Аутентификация

В предыдущем разделе понятие аутентификации уже использовалось в части противостоянии атаке «человек посередине». Иногда понятия аутентификация и идентификация отождествляют, что неверно, хотя они тесно связаны. Уточним сначала различие и близость этих понятий.

Идентификация — это процедура присвоения идентификатора (ID) и в дальнейшем процедура распознавания этого идентификатора из заданного перечня идентификаторов. В качестве идентификатора чаще всего выступают имя или логин, но также могут быть номер паспорта, СНИЛС, номер телефона, e-mail и т.п. Идентификация выполняется при попытке войти в какую-либо систему (например, в операционную систему или в сервис электронной почты) в тот момент, когда пользователь вводит своё имя, или логин, или любую другую информацию в качестве открытого имени (его могут знать многие или все пользователи этой системы, и оно уникально в рамках системы).

Аутентификация

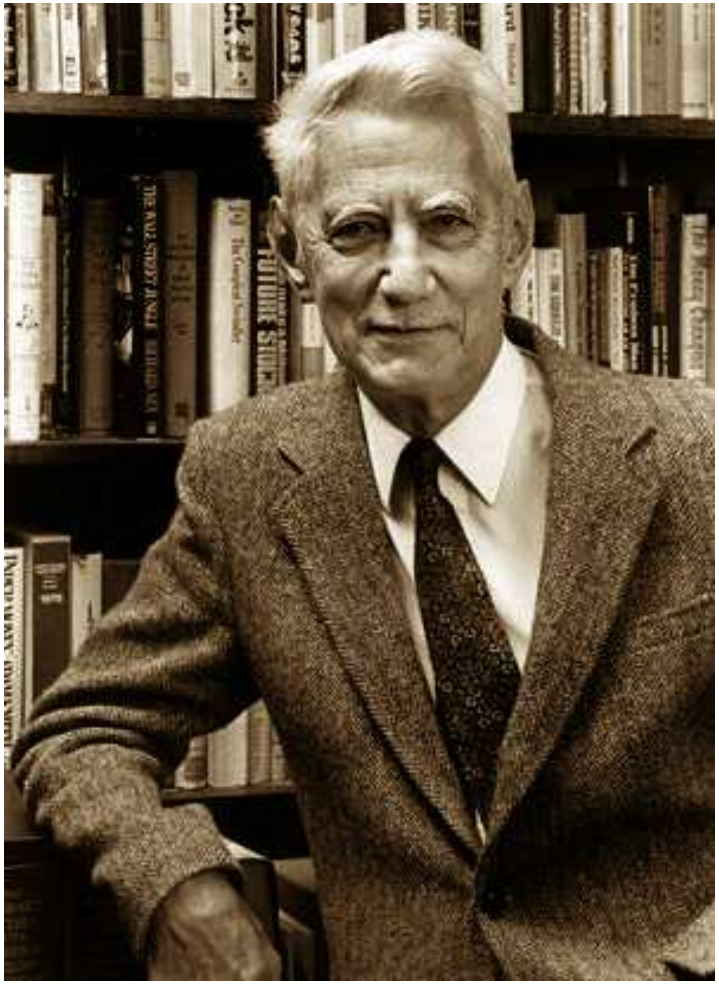
Аутентификация — это процедура проверки подлинности носителя названного идентификатора. Для проведения аутентификации пока используют три фактора: секретную информацию (слово, пароль, код, ключ и т.п.); физическое устройство или предмет, неразрывно связанный с носителем указанного имени (пластиковая карта, водительское удостоверение, паспорт, физический ключ, USB-ключ и т.п.), биометрические данные (отпечаток пальца, портрет, сетчатка глаза и т.п.). Таким образом, аутентификация при входе в систему проходит в тот момент, когда после уникального логина пользователь вводит свой секретный пароль, который известен только ему, чем подтверждает свою подлинность.

Осталось заметить, что рядом с этими понятиями часто фигурирует ещё одно — авторизация. Когда определили ID, проверили подлинность, уже можно предоставить и доступ, то есть, выполнить авторизацию. *Авторизация* – это предоставление доступа к какому-либо ресурсу (например, к электронной почте).

