А. Н. Непейвода

О длинных атаках на пинг-понг протоколы

Аннотация. Ранее автором было показано, что криптографические пинг-понг протоколы в модели угрозы Долева–Яо можно верифицировать посредством моделирования в префиксных грамматиках. В статье рассматриваются примеры атак на пинг-понг протоколы, которые удовлетворяют верхней оценке на длину пути, порожденного модельной префиксной грамматикой, приведенной в предыдущих работах автора.

Ключевые слова и фразы: криптографические протоколы, вычислительная сложность, пинг-понг протоколы, модель угрозы Долева–Яо.

Введение

Один из самых частых видов угроз при передаче секретных данных по открытому каналу — это угроза типа «человек в середине»: некто выдает себя за одного из участников передачи и от его имени отправляет сообщения другим так, что они не могут установить факт подмены. Таким образом можно получить доступ к секретной информации, даже если предполагается, что шифрование абсолютно надежно, т.е. результат $f(x)$ применения шифрующего алгоритма к слову x не несет никакой информации о самом слове x . Чтобы избежать подобных угроз, были разработаны криптографические протоколы. Криптографический протокол — это описание правил поведения пользователей при обмене сообщениями. Как правило, он составлен так, чтобы злоумышленнику было как можно труднее выдать себя за правомерного пользователя протокола. Далее мы рассматриваем лишь один класс криптографических протоколов, называемый пинг-понг протоколы.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта №14-07-00133_а и госзадания ФАНО России — НИР «Методы анализа и верификации моделей вычислительных систем и алгебраических объектов на основе средств функционального и логического программирования» (№ г/р 01201354590).

© А. Н. Непейвода, 2015

© Институт программных систем имени А. К. Айламазяна РАН, 2015

© Программные системы: теория и приложения, 2015

Пинг-понг протокол требует, чтобы передаваемое секретное сообщение было представлено единственной строкой, изменяемой посредством конечного множества операций. Участники протокола по очереди применяют операции этого множества к сообщению и пересылают друг другу результат, как будто передавая мяч при игре в пинг-понг. Допустим, что нарушитель может перехватывать все сообщения, которыми обмениваются данные пользователи, и подменять эти сообщения на другие, полученные либо применением некоторого конечного множества операций, доступных нарушителю, либо подстановкой вместо исходного сообщения другого, порожденного в ходе использования того же самого протокола теми же участниками (эта модель угрозы называется моделью угрозы Долева–Яо [1]). Долев, Ивен и Карп показали, что вопрос о безопасности пинг-понг протокола в такой модели угрозы становится алгоритмически разрешим [2].

Пинг-понг протокол можно промоделировать префиксной грамматикой [3], после чего задача его верификации сводится к проблеме пустого слова в языке, порожденном модельной грамматикой. В работе [4] была приведена оценка на длину кратчайшего пути, порожденного модельной префиксной грамматикой, заканчивающегося пустым словом (далее пустое слово всюду обозначаем Λ), в худшем случае. Оставался вопрос, существует ли протокол, атаки на который имели бы длину, согласованную с этой оценкой. В данной статье приведены несколько вариантов таких протоколов.

1. Понятие пинг-понг протокола и атаки на него

Словарем пользователя U назовем множество всех доступных для него действий, которые могут представлять собой как единичные операторы преобразования сообщений, так и их композиции. Далее обозначаем это множество как Σ_U . Прописные латинские буквы, выделенные полужирным шрифтом, в дальнейшем выступают как обозначения имен пользователей либо протоколов. Прописные латинские буквы, не выделенные полужирным — как обозначения операторов преобразования сообщений.

Здесь и далее считаем, что любые два криптографических оператора G, F не удовлетворяют никаким соотношениям, кроме, возможно, соотношений вида $G \circ F = \Lambda$. Далее всюду мы опускаем знак операции композиции.

Укажем обозначения для часто используемых криптографических операторов, которые могут быть в Σ_U , и опишем их формальные свойства.

- (1) Оператор шифрования открытым ключом E_X . Операция $E_X(M)$ зашифровывает M ключом пользователя X .
- (2) Оператор расшифровки D_X . Операция $D_X(M)$ расшифровывает M ключом пользователя X . Оператор D_X обратен E_X : выполнено как $D_X E_X = \Lambda$, так и $E_X D_X = \Lambda$ — допустимо как расшифровывать уже наложенный шифр, так и предотвращать шифрование.
- (3) Оператор приписывания имени a_X . Операция $a_X(M)$ приписывает имя пользователя X в начало сообщения M .
- (4) Оператор удаления имени d_X . Операция $d_X(M)$ стирает префикс M , соответствующий имени пользователя X . d_X является левым обратным для a_X , так что выполнено $d_X a_X = \Lambda$, но $d_X a_X \neq \Lambda$.

