Шаханова М.В., Евдокимов И.С., Шаханова Э.С. Аналитический обзор криптографических атак на SHA-512 // Международный журнал информационных технологий и энергоэффективности. – 2024. – Т. 9 № 7(45) с. 207-212

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.056

АНАЛИТИЧЕСКИЙ ОБЗОР КРИПТОГРАФИЧЕСКИХ АТАК НА SHA-512

[1]Шаханова М.В., Евдокимов И.С., Шаханова Э.С.
ФГБОУ ВО «МОРСКОЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ АДМИРАЛА Г.И. НЕВЕЛЬСКОГО», Владивосток, Россия (690003, г. Владивосток, ул. Верхнепортовая, д.50а), e-mail: [1]marinavl2007@yandex.ru

В данной статье будут рассмотрены известные аналитические атаки на SHA-512.

Ключевые слова: криптография, информационная безопасность, хеш-функции, SHA-512.

ANALYTICAL REVIEW OF CRYPTOGRAPHIC ATTACKS ON SHA-512

[1]Shakhanova M.V., Evdokimov I.S., Shakhanova E.S.
MARITIME STATE UNIVERSITY NAMED AFTER ADMIRAL G.I. NEVELSKOY, Vladivostok, Russia (690003, Vladivostok, st. Verkhneportovaya, 50a), e-mail: [1]marinavl2007@yandex.ru

This paper will consider the well-known analytical attacks on SHA-512.

Keywords: cryptography, information security, hash functions, SHA-512.

Семейство криптографических протоколов SHA-2

SHA-512 – это один из алгоритмов хеширования, используемых для вычисления криптографических хеш-сумм сообщений или данных. Он создан для обеспечения безопасности и целостности данных, а также для проверки целостности файлов и сообщений. SHA-512 является частью семейства алгоритмов SHA-2 и использует структуру Меркла-Дамгарда из функции одностороннего сжатия, которая в свою очередь использует структуру Дэвиса-Мейера из специализированного блочного шифра. Алгоритм обрабатывает сообщения блоками длиной 1024 бит и генерирует хеш-значение фиксированной длины, которое уникально для каждого входного сообщения. SHA-512 обладает высокой стойкостью к коллизиям и широко применяется в криптографических целях, таких как цифровые подписи, аутентификация и хранение паролей.

Атаки на SHA-512

SHA-512 до сих пор считается одним из самых криптостойких, а главное тщательно изученных алгоритмов хеширования. Несмотря на непрекращающиеся исследования, относительно удачные атаки (по затратам памяти и сложности вычислений) были проведены только с неполным количеством раундов хеширования. Для полного же количества раундов SHA-512 (80) по-прежнему требуется произвести 2512 операций для нахождения прообраза, или же 2256 для нахождения коллизий.

Так как каждая атака является комплексным теоретическим исследованием на десятки страниц, ниже будут приведены лишь их краткие резюме, а самое главное – динамика прогресса в криптоанализе SHA-512.

Коллизионная атака

Как было сказано ранее, все известные на сегодняшний день успешные атаки на SHA-512 были проведены только с неполным количеством раундов. И хотя нахождение коллизий при неполном количестве раундов и не несёт большой практической пользы, само их наличие говорит о потенциальных уязвимостях исследуемой хеш-функции с последующей возможностью вычислять коллизии для полного количества раундов.

Нахождение коллизий представляет собой решение огромного количества связанных между собой линейных систем уравнений. Для этого используется 2 подхода:

1. Детерминированный – как правило, заключается в определённом выборе некоторых переменных, например, значениях Wi, либо начальных условий IV.
2. Вероятностный – случайный выбор значений переменных. Он может основываться на некоторых статистических данных (например, распределение некоторой случайной величины), либо же абсолютно случайным.

Однако зачастую используется комбинированный подход с использованием и детерминированного, и вероятностного методов.

Таким образом, ключевой метрикой атак становится не только затрата памяти и количество вычислительных операций, но и вероятность их успеха.

