УДК 004.056

АУТЕНТИФИКАЦИЯ В МНОГОУРОВНЕВОЙ СИСТЕМЕ НА ОСНОВЕ КОММУТАТИВНОГО ШИФРОВАНИЯ[1]

П. А. Паутов

Рассмотрим систему, состоящую из трёх взаимодействующих подсистем: клиент, внешний сервер, внутренний сервер. Клиент взаимодействует только с внешним сервером, внешний сервер взаимодействует как с клиентом, так и с внутренним сервером (внешний сервер является клиентом внутреннего сервера). Внешний сервер взаимодействует с внутренним только для обработки запросов своих клиентов.

В таких многоуровневых системах обычно используется одна из двух следующих моделей организации аутентификации [1]:

1) модель делегирования;
2) модель доверенной подсистемы.

В модели делегирования внешний сервер взаимодействует с внутренним от имени клиента, т. е. внутренний сервер содержит учётную запись для каждого клиента внешнего сервера. В модели доверенной подсистемы внешний сервер взаимодействует с внутренним от имени фиксированного набора учётных записей, т. е. одна учётная запись внутреннего сервера соответствует нескольким клиентам внешнего сервера. В данной работе рассматривается модель доверенной подсистемы.

В модели доверенной подсистемы для взаимодействия с внутренним сервером используется учётная запись, соответствующая привилегиям клиента внешнего сервера. Например, на внешнем сервере клиенты делятся на группы по привилегиям: «гости», «операторы», «администраторы». Тогда для взаимодействия с внутренним сервером можно использовать три учётные записи, соответствующие группам клиентов. Если внешний сервер сам выбирает учётную запись для взаимодействия с внутренним сервером, то в случае компрометации первого злоумышленник сможет использовать учётную запись с максимальными привилегиями. Возникает задача разработки такой схемы аутентификации, при которой внешний сервер смог бы использовать для взаимодействия с внутренним сервером только ту учётную запись, которая соответствует клиенту, и только тогда, когда клиент взаимодействует с внешним сервером. То есть если клиент относится к группе «гости», то внешний сервер может пройти аутентификацию перед внутренним сервером только от имени учётной записи «гость». И внешний сервер не может пройти аутентификацию перед внутренним сервером от имени учётной записи «гость» без помощи клиента, относящегося к группе «гости».

В работе автора [2] предложено несколько схем для случая, когда между клиентом и внешним сервером и между внешним сервером и внутренним используется парольная аутентификация. При использовании парольной аутентификации внешнему серверу необходим пароль учётной записи внутреннего сервера, для того чтобы пройти аутентификацию от имени данной учётной записи. То есть если злоумышленник скомпрометирует внешний сервер так, что он получит доступ на чтение к памяти внешнего сервера, то когда клиент инициирует выполнение протокола аутентификации, описанного в [2], злоумышленник получит пароль учётной записи внутреннего сервера, соответствующей привилегиям клиента. После одного сеанса связи клиента

[1]Работа выполнена в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. (гос. контракт № П1010).

со скомпрометированным внешним сервером злоумышленник получает возможность использовать внутренний сервер без помощи клиента.

В данной работе предлагается схема аутентификации, при использовании которой, даже после сеанса связи клиента со скомпрометированным внешним сервером, злоумышленник не сможет использовать внутренний сервер без помощи клиента. Для её построения необходимы следующие криптографические примитивы:

1) E — коммутативный алгоритм шифрования, D — соответствующий алгоритм расшифрования;
2) H — хэш-функция.

Каждой учётной записи внутреннего сервера ставится в соответствие число S. Каждому клиенту внешнего сервера ставится в соответствие ключ K алгоритма E. На внешнем сервере для каждого клиента хранится результат шифрования $E_K(S)$, где S соответствует учётной записи внутреннего сервера для данного клиента. Алгоритм аутентификации клиента будет выглядеть следующим образом:

1. Клиент посылает внешнему серверу своё имя.
2. Внешний сервер посылает внутреннему серверу имя учётной записи, соответствующей клиенту.
3. Внутренний сервер генерирует случайный ключ шифрования K_S алгоритма E и передаёт его внешнему серверу.
4. Внешний сервер находит запись $E_K(S)$, соответствующую данному клиенту, вычисляет $E_{K_S}(E_K(S))$ и передаёт полученный результат клиенту.
5. Клиент вычисляет $H(D_K(E_{K_S}(E_K(S)))) = H(E_{K_S}(S))$ и передаёт данное значение внешнему серверу.
6. Внешний сервер передаёт полученное значение внутреннему серверу.
7. Внутренний сервер вычисляет $H(E_{K_S}(S))$ и сравнивает результат со значением, полученным от внешнего сервера. Если значения совпадают, то клиент прошёл аутентификацию перед внешним сервером, а внешний сервер перед внутренним.

Используемый коммутативный алгоритм может быть как симметричным, так и асимметричным. В асимметричном варианте каждому клиенту ставится в соответствие пара ключей: открытый K_e и закрытый K_d. На внешнем сервере для каждого клиента хранится значение $E_{K_e}(S)$ и открытый ключ клиента K_e. Алгоритм аутентификации аналогичен симметричному варианту (но на 3-м шаге внутреннему серверу достаточно сгенерировать только открытый ключ).

Для полноты схемы нужно построить алгоритмы операций по управлению пользователями. Рассматриваются следующие операции:

1) создание нового пользователя;
2) смена ключа пользователем;
3) изменение учётной записи внутреннего сервера;
4) смена учётной записи внутреннего сервера для данного пользователя системы.

Алгоритмы этих операций строятся по аналогии с [2]. В симметричном варианте для выполнения операций 3–4 необходим ключ пользователя.

ЛИТЕРАТУРА

1. *Chong F.* Trusted Subsystem Design // MSDN. 2006. `http://msdn.microsoft.com/en-us/library/aa905320.aspx`
2. *Паутов П. А.* Проблема аутентификации в многоуровневых приложениях // Прикладная дискретная математика. 2008. № 2. С. 87–90.