

А. А. Е л и с т р а т о в, Н. В. Н и к о н о в, А. О. Ш у м и л о в
(Москва, ТВП). **О паддинг-атаках на криптографические протоколы, использующие стандартные n -разрядные блочные режимы шифрования.**

Одной из первых работ, посвященных так называемым паддинг-атакам (от англ. padding-attack) на криптопротоколы TLS, IPsec и др. является работа С. Воденя [1], представленная на конференции Eurocrypt 2002. Описанная в работе [1] атака относится к классу активных атак с подобранными шифртекстами, которая позволяет без нахождения ключа найти блоки открытого текста $x_1, \dots, x_N$ исходя из соответствующих перехваченных блоков шифртекста $y_1, \dots, y_N$, полученных в режиме CBC ([2, 3]) и некоторой дополнительной информации. Основой данной атаки является знание того факта, что последний неполный блок непосредственно перед шифрованием в режиме CBC должен быть дополнен до полного блока известной последовательностью байтов — паддингом. Кроме того, для проведения атаки необходимо наличие «оракула», который некоторым способом сообщает об ошибке при расшифровании в случае получения неверного паддинга или другой информации, связанной с паддингом. Применительно к протоколу TLS таким оракулом может являться TLS Alert Protocol (протокол извещения, см. [4, 5]), а для IPsec — протокол ICMP (Internet Control Message Protocol, см. [6]).

В данных тезисах авторы обращают внимание на тот факт, что аналогичные атаки возможны в некоторых случаях и при использовании других режимов шифрования, в частности режимов гаммирования: CFB, OFB и CTR (см. [2, 7, 8, 9, 10]), которые не требуют дополнения последнего неполного блока до полного. Для TLS такая возможность может возникнуть в случае известных полей данных (например, для сокрытия истинной длины открытого текста) или при встраивании блочной шифрсистемы с режимами гаммирования именно как блочной шифрсистемы, а не поточной, что потребует присутствия однобайтового поля PL (Padding Length), содержащего 0. Для IPsec эта возможность обуславливается наличием однобайтовых полей PL и NH (Next Header). Данные поля (а, возможно и другие) по сути будут играть роль паддинга, что позволит находить не менее одного байта каждого блока открытого сообщения.

Пусть имеются зашифрованные блоки $y_1, y_2, \dots, y_N$, полученные в одном из четырех упомянутых выше режимах шифрования из блоков открытого текста $x_1, x_2, \dots, x_N$, уравнения в кратком виде для которых представлены в табл. (столбец 2).

Таблица

Режим шифрования	Уравнение зашифрования	Формируемое сообщение Y , — обозначение конкатенации блоков	Последний при расшифровании Y блок X , у которого проверяется паддинг
CBC	$y_j = E_k(x_j \oplus y_{j-1}),$ $y_0 = IV$	$r \oplus y_{j-1} y_j$	$E_k^{-1}(y_j) \oplus y_{j-1} \oplus r = x_j \oplus r$
CFB	$y_j = x_j \oplus E_k(y_{j-1}),$ $y_0 = E_k(IV)$	$y_{j-1} r \oplus y_j$	$y_j \oplus E_k(y_{j-1}) \oplus r = x_j \oplus r$
OFB	$y_j = x_j \oplus z_j,$ $z_0 = IV,$ $z_j = E_k(z_{j-1})$	$\overbrace{r' \dots r'}^{j-1} r \oplus y_j,$	$y_j \oplus z_j \oplus r = x_j \oplus r$
CTR	$y_j = x_j \oplus E_k(z_j),$ $z_0 = E_k(IV),$ $z_j = C(z_{j-1}),$ C — функция счетчика	r' — случайный блок	

Приведем общий алгоритм (не зависящий от вида паддинга и т.п.), основанный на содержимом табл., для определения последнего байта $x_j^{(b)}$ блока $x_j = x_j^{(1)}, x_j^{(2)}, \dots, x_j^{(b)}$, b — размер в байтах блока шифрсистемы E_k (блочной системы шифрования с неизвестным ключом k) при условии известного паддинга. В обозначениях алгоритма $\Omega(Y)$ — результат проверки корректности паддинга у X (см. табл., столбец 4) после расшифрования сообщения (табл., столбец 3) исходя из того или иного паддинга (1 — паддинг корректен, 0 — нет).

1. Выбрать случайно $r = r^{(1)}, r^{(2)}, \dots, r^{(b)}$;
2. Для всех возможных $r^{(b)}$ выполнить {если $\Omega(Y) = 1$, переход на 3} ;
3. Положить $x_j^{(b)} = (X^{(b)} \oplus r^{(b)})$, где $X^{(b)}$ — последний байт X .

В зависимости от вида паддинга, рассматриваемых криптопротоколов и т.п. в ряде случаев представляется возможным повторить данную процедуру для определения последующих байтов блока x_j , а в дальнейшем — и для всех блоков вплоть до первого. В этом случае трудоемкость такого алгоритма может быть оценена величиной $O(bN2^b)$.

В некоторых реализациях потребуется перебирать по два и более байтов блока для нахождения соответствующих байтов открытого текста. Кроме того, принципиальным для нахождения первого блока x_1 может оказаться знание или незнание вектора инициализации IV . Также отметим, что в реальности не всегда удастся получить точную информацию от «оракула» Ω . Так например, все ошибки в TLS (о некорректном паддинге, некорректном коде аутентичности, просроченном сертификате) шифруются. Для того, чтобы различить эти ошибки, можно использовать вторичные признаки, например, разницу откликов об этих ошибках по времени (см. [4, 5]).

СПИСОК ЛИТЕРАТУРЫ

1. Vaudenay S. Security flaws induced by CBC padding — applications to SSL, IPSEC, WTLS. — Lect. Notes Comput. Sci., 2002, v. 2332, p. 534–546.
2. FIPS 81. FIPS Publication 81. DES Modes of Operation. U.S. DoC/NIST. December 1980.

3. *Popov V., Kurepkin I., Leontiev S.* Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms. RFC 4357, 2006.
4. *Canvel B., Hiltgen A., Vaudenay S., Vuagnoux M.* Password Interception in a SSL/TLS Channel. — *Lect. Notes Comput. Sci.*, 2003, v. 2729.
5. *AlFardan N. J., Paterson K. G.* Lucky Thirteen: Breaking the TLS and DTLS Record Protocols. Egham, Surrey: Royal Holloway, Univ. London, 2013.
6. *Degabriele J.P., Paterson K. G.* Attacking the IPsec Standards in Encryption-only Configurations. — In: *IEEE Symposium on Privacy and Security*. Piscataway: IEEE Comput. Soc., 2007.
7. ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. М.: Изд-во стандартов, 1989.
8. ГОСТ Р ИСО/МЭК 10116-93. Информационная технология. Режимы работы для алгоритма n -разрядного блочного шифрования. М.: Изд-во стандартов, 1994.
9. *Ferguson N., Schneier B., Tadayoshi K.* *Cryptography Engineering. Design Principles and Practical Application*. N. Y. etc.: Wiley, 2010.
10. *Dolmatov V.* GOST 28147-89: Encryption, Decryption, and Message Authentication Code (MAC) Algorithms. RFC 5830, 2010.