Логический криптоанализ функции хэширования ГОСТ Р 34.11-2012

Маршалко Г.Б., эксперт ТК26
Мхитарян А.Г., инженер-аналитик, КРИПТО-ПРО

РусКрипто'2019

CRYPTOPRO

Основная идея

Сведение определенных задач криптографического анализа к задачам о выполнимости КНФ и последующем решении этих задач с помощью существующих программых реализаций алгоритмов нахождения выполняющих наборов КНФ (SAT-solvers).

CRYPTOPRO

Почему это интересно

Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, Yarik Markov, 2017, «The first collision for full SHA-1» («shattered»)

«As a result the construction of a very good and solvable non-linear differential path for the second near-collision attack turned out to be quite complex...

...Our final solution was to encode this problem into a satisfiability (SAT) problem and use a SAT solver to find a drop-in replacement differential path over the first eight steps that is solvable.»

Почему это интересно

Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, Yarik Markov, 2017, «The first collision for full SHA-1» («shattered»)

«As a result the construction of a very good and solvable non-linear differential path for the second near-collision attack turned out to be quite complex...

...Our final solution was to encode this problem into a satisfiability (SAT) problem and use a SAT solver to find a drop-in replacement differential path over the first eight steps that is solvable.»

CRYPTOPRO

Используемый подход

Р.Р. Гилязов, О.А. Логачев, С.В. Смышляев «Об особенностях логического криптоанализа хэш-функций»

CRYPTOPRO

1. Что такое логический криптоанализ?
2. Общая схема сведения задачи поиска прообраза к задаче о выполнимости КНФ
3. Построение полиномов Жегалкина
4. Построение итоговой КНФ
5. Результаты

Исходная задача

Рассмотрим задачу поиска прообраза некоторой хэш-функции $h: V_n \to V_m$ (т.е. вектора $\bar{x} \in V_n$, такого, что $h(\bar{x}) = \bar{z}$, где $z \in V_m$ - фиксированный вектор).

Почему не более простую задачу?

Задачи поиска коллизии и второго прообраза на практике приводят к более длинным КНФ с большим числом переменных, что усложняет работу SAT-решателя.

Примечание

Описанный далее способ легко обобщается на задачу нахождения коллизии и задачу нахождения второго прообраза.

Исходная задача

Рассмотрим задачу поиска прообраза некоторой хэш-функции $h: V_n \to V_m$ (т.е. вектора $\bar{x} \in V_n$, такого, что $h(\bar{x}) = \bar{z}$, где $z \in V_m$ - фиксированный вектор).

Почему не более простую задачу?

Задачи поиска коллизии и второго прообраза на практике приводят к более длинным КНФ с большим числом переменных, что усложняет работу SAT-решателя.

Примечание

Описанный далее способ легко обобщается на задачу нахождения коллизии и задачу нахождения второго прообраза.

1. Переход к задаче нахождения набора $\overline{x} = (x^{(1)}, ..., x^{(n)}) \in V_n$, такого, что

$$\&_{i=1,...,m} (h_i(\overline{x}) \oplus z_i \oplus 1) = 1,$$

где $h_i : V_n \to \{0, 1\}$ - i-ая компонента функции h, z_i - компоненты фиксированного вектора $z \in V_m$.

2. Получение для всех $i = 1, \ldots, m$ полиномов Жегалкина $A_i(\overline{x}, \overline{q})$, $\overline{q} = (q^{(1)}, \ldots, q^{(k)})$, а также для всех $j = 1, \ldots, k$ полиномов Жегалкина $Q_j(\overline{x}, \overline{q})$, таких, что для всякого вектора $\overline{x}$ все равенства $h_i(\overline{x}) = z_i$, $i = 1, \ldots, m$ выполнены тогда и только тогда, когда существует набор $\overline{q} = (q^{(1)}, \ldots, q^{(k)})$, для которого выполнено

$$\mathop{\&}_{i=1,\ldots,m} \overline{A_i}(\overline{x}, \overline{q}) =$$

$$= \mathop{\&}_{j=1,\ldots,k} Q_j(\overline{x}, \overline{q}) \mathop{\&}_{i=1,\ldots,m} (A_i(\overline{x}, \overline{q}) \oplus z_i \oplus 1) = 1$$