Известные атаки

1. В 2008 году исследователям из Индийского Института Статистики удалось разработать подход для нахождения коллизии в 24 раундах хеширования из 80[1].

Они использовали комбинацию вероятностных и детерминированных подходов, в частности, пользуясь тем, что SHA-512 даёт свободу выбора значений слов W0, W1, …, W15. Благодаря этому, например, можно также просчитать и последующие Wi:

$W16 = \sigma1(W14) + W9 + \sigma0(W1) + W0,$

$W17 = \sigma1(W15) + W10 + \sigma0(W2) + W1,$

$W18 = \sigma1(W16) + W11 + \sigma0(W3) + W2,$

$W19 = \sigma1(W17) + W12 + \sigma0(W4) + W3,$

$W20 = \sigma1(W18) + W13 + \sigma0(W5) + W4,$

$W21 = \sigma1(W19) + W14 + \sigma0(W6) + W5,$

…,

где σ – это линейная функция расширения сообщения.

Для выбора слов W0, W1, W2, W3, которые обеспечат локальную коллизию, предлагается использовать следующий алгоритм их нахождения:

1. Для начала выбрать W0, а также регистры a2 и a3 случайным образом.
2. Просчитать хеш-сумму с W0 и определить Ф0, где

$\Phi i = \Sigma0(ai) + fMAJ(ai , bi , ci) + \Sigma1(ei) + fIF (ei , fi , gi) + hi + Ki+1.$

3. Имея a2 и a3, определить e7 и e6, используя CDE (Cross Dependence Equation – уравнение перекрёстной зависимости), где

$ei = ai + ai-4 - \Sigma0(ai-1) - fMAJ(ai-1, ai-2, ai-3).$

4. Определить C4, используя формулу

$Ci = ei+5 - \Sigma1(ei+4) - fIF (ei+4, ei+3, ei+2) - 2ai+1 - Ki+5 + \Sigma0(ai),$

а затем D по формуле:

$D = W16 - (\sigma1(W14) + C4 + fMAJ(a4, a3, a2) - \Phi0 + W0),$ где

$W16 = \sigma1(W14) + C4 - W1 + fMAJ(a4, a3, a2) - \Phi0 + \sigma0(W1) + W0$

5. Решить уравнение D = −W1 + σ0(W1), выбирая W1 случайным образом.

6. Просчитать хеш-сумму с W1, чтобы определить регистры a1, b1, …, h1.

7. Определить Ф1 по данной выше формуле для Фi, а затем найти W2, используя:

W2 = a2 − (Σ0(a1) + fMAJ(a1, b1, c1) + h1 + Σ1(e1) + fIF (e1, f1, g1) + K2).

8. 6. Просчитать хеш-сумму с W2, чтобы определить регистры a2, b2, …, h2.

9. Определить Ф2 по данной выше формуле для Фi, а затем найти W3, используя:

W3 = a3 − (Σ0(a2) + fMAJ(a2, b2, c2) + h1 + Σ1(e2) + fIF (e2, f2, g2) + K2).

10. Рассчитать W17 и W18, используя:

W17 = σ1(W15) + C5 − W2 + fMAJ(a5, a4, a3) − Ф1 + σ0(W2) + W1

W18 = σ1(W16) + C6 − W3 + fMAJ(a6, a5, a4) − Ф2 + σ0(W3) + W2.

11. W0, W1, W2, W3 окажутся подходящими, если:

σ1(W17 + 1) − σ1(W17) = −δ1 и

σ1(W18 − 1) − σ1(W18) = δ2, где

δ1 = δWi+2, δ2 = δWi+3,

δX = X′ − X, а X′ – битовая размерность.

В качестве одного из возможных методов оптимизации предлагается использовать таблицу с заранее просчитанными возможными значениями −W1 + σ0(W1) для каждого W1 (264), что не было использовано в данной работе, однако должно уменьшить количество вычислений с 232.5 до 232.

2. В 2014 удалось исследователям из Грацкого Технического Университета найти псевдо-коллизию в 38 раундах из 80[2]. Псевдо-коллизионная атака отличается от обычной коллизионной своей целью – компрессионной функцией, и зачастую может привести к дальнейшим успехам по поискам коллизий для всей хеш-функции.

