Алгоритм шифрування "Camellia"

Д.Г. Шевченко, *студент,*
Є.В. Мелешко, *ст. викладач, канд. техн. наук*
Кіровоградський національний технічний університет

Camellia - алгоритм симетричного блокового шифрування (розмір блока - 128 біт, ключ - 128, 192, 256 біт), розробка японських компаній Nippon Telegraph and Telephone Corporation і Mitsubishi Electric Corporation. Сертифікований японською організацією CRYPTREC як рекомендований для промислового та державного використання алгоритм.

Camellia є подальшим розвитком алгоритму шифрування E2, одного з алгоритмів, представлених на конкурсі AES і з використанням елементів алгоритму MISTY1 .

Структура алгоритму заснована на класичному ланцюзі Фейстела з попереднім і фінальним забілюванням. Циклова функція використовує нелінійне перетворення (S-блоки), блок лінійного розсіювання кожні 16 циклів (побайтовая операція XOR) і байтову перестановку.

В залежності від довжини ключа має 18 циклів (128 розрядний ключ), або 24 циклу (192 і 256 розрядний ключ).

Генерація допоміжних ключів

1. Ключ (K) розбивається на дві 128-бітні частини KL і KR (Рисунок 1).

Ключ	KL	KR
128	K	0
192	K >> 64	((K & MASK64) << 64) | (~ (K & MASK64))
256	K >> 128	K & MASK128

Константа	Значення
MASK8	0xff
MASK32	0xffffffff
MASK64	0xffffffffffffffff
MASK128	0xffffffffffffffffffffffffffffffff
C1	0xA09E667F3BCC908B
C2	0xB67AE8584CAA73B2
C3	0xC6EF372FE94F82BE
C4	0x54FF53A5F1D36F1C
C5	0x10E527FADE682D1D
C6	0xB05688C2B3E6C1FD

Рисунок 1 – Розбиття ключа та значення контант

2. Обчислюємо 128-бітові числа KA і KB (Рисунок 2). Змінні D1 і D2 – 64-бітні.

3. Обчислюємо допоміжні 64-бітові ключі kw1, ..., kw4, k1, ..., k24, ke1, ..., ke6 в залежності від розміру ключа.

Шифрування

Відбувається за схемою Фейстела з 18 етапами для 128-бітового ключа і 24 етапами для 192 - і 256-бітних ключів. Кожні 6 етапів застосов. функції FL і FLINV.

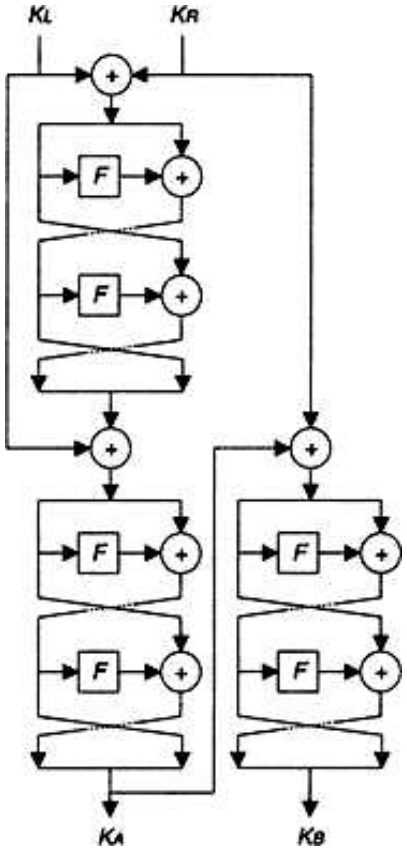

Рисунок 2 – Обчислення Ka I Kb.

Допоміжні функції F, FL, FLINV

F-, FL-і FLINV-функції на вхід отримують два 64-бітних параметра - дані F_IN і ключ KE.

Функція F використовує 16 8-бітних змінних t1, ..., t8, y1, ..., y8 і 1 64-бітну змінну. На виході функції 64-бітове число.

Функції FL і FLINV використовують 4 32-бітові змінні x1, x2, k1, k2. На виході функції 64-бітове число. Функція FLINV - обернена до FL.