3. Получение для каждого $i = 1, \ldots, m$ КНФ $C_i(\bar{x}, \bar{q}, \bar{s})$ такой, что для всяких $\bar{x}, \bar{q}$, $\overline{A_i}(\bar{x}, \bar{q}) = 1$ тогда и только тогда, когда существует такой набор $\bar{s}$, что $C_i(\bar{x}, \bar{q}, \bar{s}) = 1$.
Переход к задаче выполнимости КНФ

$$C(\bar{x}, \bar{q}, \bar{s}) = \mathop{\&}_{i=1,\ldots,m} C_i(\bar{x}, \bar{q}, \bar{s}).$$

4. Передача полученной КНФ на вход SAT-решателю.

3. Получение для каждого $i = 1, \ldots, m$ КНФ $C_i(\bar{x}, \bar{q}, \bar{s})$ такой, что для всяких $\bar{x}, \bar{q}$, $\overline{A_i}(\bar{x}, \bar{q}) = 1$ тогда и только тогда, когда существует такой набор $\bar{s}$, что $C_i(\bar{x}, \bar{q}, \bar{s}) = 1$.
Переход к задаче выполнимости КНФ

$$C(\bar{x}, \bar{q}, \bar{s}) = \mathop{\&}_{i=1,\ldots,m} C_i(\bar{x}, \bar{q}, \bar{s}).$$

4. Передача полученной КНФ на вход SAT-решателю.

CRYPTOPRO

Пусть функция $h : V_n \to V_m$ представима в следующем виде:

$$h(\bar{x}) = G_r(G_{r-1}(...G_2(G_1(\bar{x}))...));$$

$$G_p : V_{d_{p-1}} \to V_{d_p},\ p = 1, 2, ..., r;\ d_0 = n, d_r = m.$$

Обозначим за **k** число вспомогательных переменных. Требуется построить для некоторого k такие полиномы Жегалкина $A_i(\bar{x}, \bar{q}),\ i = 1, ..., m$, и $Q_j(\bar{x}, \bar{q})\ , j = 1, ..., k$, что для всякого $\bar{x}$ равенство $h(\bar{x}) = (z^{(1)}, ..., z^{(m)})$ выполнено тогда и только тогда, когда существуют такие значения набора $\bar{q}$, что

$$\&_{j=1,...,k} Q_j(\bar{x}, \bar{q}) \&_{i=1,...,m} (A_i(\bar{x}, \bar{q}) \oplus z_i \oplus 1) = 1$$

1. введение новой вспомогательной переменной q^j;
2. добавление нового проверяющего полинома $Q_j = P' \oplus q^j \oplus 1$;
3. замена полинома P' на переменную q^j всюду в процессе последующих вычислений полиномов A_i.

1. введение новой вспомогательной переменной q^j;
2. добавление нового проверяющего полинома $Q_j = P' \oplus q^j \oplus 1$;
3. замена полинома P' на переменную q^j всюду в процессе последующих вычислений полиномов A_i.

1. введение новой вспомогательной переменной q^j;
2. добавление нового проверяющего полинома $Q_j = P' \oplus q^j \oplus 1$;
3. замена полинома P' на переменную q^j всюду в процессе последующих вычислений полиномов A_i.

$$h(\bar{x}) = G_r(G_{r-1}(...G_2(G_1(\bar{x}))...));$$

$$G_p : V_{d_{p-1}} \to V_{d_p},\ p = 1, 2, ..., r;\ d_0 = n, d_r = m.$$

Возможные методы:

1. Без введения вспомогательных переменных ($k = 0$);
2. Введение вспомогательных переменных на каждом шаге ($k = \sum\limits_{p=1,...,r-1} d_p$);
3. Динамически порождаемые вспомогательные переменные. Параметр метода: **TH** (дополнительное ограничение на длину порождаемых полиномов: TH^2).

$$h(\bar{x}) = G_r(G_{r-1}(...G_2(G_1(\bar{x}))...));$$

$$G_p : V_{d_{p-1}} \to V_{d_p},\ p = 1, 2, ..., r;\ d_0 = n, d_r = m.$$

Возможные методы:

1. Без введения вспомогательных переменных ($k = 0$);
2. Введение вспомогательных переменных на каждом шаге ($k = \sum_{p=1,...,r-1} d_p$);
3. Динамически порождаемые вспомогательные переменные. Параметр метода: **TH** (дополнительное ограничение на длину порождаемых полиномов: TH^2).