В своей работе исследователи полагались на метод “guess-and-determine” (дословно – “угадай и определи”). Это одна из самых широко используемых техник в криптоанализе для восстановления неизвестных переменных в данной системе. Используя её, выбирается некоторый набор неизвестных переменных и их значения (guess), после чего с их помощью выводятся оставшиеся переменные (determine). В случае, если возникло противоречие, необходимо вернуться назад (backtracking) и выбрать другой набор переменных или их значения.

По сути, множество возможных параметров при использовании данного метода представляет из себя дерево, а сам выбор переменных и их значений – нахождение оптимального пути. Поэтому одной из главных задач становится распознавание и “отсечение” тупиковых ветвей при минимальном количестве вычислений. В этом может помочь:

1. Стратегия угадывания – определяет факторы выбора переменных. Например, выбирать их не случайным образом, а на основе некоторых характеристик.

2. Правила ветвления – определяют, каким именно образом на каждой итерации алгоритма поиска пути нужно выбирать ветвь в дереве.

3. Стратегия распространения – определяет, насколько тщательно нужно проверять новые выбранные переменные в уже просчитанных раннее уравнениях. При слишком поверхностных проверках противоречия будут возникать чаще, зато алгоритм отработает быстрее. При слишком тщательных – работа алгоритма замедлится, при этом противоречия будут выявляться на гораздо более ранних стадиях.

4. Стратегия бэктрекинга (возвращения назад) – определяет необходимые действия в случае обнаружения противоречия. Например, можно возвращаться всего на несколько шагов назад и выбирать другие переменные, либо же начинать работу алгоритма с самого начала (при этом убедиться, что программа не будет постоянно попадать на один и тот же неверный путь).

Данный метод имеет и другие возможные оптимизаций и продолжает активно развиваться и в наше время.

3. В 2016 почти той же группе исследователей из Грацкого Технического Университета удалось найти коллизию в 27 раундах из 80 и псевдо-коллизию в 39 раундах из 80[3].

На этот раз основной целью их исследования стали алгоритмы хеширования SHA-512/224 and SHA-512/256, которые схожи с основным алгоритмом SHA-512, однако применяют к результату усечение битности. Такой подход вместо, например, использования обычного SHA-256, а не SHA-512/256, применяется из-за более высокой скорости работы SHA-512 на 64-битных системах (так как длина слова тоже 64 бита).

Данное исследование во многом схоже с предыдущим, однако его более высокие результаты обусловлены несколько иным выбором характеристик на стадии “guess-and-determine”, что в подробности описано в самой статье.

Атака на прообраз

Данная атака привлекает куда меньшее количество исследователей, чем разработка методов поиска коллизий. Скорее всего, это можно объяснить тем, что шанс разработать эффективную коллизионную атаку гораздо выше, нежели атаку на прообраз. Так, известная атака на прообраз уже давно не криптостойкой хеш-функции MD5 в лучшем случае требует 239 операций, в то время как коллизионная атака – всего лишь 218.

Известные атаки на прообраз основываются на ещё одном широко используемом в криптоанализе методе “встречи посередине” (MITM, meet-in-the-middle), за счёт которого уменьшается количество требуемых операций, но увеличиваются затраты памяти. Применим не только к хеш-функциям, но и к симметричным алгоритмам шифрования. Рассмотрим его в контексте SHA-512.

Как правило, атаки на прообраз хеш-функции, построенной на основе структуры Меркла-Дамгора, основаны на атаке на псевдо-прообраз компрессионной функции. В свою очередь, многие компрессионные функции используют структуру Дэвиса-Мейера, которая базируется на блочном шифре E: $E_A(B) \oplus B$, где A и B – либо промежуточные значения хеша, либо первоначальные значения сообщения. Тогда необходимо:

1. Разделить ключ A блочного шифра E на две независимые части A1 и A2.
2. Случайным образом выбрать входное значения B шифра E.
3. Произвести “прямое” вычисление шифра, используя B и все возможные значения A1, затем сохранить полученные промежуточные значения в таблице TF.
4. Произвести “обратное” вычисление, используя $h \oplus B$ и все возможные значения A2, затем сохранить полученные промежуточные значения в таблице TB.
5. Проверить, есть ли совпадения в таблицах TF и TB, которые и означают коллизию. Если есть – псевдо-прообраз h сгенерирован. Если нет – необходимо вернуться на шаг 2 с другим значением B.