E_X , a_X , d_X — открытые, доступные всем пользователям сети операторы, а D_X — секретный, доступный лишь пользователю X . В примере 3 также используются операторы F_X и G_X , не являющиеся ни открытыми, ни секретными в обычном смысле.

ОПРЕДЕЛЕНИЕ 1. *Двухсторонний пинг-понг протокол* $P[U_1, U_2]$ — это кортеж пар $(U_k, \alpha_i[U_1, U_2])$, где $k = 1$, если i нечетно, и $k = 2$ в противном случае, и $\alpha_i[U_1, U_2]$ — последовательность операторов из словаря U_k .

Пусть U_1, U_2 — легальные пользователи сети, I — злоумышленник. Протокол $P[U_1, U_2]$ *ненадежен*, если существует последовательность строк $\beta_1, \dots, \beta_k$ такая, что

- (1) $\beta_k \dots \beta_1 \alpha_1[U_1, U_2] = \Lambda$.
- (2) Выполнено одно из двух условий:
 - (a) $\beta_i \in \Sigma_I$, или
 - (b) $\beta_i = \alpha_j[X, Y]$, $X \in \{U_1, U_2, I\}$, $Y \in \{U_1, U_2, I\}$, $X \neq Y$, $j \equiv 1 \pmod{2} \Rightarrow \beta_i \in \Sigma_X$, $j \equiv 0 \pmod{2} \Rightarrow \beta_i \in \Sigma_Y$.

Здесь первое условие соответствует применению злоумышленником какого-либо действия из своего словаря к перехваченной строке операторов. Второе действие соответствует применению пользователями X и Y j -ого правила протокола $P[U_1, U_2]$.

Рис. 1. Легальное взаимодействие по протоколу $P_{Double}[X, Y]$ Рис. 2. Схема взлома протокола $P_{Double}[X, Y]$

ПРИМЕР 1. Обратимся к протоколу $P_{Double}[X, Y]$, описанному в [2]:

$$P_{Double}[X, Y] = ((X, E_Y a_X E_Y), (Y, E_X D_Y d_X D_Y)).$$

Взаимодействие двух легальных пользователей по $P_{Double}[X, Y]$ изображено на рис. 1. $P_{Double}[X, Y]$ ненадежен. Одна из атак на него показана на рис. 2. В ней участвуют легальные пользователи **A** и **B** и злоумышленник **I**. Злоумышленник перехватывает $E_B a_A E_B(M)$, приписывает к нему свое имя и шифрует опять ключом E_B . **B** считает пересланное сообщение продолжением некоего правомерного взаимодействия с **I** по тому же протоколу и отправляет ему $E_I a_A E_B(M)$. **I** расшифровывает сообщение ключом D_I , стирает a_A и повторяет свои предыдущие действия. После ответа **B** злоумышленник **I** получает $E_I(M)$ и читает M .

Рассмотрим другую последовательность действий **A**, **B** и **I**, изображенную на рис. 3. И опять злоумышленник получает доступ к M . Но теперь после перехвата $E_A(M)$ нарушитель **I** начинает протокольное взаимодействие с **A**.

Видно, что один и тот же протокол может содержать целый ряд уязвимостей. Возникает вопрос: можно ли их обнаружить переборным поиском атак вплоть до некоторой (разумной) величины?

Рис. 3. Еще одна схема взлома протокола $P_{Double}[\mathbf{X}, \mathbf{Y}]$

2. Простейший пример протокола с экспоненциально длинной атакой

Верхняя оценка на длину кратчайшей атаки на пинг-понг протокол приведена в [4] — она экспоненциальна от числа правил переписывания в префиксной грамматике, моделирующей протокол. Там же приведен класс грамматик, для которых эта оценка оказывается точной. Однако остается вопрос, сохранится ли эта оценка при переходе от модели префиксных грамматик к классическим моделям пинг-понг протоколов? Здесь важно понять, можно ли при построении «экспоненциально трудных» грамматик ограничиться стандартными криптографическими операторами, поскольку свойства этих операторов налагают ограничения на правила грамматики, моделирующей их применения.

Мы опишем класс двухсторонних пинг-понг протоколов, атаки на которые также окажутся экспоненциально длинными от числа правил в протоколах (а значит, и в соответствующих префиксных грамматиках).