Аутентификация

Аналогично теории связи секретных систем Шеннона в 1985 году была построена теория аутентификации Симмонса.

Клод Э́лвуд Ше́ннон (англ. Claude Elwood Shannon; 30 апреля 1916, Петоски, Мичиган, США — 24 февраля 2001, Медфорд, Массачусетс, США) — американский инженер, криптоаналитик и математик. Считается «отцом информационного века».



Является основателем теории информации, нашедшей применение в современных высокотехнологических системах связи. Предоставил фундаментальные понятия, идеи и их математические формулировки, которые в настоящее время формируют основу для современных коммуникационных технологий. В 1948 году предложил использовать слово «бит» для обозначения наименьшей единицы информации (в статье «Математическая теория связи»). Кроме того, понятие энтропии было важной особенностью теории Шеннона. Он продемонстрировал, что введённая им энтропия эквивалентна мере неопределённости информации в передаваемом сообщении. Статьи Шеннона «Математическая теория связи» и «Теория связи в секретных системах» считаются основополагающими для теории информации и криптографии. Клод Шеннон был одним из первых, кто подошёл к криптографии с научной точки зрения, он первым сформулировал её теоретические основы и ввёл в рассмотрение многие основные понятия. Шеннон внёс ключевой вклад в теорию вероятностных схем, теорию игр, теорию автоматов и теорию систем управления — области наук, входящие в понятие «кибернетика».

Аутентификация

Аналогично теории связи секретных систем Шеннона в 1985 году была построена теория аутентификации Симмонса.

Густавус Джеймс Си́ммонс (англ. Gustavus James Simmons, 27 октября 1930) — американский криптограф, бывший руководитель факультета прикладной математики, старший научный сотрудник Национальной лаборатории «Сандия». Он занимался в основном теорией аутентификации, разрабатывал криптографические методы для решения проблем «взаимного недоверия». Так же поле его деятельности включало в себя разработку протоколов чьи функции могут быть доверенными, даже если некоторые участники (или блоки входных данных) доверенными не являются.

Спустя 40 лет с начала своей работы по «теории аутентификации» Густавус стал известен как «отец теории аутентификации». Его работа нашла применение не только в сфере национальной безопасности, но и в сферах коммерческой и интернет безопасности.

Его вклад в области криптографии включает: развитие шифр-каналов, которые препятствуют искажению информации в электронном документе с использованием закрытого ключа электронно-цифровой подписи; им было сформулировано математическое определение аутентификации канала, включая определение шифр-канала сформулированное ранее Клодом Шенноном в 1948 году. В 1980-х он помог сформировать Международную ассоциацию по криптографическим исследованиям (IACR). Он также является создателем математической игры Sim, основанной на теории графов/теории Рамсея.

Аутентификация

Итак, *аутентификация* — это проверка подлинности либо названного пользователя сети (личности), либо названного сообщения, либо названной сети, либо чего-то ещё названного.

Для проведения аутентификации необходим избыток информации, дополнительная информация, *аутентификатор*.

В качестве аутентификатора может выступать либо секретный пароль, код, ключ, цифровая подпись и т.д., любая дополнительная информация, неразрывно связанная с предметом аутентификации и допускающая использование её для проведения аутентификации.

Аутентификация

1. Аутентификация при входе в систему

В этом разделе в протоколах появляется ещё одно действующее лицо — Хост, обобщающее образ принимающей стороны.

Хост (от англ. host — «хозяин, принимающий гостей») — любое устройство, предоставляющее сервисы формата «клиент-сервер» в режиме сервера по каким-либо интерфейсам и уникально определённое на этих интерфейсах.

Хост проводит аутентификацию с целью опознания реальных клиентов от самозванцев. При этом, как правило, используется парольная система доступа.

Рис. 7. Типичная процедура идентификации, аутентификации и авторизации.