$$h(\bar{x}) = G_r(G_{r-1}(...G_2(G_1(\bar{x}))...));$$

$$G_p : V_{d_{p-1}} \to V_{d_p},\ p = 1, 2, ..., r;\ d_0 = n, d_r = m.$$

Возможные методы:

1. Без введения вспомогательных переменных ($k = 0$);
2. Введение вспомогательных переменных на каждом шаге ($k = \sum_{p=1,...,r-1} d_p$);
3. Динамически порождаемые вспомогательные переменные. Параметр метода: **TH** (дополнительное ограничение на длину порождаемых полиномов: TH^2).

Для преобразования получившегося таким образом выражения в КНФ достаточно преобразовать в КНФ каждый из полиномов $A_i, i = 1, 2, ..., m$ и $Q_j, j = 1, 2, ..., k$. Таким образом, задачу можно сформулировать следующим образом: Пусть дан полином P вида

$$P(x^{(1)}, ..., x^{(n)}) = M_1(x^{(1)}, ..., x^{(n)}) \oplus ... \oplus M_r(x^{(1)}, ..., x^{(n)}),$$

где все $M_i, i = 1, 2, ..., r$ - мономы, и требуется построить КНФ $C(x^{(1)}, x^{(2)}, ..., x^{(n)}, \bar{s})$ такую, что для всякого $\bar{x} = (x^{(1)}, x^{(2)}, ..., x^{(n)})$ равенство $P(\bar{x}) = 1$ выполняется тогда и только тогда, когда существует набор $\bar{s}$, такой, что $C(\bar{x}, \bar{s}) = 1$.

Задача о разбиении полинома Жегалкина

Вход: дан полином Жегалкина $P(\overline{x})$, заданный бесповторным списком мономов $\overline{P}$, дано некоторое число T, $T \leqslant n$, где n - число переменных в P.

Вопрос: существуют ли для какого-то $m \leqslant \#\overline{P}$ полиномы Жегалкина $P_i(x^{(i1)}, x^{(i2)}, ..., x^{(iT)}), i = 1, 2, ..., m$, такие, что полином $P(\overline{x})$ представим для некоторого $r \leqslant \#\overline{P}$ в следующем виде:

$$P(\overline{x}) = \bigoplus_{i=1,2,...,m} P_i(x^{(i1)}, x^{(i2)}, ..., x^{(iT)}) \bigoplus_{i=1,2,...,r} M_i(\overline{x}),$$

где $M_i(\overline{x}), i = 1, ..., r$ - мономы степени $> T$, а $\#\overline{P}$ - число мономов в $\overline{P}$.

CRYPTOPRO

Построение итоговой КНФ

1. Табличные методы построения КНФ (для полиномов, существенно зависящих от $\leqslant$ T переменных);
2. Перевод в КНФ мономов высокой степени ($>$ T);
3. Построение КНФ линейной функции.

CRYPTOPRO

Построение итоговой КНФ

1. Табличные методы построения КНФ (для полиномов, существенно зависящих от $\leqslant$ T переменных);
2. Перевод в КНФ мономов высокой степени ($>$ T);
3. Построение КНФ линейной функции.

CRYPTOPRO

Построение итоговой КНФ

1. Табличные методы построения КНФ (для полиномов, существенно зависящих от $\leqslant$ T переменных);
2. Перевод в КНФ мономов высокой степени ($>$ T);
3. Построение КНФ линейной функции.

CRYPTOPRO

1 Что такое логический криптоанализ?

2 Общая схема сведения задачи поиска прообраза к задаче о выполнимости КНФ

3 Построение полиномов Жегалкина

4 Построение итоговой КНФ

5 Результаты

CRYPTOPRO

Р.Р. Гилязов, О.А. Логачев, С.В. Смышляев «Об особенностях логического криптоанализа хэш-функций»

CubeHash (16 итераций, TH = 5)

	Длина	Кол-во переменных	Ранг
T = 12	58 250	59 138	1 979 263
T = 15	66 540	51 630	2 071 679

CubeHash, 16 итераций, TH = 5

	Длина	Кол-во переменных	Ранг
T = 12	58 250	59 138	1 979 263
T = 15	66 540	51 630	2 071 679