ПРИМЕР 2. Допустим, каждый пользователь некоторой сети имеет не один ключ шифрования, а набор таких ключей $\{E_X^1, E_X^2, \dots, E_X^N\}$. Все ключи являются открытыми и доступны любому пользователю сети. D_X^i — это операторы дешифрования ключей E_X^i , доступные лишь пользователю $\mathbf{X}$. Тогда система протоколов с экспоненциальным разрастанием длины атак будет выглядеть как

$$\mathbf{P}_{\langle n \rangle \text{EXP}}[\mathbf{X}, \mathbf{Y}] = ((\mathbf{X}, \alpha_1[\mathbf{X}, \mathbf{Y}]), \dots, (\mathbf{U}, \alpha_n[\mathbf{X}, \mathbf{Y}])),$$

где $\mathbf{U} = \mathbf{X}$, если n нечетно, и $\mathbf{U} = \mathbf{Y}$ в противном случае, а также

$$\begin{aligned}
\alpha_1[\mathbf{X}, \mathbf{Y}] &= E_{\mathbf{Y}}^1; \\
\alpha_{2k}[\mathbf{X}, \mathbf{Y}] &= E_{\mathbf{Y}}^{2k} E_{\mathbf{Y}}^{2k} D_{\mathbf{Y}}^{2k-1}, \text{ если } 2k \neq n; \\
\alpha_{2k}[\mathbf{X}, \mathbf{Y}] &= D_{\mathbf{Y}}^{2k-1}, \text{ если } 2k = n; \\
\alpha_{2k+1}[\mathbf{X}, \mathbf{Y}] &= E_{\mathbf{X}}^{2k+1} E_{\mathbf{X}}^{2k+1} D_{\mathbf{X}}^{2k}, \text{ если } 2k+1 \neq n; \\
\alpha_{2k+1}[\mathbf{X}, \mathbf{Y}] &= D_{\mathbf{X}}^{2k}, \text{ если } 2k+1 = n.
\end{aligned}$$

Чтобы расшифровать $E_{\mathbf{X}}^i$ ($\mathbf{X} \neq \mathbf{I}$, $i \neq n$), злоумышленник должен дважды расшифровать $E_{\mathbf{X}}^{i+1}$, и так далее, пока не дойдет до $E_{\mathbf{X}}^n$.

Из примера 2 видно, что протоколы с экспоненциальной нижней оценкой на длину атаки теоретически существуют. Однако протоколы, подобные $\mathbf{P}_{\langle n \rangle \text{EXP}}[\mathbf{X}, \mathbf{Y}]$, практически вряд ли будут использоваться. Для того, чтобы возникло экспоненциальное разрастание длины атаки, необходимо, чтобы участники протокола использовали много различных ключей для одного и того же взаимодействия.

Заметим также, что слово протокола $\alpha_n[\mathbf{X}, \mathbf{Y}]$, представляющее собой простую расшифровку оператора $E_{\mathbf{X}}^n$, на первый взгляд выглядит иррационально — это открытое применение дешифрующего оператора без каких-либо предварительных проверок. Но $\alpha_n[\mathbf{X}, \mathbf{Y}]$ можно считать метапредставлением некоторой атаки на $E_{\mathbf{X}}^n$, не включающей операции с прочими ключами $E_{\mathbf{X}}^i$.

Далее мы покажем, какие практически возможные ситуации могут приводить к появлению экспоненциально длинных атак.

3. Атака на протокол как путь обхода вершин n -мерного куба

Ниже приведен пример разумно выглядящего протокола, атака на который представляет собой алгоритм решения задачи о Ханойской башне. Этот протокол может дать подсказку, какого рода ситуации могут привести к возникновению подобных длинных атак.

ПРИМЕР 3. Предположим, что на некотором сайте администраторы ввели собственную систему шифрования: прежде зашифровки обычным открытым ключом, она применяет к сообщению некоторый кодовый шифр пользователя $F_{\mathbf{X}}$, являющийся подтверждением того, что пересылающий сообщение пользователь $\mathbf{X}$ является легальным пользователем сайта. Кодовый шифр пользователя ему неизвестен, но система шифрования сайта сама по себе открытая, то есть композиция $E_{\mathbf{X}} F_{\mathbf{X}}$ доступна всем. Оператор, обратный $F_{\mathbf{X}}$, обозначим $G_{\mathbf{X}}$; считаем, что он односторонний, то есть $F_{\mathbf{X}} G_{\mathbf{X}} \neq \Lambda$, $G_{\mathbf{X}} F_{\mathbf{X}} = \Lambda$. Ни одному обычному пользователю сайта операторы $G_{\mathbf{X}}$ не доступны.