Аутентификация

1. Аутентификация при входе в систему

Майк Гай (Michael J. T. Guy) и **Роджер Нидхем** (Roger Michael Needham) в 1967г., работая над системой регистрации в системе Титан для университетского компьютера в Кембридже, первыми догадались, что Хосту совсем не обязательно знать сами пароли, а достаточно уметь *отличать достоверные пароли от недостоверных*, а именно пришла идея использовать для шифровки паролей односторонний шифр, который в дальнейшем получил название односторонней функции или хэш-функции.

Роджер Нидхем (также Нидэм, Нидем; англ. Roger Needham; 9 февраля 1935, Шеффилд, Саут-Йоркшир, Великобритания — 1 марта 2003, Уиллингем, Кембриджшир, Великобритания) — Британский учёный в области теории вычислительных систем, эксперт в области компьютерной безопасности, Командор ордена Британской империи, член Лондонского королевского общества, член Королевской инженерной академии наук Великобритании.

Аутентификация

1. Аутентификация при входе в систему

1. $A \rightarrow H$: $\{A, P_A\}$, Алиса посылает свой пароль хосту,
2. H: $f(P_A)$, Хост рассчитывает значение однонаправленной функции для данного пароля. Затем Хост сравнивает результат расчета однонаправленной функции со значением, хранящимся в его памяти.

Поскольку на хосте более не хранится таблица достоверных паролей всех пользователей, угроза взлома хоста и кражи таблицы паролей уменьшается. Список паролей, обработанных однонаправленной функцией, бесполезен, так как однонаправленную функцию невозможно инвертировать для восстановления паролей.

Аутентификация

1. Аутентификация при входе в систему

Атака по словарю и привязка

Мэллори (активная взломщица) составляет список из миллиона слов, часто используемых в качестве паролей. Затем она обработает весь миллион слов однонаправленной функцией и сохраняет результаты. Если длина каждого пароля составляет около 8 байт, размер итогового файла не превысит 8 Мбайт. Далее Мэллори необходимо как-то украсть зашифрованный файл паролей и сравнивает этот файл со своим приготовленным файлом зашифрованных вероятных паролей, отыскивая совпадения.

Аутентификация

1. Аутентификация при входе в систему

Атака по словарю и привязка

В противостоянии этой, часто весьма успешной, атаке используют привязку (salt). т.е. случайную строку, которая конкатенируется с паролями перед их обработкой однонаправленной функцией. Затем в базе данных хоста сохраняются как значение привязки, так и результат расчета однонаправленной функции, H: $\{f(P_A S_A), S_A\}$. Теперь Мэллори придется вычислять значения однонаправленной хэш-функции для каждого возможного значения привязки, и увеличение этих вычислений напрямую зависит от числа бит в привязки.

Аутентификация

1. Аутентификация при входе в систему

Атака по словарю и привязка

Однако следует помнить, что привязка — не панацея. Она предохраняет только от самых распространенных методов вскрытий файла паролей с использованием словаря, защищает людей, использующих один и тот же пароль на различных хостах. Но привязка не защищает от направленной попытки вскрытия единственного пароля, нисколько не улучшает неудачно выбранный пароль.

Аутентификация

1. Аутентификация при входе в систему

S/Key — система одноразовых паролей

S/Key представляет собой систему генерирования одноразовых паролей на основе стандартов MD4 и MD5. Она предназначена для борьбы с так называемыми «повторными атаками», когда хакер подслушивает канал, выделяет из трафика аутентификатор пользователя и его пароль и в дальнейшем использует их для несанкционированного доступа.

Принцип ее работы несложен:

1. $A \rightarrow H$: $\{A, R_A\}$, Алиса задаёт своё случайное число.
2. $H : f(R_A) = x_1, f(f(R_A)) = x_2, f(f(f(R_A))) = x_3, \ldots, x_n, f(x_n) = x_{n+1}$. H вычисляет значения $x_1, x_2, x_3, \ldots, x_n, x_{n+1}$. H сохраняет в незашифрованном виде значение x_{n+1} в регистрационной базе данных напротив имени Алисы.
3. $H \rightarrow A$: $\{x_1, x_2, x_3, \ldots, x_n\}$.