S - блоки

Значення функції визначається з таблиці, зображеної на Рисунку 3.

Розшифрування

Алгоритм розшифрування ідентичний шифруванню, з тією лише різницею, що допоміжні ключі міняються місцями за такою схемою, в залежності від довжини вихідного ключа.

Застосування

Підтримка Camellia була додана в фінальній версії Mozilla Firefox 3 в 2008 році. Пізніше в тому ж році команда розробників FreeBSD оголосила, що підтримка цього шифрування також була включена в FreeBSD 6.4-RELEASE. У вересні 2009 року GNU Privacy Guard додали підтримку Camellia у версії 1.4.10. Крім того, багато популярних бібліотек безпеки, такі як Crypto++, GnuTLS, PolarSSL і OpenSSL також включають в себе підтримку Camellia.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	112	130	44	236	179	39	192	229	228	133	87	53	234	12	174	65
1	35	239	107	147	69	25	165	33	237	14	79	78	29	101	146	189
2	134	184	175	143	124	235	31	206	62	48	220	95	94	197	11	26
3	166	225	57	202	213	71	93	61	217	1	90	214	81	86	108	77
4	139	13	154	102	251	204	176	45	116	18	43	32	240	177	132	153
5	223	76	203	194	52	126	118	5	109	183	169	49	209	23	4	215
6	20	88	58	97	222	27	17	28	50	15	156	22	83	24	242	34
7	254	68	207	178	195	181	122	145	36	8	232	168	96	252	105	80
8	170	208	160	125	161	137	98	151	84	91	30	149	224	255	100	210
9	16	196	0	72	163	247	117	219	138	3	230	218	9	63	221	148
a	135	92	131	2	205	74	144	51	115	103	246	243	157	127	191	226
b	82	155	216	38	200	55	198	59	129	150	111	75	19	190	99	46
c	233	121	167	140	159	110	188	142	41	245	249	182	47	253	180	89
d	120	152	6	106	231	70	113	186	212	37	171	66	136	162	141	250
e	114	7	185	85	248	238	172	10	54	73	42	104	60	56	241	164
f	64	40	211	123	187	201	67	193	21	227	173	244	119	199	128	158

Рисунок 3 – Таблиця значень функції

Криптоаналіз алгоритму

Первинний аналіз алгоритму Camellia був виконаний його розробниками. Була показана стійкість алгоритму до лінійного і диференціального криптоаналізу, а також до використання зрізаних і неможливих диференціалів, методу бумеранга, методу інтерполяції, зсувними атакам і ряду інших атак.

Перші відгуки відомих експертів про алгоритм також були вкрай позитивними. Відзначена виключно висока крипостійкість. Ларс Кнудсен стверджує наступне:

- Будь-які практично здійсненні атаки на Camellia будуть можливі тільки після принципових проривів в області криптоаналізу;
- Camellia - один з найбільш стійких сучасних алгоритмів шифрування;

В оцінці ресурсоємності та швидкодії алгоритму експерти були далеко не так одностайні:

- Визначилось, що Camellia істотно програє у швидкості алгоритму Rijndael;
- Пред'являє досить високі вимоги до оперативної і енергонезалежної пам'яті;
- І навпаки, автори звіту CRYPTREC відзначили високу швидкість шифрування (особливо на серверних платформах), швидку процедуру розширення ключа і досить невисокі вимоги до енергонезалежної пам'яті.

Camellia - один з алгоритмів-переможців конкурсу NESSIE. В рамках досліджень, не було виявлено будь-яких проблем з криптостійкістю даного алгоритму. В даний час також не знайдено яких-небудь серйозних вразливих місць в алгоритмі, однак активний аналіз цього алгоритму буде продовжуватись.

Список джерел

1. Camellia (алгоритм) – Википедия: http://ru.wikipedia.org/wiki/Camellia_(алгоритм).
2. Алгоритм Camellia | Блог о шифровании. Статьи описывающие алгоритмы и способы реализации систем шифрования и безопасности: http://crypto.pp.ua/2011/01/algoritm-camellia/