Рис. 4. Схема протокола P_{LA}

Чтобы послать администратору **B** запрос M о смене логина, пользователь **X** должен зашифровать его обычным ключом администратора E_B , а затем собственным ключом пользователя F_X . Это можно сделать, применив открытую систему шифрования сайта, выражающуюся композицией $E_X F_X$, а затем дешифровав собственный открытый ключ. Таким образом **X** подтверждает, что сообщение послано именно от него. Администратор, зная, кому принадлежит какой кодовый шифр, расшифровывает $F_X E_B$ и возвращает пользователю его исходное сообщение M , зашифрованное его открытым ключом E_X . Имея D_X , пользователь **X** может убедиться, что ему пришло то сообщение, которое он посылал.

Формально протокол имеет вид

$$P_{LA}[X, Y] = ((X, F_X E_Y), (Y, E_X D_Y G_X)).$$

Обратим внимание, что если в роли **X** может выступать любой пользователь, то в роли **Y** — лишь тот, в чьем словаре есть G_X (в нашем случае это администратор **B**). Схема протокола изображена на рис. 4.

Представим себе, что помимо администратора, на сайте есть еще и программист **A**. С обычными пользователями он не общается, а с администратором взаимодействует по вышеописанной схеме, причем имеет возможность проверить кодовый шифр администратора (то есть владеет оператором G_B). В некоторый момент нарушителю **I** удалось перехватить сообщение $E_A(M)$, соответствующее реплике **B** на сообщение **A** вида $F_A E_B(M)$, в котором предположительно содержится база данных паролей. Сумеет ли **I** взломать это сообщение?

Сообщение $E_B(M)$ злоумышленнику взломать нетрудно: достаточно подписать к нему свой код F_I и послать **B** под видом собственного запроса. Получив реплику $E_I(M)$, **I** может прочитать M .

Но с $E_A(M)$ эта схема не сработает — **A** не знает G_I . Злоумышленник будет вынужден действовать через **B** по схеме, указанной на рис. 5. Эта схема соответствует схеме решения задачи о Ханойской башне,

Рис. 5. Схема взлома протокола P_{LA} Рис. 6. Схема взлома протокола P_{LA} как схема обхода вершин куба

которую можно представить себе как схему обхода вершин трехмерного куба, если сопоставить именам пользователей-адресатов **I**, **A** и **B** перемещения по координатам y , z и x (см. рис. 6, на котором двойной чертой выделен путь обхода).

Если имеет место взаимодействие четырех участников — **I**, **A**, **B** и **C**, — где **A** взаимодействует лишь с **C**, а **C** — лишь с **B**, то, чтобы расшифровать $E_A(M)$, злоумышленнику придется дважды обращаться к схеме рис. 5 (но теперь в роли **A** в ней выступает **C**): сначала — чтобы получить $F_C(M)E_A(M)$ из $E_C F_C E_A(M)$, а затем — чтобы расшифровать $E_C(M)$. Здесь происходит то же самое, что и при решении задачи о Ханойской башне: чтобы переложить на соседний штырь

пирамиду высоты 4, приходится перекладывать сначала пирамиду высоты 3 (расшифровать E_C), затем — самый большой нижний диск (от имени **C** провоцировать **A** на реплику $E_C(M)$), и затем уже на него опять перекладывать пирамиду высоты 3. Добавление любого очередного участника такой цепочки удваивает количество действий, которые необходимо произвести злоумышленнику.

4. Заключение

Заодно с ответом на вопрос, возможны ли экспоненциально длинные атаки на простые протоколы, из последнего примера можно получить представление о вероятном источнике таких атак. А именно, если в цепочке доверенных пользователей канала есть хотя бы одно слабое звено (в нашем случае — администратор **B**), пострадать от его беспечности может каждый пользователь, даже такой, который не взаимодействует ни с какими пользователями сети, кроме тех, в ком уверен. Косвенным показателем того факта, что идет процесс взлома, для такого осторожного участника сети могут служить странные (например, нечитаемые) или повторные сообщения от его друзей. Будет уместно игнорировать такие сообщения, чтобы случайно не дать возможность взломать себя.

