бит $S\,4_7$ — битом 30 и т. д. (рис. 3).

После этого обрабатываемый субблок накладывается на необработанный операцией XOR.

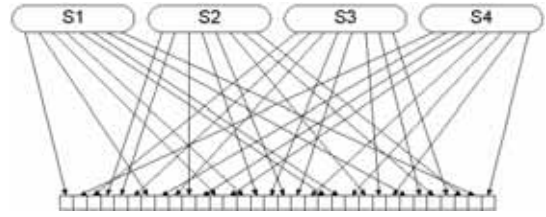

Рис. 3. Перестановка P

Как видно, функция раунда алгоритма ICE весьма похожа на функцию раунда алгоритма DES; принципиальное отличие состоит лишь в том, что в DES отсутствует зависящая от ключа перестановка.

ВАРИАНТЫ АЛГОРИТМА

Выше описан основной вариант алгоритма ICE. Кроме того, в спецификации алгоритма описаны следующие варианты.

1. Thin-ICE — версия с уменьшенным числом раундов (и более низкой криптостойкостью), предназначенная для высокоскоростных применений, в которых не предъявляются требования к повышенной защищенности данных. В данном варианте алгоритма выполняется 8 раундов вместо 16.

2. ICE-n — версия с увеличенным (потенциально неограниченным) числом раундов, в которой выполняется n * 16 раундов шифрования. В данном варианте отличается и размер ключа алгоритма — вместо 64-битного используется (64 * n)-битный ключ.

ПРОЦЕДУРА РАСШИРЕНИЯ КЛЮЧА

Задача процедуры расширения ключа состоит в формировании 16 ключей раунда по 60 бит. Данная процедура является достаточно сложной и выполняется в несколько шагов.

Шаг 1. 64-битный ключ шифрования используется для инициализации массива временных переменных KB:

$$KB\,[0] = K_{63} \ldots K_{48},$$
$$KB\,[1] = K_{47} \ldots K_{32},$$
$$KB\,[2] = K_{31} \ldots K_{16},$$
$$KB\,[3] = K_{15} \ldots K_{0},$$

где K_n — n-й бит расширяемого ключа шифрования.

Шаг 2. Обнуляются регистры $SK1 \ldots SK3$, предназначенные для хранения текущего ключа раунда. Предполагается, что назначение подключей таково:

$SK1$ и $SK2$ — предназначены для наложения ключа операцией XOR,

$SK3$ — управляет перестановкой KP (рис. 2).

Шаг 3. Для каждого раунда процедуры расширения ключа (соответствуют раундам алгоритма) j поочередно для $SK1$, $SK2$ и $SK3$ (текущий регистр ниже обозначен как SK) пятикратно выполняется цикл по i от 0 до 3 (то есть всего 4 уровня вложенности циклов), в котором производятся следующие действия:

$$B = KB[(i + KR[j]) \bmod 4]_0,$$
$$SK << 1,$$
$$SK_0 = B,$$
$$KB[(i + KR[j]) \bmod 4] >> 1,$$
$$KB[(i + KR[j]) \bmod 4]_{19} = \sim B,$$

где ***B*** — временная 1-битовая переменная,
i — операции сдвига на указанное число бит влево и вправо соответственно;
~ B — обратное значение для ***B***;
KR[j] — ***j***-й элемент массива ***KR***, определенного следующим образом:

Раунд j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
KR[j]	0	1	2	3	2	1	3	0	1	3	2	0	3	1	0	2

Выше описана процедура расширения ключа для основного варианта алгоритма ICE, в котором выполняется 16 раундов шифрования.

Для других вариантов алгоритма расширение ключа выполняется следующим образом:

- для 8-раундового Thin-ICE просто выполняются первые 8 раундов описанного выше цикла шага 3 процедуры расширения ключа;
- для ICE-n выполняется описанная выше процедура расширения ключа, затем массив ***KB*** инициализируется следующим 64-битным фрагментом ключа, на основе которого вычисляются еще 16 ключей раунда и т. д. (в цикле всего ***n*** раз).

КРИПТОАНАЛИЗ АЛГОРИТМА

В 1998 году несколько экспертов из католического университета Лювена (Бельгия) предложили метод вскрытия алгоритма ICE с помощью дифференциального криптоанализа[1] (Van Rompay B., Knudsen L.R., Rijmen V. Differential cryptanalysis of the ICE encryption algorithm http://www.cosic.esat.kuleuven.ac.be — 1998).

Данный метод позволяет добиться следующих результатов:

Алгоритм Thin-ICE вскрывается (то есть вычисляется используемый ключ шифрования) при наличии 2^{23} выбранных открытых текстов и соответствующих им шифротекстов[2] с вероятностью 25%. При увеличении числа выбранных открытых текстов до 2^{27} вероятность успеха повышается до 95%.

Существует вероятность вскрытия и стандартного алгоритма ICE (правда, достаточно небольшая) при наличии не менее чем 2^{62} выбранных открытых текстов. Данная атака, однако, весьма сложно реализуема на практике.

Более подробные сведения об алгоритме ICE можно найти на домашней странице ICE по адресу: http://www.darkside.com.au/ice.

1 Анализ зависимостей между соотношениями двух или более открытых текстов и соответствующих им шифротекстов.

2 Атака на основе выбранного открытого текста.