УДК 004.45 **ТЕХНИЧЕСКИЕ НАУКИ**

Стрелец Андрей Иванович, *магистр*
кафедры «Компьютерные системы и технологии»,
Национальный исследовательский ядерный университет «МИФИ»
Россия, г. Москва
Морочев Георгий Михайлович, *магистр*
кафедра «Компьютерные системы и технологии»,
Национальный исследовательский ядерный университет «МИФИ»
Россия, г. Москва
Дороничев Никита Андреевич, *магистр*
кафедры «Компьютерные системы и технологии»,
Национальный исследовательский ядерный университет «МИФИ»
Россия, г. Москва
Сычев Максим Игоревич, *магистр*
кафедры «Компьютерные системы и технологии»,
Национальный исследовательский ядерный университет «МИФИ»
Россия, г. Москва

АЛГОРИТМ ПРОВЕРКИ ЦЕЛОСТНОСТИ ОДНОРАЗОВОЙ ПРОГРАММЫ

Аннотация: Данная статья описывает ключевые моменты алгоритма проверки целостности одноразовой программы. Этот алгоритм используется для верификации и выполнения одноразовой программы в защищенной среде.

Ключевые слова: одноразовые программы, информационная безопасность, разработка, верификация.

Abstract: This article is about most important parts of algorithm of one-time program consistency. This algorithm is often used to verify and control the one-time program environment.

Key words: one-time program, media-content, information security, development, verification.

Введение

Одной из ключевых концепция в области производства защищенного программного обеспечения для банковского сектора и областей с повышенными требованиями к информационной безопасности является концепция программ одноразового выполнения (one-time program) [1]. В основе данных программ лежит программно-аппаратный комплекс, обеспечивающий выполнение программы заданное число раз [2]. При этом данный комплекс предназначен для недопущения несанкционированного копирования и воспроизведения целевой программы [3].

Для корректной работы данной программы используются специализированное программное обеспечение, работающее в соответствии с алгоритмом, предназначенным для поддержания консистентности среды выполнения одноразовой программы [4]. В данной статье рассмотрены ключевые моменты данного алгоритма.

Алгоритм проверки целостности одноразовой программы

Система одноразового доказательства для NP языка L состоит из трех объектов: (i) владелец свидетеля, имеющий свидетельство о принадлежности какого-либо элемента X к языку L. (ii) проверяющий и (iii) верификатор где проверяющий и верификатор знают ввод X, но не знают свидетеля членства X в L. Система одноразового доказательства позволяет владельцу свидетеля (эффективно) преобразовать своего NP-свидетеля в аппаратный токен доказательства. Токен доказательства может позже использоваться эффективным проверяющим (который не знает свидетеля), чтобы убедить верификатора ровно один раз, что вход X находится в языке. Предполагается,

что владелец свидетеля и проверяющий являются «честными» и следуют предписанным протоколам, тогда как проверяющий может быть злонамеренным и произвольно отклоняться.

При однократном доказательстве средство проверки убеждает верификатора с помощью стандартной интерактивной системы проверки. В частности, верификатору не требуется физический доступ к токену проверки (этот доступ нужен только проверяющему). После запуска интерактивного доказательства и убедительного (честного) верификатора один раз токен доказательства становится бесполезным и не может использоваться снова. Дело в том, что (i) владелец свидетеля не должен участвовать в доказательстве, помимо предоставления токена (следовательно, система доказательства отключена), и (ii) проверяющий, даже если он убеждает верификатора, ничего не узнает от взаимодействия с оборудованием, и, в частности, не может убедить проверяющего во второй раз.

Таким образом, для любого заявления N P одноразовые доказательства позволяют владельцу свидетеля давать другим сторонам возможность доказать контролируемое утверждение, не раскрывая им свидетеля. Система одноразовой проверки дает эту «однократную гарантию проверки», а также более стандартные гарантии полноты и надежности и гарантию нулевого знания, утверждая, что все, что можно извлечь из маркера проверки, может быть изучено без него. Наконец, пользователь, который хочет использовать одноразовый токен доказательства, чтобы убедить себя, что x ∈L может сделать это без какого-либо взаимодействия, запустив интерактивное доказательство в своей голове (в этом случае проверяющий и верификатор – это одна и та же сущность). Обратите внимание, что проверяющий, который каким-то образом знает принадлежность x, может убедить проверяющего столько раз, сколько ему нужно. Одноразовое доказательство захватывает это требование, требуя, чтобы любой проверяющий, который может использовать один токен доказательства, чтобы убедить проверяющего более одного раза, должен фактически знать свидетеля о принадлежности X к языку.

Формально свидетель может быть извлечен из проверяющего за полиномиальное время. В частности, это означает, что если проверяющий может убедить верификатор более одного раза, используя один проверочный токен, и тогда проверяющий может убедить верификатора столько раз, сколько ему нужно, даже не увидев проверочный токен. Другими словами, проверочный токен не помогает проверяющему доказать утверждение более одного раза. Еще одна естественная ситуация, когда появляются единовременные доказательства, — это системы голосования, цель которых состоит в том, чтобы избиратели могли голосовать только один раз. В этом случае каждому избирателю будет предоставлено одноразовое доказательство наличия права голоса (конечно, необходимо также убедиться, что жетоны подтверждения не могут быть переданы от одного избирателя к другому). Аналогичное применение одноразовых доказательств относится к электронным жетонам метрополитена. Здесь оператор метрополитена хочет продать пассажирам электронные жетоны метрополитена, где жетоны должны быть проверены турникетами станции метро. Пассажир должен иметь возможность использовать жетон только один раз, чтобы войти в Метро, и после этого он становится бесполезным. Эта цель легко реализуется естественным путем с помощью одноразовых доказательств. Оператор метро генерирует жесткий криптографический экземпляр, скажем, продукт в n=pq из двух простых чисел. Жетоны метро являются одноразовыми доказательствами для доказательства того, что n на самом деле является произведением двух больших простых чисел. Пассажиры играют роль свидетеля. Турникеты являются проверяющим и позволяют только проверяющим, которые могут доказать им, что n - произведение двух простых чисел на станцию метро. Любой пассажир, который может использовать один жетон, чтобы получить вход более одного раза, также может быть использован для поиска свидетеля или факторизации n, задача, которую мы считаем невозможной для эффективных пассажиров.