Примеры, приведенные в статье, свидетельствуют, что переборный способ поиска атаки на протокол по ее длине (например, как в [5]) ненадежен в случае, если канал связи рассчитан на много пользователей. Не удастся разумным образом ввести ограничение на длину атаки даже на протокол всего с двумя правилами, например, если он используется в большой социальной сети.

Список литературы

- [1] D. Dolev, A. C. Yao. “On the security of public key protocols”, *Transactions on Information Theory*, **29** (1983), pp. 198–208. ^{↑342}
- [2] D. Dolev, S. Even, R. M. Karp. “On the security of ping-pong protocols”, *Information and Control*, **55** (1982), pp. 57–68. ^{↑342,344}
- [3] A. Nepeivoda, “Ping-Pong protocols as prefix grammars and Turchin’s relation”, VPT 2013. First International Workshop on Verification and Program Transformation (July 13th and 14th, 2013, Saint Petersburg, Russia), EPIc Series, vol. **16**, eds. A. Lisitsa, A. Nemytykh, EasyChair, 2013, pp. 74–87. ^{↑342}

- [4] A. Nepeivoda, “Verification of Multi-Party Ping-Pong Protocols via Program Transformation”, VPT 2014. Second International Workshop on Verification and Program Transformation (July 17th and 18th, 2014, Vienna, Austria), EPiC Series, vol. **28**, EasyChair, 2014, pp. 84–98. [↑][342,345](#)
- [5] A. Ahmed, A. Lisitsa, A. Nemytykh, “Cryptographic Protocol Verification via Supercompilation (A Case Study)”, VPT 2013. First International Workshop on Verification and Program Transformation (July 13th and 14th, 2013, Saint Petersburg, Russia), EPiC Series, vol. **16**, eds. A. Lisitsa, A. Nemytykh, EasyChair, 2013, pp. 16–29. [↑][349](#)

Рекомендовал к публикации

к.т.н. С. М. Пономарёва

Об авторе:

Антонина Николаевна Непейвода

Младший научный сотрудник. Работает в сфере суперкомпиляции, частичных вычислений и вычислительной сложности процессов преобразования программ

e-mail:

a_nevod@mail.ru

Пример ссылки на эту публикацию:

А. Н. Непейвода. «О длинных атаках на пинг-понг протоколы», *Программные системы: теория и приложения*, 2015, **6**:4(27), с. 341–351.

URL:

http://psta.psiras.ru/read/psta2015_4_341-351.pdf

Antonina Nepeivoda. *A remark about long attacks on ping-pong protocols.*

ABSTRACT. The paper presents some long attacks on cryptographic ping-pong protocols in the Dolev–Yao intruder model. The length of these attacks reaches the upper bound on the attack length proven in the previous works of the author. (*In Russian*).

Key Words and Phrases: Computational complexity, ping-pong protocols, protocol security, Dolev–Yao intruder model.

References

- [1] D. Dolev, A. C. Yao. “On the security of public key protocols”, *Transactions on Information Theory*, **29** (1983), pp. 198–208.
- [2] D. Dolev, S. Even, R. M. Karp. “On the security of ping-pong protocols”, *Information and Control*, **55** (1982), pp. 57–68.
- [3] A. Nepeivoda, “Ping-Pong protocols as prefix grammars and Turchin’s relation”, VPT 2013. First International Workshop on Verification and Program Transformation (July 13th and 14th, 2013, Saint Petersburg, Russia), EPiC Series, vol. **16**, eds. A. Lisitsa, A. Nemytykh, EasyChair, 2013, pp. 74–87.
- [4] A. Nepeivoda, “Verification of Multi-Party Ping-Pong Protocols via Program Transformation”, VPT 2014. Second International Workshop on Verification and Program Transformation (July 17th and 18th, 2014, Vienna, Austria), EPiC Series, vol. **28**, EasyChair, 2014, pp. 84–98.
- [5] A. Ahmed, A. Lisitsa, A. Nemytykh, “Cryptographic Protocol Verification via Supercompilation (A Case Study)”, VPT 2013. First International Workshop on Verification and Program Transformation (July 13th and 14th, 2013, Saint Petersburg, Russia), EPiC Series, vol. **16**, eds. A. Lisitsa, A. Nemytykh, EasyChair, 2013, pp. 16–29.

Sample citation of this publication:

Antonina Nepeivoda. “A remark about long attacks on ping-pong protocols”, *Program systems: theory and applications*, 2015, **6**:4(27), pp. 341–351. (*In Russian*). URL: http://psta.psir.ru/read/psta2015_4_341-351.pdf