В свою очередь, метод MITM может иметь различные вариации и оптимизации, сокращающие количество необходимых вычислений.

В 2009 году группе исследователей из Японского Университета Электро-Коммуникаций удалось разработать атаку на прообраз в 46 раундах из 80[4]. При этом для её реализации требуется произвести 2511.5 операций, что лишь немногим лучше полного перебора в 2512 операций. Атака полагается на уже описанный выше метод MITM, а также некоторые другие специфические техники.

На сегодняшний день алгоритм хеширования SHA-512 является криптостойким – не слишком многие известные теоретические атаки работают лишь при неполном количестве раундов. Однако, как уже было сказано, само их наличие уже указывает на возможные уязвимости в работе хеш-функции. К тому же, семейство SHA-2 по принципу своей работы во многом совпадает с семейством SHA-1, которое давно признано некриптостойким. Поэтому, вполне вероятно, что в ближайшие годы появятся гораздо более эффективные теоретические атаки, а возможно даже и практические. В то же время, NIST активно участвует в продвижении семейства SHA-3 – на данный момент в претендентах десятки различных хеш-функций, к которым предъявляются более строгие требования, чем к SHA-2. Именно их криптоанализом сейчас и занимается множество различных исследователей со всего мира. Впрочем SHA являются далеко не единственными криптостойкими функциями хеширования, среди популярных альтернатив – Bcrypt, Argon2 и другие, хотя у них и есть свои особенности применения

Список литературы

1. Кристоф Добрауниг, Мария Эйхлседер, Флориан Мендель, Технологический университет Граца – Оценка безопасности SHA-224, SHA-512/224 и SHA-512/256.
2. Сомитра Кумар Санадхья, Палаш Саркар, Отдел прикладной статистики, Индийский статистический институт – Новые коллизионные атаки на 24-ступенчатый SHA-2
3. Мария Эйхлседер, Флориан Мендель, Мартин Шлаффер, Технологический университет Граца - Ветвящиеся эвристики в дифференциальном поиске коллизий с приложениями к SHA-512
4. Кристоф Добрауниг, Мария Эйхлседер, Флориан Мендель, Технологический университет Граца – Анализ SHA-512/224 и SHA-512/256.
5. Ю Сасаки, Лей Ван, Кадзумаро Аоки, Университет электрокоммуникаций - Прообразы атак на 41–ступенчатый SHA-256 и 46-ступенчатый SHA-512.
6. Т. Хансен, Internet Engineering Task Force - SHA-512 RFC.
7. Марк Стивенс, Арьен К. Ленстра, Бенне де Вегер – Предсказание победителя президентских выборов в США в 2008 году.

References

1. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, Graz University of Technology – Security Evaluation of SHA-224, SHA-512/224, and SHA-512/256
2. Somitra Kumar Sanadhya, Palash Sarkar, Applied Statistics Unit, Indian Statistical Institute – New Collision attacks Against Up To 24-step SHA-2
3. Maria Eichlseder, Florian Mendel, Martin Schlaeffer, Graz University of Technology – Branching Heuristics in Differential Collision Search with Applications to SHA-512

4. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, Graz University of Technology – Analysis of SHA-512/224 and SHA-512/256
5. Yu Sasaki, Lei Wang, Kazumaro Aoki, The University of Electro-Communications – Preimage Attacks on 41-Step SHA-256 and 46-Step SHA-512
6. T. Hansen, Internet Engineering Task Force – SHA-512 RFC.
7. Marc Stevens, Arjen K. Lenstra, Benne de Weger – Predicting the winner of the 2008 US Presidential Elections