Стрибог, 1 раунд LPS-преобразования, TH = 5

	Длина	Кол-во переменных	Ранг
T = 12	5 823 264	46 464	65 932 128
T = 15	23 075 616	40 192	350 638 048

CubeHash, 16 итераций, TH = 5

	Длина	Кол-во переменных	Ранг
T = 12	58 250	59 138	1 979 263
T = 15	66 540	51 630	2 071 679

Стрибог, 1 раунд LPS-преобразования, TH = 5

	Длина	Кол-во переменных	Ранг
T = 12	5 823 264	46 464	65 932 128
T = 15	23 075 616	40 192	350 638 048

CubeHash, 16 итераций, TH = 5

	Длина	Кол-во переменных	Ранг
T = 12	58 250	59 138	1 979 263
T = 15	66 540	51 630	2 071 679

Стрибог, 2 раунда LPS-преобразования, TH = 5

	Длина	Кол-во переменных	Ранг
T = 12	46 410 961	261 082	501 032 405
T = 15	65 518 361	254 402	813 545 517

CRYPTOPRO

Попробуем посчитать в упрощенной модели объем памяти для записи итоговой КНФ

813 545 517 переменных $\approx$ { хранение каждой переменной + знак $\approx$ 8 байт } $\approx 6,2$ Гб памяти

Yusuf M Motara, Barry V Irwin, 2017, «SHA-1, SAT-solving, and CNF» - задача поиска прообраза 80 раундов SHA1

SHA1, 80 итераций

Кодирование	Длина	Кол-во переменных
Espresso	478 476	13 408
CryptLogVer	248 220	44 812
Simple	223 551	56 108
Hand-crafted	491 791	12 779
simplified	375 195	12 771

Yusuf M Motara, Barry V Irwin, 2017, «SHA-1, SAT-solving, and CNF» - задача поиска прообраза 80 раундов SHA1

Границы практической применимости: 80 раундов SHA1

Число свободных бит	SAT-решатель	Результат
18	Glucose	+
	Plingeling	+
	CryptoMiniSat	+
20	Glucose	+
	Plingeling	-
	CryptoMiniSat	-
22	Glucose	-
	Plingeling	-
	CryptoMiniSat	-

SHA1, 80 итераций

Кодирование	Длина	Кол-во переменных
Espresso	478 476	13 408
CryptLogVer	248 220	44 812
Simple	223 551	56 108
Hand-crafted	491 791	12 779
simplified	375 195	12 771

Стрибог, 2 раунда LPS-преобразования, TH = 5

	Длина	Кол-во переменных
T = 12	46 410 961	261 082
T = 15	65 518 361	254 402

CRYPTOPRO

Выводы

1. Для полнораундового Стрибога применение метода не имеет смысла вследствие быстрого увеличения длины итоговой КНФ и числа вспомогательных переменных;
2. Даже при значительно меньшей длине КНФ и меньшем количестве переменных SAT-решатель не справляется с задачей в случае, если число свободных бит > 20;
3. Таким образом, показано, что задача построения прообраза для хэш-функци ГОСТ Р 34.11-2012 с помощью указанного метода является невыполнимой за приемлемое время.

Выводы

1. Для полнораундового Стрибога применение метода не имеет смысла вследствие быстрого увеличения длины итоговой КНФ и числа вспомогательных переменных;
2. Даже при значительно меньшей длине КНФ и меньшем количестве переменных SAT-решатель не справляется с задачей в случае, если число свободных бит > 20;
3. Таким образом, показано, что задача построения прообраза для хэш-функци ГОСТ Р 34.11-2012 с помощью указанного метода является невыполнимой за приемлемое время.

CRYPTOPRO

Выводы

1. Для полнораундового Стрибога применение метода не имеет смысла вследствие быстрого увеличения длины итоговой КНФ и числа вспомогательных переменных;
2. Даже при значительно меньшей длине КНФ и меньшем количестве переменных SAT-решатель не справляется с задачей в случае, если число свободных бит > 20;
3. Таким образом, показано, что задача построения прообраза для хэш-функци ГОСТ Р 34.11-2012 с помощью указанного метода является невыполнимой за приемлемое время.

CRYPTOPRO

Спасибо за внимание!

Вопросы?