В более общем смысле, одноразовые доказательства можно рассматривать как естественное обобщение проблем контроля доступа с

ограниченным количеством. В частности, мы можем конвертировать любую 3-раундовую идентификационную схему (или любой Х-протокол) в одноразовое подтверждение личности. Одноразовые доказательства отличаются от не интерактивных доказательств с нулевым знанием (NIZK). В обоих случаях владелец свидетеля не должен присутствовать при проверке доказательства. Однако в системах доказательства NIZK либо доказательство может быть проверено произвольными верификаторами неограниченное количество раз, и, в частности, оно также не может быть отклонено (например, системы доказательства NIZK в CRS-подобной модели), или доказательства должны быть адаптированы к конкретному верификатору и бесполезны для других верификаторов (например, системы NIZK-доказательств в модели предварительной обработки). С другой стороны, одноразовые доказательства с нулевым знанием могут быть проверены только один раз, но любым пользователем, и позже могут быть отклонены. Они также не нуждаются в надежной установке, инфраструктуре открытых ключей или предварительной обработке, но, с другой стороны, они используют защищенное аппаратное обеспечение. Любое утверждение N P имеет эффективное одноразовое доказательство с использованием ОТМ. Необходимо преодолеть небольшие проблем устойчивости, возникающие при параллельной композиции доказательства с нулевым знанием (но в не интерактивной среде). Это достигается с помощью безопасного аппаратного обеспечения, позволяющего использовать чувствительный аргумент моделирования. Хотя одноразовый компилятор общего назначения из предыдущего раздела можно использовать для создания одноразового доказательства, это приводит к значительно менее эффективным (и не интуитивно привлекательным) схемам.

Пусть K является параметром безопасности и k - параметром надежности. Предположим, что существует односторонняя перестановка на K-битных входах. Каждый NP язык L имеет одноразовое доказательство с нулевым знанием, с совершенной полнотой и достоверностью 2-k. Токен доказательства

использует k OTM (каждый из которых имеет размер poly (n,k,K), где n - длина ввода).

Одноразовое доказательство построено для NP-полного языка Гамильтонов граф, из которого можно получить одноразовое доказательство для любого языка NP. Входные данные представляют собой граф G = (V,E), у производителя есть свидетель w, описывающий гамильтонов цикл в графе. Единовременное доказательство использует k OTM, чтобы получить доказательство с достоверностью 2-k (и совершенной полнотой). Основная идея доказательства с нулевым знанием состоит в том, чтобы проверяющий мог совершить случайную перестановку графа и отправить это обязательство верификатору. Затем верификатор может спросить проверяющего, отправлять ли ему перестановку и все декомпозиции (открытия всех обязательств), или посылать отмены обязательств к гамильтонову циклу в переставленном графе с достоверностью 1/2. Естественный подход к нашей настройке заключается в том, что владелец-свидетель должен сгенерировать пробный токен, в котором в качестве компонента программного обеспечения используется выделенный пермутированный граф. Пробный токен также включает в себя OTM, первым секретом которого является перестановка и все изъятия, а также второй секрет - разложение в гамильтонов цикл в перестановочном графе (ответ на второй запрос верификатора). Это действительно дает (простое) одноразовое доказательство с помощью достоверности 1/2 через стандартные аргументы: единственное, что проверяющий может извлечь из токена является одной из двух возможных последовательностей отмены обязательств, и мы знаем, что проверяющий может сгенерировать это самостоятельно. С другой стороны, как это ни парадоксально, этот токен доказательства позволяет проверяющему убедить верификатора в том, что граф имеет гамильтонов цикл в интерактивном доказательстве с совершенной полнотой и достоверностью 1/2.

Чтобы усилить устойчивость до 2-k на первый взгляд эффективная идея состоит в том, чтобы продюсер создал k таких подтвержденных графов и k таких соответствующих OTM, каждый из которых содержит пару секретов,

соответствующих новой приверженности случайной перестановке своего графа. Эта идея, однако, проблематична. Сложность состоит в том, что имитация одноразового доказательства становится такой же сложной, как симуляция параллельного выполнения протокола нулевого знания Блюма.

Заключение

В данной статье рассмотрен алгоритм проверки целостности одноразовой программы. Данный алогоритм является ключевым компонентом комплексной системы, обеспечивающей консистентность защищённой среды выполнения одноразовой программы. Алгоритм проверки целостности программы может быть использован для реализаций систем, обеспечивающих одноразовое выполнение программ.

Библиографический список:

1. Артем Генкин, Алексей Михеев. Блокчейн. Как это работает и что ждет нас завтра. Мю: Альпина Паблишер, – 2017. – 592 с.
2. Джон Хопкрофт, Раджив Мотвани, Джеффри Ульман. Введение в теорию автоматов, языков и вычислений. М.: Вильямс, – 2002. – 57 с.
3. Чернодуб А. Н., Дзюба Д. А. Обзор методов нейроуправления // Проблемы программирования. — 2011. — No 2. — С. 79—94.