Эти обмены проходят по защищённому каналу. В дальнейшем при каждом входе в систему проводится следующий протокол аутентификации.

Аутентификация

1. Аутентификация при входе в систему

S/Key — система одноразовых паролей

ПРОТОКОЛ:

1. $A \rightarrow H$: $\{A, x_n\}$, при первом входе в систему Алиса вводит свое имя и значение x_n.
2. H : $f(x_n) = x_{n+1}$. H вычисляет $f(x_n)$ и сравнивает его со значением с x_{n+1}, которое хранится в регистрационной базе. Если значения совпадают, подлинность Алисы подтверждается. Затем H заменяет в базе данных x_{n+1} на x_n.

A : удаляет из своего списка x_n.

Далее при каждом входе в систему Алиса вводит последнее не вычеркнутое число из своего списка, например, x_i. H вычисляет $f(x_i)$ и сравнивает его со значением x_{i+1}, хранящемся в базе данных. Так как каждый номер используется только один раз, Ева не сумеет получить никакой полезной информации. Точно так же база данных бесполезна и взломщику. Разумеется, когда Алиса исчерпает числа из своего списка, ей придется заново инициализировать систему.

Аутентификация

1. Аутентификация при входе в систему

Аутентификация средствами криптографии с открытым

У предыдущих протоколов, когда Алиса посылает свой пароль Хосту, пароль может прочесть любой, у кого есть доступ к маршруту прохождения данных, т.е. Ева может узнать пароль до его хэширования Хостом.

Эту проблему можно решить средствами криптографии с открытым ключом. Хост сохраняет файл с открытыми ключами всех пользователей, а пользователи хранят свои закрытые ключи. Ниже описана упрощенная схема двухпроходного протокола с открытым ключом:

1. $H \rightarrow A$: $\{R_H\}$, Хост отправляет Алисе случайную строку.
2. $A \rightarrow H$: $\{A, E_A(R_H)\}$, Алиса шифрует строку с помощью своего закрытого ключа и отсылает ее обратно вместе со своим именем.
3. $H : D_A(E_A(R_H)) = R_H$, Хост отыскивает в базе данных открытый ключ Алисы и расшифровывает сообщение этим открытым ключом. Если расшифрованная строка совпадет с той, что хост отправил на этапе 1, хост открывает Алисе доступ к системе.

Аутентификация

1. Аутентификация при входе в систему

Аутентификация средствами криптографии с открытым ключом

Итак, Алиса никогда не отправляет на хост свой закрытый ключ. Поэтому Ева, подслушивая обмен данными, не получит никаких сведений, которые позволили бы ей выдать себя за Алису.

Учитывая способы атак на систему RSA в этих обстоятельствах не рекомендуется шифровать случайные строки, причем не только посланные подозрительным автором — любые.

Аутентификация

1. Аутентификация при входе в систему

Аутентификация средствами криптографии с открытым

Надежные протоколы аутентификации с открытым ключом имеют следующую, несколько более сложную форму:

1. $A \rightarrow H$: $\{E_A(R_A)\}$, где $E_A(R_A)$ — случайное число Алисы R_A, зашифрованное её закрытым ключом;
2. $H \rightarrow A$: $\{R_H\}$, где R_H — случайное число Хоста;
3. $A \rightarrow H$: $\{E_A(f(R_A, R_H))\}$, где f — однонаправленная функция общего пользования;

H: $D_A(E_A(R_A)) = R_A$, Хост расшифровывает первое сообщение Алисы её открытым ключом, вычисляет $f(R_A, R_H)$, и сравниваем полученное значение с расшифрованным сообщением $D_A(E_A(f(R_A, R_H))$ Алисы, полученным на шаге 3. Если они совпадают, то значит Алисе известен её закрытый ключ, и аутентификация прошла успешно.

Аутентификация

1. Аутентификация при входе в систему

Аутентификация средствами криптографии с открытым ключом

Если Алиса доверяет Хосту не в большей мере, чем тот доверяет Алисе, она должна аналогичным образом потребовать, чтобы Хост также подтвердил свою подлинность.

Вопрос: От какой атаки потребовался шаг 1?